MS500

Ultra-Low Power,
Advanced Security,
High Scalability

DATASHEET

— Version 1.2

COPYRIGHT NOTICE

No Warranty Notice

eWBM Co., Ltd. makes no warranty of any kind in regard to this material which is delivered to you as is, including, but not limited to, the implied warranties as to its accuracy or fitness for a specific purpose. Any use of this technical documentation or the information contained therein is at the risk of the user. eWBM Co., Ltd. shall not be liable for errors contained therein or for incidental consequential damages in connection with the furnishing, performance or use of the material.

The information contained in this document is subject to change without notice.

TABLE OF CONTENTS

TABLE LIST

FIGURE LIST

PREFACE

This datasheet provides reference information for the MS500 microcontroller, describing the functional blocks of the system-on-chip (SoC) device designed around the ARM® Cortex™-M0 core.

The purpose of this book is to provide working knowledge of the MS500 microcontroller.

Using this foundation, the reader will be equipped to understand and implement the MS500 microcontroller features.. For better understanding of MS500, the major technical concepts of the MS500 are selected and introduced gradually over several chapters with preactical examples supporting theory.

AUDIENCE

This book is intended for system software developers, hardware designers, and application developers.

DOCUMENT REVISION AND REFERENCE

Revision History

Revision	Date	Description
1.0	22-Aug-2016	1st Release
1.01	12-Dec-2016	RTC register added / SDIO Slave added
1.02	13-Dec-2016	Detail describe Boot mode / Correction typing errors
1.03	15-Dec-2016	Add PMU(power management unit) description
1.04	29-Dec-2016	Add Power consumption table
1.05	03-Jan-2017	Add syscon register & power table classified
1.06	12-Jan-2017	Correction PMU(LPM3) wakeup condition
1.07	24-Jan-2017	Correction BKSRAM mode register
1.08	06-Apr-2017	Match Table 2 and package pin map (PC12~PC15 deleted)
1.09	10-May-2017	Table 17 updated (I2C)
1.10	24-May-2017	Update Figure 1 / Add Flash memory density (4MB)
1.11	09-Sep-2017	Correction Table 36 comments
1.12	24-Nov-2017	Correction BOOT_SRC register
1.13	01-Feb-2018	Change SPI3 operation mode
1.14	06-Mar-2018	Update Block Diagram/Add Crystal operation condition and change register name (STBY_WU_REC -> LPM_WU_REC) on the syscon part
1.15	02-Apr-2018	Update Pin & package configuration and PAD name change
1.2	30-May-2019	Add LoRa Application info

INTRODUCTION

OVERVIEW

The MS500 is an ARM® Cortex-M0™ based RISC microcontroller with Hardware Security Block for embedded applications featuring a high level of integration and low-power consumption.

The MS500 operates at CPU frequency up to 100MHz. The MS500 features a security block called Crypto Block. The Crypto Block consists of Asymmetric Crypto Accelerator (ACA), Symmetric Crypto Accelerator (SCA), and TRNG (True Random Number Generator). It protects the device and its data at boot time, run time and during the communication with other devices or with the cloud.

The peripheral complement of the MS500 includes up to 64 KB of internal SRAM, 8 KB Boot ROM, 4MB Internal flash memory, Non-volatile eFuse memory, Quad SPI interface (SPIF), three SPI controllers, 2-channel DMA controller, two General Purpose Timers, Watchdog Timer, SDMMC interface, three UARTs, four I2Cs, and up to 41 fast general purpose I/O pins.

With its security features as well as low-power, high performance, and diverse connectivity options., the MS500 is ideal for IoT applications such as Smart home applications, Smart metering, Tele-monitoring, Remote Healthcare, and other Electronic devices linked to the Internet.

FEATURES

1. PROCESSOR CORE

- 32-bit hardware multiplier providing Up to 100 MHz operation frequency.
- Built-in Nested Vectored Interrupt Controller (NVIC) for fast deterministic interrupt processing.
- Wake-up Interrupt Controller (WIC) allows automatic wake from any priority interrupt at ultra-low power sleep mode support
- AMBA AHB-Lite Interface
- System Tick Timer.
- 8KB Level 1, 4 way set-associative Instruction Cache for XIP (execute in place).

2. ON-CHIP MEMORIES

- 64 KB SRAM for general purpose such as data buffer supporting security and non-security access
- 8 KB Boot ROM, supporting system code downloaded by UART interface
- 1024 bits Non-Volatile e-Fuse providing storage for the secret keys.
- 4MB (32Mbits) flash Memory

3. CLOCK GENERATOR

- Embedded 32KHz Low-speed Internal RC oscillator for Ultra low power mode.
- Embedded 6~15 MHz RC oscillator allowing operation without external Oscillator or external Crystal. (User Selectable)
- Internal PLL for high frequency clock generation.
- One oscillator with 24MHz clock input and embedded PLLs
- Support global soft-reset control for entire SOC, or individual soft-reset for every components

4. POWER

- Single 3.3V (1.7V~3.6V) power supply with on-chip internal voltage regulator for the core supply and the always-on (AON) power domain.
- Three Low-power modes.: Sleep, Deep-sleep, Standby.
- Built-in Brown-out detection (BOD) circuit for monitoring 3 supply voltage levels.
- Power-on Reset (POR).
- Power Management Unit (PMU) to minimize power consumption during {Sleep} mode.

5. SERIAL INTERFACE

- Quad SPI Flash Interface (SPIF) with 1-, 2- and 4-bit data at rate of up to 50MB per second.
- Three UARTs supporting for DMA and full modem control.
- Four I^{2}C bus interfaces supporting for DMA, master and slave operations, Fast mode plus with data rates of up to 400kbit/s.
- Three SPI controllers; One SPI controller and two SSP controllers supporting FIFO, multi-protocols, DMA, and master and slave operations.
- Support three chip-selects output, serial-master and serial-slave mode, and software configurable.

6. DIGITAL PERIPHERALS

- 2-channel Direct Memory Access (DMA) controller which can access all memories on the AHB-lite and all peripherals which have DMA support.
- Two 2-channel General Purpose Timers.
- Programmable Watchdog Timer.
- Ultra-low power Real-Time Clock (RTC).
- SDMMC card interface with support for eMMC 4.41, SD 3.01 and SDIO 3.0 Host controller.
- SDIO 3.0 Slave controller.
- Up to 41 Fast GPIO pins with configurable.

7. SECURITY BLOCK (HARDWARE CRYPTO BLOCK)

- Asymmetric Crypto Accelerator (ACA)
  - Features:
    - Offloads public key cryptography processing.
    - RSA up to 2048 bit private/public keys
    - ECC up to 512 bit private/public keys

- Symmetric Crypto Accelerator (SCA)

SCA provides a framework including a programmable sequencer, DMA engine, and cryptographic/hashing resources that may handle a variety of protocols involving a cipher and a hash. Also, SCA optimizes cryptographic offload for bulk processing of block cipher and hash algorithms.

- Features:
  - AES128/256(ECB, CBC, CTR, CCM, GCM Modes),
  - SHA1/SHA256 (HASH and HMAC modes)
  - ARIA 128/192/256
  - DMA provides processor offload

- TRNG (True Random Number Generator)

The TRNG generates random data that is intended to be statistically equivalent to a uniformly distributed random data stream.

- Features:
  - Complies NIST SP 800-90a/b/c.FIPS 140-2 and FIPS 140-3 (draft).

- High speed operation
  - 25Mbps at 100MHz

8. PACKAGE INFORMATION

- 60-pin LGA
- Ball Pitch: 0.35 ㎜
- Package Width × Length : 6 ㎜ × 6 ㎜

BLOCK DIAGRAM

The following subsections provides an architectural overview of the MS500.

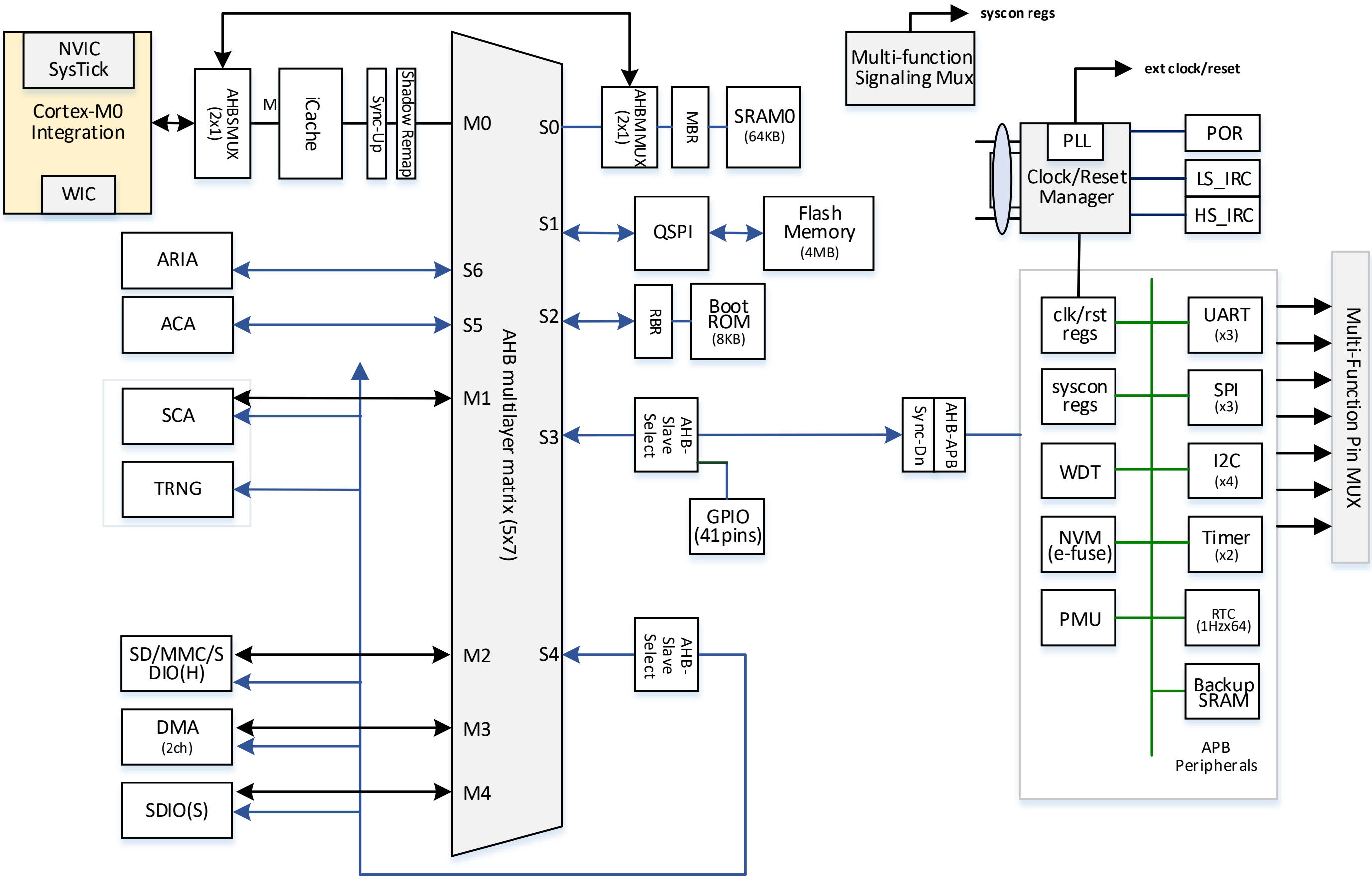

Figure 1 MS500 Block Diagram

PIN & PACKAGE CONFIGURATION

PIN MAP (TOP VIEW)

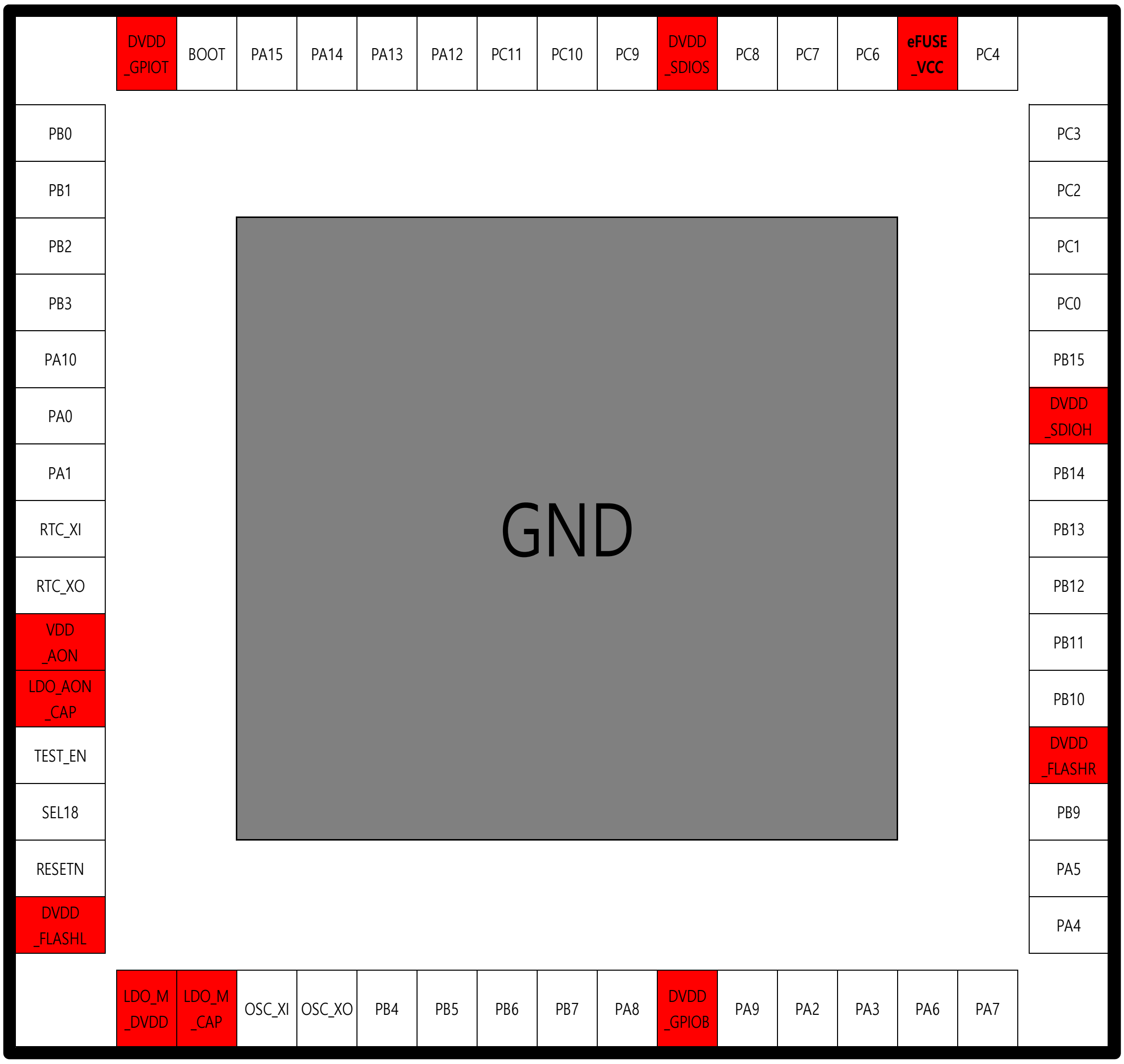

Figure 2 MS500 Pin Map (Top View)

Package Dimension

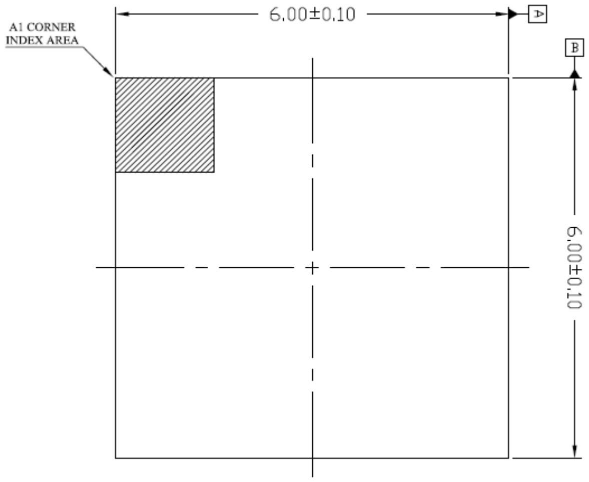

Figure 3 MS500 Package View (TOP VIEW)

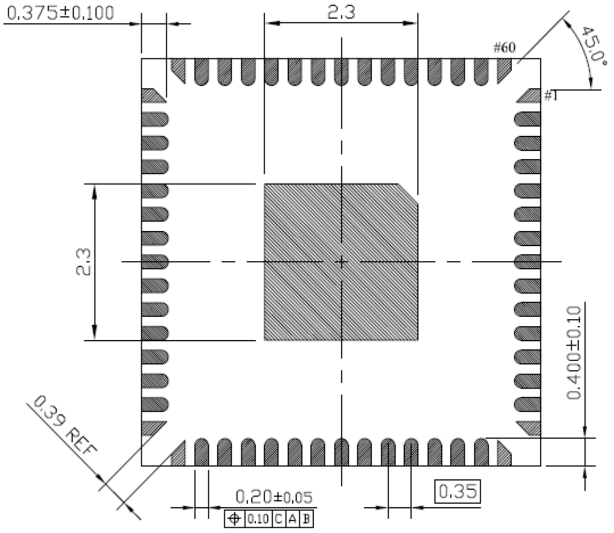

Figure 4 MS500 Package View (BOTTOM VIEW)

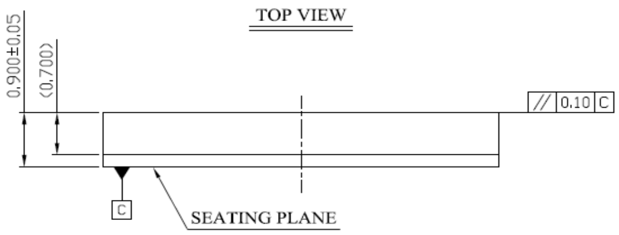

Figure 5 MS500 Package View (SIDE VIEW)

PIN DESCRIPTION

1. SYSTEM CONFIGURATION PINS

Table 1 System Configuration Pins

Pin	Description
OSC_XI	Crystal oscillator XI for system operation, Recommend 24MHz or 12MHz
OSC_XO	Crystal oscillator XO for system operation, Recommend 24MHz or 12MHz
RTC_XI	RTC oscillator XI for Real time clock only(option), use 32.768KHz
RTC_XO	RTC oscillator XO for Real time clock only(option), use 32.768KHz
TEST_EN	Chip test mode enable only, Tie to GND when normal operation
BOOT	Boot mode bit 0
PA6(Boot1)	Boot mode bit 1 when reset release
PA7(Boot2)	Boot mode bit 2 when reset release
SEL18	Select VDDIO voltage range (input 0 : IO voltage is 3.3V, input 1 : IO voltage is 1.8V)
RESETN	Chip reset (Active Low)
PA0	Wake up signal input to wake up from standby mode (Active High)
PA1	When MS500 go to standby mode, this pin output HIGH

Pin Multiplexing Matrix

Table2 shows the pin mux matrix of the MS500, GPIO_i means GPIO input mode.

Table 2 Pin Mux Matrix

Pin	Default	MF0 (SYSTEM)	MF1 (SPI)	MF2 (UART)	MF3 (I2C)	MF4 (TIMER)	MF5 (SDMMC)
PA0	WAKEUP[0]						
PA1	STANDBY_O						
PA2	GPIO_i				I2C3_SCL		SDIO_PULLUP
PA3	TDO_TRACESWO	TDO_TRACESWO			I2C3_SDA		
PA4	GPIO_i				I2C2_SCL		
PA5	GPIO_i				I2C2_SDA		
PA6	BOOT[1]	BOOT[1]					
PA7	BOOT[2]	BOOT[2]					
PA8	GPIO_i	EXTOUT_CLK1		UART1_TX			
PA9	GPIO_i	EXTOUT_CLK2		UART1_RX			
PA10	WAKEUP[1]	WAKEUP[1]		UART1_RTSN			
PA12	TCK_SWCLK	TCK_SWCLK	SSP2_SCK	UART2_RTSN			
PA13	TMS_SWDIO	TMS_SWDIO	SSP2_SSN	UART2_CTSN			
PA14	TDI	TDI	SSP2_MISO	UART3_RTSN			
PA15	TRSTN	TRSTN	SSP2_MOSI	UART3_CTSN			
PB0	GPIO_i		SSP1_SCK		I2C4_SCL	TMR2_MAT[2]	
PB1	GPIO_i		SSP1_SSN		I2C4_SDA	TMR2_CAP[2]	
PB2	GPIO_i		SSP1_MISO		I2C1_SCL	TMR2_MAT[1]	
PB3	GPIO_i		SSP1_MOSI		I2C1_SDA	TMR2_CAP[1]	
PB4	GPIO_i		SSP3_SCK	UART2_TX			
PB5	GPIO_i		SSP3_SSN	UART2_RX			
PB6	GPIO_i		SSP3_MISO	UART3_TX	I2C2_SCL		
PB7	GPIO_i		SSP3_MOSI	UART3_RX	I2C2_SDA		
PB9	GPIO_i				I2C1_SDA		
PB10	GPIO_i						SDHC_CLK
PB11	GPIO_i		SSP1_SCK				SDHC_CMD
PB12	GPIO_i		SSP1_SSN				SDHC_DAT[0]
PB13	GPIO_i		SSP1_MISO				SDHC_DAT[1]
PB14	GPIO_i		SSP1_MOSI				SDHC_DAT[2]
PB15	GPIO_i						SDHC_DAT[3]
PC0	GPIO_i		SSP2_SCK				SDHC_DAT[4]
PC1	GPIO_i		SSP2_SSN				SDHC_DAT[5]

PC2	GPIO_i	SSP2_MISO		SDHC_DAT[6]
PC3	GPIO_i	SSP2_MOSI		SDHC_DAT[7]
PC4	GPIO_i		SDHC_CDN	eMMC_RSTn
PC6	GPIO_i			SDIO_CLK
PC7	GPIO_i			SDIO_CMD
PC8	GPIO_i	SSP3_SCK		SDIO_DAT[0]
PC9	GPIO_i	SSP3_SSN		SDIO_DAT[1]
PC10	GPIO_i	SSP3_MISO		SDIO_DAT[2]
PC11	GPIO_i	SSP3_MOSI		SDIO_DAT[3]

FUNCTIONAL DESCRIPTION

ARM CORTEX-M0 INTEGRATION

The ARM Cortex-M0 is a 32-bit microprocessor with a simple, easy-to-use programmers model and excellent code density, designed for low-power operation and high performance interrupt handling.

1. ARM CORTEX-M0 PROCESSOR

The ARM Cortex-M0 incorporates a 3-stage pipeline von Neumann architecture with separate buses for instruction cache, data sync-up cache, and cache of shadow remap, making it ideal for demanding embedded application. The Cortex-M0 processor also implements the ARMv6-M architecture, which is based on the 16-bit Thumb instruction set that provides the exceptional performance expected of a modern 32-bit architecture, with a higher code density than other 8-bit and 16-bit microcontrollers. In addition, Cortex-M0 closely integrates a configurable Nested Vectored Interrupt Controller (NVIC), to deliver industry-leading interrupt performance

2. NESTED VECTORED INTERRUPT CONTROLLER (NVIC)

The NVIC and the processor core interface are tightly coupled to enable low interrupt latency and efficient processing of late arriving interrupts. It includes 32 interrupt inputs and a Wakeup Interrupt Controller (WIC) When the main processor is in Deep Sleep mode the WIC can wake up the processor. This allows the power to be switched off to the main processor when it is not in use.

Table 3 NVIC Interrupts

Position	Priority	Type of priority	Acronym	Description	Address
0	7	settable	syscon	System Int (Clock Status, WAKEUP pin), plus event which can be used as NMI	0x0000 0040
1	8	settable	RTC	RTC (alarm or wakeup)	0x0000 0044
2	9	settable	DMA	DMA IRQ	0x0000 0048
3	10	settable	DMA	DMA Error IRQ	0x0000 004C
4	11	settable	SDIO	SDIO global interrupt	0x0000 0050
5	12	settable	SDMMC	SDMMC global interrupt	0x0000 0054
6	13	settable	TRNG	Security Module (TRNG)	0x0000 0058

7	14	settable	ARIA	ARIA interrupt	0x0000 005C
8	15	settable	SCA	Security Engine (SCA)	0x0000 0060
9	16	settable	Reserved		0x0000 0064
10	17	settable	ACA	ACA global Interrupt	0x0000 0068
11	18	settable	clk_reset_mgr	Clock reset manager interrupt	0x0000 006C
12	19	settable	WatchDog	WDT interrupt	0x0000 0070
13	20	settable	GPIO1	GPIO1 global interrupt	0x0000 0074
14	21	settable	GPIO2	GPIO2 global interrupt	0x0000 0078
15	22	settable	GPIO3	GPIO3 global interrupt	0x0000 007C
16	23	settable	I2C1	I2C1 global interrupt	0x0000 0080
17	24	settable	I2C2	I2C2 global interrupt	0x0000 0084
18	25	settable	I2C3	I2C3 global interrupt	0x0000 0088
19	26	settable	I2C4	I2C4 global interrupt	0x0000 008C
20	27	settable	SSP1	SSP1 global interrupt	0x0000 0090
21	28	settable	SSP2	SSP2 global interrupt	0x0000 0094
22	29	settable	SSP3	SSP3 global interrupt	0x0000 0098
23	30	settable	UART1	UART1 global interrupt	0x0000 009C
24	31	settable	UART2	UART2 global interrupt	0x0000 00A0
25	32	settable	UART3	UART3 global interrupt	0x0000 00A4
26	33	settable	Timer1	Timer1 global interrupt	0x0000 00A8
27	34	settable	Timer2	Timer2 global interrupt	0x0000 00AC
28	35	settable	Reserved		0x0000 00B0
29	36	settable	Reserved		0x0000 00B4
30	37	settable	Reserved		0x0000 00B8
31	38	settable	Reserved		0x0000 00BC

3. SYSTICK TIMER

The ARM Cortex-M0 processor has a 24-bit count-down timer. This can be configured as a Real Time Operating System (RTOS) tick timer or as a simple counter.

4. BUS SYSTEM

The ARM Cortex-M0 processor within the MS500 implements the AHB-Lite bus interface for communication with internal memories and on-chip peripherals.

MEMORIES

The MS500 contains boot ROM, on-chip SRAM and Non-volatile e-Fuse memory.

1. BOOT ROM

The MS500 contains internal ROM memory which is used to store the boot code for the ARM Cortex-M0. When the ARM Cortex-M0 processor is released from reset, it immediately starts execution of the boot code stored in this ROM.

- The size of ROM is 8 KB

2. ON-CHIP SRAM

The MS500 contains 64 KB of on-chip SRAM which is used by the ARM Cortex-M0 for general purpose memory (stack, heap and data butters). Certain peripherals with DMA capability can also access this memory directly for high speed data processing.

3. NON-VOLATILE MEMORY (NVM)

eFuse is a non-volatile one-time programmable memory used to permanently store device/system configuration settings, software-specific configuration information and user-defined information.

The eFuse memory also contains device specific information including device ID (DID), platform key (PFK), device unique key (DUK), and a user definable key. For details on how to program eFuse with customer specific information, please contact eWBM's technical support.

- 1024-bits high-density electrical Fuse.
- Can be programmed by eWBM for Customer with special condition.

4. INSTRUCTION CACHE

MS500 contains a special instruction cache to increase CPU performance when an application is executing directly from the flash memory connected to the QSPI (Quad SPI) flash interface.

The instruction cache has the following features:

- 8 KB cache memory
- 4-way set-associativity
- 64byte cache line size
- Provide zero-wait read cycle on cache hit
- Pseudo-LRU replacement policy
- Provide optional prefetching mode
- Up to 256MB of cacheable address space

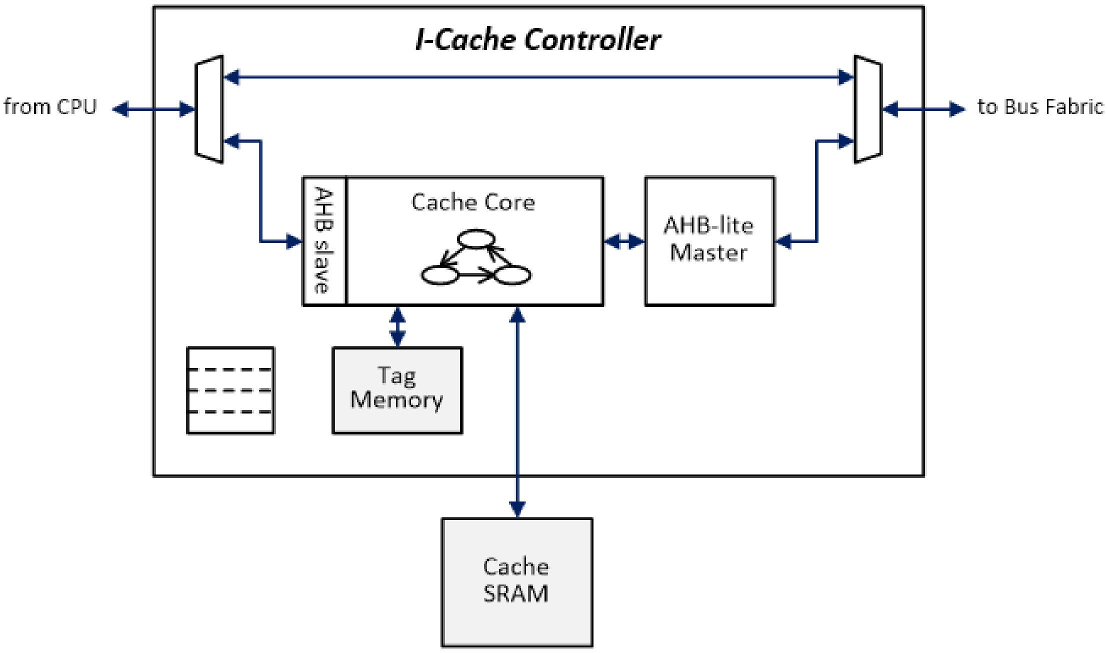

Figure 6 Instruction Cache Block Diagram

Boot Modes

The MS500 supports various boot modes for both normal boot and secure boot. These modes can be selected by the internal eFuse boot configuration settings (Boot SRC[2:0]) as shown in table 4 or Boot mode pin as table5.

Table 4 Boot Modes select by the eFuse's bit(When eFuse 's address 0, bit[2:0] programmed)

Boot Mode	Boot_SRC[2]	Boot_SRC[1]	Boot_SRC[0]	Description
ROM Boot	0	0	0	Boot mode select by external boot mode pin (Not programmed)
UART-to-SRAM	0	0	1	UART2 to SRAM boot code download and boot from SRAM. The Dedicated pins are PB4/PB5.
UART-to-Flash	0	1	0	UART2 to Flash memory boot code download (using HS-IRC), and dedicated pins are PB4/PB5.
SPI-to-SRAM (same boot_src)	0	1	1	SPI3 to SRAM boot code download and use HS-IRC This is default, when eFuse bit does not programmed. The dedicated pins are PB0, PB1, PB2, PB3.
eMMC-to-SRAM (same boot_src)	0	1	1	eMMC to SRAM boot code download and use HS-IRC To enable eMMC, it needs program dedicated eFuse bits. (Address 0x1, bit [14:12] as 001) The eMMC Host pin use to this mode.
SPI Direct Boot1	1	0	0	Boot loading from internal SPI Flash memory directly. (SPI 1-bit mode, using HS_IRC)
SPI Direct Boot2	1	0	1	Boot loading from internal SPI Flash memory directly. (SPI 4-bit mode, using XTAL)
SPI Direct Boot3	1	1	0	Boot loading from internal SPI Flash memory directly. (SPI 4-bit mode, using HS_IRC)

If the Boot_SRC[2:0] is set to 000, the default setting, then the external system pins (Boot [2:0]) are used to select the boot mode as shown in Table 6. This mean that boot Modes selected by boot pins.

Table 5 Boot Modes selected by boot pins (When eFuse 's address 0, bit[2:0] doesn't programmed)

Boot Mode	Boot[2]	Boot[1]	Boot[0]	Description
ROM Boot	0	0	0	Boot loading from ROM.
UART-to-SRAM	0	0	1	UART2 to SRAM boot code download and boot from SRAM The Dedicated pins are PB4/PB5.
UART-to-Flash	0	1	0	UART2 to Flash memory boot code download (SPI 4-bit mode, using HS-IRC) , and dedicated pins are PB4/PB5.

SPI-to-SRAM (same boot_src)	0	1	1	SPI3 to SRAM boot code download and use HS-IRC This is default, when eFuse bit does not programmed.
eMMC2SRAM (same boot_src)	0	1	1	eMMC to SRAM boot code download and use HS-IRC To enable eMMC, it needs program dedicated eFuse bits (Address 0x1, bit [14:12] as 001)
SPI Direct Boot1	1	0	0	Boot loading from internal SPI Flash memory directly.(SPI 1-bit mode, using HS_IRC)
SPI Direct Boot2	1	0	1	Boot loading from internal SPI Flash memory directly.(SPI 4-bit mode, using XTAL)
SPI Direct Boot3	1	1	0	Boot loading from internal SPI Flash memory directly. (SPI 4-bit mode, using HS_IRC)

1. SECURE BOOT

The MS500 is designed specifically to meet the need for the strong security in IoT devices by providing enhanced features such as secure boot, hardware protected secret keys and high performance hardware crypto modules.

The secure boot process is contained within the boot ROM. Once the secure boot has been enabled in the eFuse configuration bits, it cannot be modified and the main application will always be securely booted on power up. If the secure boot process fails, the MS500 will halt and not allow an invalid firmware application to execute.

The secure boot mode supported by the MS500 provides integrity check and authentication of the application firmware.

The boot ROM coordinates the SCA and ACA hardware modules to perform the secure boot process:

Step 1: the SCA hardware module performs SHA256 HMAC integrity check.

Step 2: the ACA hardware module performs an ECC verification of the HMAC to ensure authenticity.

MEMORY MAP

The address map for the MS500 memory and peripherals are defined in Table 7.

Table 6 Memory Mapping

Addr[31:28]	Name	Start Addr	End Addr	Size	Brief Description
0x0	Shadow Region	0x0000_0000	0x0FFF_FFFF	256M	
0x1	ROM	0x1FFF_E000	0x1FFF_FFFF	8KB	On-chip Boot Memory
0x2	On-chip Memory	0x2000_0000	0x2000_FFFF	64K	On-chip SRAM1
		-	-	-	Reserved
0x3	QSPI	0x3000_0000	0x3FFF_FFFF	256M	Flash memory
0x4	APB1 Peripheral	0x4000_0000	0x4000_07FF	2K	System Control Register
		0x4000_0800	0x4000_0FFF	2K	NVM Control Register
		0x4000_1000	0x4000_1FFF	4K	Clock/Reset Manager
		0x4000_2000	0x4000_2FFF	4K	Watchdog Timer
		0x4000_3000	0x4000_37FF	2K	Timer1
		0x4000_3800	0x4000_3FFF	2K	Timer2
		0x4000_4000	0x4000_4FFF	4K	SPI1
		0x4000_5000	0x4000_5FFF	4K	SPI2
		0x4000_6000	0x4000_6FFF	4K	UART1
		0x4000_7000	0x4000_7FFF	4K	UART2
		0x4000_8000	0x4000_8FFF	4K	I2C1
		0x4000_9000	0x4000_9FFF	4K	I2C2
		0x4000_A000	0x4000_AFFF	4K	I2C3
		0x4000_B000	0x4000_BFFF	4K	I2C4
		0x4000_C000	0x4000_CFFF	4K	SSP3
		0x4000_D000	0x4000_DFFF	4K	UART3
		0x4000_E000	0x4000_EFFF	4K	PMU
		0x4000_F000	0x4000_F7FF	2K	RTC
		0x4000_F800	0x4000_FFFF	2K	Backup SRAM(64x32bit)
	AHB Peripheral	0x4002_0000	0x4002_0FFF	4K	GPIO1 (16 I/Os)
		0x4002_1000	0x4002_1FFF	4K	GPIO2 (16 I/Os)
		0x4002_2000	0x4002_2FFF	4K	GPIO3 (16 I/Os)
		-	-	-	Reserved
	Reserved	-	-	-	-
0x5	AHB Controller	0x5000_0000	0x5000_0FFF	4K	DMA
		0x5000_1000	0x5000_1FFF	4K	Reserved
		0x5000_2000	0x5000_2FFF	4K	Reserved
		0x5000_3000	0x5000_3FFF	4K	SD/MMC/SDIO Host
		0x5000_4000	0x5000_4FFF	4K	SD/SDIO Slave Controller
		0x5000_5000	0x5000_5FFF	4K	SDIO CIS slave(128Words)
		0x5000_6000	0x5000_6FFF	4K	Security Module (TRNG)
		0x5000_7000	0x5000_7FFF	4K	
		-	-	-	Reserved
		0x5001_0000	0x5001_FFFF	64K	SCA

		-	-	-	Reserved
	Reserved	-	-	-	-
0x6	AHB	0x6000_0000	0x6000_7FFF	32K	ACA
	Reserved	-	-	-	-
0x7	AHB	0x7000_0000	0x7000_7FFF	32K	ARIA
	Reserved	-	-	-	-
0x8~0xD	Reserved	-	-	-	-
0xE	Cortex-M0	0xE000_0000	0xE00F_FFFF	1M	Cortex-M0 Private Bus
		-	-	-	Reserved
0xF	Cache	0xFFFF_F000	0xFFFF_FFFC	4K	Cache

Power Management

1. POWER DOMAIN

The MS500 has two power domains, Always-on power domain which must always be supplied with power for normal operation, and Main power domain which can be turned on or off, independent of the Always on power domain. When the Main power domain power is off, the MS500 will be in standby mode and consume minimal power. To allow for standby mode to be enabled, separate power supplies should be connected to the external VDD_AON (Always-On power supply) and LDO_M_DVDD (Main power supply) pins or the same power is supplied to VDD_AON and LDO_M_DVDD. When same power is supplied, PMU(Power Management Unit) can shut-down LDO_M and enter standby mode.

The MS500 provides assurance that the required voltage levels are met for each respective mode. If the voltage level is not correct for a specific mode, such as power on reset (POR), the MS500 will perform one of the following actions:

- delay entry into that mode until the correct voltage level is available
- generate a reset (brown-out detection (BOD))

Transitions between these modes are managed by the power management unit which response to CPU command.

The MS500 supports the following power modes in the order from highest to lowest power consumption:

- Active mode
- Sleep mode
  - stops the processor clock
- System power-down modes;
  - Deep-sleep mode stops the system clock and peripheral clock
  - Turn off peripheral power
  - Alive Always-On domain only

2. POWER MODE OPERATION

The MS500 has active mode, sleep mode(LPM1, LPM2) and standby mode(LPM3). For the low power operation, CPU can function module power on/off control in the active mode. The LPM1,LPM2, LPM3 mode define which module on and off and this can easy control by power management register. The LPM3 mode which is Main power domain is off and Always on power domain power on only. When enter standby mode, MS500 power consumption is very low.

The power mode control flow started by CPU settings as figure 7. When power mode set such as LPM1, LPM2, LPM3, then MS500's PMU control isolation and power gating cells to entered pre-defined power mode as Table 7. The wakeup process depends on the which power mode. For example, wakeup from standby mode(LPM3), it needs input from the wake-up pin or chip reset or RTC interrupt.

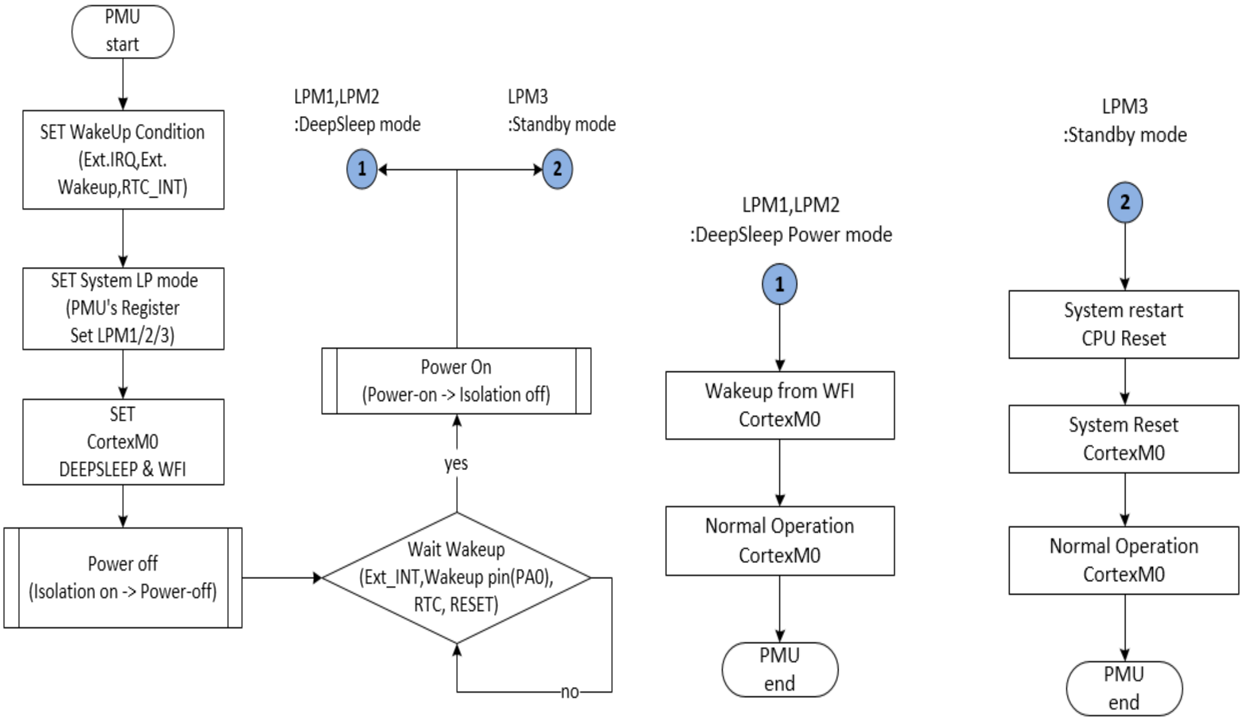
PMU start
SET WakeUp Condition (Ext.IRQ,Ext. Wakeup,RTC_INT)
SET System LP mode (PMU's Register Set LPM1/2/3)
SET CortexM0 DEEPSLEEP & WFI
Power off (Isolation on -> Power-off)
LPM1,LPM2 :DeepSleep mode
LPM3 :Standby mode
1
2
Power On (Power-on -> Isolation off)
yes
Wait Wakeup (Ext_INT,Wakeup pin(PA0), RTC, RESET)
no
LPM1,LPM2 :DeepSleep Power mode
1
Wakeup from WFI CortexM0
Normal Operation CortexM0
PMU end
LPM3 :Standby mode
2
System restart CPU Reset
System Reset CortexM0
Normal Operation CortexM0
PMU end

Figure 7 Power mode operation

As show in the Table 7, MS500 power mode is defined as Active, LPM0, LPM1, LPM2, LPM3. In the Active mode, some power group's power can be controlled by CPU. These power groups are SDMMC host, SDIO slave, SCA, ACA, ARIA.

To wakeup from LPM0, LPM1, LPM2, LPM3, it needs input from outside such as pin interrupt or wakeup pin (PA1).

In the LPM3 mode, PMU operate main power domain off, so wakeup from LPM3 start from ROM boot.

Table 7 Low Power mode and power domain table

Power Source	Power group /Power mode	Active	LPM0	LPM1	LPM2	LPM3
Main LDO Power	Core Retention F/F	On	On	On	On	Off
	Core Combination Logic	Off	On(clock off)	Off	Off	Off
	Outside of Core(WIC..)	On	On	On	On	Off
	CRM	On	On	On	On	Off
	IO Mux	On	On	On	On	Off
	SYSCON, WDT CACHE QSPI e-Fuse, NVM I/F	On	On	On	On	Off
	(Arch. Group) BUSMATRIX DMA UART1, UART2 SPI1, SPI2 SPI3, SPI4 Timer1, Timer2 I2C1, I2C2, I2C3, I2C4 GPIO1, GPIO2, GPIO3	On	On	On	Off	Off
	ROM	On	On	On	On	Off
	SRAM	On	On	On	On	Off
	SDMMC Host	On (CPU can set On/Off)	On (CPU can set On/Off)	Off	Off	Off
	SDIO Slave	On (CPU can set On/Off)	On (CPU can set On/Off)	Off	Off	Off
	SCA	On (CPU can set On/Off)	On (CPU can set On/Off)	Off	Off	Off
	ACA	On (CPU can set On/Off)	On (CPU can set On/Off)	Off	Off	Off
	ARIA	On (CPU can set On/Off)	On (CPU can set On/Off)	Off	Off	Off
Always On LDO Power	RTC PMU, Backup-SRAM	On	On	On	On	On

Table 8 Entrance of Low Power Mode Condition

Low Power Mode	Condition	Entrance Source
LPM0 (without PMU)	CPUWFI or with SLEEPDEEP	CPU SLEEPING or SLEEPDEEP

LPM1(with PMU)	CPUWFI with SLEEPDEEP
	HWREG(NVIC_SYS_CTRL) |= NVIC_SYS_CTRL_SLEEPDEEP;
	CPUwfi();
	HWREG(NVIC_SYS_CTRL) &=
	~(NVIC_SYS_CTRL_SLEEPDEEP);
		CPU SLEEPDEEP

LPM2(with PMU)		
LPM3(with PMU)		

Table 9 Wakeup Condition of Low Power Mode

Low Power Mode	Condition	Wakeup Source
LPM0 (without PMU)	wakeup by CPU wait-for-interrupt	All Interrupts
LPM1 (with PMU)	wakeup by SYSCON Wakeup signals	SYSCON wakeup signal (RTC/WAKEUP/RESETN) *RESETN : not recommand
LPM2 (with PMU)		
LPM3 (with PMU)	wakeup by AON_PMU	RTC/PA0/RESETN

Table 10 Power Management Register Description

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
0x00	**LPM_STT**	32	31:0	**This register show the current Low Power Status**		R
	volt_domain_stt	9	8:0	Power On/Off status of Voltage Domains 0- Power Down state 1- Power Up state bit[0] : CPU Retention F/F Power, Outside of Core(WIC,.. etc) bit[1] : CPU's Core internal Power bit[2] : System Operation Group Power(CRM, SYSCON, WDT, CACHE, QSPI) bit[3] : Peripherals IP Group Power bit[4] : SD/MMC Group Power bit[5] : SDIO Group power bit[6] : Cryptographer Group Power bit[7] : PKA Group Power bit[8] : ARIA Group Power	-	
		3	*11:9*	*Reserved*		
	lpm_stt	3	14:12	Low Power Mode Status 000 : Normal Operation 001 : Low Power mode 1 010 : Low Power mode 2 011 : Low Power mode 3(Standby Mode) reverved : Normal Operation	-	
	-	17	31:15	*Reserved*		
0x04	**LPM_CONF**	32	31:0	**This register is enable the Low Power Mode.** **When the write command must be run at a time. and then must follow immediately the command of Cortex CPU WIC(Wait for Interrupt Command).** **For Example with controlling Flash Device's PowerDown and LPM3** **First, write 0x0001_B000 to LPM_CONF's Register and write wait for interrupt command to CPU.** **If you want to control without Flash Device Power, write 0x0000_B0000 to this register.**		R/W

	volt_domain_conf	9	8:0	Power On/Off setting of Voltage Domains, when volt_domain_en is set and lpm_conf is all zero. 0- Power Off 1- Power On bit[3:0] : set to 1 (default) bit[4] : On/Off of SD/MMC Group Power bit[5] : On/Off of SDIO Group power bit[6] : On/Off of Cryptographer Group Power bit[7] : On/Off of PKA Group Power bit[8] : On/Off of ARIA Group Power when lpm_conf is not zero, this configure is ignored. According to lpm_conf, volt_comain_conf is controlled by Hardware as below. LPM1 : bit[8:0]=9'b0_0000_1101 LPM2 : bit[8:0]=9'b0_0000_0101 LPM3 : bit[8:0]=9'b0_0000_0000	1FFh	
		3	*11:9*	*Reserved*		
	lpm_conf	2	13:12	Low Power Mode configuration 00 : Normal Operation 01 : Low Power mode 1 10 : Low Power mode 2 11 : Low Power mode 3(Standby Mode)	0h	
		1	*14*	*Reserved*		
	volt_domain_en	1	15	Voltage Domain Power On/Off enable on normal operation 0 : power control disable 1: power control enable **NOTE. This Register is start indicator. This set value does not auto-refresh.** **It should be write disable first, if CPU send command again to Low Power Mode.**	0h	
	peri_handshake_en	1	16	Flash Device Power Down/On handshake enable Register 0: handshake disable 1: handshake enable **NOTE1. If SPI-Flash Boot Mode is set, this must be always disable.**	0h	
		7	*23:17*	*Reserved*		
	mem_ret[2:0]	3	26:24	Memory Retention Control by itself (Active Low) 0 - Normal, 1- Retention mode (internal clock off) bit[0] : always set to 0 bit[1] : RAM retention bit[2] : CACHE RAM Retention	0h	
		1	*27*	*Reserved*		
	mem_pg_en[2:0]	3	30:28	Memory Power Gate enable by itself (Active High) 0-Normal, 1- Power-Down bit[0] : ROM Power Down enable (SPI flash direct boot) bit[1] : RAM Power Down enable bit[2] : CACHE RAM Power Down enable	0h	
	-	*1*	*31*	*Reserved*		
0x08	**LPM_WAKEUP_CNT**	32	31:0	**This register is the value of power up switching delay.**	-	RW
	lpm_wakeup_cnt	13	12:0	when escape from Low Power Mode, Switching Delay time value. switch delay time = lpm_conf x SYS_CLK(referenced max frequency is 12MHz=83ns)	0h	
	-	19	31:13	*Reserved*		
0x0C	**Reserved**	31	31:0		-	
	-	31	31:8	*Reserved*		

CLOCKS

1. CLOCK GENERATION UNIT

The MS500 clock generation unit is responsible for providing the clock signals to all subsystems and ensuring clean, glitch free switching between the various clock sources.

The MS500 supports many clock sources which can be any one of the following:

- External crystal
- High-speed internal RC oscillator (12MHz)
- Low-speed internal RC oscillator (32KHz)
- A phase-locked loop (PLL),

The clock generation unit also features safe Auto-switching control (to safe clock) and a separate Always-On (AON) Clock Generation Unit for standby mode.

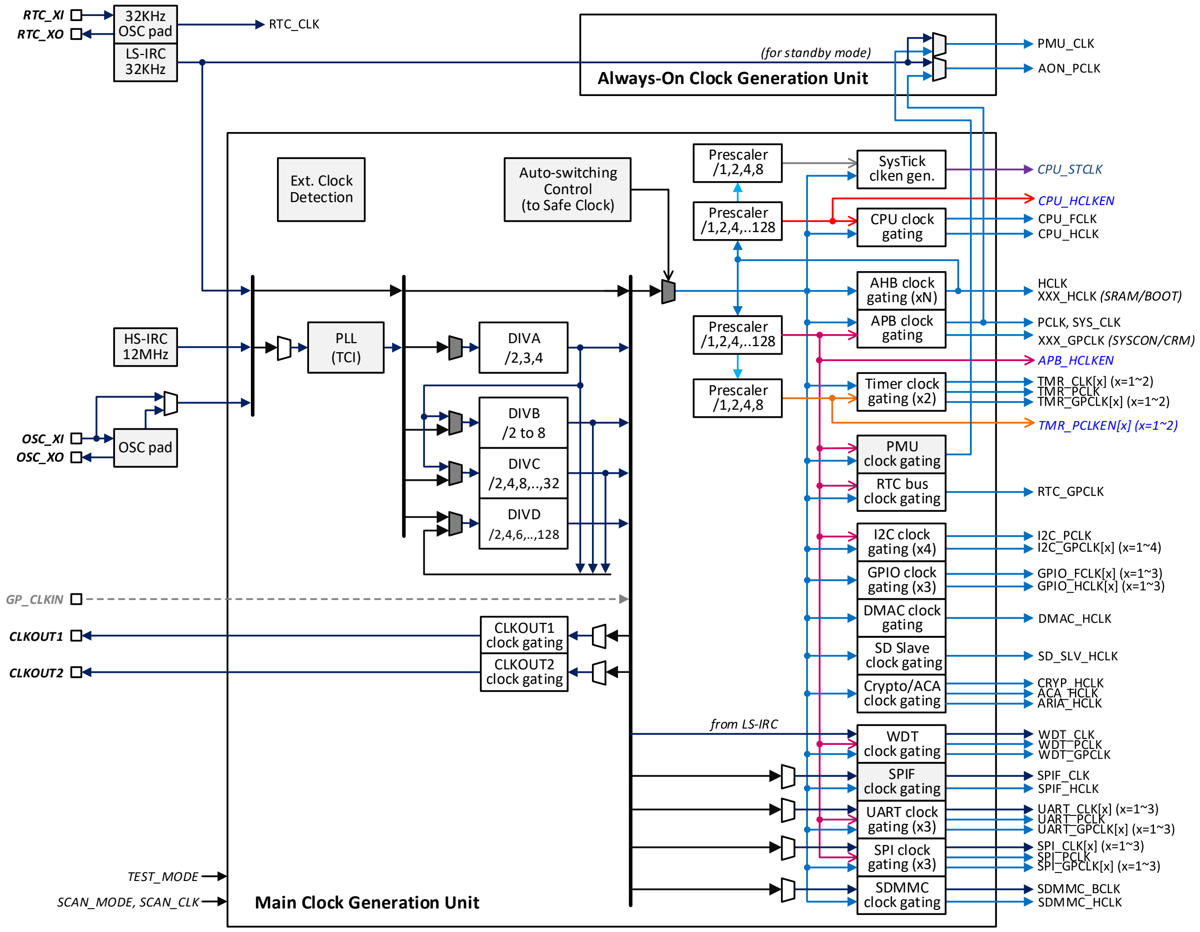

Figure 8 MS500 Clock Generation

2. CRYSTAL

32 KHz Crystal Oscillator

The RTC oscillator is specifically designed for real time clock (RTC) applications, generates an on-chip asynchronous clock signal using a 32.786 kHz external crystal. It is characterized with crystal loading capacitors ranging from 4 pF to 25 pF.

50 MHz Low Power Crystal Oscillator

The design has been optimized for stability and minimum jitter and can be used over a frequency range of <1MHz to 50MHz.

This oscillator is designed to generate an asynchronous on-chip clock signal with an appropriate external crystal. The design has been optimized for stability and minimum jitter and can be used over a frequency range of <1MHz to 50MHz. Designed to operate on core power only, this low power oscillator has a fixed drive strength with a maximum figure of effort of EF = 0.7 to guarantee stable oscillation.

$$EF = f * C^{0.8} * R^{0.61}$$

Where

EF = figure of effort
f = frequency of oscillation
C = capacitive loading on XI and XO
R = crystal equivalent series resistance

3. INTERNAL HIGH-SPEED CLOCK SOURCE

The High-speed(HS) IRC is the high speed clock source of internal clock in the MS500.

The High-speed(HS) IRC is the clock source for the internal high speed clock of the MS500. If the user decides to use this HS-IRC it can be trimmed during testing to achieve the specified accuracy (10%) over the entire voltage and temperature range. Trim values are permanently stored in the internal eFuse area. Additional trim settings can also be stored in flash and can be used to compensate for the changes in operating conditions. The HS IRC default frequency is 12 MHz and it can be select 6MHz or 12MHz by register set.

4. INTERNAL LOW-SPEED CLOCK SOURCE

The Low-speed(LS) IRC is a very low power oscillator, nominally 32KHz, which is primarily used to generate the clocks for the internal peripheral blocks when operating in standby mode or Always-On mode. This does not use RTC clock, RTC clock should come from RTC X'tal (32.768KHz).

5. PLL

The PLL accepts an input clock frequency from an external oscillator or internal IRC.

– Divided reference frequency range	4.69MHz - 600MHz
– 1 output frequency range (VCO output internally divided by 2 for 50% DC)	120MHz - 600MHz
– Fout = Fref * NF / NR / OD	
– Reference divider values (NF)	1~16
– Feedback divider values(NR)	1~64
– Output divider values(OD)	1~16
– Output duty cycle (nom, tol)	50%, +/-2%
– Static phase error (max)	+/-1.25% div. reference cycle
– Period jitter (P-P) (max)	+/-3% output cycle
– Input-to-output jitter (P-P) (max) (jitter numbers are worst-case estimates with supply and substrate noise levels below - actual results will be better)	+/-1.5% div. reference cycle
– Power dissipation (nom)	1.5mA @ 300MHz (/1 output)
– Reset pulse width (min)	5µs
– Reset /1 output frequency range	10MHz - 100MHz
– Lock time (min allowed) (actual lock time will be much smaller)	500 div. reference cycles
– Freq. overshoot (full-~/half-~) (max)	40%/50%

6. ALWAYS-ON CLOCK GENERATION UNIT

The external VDD_AON (Always-On power supply) supplies the power to the power domain and the LDO_M_DVDD (Main power supply) provide the main power domain.

When the main power domain is off, the always-on domain will remain alive with a low frequency clock such as the internal low speed cock.

To wake up from standby mode, a wake up signal must be provided to the WAKEUP pin (Active High) or RESETN pin (Active Low).

RESET

A power on reset is required to place the device in a known good state after power-up. The MS500 is reset by internal Power-on Reset (PoR) module, the external reset signal, or the Watchdog Timer. In all cases, the MS500 restart program execution from the address provided by the reset entry point. A register can be later referenced to determine the source of the reset. The reset signal also propagates to the CPU peripherals and the rest of the CPU subsystems.

1. RESET GENERATION DIAGRAM

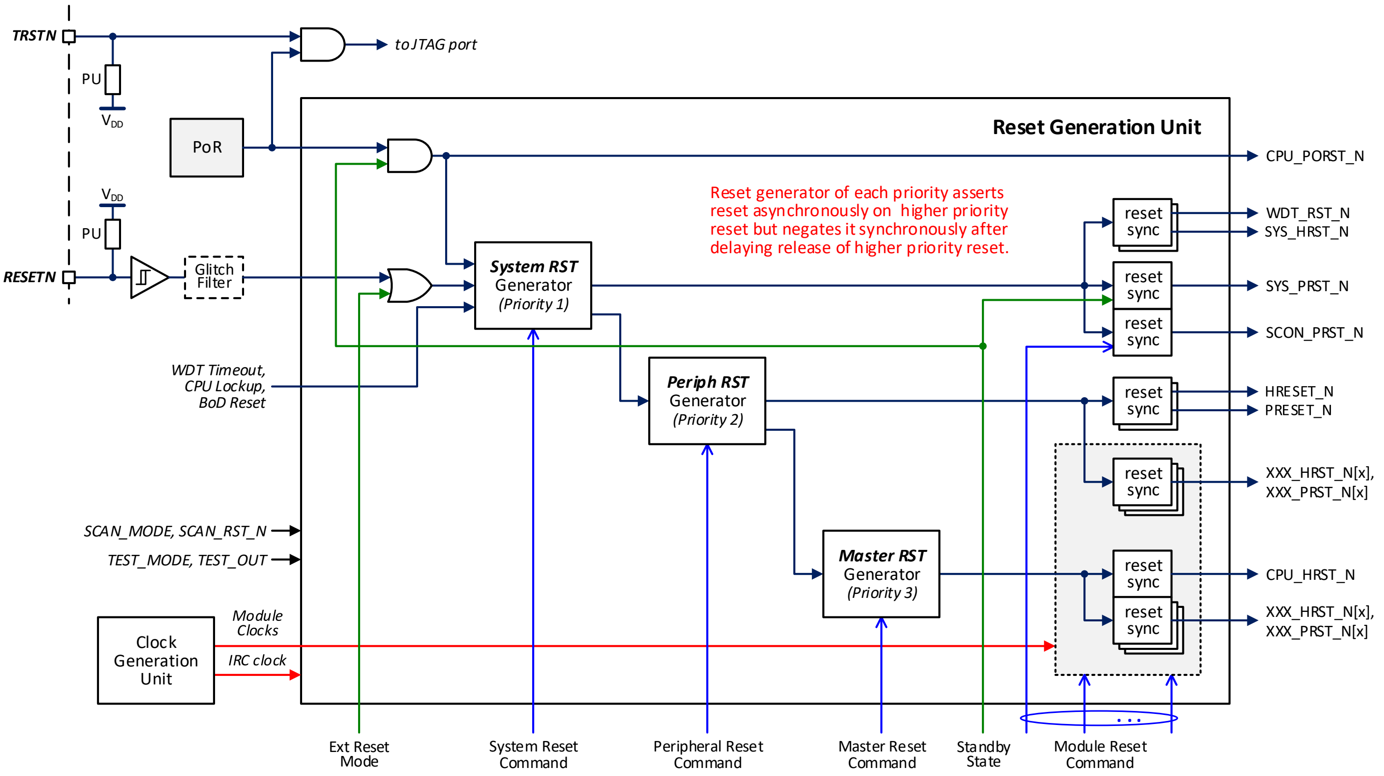
TRSTN
PU
VDD
to JTAG port
PoR
Reset Generation Unit
CPU_PORST_N
VDD
PU
RESETN
Glitch Filter
Reset generator of each priority asserts reset asynchronously on higher priority reset but negates it synchronously after delaying release of higher priority reset.
reset sync
WDT_RST_N
SYS_HRST_N
System RST Generator (Priority 1)
SYS_PRST_N
SCON_PRST_N
WDT Timeout, CPU Lockup, BoD Reset
Periph RST Generator (Priority 2)
HRESET_N
PRESET_N
XXX_HRST_N[x], XXX_PRST_N[x]
SCAN_MODE, SCAN_RST_N
TEST_MODE, TEST_OUT
Master RST Generator (Priority 3)
CPU_HRST_N
XXX_HRST_N[x], XXX_PRST_N[x]
Clock Generation Unit
Module Clocks
IRC clock
Ext Reset Mode
System Reset Command
Peripheral Reset Command
Master Reset Command
Standby State
Module Reset Command

Figure 9 Reset Generation Block Diagram

2. RESET SOURCES

- Power on Reset (PoR)

Power-on reset is initiated by the PoR (power on reset module) inside of MS500. Power on reset sets all of the device internal logic to its default state. This PoR operates followed by power on stage as automatically.

- External Reset Input Pin (RESETN) assertion

The external reset pin (RESETN) is resets the MS500 including the core and all the on-chip peripherals except JTAG.

- A Watchdog Timer

Watchdog timer consists of a 32-bit down counter with a programmable timeout interval that has the capability to generate an interrupt and a reset signal on timing out. It is used to apply a reset to a system in the event of a software failure.

3. RESET GENERATION STRUCTURE

Table 11 MS500 Reset Hierarchy

Reset Hierarchy	Target							Reset Source
	CRM	SysCon, PMU	WDT	Bus	Peri	CPU	AON Domain	
Power Reset	Yes	Yes	Yes	Yes	Yes	Yes	Yes	V_{DD} Power on Reset (PoR)
System Reset	Yes	Yes	Yes	Yes	Yes	Yes	-	Higher Priority (0) Reset External reset pin Watchdog Timeout Reset, CPU Lockup Detection, BoD Reset Reset Command (Group or Module)
Peripheral Reset	-	-	-	Yes	Yes	Yes	-	Higher Priority (1) Reset Reset Command (Group or Module)
Wakeup reset	Yes	Yes	Yes	Yes	Yes	Yes		Wakeup from Standby mode (Wake-up pin)

- CRM : Clock reset module
- PMU : Power Management Unit
- WDT : Watchdog
- AON Domain : Always on Power domain

4. RESET GENERATION SIGNAL

Table 12 RESET GENERATION SIGNAL

Module	Hierarchy	Clock Name	Brief Description	Reset Source	Sync Clock (@Release)	SW Command
CPU	Power Reset	CPU_PORSTN	Cortex-M0 power reset Cortex-M0 includes reset sync logic for power reset.	PoR (V_{DD})	-	-
	Module Reset	CPU_HRSTN	Cortex-M0 reset	ResetGen	CPU_HCLK	CM0
System	System Reset	SYS_RSTN	System reset (for APB clock domain) This is not asserted by standby.	ResetGen	PCLK	CM0+CRM
SysCon	Module Reset (System)	SYSCON_RSTN	SysCon reset This is not asserted at standby wakeup and is not asserted by peripheral group reset.	ResetGen	PCLK	CRM
AHB Fabric & Memory	Peripheral Reset	HRESETN	AHB reset	ResetGen	HCLK	CM0+CRM
APB Fabric	Peripheral Reset	PRESETN[x]	APB reset	ResetGen	PCLK	CM0+CRM
Timer	Module Reset	TMR_PRSTN[x]	Timer reset	ResetGen	HCLK	
UART	Module Reset	UART_RSTN[x]	UART reset	ResetGen	HCLK	CRM
		UART_PRSTN	UART APB reset	ResetGen	PCLK	CRM
SPI	Module Reset	SPI_RSTN[x]	SPI reset	ResetGen	HCLK	CRM
		SPI_PRSTN	SPI APB reset	ResetGen	PCLK	CRM
I2C	Module Reset	I2C_PRSTN[x]	I2C reset	ResetGen	PCLK	CRM
GPIO	Module Reset	GPIO_HRSTN[x]	GPIO reset	ResetGen	HCLK	CRM
SDMMC	Module Reset	SDMMC_HRSTN	SDMMC reset	ResetGen	HCLK	CRM
DMAC	Module Reset	DMAC_HRSTN[x]	DMAC reset	ResetGen	HCLK	CRM
CRYP	Module Reset	CRYP_HRSTN	Crypto reset	ResetGen	HCLK	CRM
ACA	Module Reset	ACA_HRSTN	ACA reset	ResetGen	HCLK	CRM
ARIA	Module Reset	ARIA_HRSTN	Aria reset	ResetGen	HCLK	CRM
SDSLV	Module Reset	SDSLV_HRSTN	SD Slave reset	ResetGen	HCLK	CRM
WDT	System Reset	WDT_RSTN	Watchdog timer reset	ResetGen	WDT_CLK	-
		WDT_PRSTN	Watchdog timer APB reset	ResetGen	PCLK	CRM
RTC	Power Reset	RTC_PORSTN	RTC power reset This needs reset sync logic in RTC module.	PoR (AON domain)	-	-
	Module Reset	-	RTC reset	-	WDT_CLK	RTC
CM0 JTAG	Test Reset	CM0_TRSTN	Cortex-M0 JTAG-TRST This is asserted on PoR or external pin.	External Pin or PoR (V_{DD})		
NVM JTAG	Test Reset	MVM_TRSTN	NVM interface JTAG-TRST This is asserted on PoR or external pin.	External Pin or PoR (V_{DD})		

Table 13 MS500 Clock/Reset Manager Register Description

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
Note 1: System clock and reset is synchronized clock with APB clock, thus they are used for always-on domain. System clock is not stopped during all low-power modes and system reset is asserted Note 2: Default clock after power-on reset (which is also called as safe clock) is LS-IRC clock. If internal clock monitor detects the fault of system clock and its source, system clock is automatically switched to safe clock. Program can recognize this by reading SysCon register when clock fault interrupt is occurred as NMI. Note 3: If a PLL and clock divider is used as clock source at output port, it cannot be disabled by program Note 4: Clock switching using glitchless mux is executed only when current and new clock are running. If current clock is stopped, clock switching can be executed using force mode. But new clock is not running, clock switching is never occurred. Program can check the actual state of clock selection by reading clock select bits. Note 5: Clock switching using general mux is executed in independent of the both clock states. In order to avoid glitch clock, switching can be executed only during the output is disabled.						
0x00	**CLK_STS**	32	31:0	**Clock Status Register** Note: Changing values of PLL/Divider/Clock Control Register causes the corresponding bit goes to 0 immediately Note: Changing values of Divider/Clock Control Register is not allowed during unready state by previous writing to Divider/Clock Control Register	0x0	RO
	-	1	0	*Reserved*		
	XTAL_RDY	1	1	X'tal clock ready - clock detection		
	-	2	3:2	*Reserved (for GP_CLKIN)*		
	PLL_LOCK	1	4	PLL Lock detection		
	PLL_UNLOCK	1	5	PLL Unlock detection (for Interrupt)		
	-	2	7:6	*Reserved*		
	DIVA_RDY	1	8	Divider A clock ready - Clock change		
	DIVB_RDY	1	9	Divider B clock ready - Clock change		
	DIVC_RDY	1	10	Divider C clock ready - Clock change		
	DIVD_RDY	1	11	Divider D clock ready - Clock change		
	-	4	15:12	*Reserved*		
	AHB_CLK_RDY	1	16	AHB clock ready - Clock change		
	CPU_CLK_RDY	1	17	CPU clock ready - Clock change		
	APB_CLK_RDY	1	18	APB clock ready - Clock change		
	-	1	19	*Reserved*		
	ST_CLK_RDY	1	20	CM0 SysTick clock ready - Clock change		
	-	1	21	*Reserved*		
	TMR_CLK_RDY	1	22	Timer clock ready - Clock change		
	-	9	31:23	*Reserved*		
0x04	**CLK_INT_EN**	32	31:0	**Clock Interrupt Enable Register** Note: Each bit is corresponding to each bit of **CLK_STS** register. 1b is enabled state, thus enabled signal source is used for interrupt generation	0x0	RW
0x08 (Read)	**CLK_INT_STS**	32	31:0	**Clock Interrupt Status Register** Note: Interrupt event is only the change from low(0b) to high(1b) in **CLK_STS** Note: The only bit enabled by **CLK_INT_EN** can be set from **CLK_STS**, otherwise it will be read as zero	-	RW1C
0x08 (Write)	**CLK_INT_CLR**	32	31:0	**Clock Interrupt Clear Register** Note: Each bit is corresponding to each bit of **CLK_INT_STS** register. Writing 1b to a bit clear the bit, which of **CLK_INT_STS** is in same position		
0x0C	***reserved***	32	31:0	-	-	Rsvd
0x10	**CLK_CTRL**	32	31:0	**Clock Control Register**	0x0	RW

	XTAL_EN	1	0	X'tal clock enable		
	XTAL_BYPASS	1	1	X'tal Bypass Mode enable 0: Crystal Connected (default), 1: Use external clock source instead of crystal		
	HS_IRC_EN	1	2	HS-IRC enable		
	-	29	31:3	*Reserved*		
0x14	***reserved (for FREQ_MON)***	32	31:0	-	-	RW
0x18	**PLL_CTRL**	64	31:0	**PLL Control Register**	0x0	RW
	PLL_CLK_SEL	2	1:0	Clock source selection for PLL 00: LS-IRC (default) 01: HS-IRC 10: X'tal Osc 11: *reserved (for GP_CLKIN)*		
	-	2	3:2	*Reserved*		
	PLL_USED	1	4	Clock output used indicator 0: Not used, 1: at least one branch clock running		RO
	PLL_EN	1	5	PLL enable 0: Disable, 1: Enable		
	PLL_BYPS	1	6	PLL bypass mode enable 0: Normal, 1: Bypass		
	PLL_RST	1	7	PLL Reset		
	PLL_FB_DIV	6	13:8	Feedback divider value 0x00: divided by 1 (not divided) ... N: divided by N+1 ... 0x3F: divided by 64		
	-	2	15:14	*Reserved*		
	PLL_PRE_DIV	4	19:16	Input divider value 0x0: divided by 1 (not divided) ... N: divided by N+1 ... 0xF: divided by 16		
	PLL_POST_DIV	4	23:20	Output divider value 0x0: divided by 1 (not divided) ... N: divided by N+1 ... 0xF: divided by 16		
	PLL_BWA	6	29:24	Loop Filter Bandwidth Adjust		
	-	2	31:30	*Reserved*		Rsvd
0x1C	***reserved (for PLL2)***	32	31:0	-	-	Rsvd
0x20	**DIVA_CTRL**	32	31:0	**Divider A Control Register**	0x0	RW
	DIVA_CLK_SEL	2	1:0	Clock source selection for Divider A 00: LS-IRC (default) 01: HS-IRC 10: X'tal Osc 11: PLL output Note: Clock switching is glitchless type		
	-	2	3:2	*Reserved*		
	DIVA_USED	1	4	Clock output used indicator 0: Not used, 1: at least one branch clock running		RO
	-	3	7:5	*Reserved*		
	DIVA_VAL	2	9:8	Divider value 00: Clock disabled		

				01: divided by 2 10: divided by 3 11: divided by 4		
	-	22	31:10	*Reserved*		Rsvd
0x24	**DIVB_CTRL**	32	31:0	**Divider B Control Register**	0x0	RW
	DIVB_CLK_SEL	3	2:0	Clock source selection for Divider B 000: LS-IRC (default) 001: HS-IRC 010: X'tal Osc 011: PLL output 100: Divider A Others: *reserved* Note: Clock switching is glitchless type		
	-	1	3	*Reserved*		
	DIVB_USED	1	4	Clock output used indicator 0: Not used, 1: at least one branch clock running		RO
	-	3	7:5	*Reserved*		
	DIVB_VAL	3	10:8	Divider value 000: Clock disabled 001: divided by 2 ... N: divided by N+1 ... 111: divided by 8		
	-	21	31:11	*Reserved*		Rsvd
0x28	**DIVC_CTRL**	32	31:0	**Divider C Control Register**	0x0	RW
	DIVC_CLK_SEL	3	2:0	Clock source selection for Divider C 000: LS-IRC (default) 001: HS-IRC 010: X'tal Osc 011: PLL output 100: Divider A Others: *reserved* Note: Clock switching is glitchless type		
	-	1	3	*Reserved*		
	DIVC_USED	1	4	Clock output used indicator 0: Not used, 1: at least one branch clock running		RO
	-	3	7:5	*Reserved*		
	DIVC_VAL	3	10:8	Divider value (by power of 2) 000: Clock disabled 001: divided by 2 ... N: divided by 2^N ... 101: divided by 32 Others: *forbidden*		
	-	21	31:11	*Reserved*		Rsvd
0x2C	**DIVD_CTRL**	32	31:0	**Divider D Control Register**	0x0	RW
	DIVD_CLK_SEL	3	2:0	Clock source selection for Divider D 000: LS-IRC (default) 001: HS-IRC 010: X'tal Osc 011: PLL output 100: Divider A 101: Divider B		

				110: Divider C 111: *reserved* Note: Clock switching is glitchless type		
	-	1	3	*Reserved*		
	DIVD_USED	1	4	Clock output used indicator 0: Not used, 1: at least one branch clock running		RO
	-	3	7:5	*Reserved*		
	DIVD_VAL	6	13:8	Divider value (by even number) 0x0: divided by 2 0x1: divided by 4 ... N: divided by 2x(N+1) ... 0x3F: divided by 128		
	DIVD_EN	1	14	Diver Enable		
	-	17	31:15	*Reserved*		Rsvd
0x30	**SYS_CLK_CTRL**	32	31:0	**System Clock Control Register** Note: Bus clock cannot be disabled by program	0x0	RW
	SYS_CLK_SEL	3	2:0	Clock source selection for system clock 000: LS-IRC (default) 001: HS-IRC 010: X'tal Osc 011: PLL output 100: Divider A 101: Divider B 110: Divider C 111: Divider D Note: Clock switching is glitchless type		
	-	1	3	*Reserved*		
	CPU_PRES	3	6:4	CPU clock Prescaler value (by power of 2) 000: divided by 1 (not divided) 001: divided by 2 ... N: divided by 2^N ... 111: divided by 128		
	-	1	7	*Reserved*		
	APB_PRES	3	10:8	APB clock Prescaler value (by power of 2) 000: divided by 1 (not divided) 001: divided by 2 ... N: divided by 2^N ... 111: divided by 128		
	-	6	16:11	*Reserved*		
	AHB_CLK_LPEN	1	17	AHB clock stop during (deep-)sleep		
	APB_CLK_LPEN	1	18	APB clock stop during (deep-)sleep		
	-	1	19	*Reserved*		
	SRAM_CLK_DIS	1	20	SRAM clock disable (x = 1~4; bit position - 19) Note: This is executed at the next CPU reset		
	-	3	23:21	*Reserved*		
	SRAM_CLK_LPEN	1	24	SRAM clock stop during (deep-)sleep (x = 1~4; bit position - 23)		
	-	2	26:25	*Reserved*		
	SCPMU_CLK_MODE	1	27	SysCon & PMU Clock Mode (for safety option) 0: Enable automatic clock gating, 1: Disable automatic clock gating	0b	

	BOOT_CLK_DIS	1	28	Boot Memory clock disable Note: This is executed at the next CPU reset		
	BOOT_CLK_LPEN	1	29	Boot Memory clock stop during (deep-)sleep		
	DS_CLK_MODE	2	31:30	Deep-sleep Clock Mode 00: Disable auto-switching (default) 01: Auto-switched to LS-IRC 10: Auto-switched to HS-IRC 11: Auto-switched to X'tal Osc Note: If enabled, PLL and divider is automatically disabled, and system clock and the clocks enabled during low-power mode are switched to IRC or XTAL when deep-sleep signal is occurred Note: If disabled, program must disable PLL and change output frequency before executing WFI or WFE instructions		
0x34	**SPIF_CLK_CTRL**	32	31:0	**SPI Flash Clock Control Register**	0x0	RW
	SPIF_CLK_SEL	3	2:0	Clock source selection for SPI Flash clock 000: LS-IRC (default) 001: HS-IRC 010: X'tal Osc 011: PLL output 100: Divider A 101: Divider B 110: Divider C 111: Divider D		
	-	1	3	*Reserved*		
	SPIF_CLK_EN	1	4	SPI Flash clock enable Note: If enabled, this cannot be disabled by program		
	SPIF_CLK_LPEN	1	5	SPI Flash clock stop during (deep-)sleep		
	-	26	31:6	*Reserved*		
0x38	**ST_CLK_CTRL**	32	31:0	**SysTick Clock Control Register**	0x0	RW
	-	4	3:0	*Reserved*		
	ST_CLK_PRES	2	5:4	ST clock Prescaler value (up to 8 by power of 2) 00: Not used (regarded as 01b) 01: divided by 2 10: divided by 4 11: divided by 8 Note: If un-divided clock is needed, use FCLK by disabling **ST_CLK_EN**		
	ST_CLK_EN	1	6	Reference clock enable for SysTick timer 0: Disable, 1: Enable Note: If disabled, signal for STCALIB[25] goes to HIGH which indicates that no reference clock source has been integrated. In that case, FCLK (CPU free-running clock) will be used instead of the reference clock by Cortex-M0		
	ST_CLK_SKEW	1	7	Skew indication for SysTick clock - STCALIB[24] for Cortex-M0 Integration 0: Indicates that the reference clock, or FCLK as indicated by STCLIB[25], can gurantee an exact multiple of 10ms 1: Otherwise		
	ST_CLK_CALIB	24	31:8	Calibration value of SysTick clock - STCALIB[23:0] for Cortex-M0 Integration Note: Provides and integer value to compute a 10ms (100Hz) delay from either the reference clock, or FCLK if the reference clock is not supplied		

0x3C	**TMR_CLK_CTRL**	32	31:0	**Timer Clock Control Register**	0x0	RW
	-	8	7:0	*Reserved*		
	TMRx_CLK_EN	2	9:8	Timer clock enable (x=1~2; bit position -7)		
	-	6	15:10	*Reserved*		Rsvd
	TMRx_CLK_LPEN	2	17:16	Timer clock stop during (deep-)sleep (x=1~2; bit position -15)		
	-	14	31:18	*Reserved*		Rsvd
0x40	**CLK_CTRL_LOCK**	32	31:0	**Clock Control Lock Register** [Sequence 1] Write **CLK_LOCK_EN** and **CLK_LOCK_CODE** (any value) with **CLK_LOCK_KEY** = 1b [Sequence 2] Write **CLK_LOCK_EN** and **CLK_LOCK_CODE** (of sequence 1) with **CLK_LOCK_KEY** = 0b [Sequence 3] Write **CLK_LOCK_EN** and **CLK_LOCK_CODE** (of sequence 1) with **CLK_LOCK_KEY** = 1b [Sequence 4] Read **CLK_LOCK_KEY** (optional)		RW
	CLK_LOCK_EN	1	0	Clock Register Write Protection 0: Off, 1: On		
	CLK_LOCK_CODE	15	15:1	Lock Code for Clock Register Write Protection		
	CLK_LOCK_KEY	1	16	Lock Key forClock Register Write Protection when writing: lock sequence key when reading: lock sequence status - 1b means successful		
	-	15	31:17	*Reserved*		
0x44	**PERI_CLK_SEL**	28	31:0	**Peripheral Clock Select Register** Bit fields for clock source selection has the following values: 000: LS-IRC (default) 001: HS-IRC 010: X'tal Osc 011: PLL output 100: Divider A 101: Divider B 110: Divider C 111: Divider D	0x0	RW
	-	4	3:0	*Reserved*		
	-	4	7:4	*Reserved*		
	UART_CLK_SEL	3	10:8	Clock source selection for UART clock		
	-	1	11	*Reserved*		
	SPI_CLK_SEL	3	14:12	Clock source selection for SPI clock		
	-	1	15	*Reserved*		
	SDMMC_BCLK_SEL	3	19:16	Clock source selection for the base clock of SD/MMC Host		
	-	1	20	*Reserved*		
	CLKOUT1_SEL	3	26:24	Clock source selection for CLKOUT1		
	-	1	27	*Reserved*		
	CLKOUT2_SEL	3	30:28	Clock source selection for CLKOUT2		
	-	1	31	*Reserved*		
0x48	**PERI_CLK_EN**	32	31:0	**Peripheral Clock Enable Register**	0x0	RW
	DMAC_CLK_EN	1	0	DMAC clock enable		
	-	1	1	*Reserved*	—	—
	-	1	2	*Reserved*		
	WDT_CLK_EN	1	3	Watchdog Timer clock enable Note: If enabled once, this cannot be disabled by program		
	GPIOx_CLK_EN	3	6:4	GPIO x clock enable (x = 1~3; bit position - 3)		
	-	5	11:7	*Reserved*		
	UARTx_CLK_EN	3	14:12	UART x clock enable (x= 1~3; bit position - 11)		

	-	1	15	*Reserved*		
	SPIx_CLK_EN	3	18:16	SPI x clock enable (x = 1~3; bit position - 15)		
	-	1	19	*Reserved*		
	I2Cx_CLK_EN	4	23:20	I2C x clock enable (x = 1~4; bit position - 19)		
	-	1	24	*Reserved*		
	ARIA_CLK_EN	1	25	ARIA Engine clock enable		
	SDSLV_CLK_EN	1	26	SD Slave clock enable		
	SDMMC_CLK_EN	1	27	SD/MMC Host clock enable		
	CRYP_CLK_EN	1	28	Crypto Processor clock enable		
	ACA_CLK_EN	1	29	ACA Engine clock enable		
	CLKOUT1_EN	1	30	CLKOUT1 pin clock enable		
	CLKOUT2_EN	1	31	CLKOUT2 pin clock enable		
0x4C	**PERI_CLK_LPEN**	32	31:0	**Peripheral Clock Low-Power Enable Register**	0x0	RW
	DMAC_CLK_LPEN	1	0	DMAC clock stop during (deep-)sleep		
	-	1	1	*Reserved*		
	-	2	3:2	*Reserved*		
	GPIOx_CLK_LPEN	3	6:4	GPIO x clock stop during (deep-)sleep (x = 1~3; bit position - 3)		
	-	5	11:7	*Reserved*		
	UARTx_CLK_LPEN	3	14:12	UART x clock stop during (deep-)sleep (x= 1~3; bit position - 11)		
	-	1	15	*Reserved*		
	SPIx_CLK_LPEN	3	18:16	SPI x clock stop during (deep-)sleep (x = 1~3; bit position - 15)		
	-	1	19	*Reserved*		
	I2Cx_CLK_LPEN	4	23:20	I2C x clock stop during (deep-)sleep (x = 1~4; bit position - 19)		
	-	1	24	*Reserved*		
	ARIA_CLK_LPEN	1	25	ARIA Engine clock stop during (deep-)sleep		
	SDSLV_CLK_LPEN	1	26	SD Slave clock stop during (deep-)sleep		
	SDMMC_CLK_LPEN	1	27	SD/MMC Host clock stop during (deep-)sleep		
	CRYP_CLK_LPEN	1	28	Crypto Processor clock stop during (deep-)sleep		
	ACA_CLK_LPEN	1	29	ACA Engine clock stop during (deep-)sleep		
	CLKOUT1_LPEN	1	30	CLKOUT1 pin clock stop during (deep-)sleep		
	CLKOUT2_LPEN	1	31	CLKOUT2 pin clock stop during (deep-)sleep		
0x50 ~0x5B	***reserved***	-	-	-	-	Rsvd
0x5C	**AON_CLKRST_CTRL**	32	31:0	**Always-on Domain Clock/Reset Control Register**		RW
	RTC_CLK_SRC	1	0	*RTC clock source* *0 : RTC X’tal OSC (default)*		
		1	1	*Reserved*		
	RTC_XTAL_EN	1	2	RTC X'tal clock enable		
	RTC_XTAL_BYPASS	1	3	RTC X'tal Bypass Mode enable 0: Crystal Connected (default), 1: Use external clock source instead of crystal		
	RTC_RST	1	4	RTC reset command & status		
	-	27	31:5	*Reserved*		
0x60	**RST_STATE**	32	31:0	**Reset State Register** Note: Program can know reset reason by clearing the flag before the reset event Note: Flag bits of each priority is cleared when higher priority reset is occurred	-	ROAC

	CPU_RST_FLAG	1	0	Record for the occurrence of CPU Reset		
	POR_FLAG	1	1	Record for the occurrence of Power-on Reset (Priority 0) Note: This bit is set only when PoR is released.		
	STBY_RST_FLAG	1	2	Record for the occurrence of Standby Wakeup Reset (Priority 0)		
	EXT_RST_FLAG	1	3	Record for the occurrence of External Reset (Priority 1) Note: If external reset pin is used for NMI, this bit does not work		
	WDT_RST_FLAG	1	4	Record for the occurrence of Watchdog Timeout Reset (Priority 1)		
	CPU_LOCKUP_FLAG	1	5	Record for the occurrence of CPU Lockup Reset (Priority 1) Note: If RAM boot mode by JTAG is used, CPU lockup reset is disabled		
	BOD_RST_FLAG	1	6	Record for the occurrence of BoD Reset (Priority 1)		
	-	1	7	*Reserved*		
	PERI_RST_FLAG	1	8	Record for the occurrence of Peripheral Group Reset (Priority 2)		
	MST_RST_FLAG	1	9	Record for the occurrence of Master Group Reset (Priority 3)		
	-	22	31:10	*Reserved*		Rsvd
0x64	**GRP_RST_CTRL**	32	31:0	**Group Reset Control Register**		RW
				[Reset Group] - Master Group: CPU, AHB masters (DMAC, SDMMC, CRYP, PCP, etc.) - Peripheral Group: Bus fabric, APB peripherals, GPIO, EMC, etc. Note: Group Reset is asserted by CPU reset using AIRCR bit of Cortex-M0 when Group Reset is enabled [Sequence 1] Write **GRP_RST_MODE** and **RST_LOCK_CODE** (any value) with **RST_LOCK_KEY** = 1b [Sequence 2] Write **GRP_RST_MODE** and **RST_LOCK_CODE** (of sequence 1) with **RST_LOCK_KEY** = 0b [Sequence 3] Write **GRP_RST_MODE** and **RST_LOCK_CODE** (of sequence 1) with **RST_LOCK_KEY** = 1b [Sequence 4] Read **RST_LOCK_KEY** (optional)		
	GRP_RST_MODE	2	1:0	Group Reset Mode 00: Disabled (default) 01: Master Group Reset 10: Peripheral Group Reset 11: System Reset		
	RST_LOCK_CODE	14	15:2	Lock Code for Group Reset Mode		
	RST_LOCK_KEY	1	16	Lock Key for Group Reset Mode when writing: lock sequence key when reading: lock sequence status - 1b means successful		
	-	15	31:17	*Reserved*		
0x68	**PERI_RST**	32	31:0	**Peripheral Reset Register** Meaning of read value is the following: 0: Reset pending, 1: Reset done Note: Writing 1 generate reset command, this is automatically cleared after the reset is done. (There is no need to write 0 to clear reset command)	0x0	RWAC
	DMAC_RST	1	0	DMAC reset command & status		
	-	1	1	*Reserved*		
	TMRx_RST	2	3:2	Timer reset command & status (x = 1~2; bit position - 1)		
	GPIOx_RST	3	6:4	GPIO x reset command & status (x = 1~3; bit position - 3)		
	-	5	11:7	*Reserved*		
	UARTx_RST	3	14:12	UART x reset command & status (x= 1~3; bit position - 11)		
	-	1	15	*Reserved*		

	SPIx_RST	3	18:16	SPI x reset command & status (x = 1~3; bit position - 15)		
	-	1	19	*Reserved*		
	I2Cx_RST	4	23:20	I2C x reset command & status (x = 1~4; bit position - 19)		
	-	1	24	*Reserved*		
	ARIA_RST	1	25	ARIA Engine reset command & status		
	SDSLV_RST	1	26	SD Slave reset command & status		
	SDMMC_RST	1	27	SD/MMC reset command & status		
	CRYP_RST	1	28	Crypto Processor reset command & status		
	ACA_RST	1	29	ACA Engine clock reset command & status		
	-	1	30	*Reserved*		
	SYSCON_RST	1	31	SysCon reset command & status Note: SysCon Register is cannot be reset by Peripheral Group Reset		
0x6C ~0x7B	***reserved***	-	-	-	-	Rsvd
0x7C	**PERI_ID**	32	31:0	**Clock/Reset Manager Peripheral ID Register**	HwInit	RO
	PART_NUM	12	11:0	Clock/Reset Manager Part Number	0xC00	
	CONFIG_NUM	4	15:12	Clock/Reset Manager Configuration Number	0x2	
	MAJ_REV	4	19:16	Clock/Reset Manager Major Revision		
	MIN_REV	4	23:20	Clock/Reset Manager Minor Revision		
	RSVD_NUM	8	31:24	Clock/Reset Manager Reserved Number		

SYSTEM CONTROL

1. SYSTEM CONTROL

The System Control(SysCon) takes various control related system operation in the MS500.

The SysCon include the following features:

- System Status report
- System Interrupt control and define such as RTC, Timer
- IO Pin's pull-up, pull-down, drive strength, Schmitt trigger control
- IO pin's multi-function Select
- IO voltage level select 1.8V or 3.3V

Table 14 Syscon register

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
0x00	**SYS_STS**	32	31:0	**System Status Register**		RO
	BOOT_MODE	3	2:0	Sampling data from BOOT_MODE[2:0] Pin (@Rising-edge of PoR signal) 000: ROM boot 001: UART SRAM boot 010: UART S-Flash Update 100: SPI Direct Boot Mode1(using HS-IRC, 1bit SPI) 101: SPI Direct Boot Mode2(using XTAL, Quad-SPI) 110: SPI Direct Boot Mode3(using HS-IRC, Quad-SPI) 111: SRAM1 booting Others: ROM Boot		
	-	1	3	*Reserved*		
	BOOT_SRC	3	6:4	Sampling data from eFuse address0 bit[2:0] (@Rising-edge of PoR signal) when eFuse address 0 programmed. 000(not program): selected by External Pins (Boot Mode[2:0]) 100: SPI Direct Boot Mode1 (using HS-IRC, 1bit SPI) 101: SPI Direct Boot Mode2 (using XTAL, Quad-SPI) 110: SPI Direct Boot Mode3 (using HS-IRC, Quad-SPI) 111: SRAM Boot Mode Others: ROM Boot		
	-	1	7	*Reserved*		

	LPM_WU_REC	2	9:8	Record for the reason of wakeup on low power mode 00: None (default state) 01: RTC interrupt 10: WAKEUP pin(unstable on LPM3 as external PA0 active time) 11: Reserved Note: Reading automatically clears this to default state		ROAC
	BOD_STS	1	10	Brown-out Detection Status 0: no detect 1: detected low level power		RO
	-	1	11	*Reserved*		
	SEC_BOOT	1	12	Secure Boot Mode (from NVM data)		
	-	3	15:13	*Reserved*		
	VBAT_POR	1	16	Power-on Reset Status for Battery-domain Voltage 0: Off, 1: On		RO
	-	1	17	*Reserved*		
	GK1_LOCK	1	18	Generic Key 1 lock state 0: readable 1: not readable		RO
	GK2_LOCK	1	19	Generic Key 2 lock state 0: readable 1: not readable		RO
	GK3_LOCK	1	20	Generic Key 2 lock state 0: readable 1: not readable		RO
	-	11	31:21	*Reserved*		
0x04	**SHADOW_MAP**	32	31:0	**Shadow Memory Mapping Register**		RW
				Note: The default value of the shadow pointer is 0x1FFF_E000, ensuring that the code contained in the Boot ROM is executed at reset. If SRAM boot mode is enabled by boot mode pins and NVM's system configuration, this value becomes 0x2000_0000 which is the base address of SRAM1 Note: Change of shadow memory is executed at next CPU reset		
	-	13	12:0	*Reserved*		
	SHADOW_ADDR	19	31:13	Shadow Address High 19 bits [31:13] address when accessing memory at address 0x0000_0000 Note: Write address is the next shadow address after CPU reset, read address is the current shadow address		
0x08	**SYS_CTRL**	32	31:0	**System Control Register**	32'h40000010	RW
	EXT_RST_MODE	1	0	Use Mode for External Reset Pin (RESETN) 0: Reset (default), 1: NMI		
	-	3	3:1	*Reserved*		
	BOD_DIS	1	4	Brown-out Detection Disable 0: Disable, 1: Enable	1'b1	
	BOD_MODE	1	5	Use Mode for Brown-out Detection 0: Reset (default), 1: NMI		

	BOD_LVL	1	6	Brown-out Detection Setting Value 0: no detect 1: detected low level power (clk_mgr bor reset)		
	-	1	7	*Reserved*		
	WKUP_PIN_SEL	2	9:8	Wakeup Pin Select 00: WAKEUP[0] 01: WAKEUP[1] other: No operation.		
	EXT_INT_TYPE	1	10	External Interrupt type 0: Edge-triggered, 1: Level-sensitive		
	EXT_INT_LVL	1	11	External Interrupt level 0: Low (or falling-edge), 1: High (or rising-edge)		
	EXT_INT_SEL	7	18:12	External Pin Select for Interrupt 0x00: PA[0] 0x01: PA[1] ... 0x4F: PE[15] Others: *reserved*		
	TEST_NMI	1	19	Test NMI Generation		WOAC
	BUS_TOUT	10	29:20	Number of Timeout count for Slave Ready in AHB matrix Note: Zero value disable timeout monitoring		
	CPU_LOCKUP	1	30	CM0 boot lockup disable signals 0: CM0 lockup enable 1: CM0 lockup disable(default)	1	
	-	1	31	*Reserved*		
0x09 ~0x0F	***reserved***	-	-	-	-	Rsvd
0x10	**NMI_STS**	32	31:31	**NMI Status Register**	-	ROAC
	EXT_RESET	1	31:32	External Reset Note: If **EXT_RST_MODE** is 0 (default), this is not occurred		
	WDT_TOUT	1	31:33	Watchdog Timeout Interrupt		
	BOD_SIG	1	31:34	Brown-out detection signal Note: If **BOD_MODE** is 0 (default), this is not occurred		
	CLK_FAULT	1	31:35	Clock Fault (including PLL unlock after locked)		
	EXT_INT	1	31:36	External Interrupt (selected by **EXT_INT_SEL**)		
	-	2	31:37	*Reserved*		
	TEST	1	31:38	Generated by **TEST_NMI**		
	-	24	31:39	*Reserved*		
0x14	**NMI_EN**	32	31:40	**NMI Enable Register**		RW
				Note: Each bit is corresponding to each bit of **NMI_STS** register. 1 is enabled state, thus enabled signal source is used for interrupt generation		

0x18	**SYS_INT_STS**	32	31:0	**System Interrupt Status Register**	-	ROAC
	…	8	7:0	*~ Same as the description of NMI_STS*		
	CLK_STS_INT	1	8	Clock Status Interrupt (from Clock/Reset Manager)		
	-	23	31:9	*Reserved*		
0x1C	**SYS_INT_EN**	32	31:0	**System Interrupt Enable Register**		RW
				Note: If a bit in the bits[7:0] is used in same bit position of **NMI_EN**, that bit is not writeable. Thus it can not be used as system interrupt source Note: Each bit is corresponding to each bit of **SYS_INT_STS** register. 1 is enabled state, thus enabled signal source is used for interrupt generation		
0x20	**PA_PORT_MODE**	32	31:0	**PA Port Mode Register**	32'hF4C9	RW
				Note: Reset values are 0x000000E0		
	MF_EN	16	15:0	MF Enable for port x (= bit position) 0: Disable (used as GPIO) 1: Enable (used ad MF I/O)		
	MF_TYPE	16	31:16	MF Type for port x (= bit position - 16) 0: Digital (default) 1: Analog (reserved)		
0x24	**PB_PORT_MODE**	32	31:0	**PB Port Mode Register**	32'h0000	RW
				~ Same as the description of above register		
				Note: Reset values are 0x00008000		
0x28	**PC_PORT_MODE**	32	31:0	**PC Port Mode Register**	32'h0000	RW
				~ Same as the description of above register		
				Note: Reset values are 0x00008000		
0x25 ~0x3F	***reserved***	-	-	-	-	Rsvd
0x40	**PA_DRV_STR**	32	31:0	**PA Drive Strength Register**		RW
				0: 2mA, 1: 4mA 2: 8mA 3: 12mA Note: each port's drive control is assigned by 2 bits.		
0x44	**PB_DRV_STR**	32	31:0	**PB Drive Strength Register**		RW
		20	19:0	*~ Same as the description of above register*		
		12	31:20	00: 2mA [SDMMC HOST CONTROL 2'b11(D)] 01: 4mA [SDMMC HOST CONTROL 2'b10(C)] 10: 8mA [SDMMC HOST CONTROL 2'b00(B)] 11: 12mA [SDMMC HOST CONTROL 2'b01(A)]		Rsvd
0x48	**PC_DRV_STR**	32	31:0	**PC Drive Strength Register**		RW
		10	9:0	00: 2mA [SDMMC HOST CONTROL 2'b11(D)] 01: 4mA [SDMMC HOST CONTROL 2'b10(C)] 10: 8mA [SDMMC HOST CONTROL 2'b00(B)] 11: 12mA [SDMMC HOST CONTROL 2'b01(A)]		Rsvd
		22	31:10	*~ Same as the description of above register*		
0x4C	**PD_DRV_STR**	32	31:0	**PD Drive Strength Register**		RW
			11:0	0: 2mA, 1: 4mA 2: 8mA 3: 12mA Note: each port's drive control is assigned by 2 bits.		
	-	20	31:12	*Reserved*		

0x4D ~0x5F	*reserved*	-	-	-	-	Rsvd
0x60	**PA_PUPD_EN**	32	31:0	**PA Pull-up/Pull-down Register**		RW
		1	0	Port A[0] 0: Normal 1: pull-up Enable Note:pull-up Enable and pull-down Enable A bus-holder (or Bus-keeper) is a weak latch circuit which holds last value on a tri-state bus.		
				…		
		1	15	Port A[15] 0: Normal 1: pull-up Enable Note:pull-up Enable and pull-down Enable A bus-holder (or Bus-keeper) is a weak latch circuit which holds last value on a tri-state bus.		
		1	16	Port A[16] 0: Normal 1: pull-down Enable Note:pull-up Enable and pull-down Enable A bus-holder (or Bus-keeper) is a weak latch circuit which holds last value on a tri-state bus.		
				…		
		1	31	Port A[31] 0: Normal 1: pull-down Enable Note:pull-up Enable and pull-down Enable A bus-holder (or Bus-keeper) is a weak latch circuit which holds last value on a tri-state bus.		
0x64	**PB_PUPD_EN**	32	31:0	**PB Pull-up/Pull-down Register**		RW
				~ Same as the description of above register		
0x68	**PC_PUPD_EN**	32	31:0	**PC Pull-up/Pull-down Register**		RW
				~ Same as the description of above register		
0x6C	**PD_PUPD_EN**	32	31:0	**PD Pull-up/Pull-down Register**		RW
			11:0	*~ Same as the description of above register*		
	-	20	31:12	*Reserved*		
0x6D ~0x7F	*reserved*	-	-	-	-	Rsvd
0x80	**PA_MFSEL_LO**	32	31:0	**PA Multi-Function Select Register - Low Byte** Note: Valid value of each field is 0x0~0x7		RW
	Pin0_MF_SEL	4	3:0	Multi-Function Select for Pin 0		
	Pin1_MF_SEL	4	7:4	Multi-Function Select for Pin 1		
	Pin2_MF_SEL	4	11:8	Multi-Function Select for Pin 2		
	…			…		
	Pin7_MF_SEL	4	31:28	Multi-Function Select for Pin 7		
0x84	**PA_MFSEL_HI**	32	31:0	**PA Multi-Function Select Register - High Byte** Note: Valid value of each field is 0x0~0x7		RW
	Pin8_MF_SEL	4	3:0	Multi-Function Select for Pin 8		
	…			…		
	Pin15_MF_SEL	4	31:28	Multi-Function Select for Pin 15		

0x88	**PB_MFSEL_LO**	32	31:0	**PB Multi-Function Select Register - Low Byte**		RW
				~ Same as the description of PA_MFSEL_LO		
0x8C	**PB_MFSEL_HI**	32	31:0	**PB Multi-Function Select Register - High Byte**		RW
				~ Same as the description of PA_MFSEL_HI		
0x90	**PC_MFSEL_LO**	32	31:0	**PC Multi-Function Select Register - Low Byte**		RW
				~ Same as the description of PA_MFSEL_LO		
0x94	**PC_MFSEL_HI**	32	31:0	**PC Multi-Function Select Register - High Byte**		RW
				~ Same as the description of PA_MFSEL_HI		
0x95 ~0xBF	***reserved***	-	-	-	-	Rsvd
0xC0	**PORT_LOCK**	32	31:0	**Port Write Lock Register**		RW
				[Sequence 1] Write **PORT_LOCK_EN** and **PORT_LOCK_CODE** (any value) with **PORT_LOCK_KEY** = 1 [Sequence 2] Write **PORT_LOCK_EN** and **PORT_LOCK_CODE** (of sequence 1) with **PORT_LOCK_KEY** = 0 [Sequence 3] Write **PORT_LOCK_EN** and **LOCK_CODE** (of sequence 1) with **PORT_LOCK_KEY** = 1 [Sequence 4] Read **PORT_LOCK_KEY** (optional)		
	PORT_LOCK_EN	1	0	Write Protection for Port Register 0: Disable, 1: Enable		
	PORT_LOCK_CODE	15	15:1	Lock Code for Port Write Protection		
	PORT_LOCK_KEY	1	16	Lock Key for Port Write Protection when writing: lock sequence key when reading: lock sequence status - 1 means successful		
	-	15	31:17	*Reserved*		
0xC4	**EVENT_ROUTER**	32	31:0	**Event Routing Register**		RW
	-	8	7:0	*Reserved*		
	RTC_TRIG_SEL	2	9:8	RTC Trigger Select 0x0: Disabled 0x1: Standby Wakeup 0x2: NMI 0x3: System Interrupt		
	-	6	15:10	*Reserved*		
	TMR1_TRIG_SEL	4	19:16	Timer 1 Trigger Select 0x0: Disabled 0x1: EXT_INT 0x2 - 0x7: *reserved* 0x8 - 0xC: GPIO x Interrupt (x=1~5) 0xD - 0xF: *reserved*		
	TMR2_TRIG_SEL	4	23:20	Timer 2 Trigger Select 0x0: Disabled 0x1: EXT_INT 0x2 - 0x7: *reserved* 0x8 - 0xC: GPIO x Interrupt (x=1~5) 0xD - 0xF: *reserved*		
	-	8	31:24	*Reserved*		
0xC5 ~0xCC	***reserved***	-	-	-	-	Rsvd
0xD0	**PDV_SEL18**	32	31:0	**Power domain I/O voltage select Register**		RW

	PDV_SEL18_MAIN	1	0	MAIN power domain I/O voltage select 0: 3.3V 1: 1.8V		
	PDV_SEL18_SDIO	1	1	SDIO power domain I/O voltage select 0: 3.3V 1: 1.8V		
	PDV_SEL18_SDMMC	1	2	SDMMC power domain I/O voltage select 0: 3.3V 1: 1.8V Note: If PDV_SEL18_SDMMC_CTRL is 1, this is controlled.		
	PDV_SEL18_FLASH	1	3	FLASH power domain I/O voltage select 0: 3.3V 1: 1.8V		RO
	PDV_SEL18_RTC	1	4	RTC power domain I/O voltage select 0: 3.3V 1: 1.8V		
	-	27	31:5	*Reserved*		
0xD4	**Reserved**			*Reserved*		
0xD8	**MISCELLOUNIOUS_IF**	32	31:0	**Miscellaneous Interface Register**		RW
	HSIRC_FREQ_SEL	1	0	HS_IRC frequency select 0x0: 12MHz 0x1: 6MHz		
	DMA_REQ0809_SEL	1	1	DMA Requests 9 Signal select 0: I2C2 1: SDMMC HOST		
	DMA_REQ1011_SEL	1	2	DMA Requests 10, 11 Signal select 0: I2C3,I2C4 1: ARIA Transmit, ARIA Receive		
	PDV_SEL18_SDMMC_CTRL	1	3	SDMMC CM0 or SDMMC select Control 0: SDMMC 1: CM0(SW)		
	-	28	31:4	*Reserved*		
0xDC	**PAB_SMT**	32	31:0	**GPIO A/B SMT(Schmitt trigger) Interface Register**	32'hFFFFFFFF	RW
	PA_SMT0	1	0	PA_SMT[0]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PA_SMT1	1	1	PA_SMT[1]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	…			…		
	PA_SMT15	1	15	PA_SMT[15]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PB_SMT0	1	16	PB_SMT[0]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PB_SMT1	1	17	PB_SMT[1]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	

	…			…		
	PB_SMT15	1	31	PB_SMT[15]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
0xE0	**PCD_SMT**	32	31:0	**GPIO C/D SMT(Schmitt trigger) Interface Register**	32'h3FFFFF	RW
	PC_SMT0	1	0	PC_SMT[0]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PC_SMT1	1	1	PC_SMT[1]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	…			…		
	PC_SMT15	1	15	PC_SMT[15]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PD_SMT0	1	16	PD_SMT[0]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PD_SMT1	1	17	PD_SMT[1]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PD_SMT2	1	18	PD_SMT[2]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PD_SMT3	1	19	PD_SMT[3]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PD_SMT4	1	20	PD_SMT[4]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	PD_SMT5	1	21	PD_SMT[4]: GPIO PA Port Schmitt trigger control signal 0: No hysteresis 1: Schmitt trigger enabled	1'b1	
	-	10	31:22	*Reserved*		
0xE1 ~0xEF	***Reserved***	-	-	-	-	Rsvd
0xF0 ~0xFF	**CHIP_ID**	128	127:0	**CHIP ID Register (16 characters in ASCII format)**	HwInit	RO

SERIAL INTERFACES

1. QUAD SPI FLASH INTERFACE (SPIF)

The SPI Flash Interface allows low-cost serial flash memories to be connected to the processor with little performance penalty compared to parallel flash devices with higher pin count.

The SPIF provides sufficient flexibility to be compatible with common flash devices and includes extensions to help insure compatibility with future devices.

The SPIF include the following features:

- Interfaces to serial flash memory
- Supports classic and 4-bit bidirectional serial protocols.
- Quad SPI Flash Interface with 1-, 2-, or 4-bit data at rate of up to 50MB per second.
- Supports DMA access.

Table 15 QSPI Register Description

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
0x00	**BSPI_R_ADDR**	32	31:0	**Indirect Access Mode Address or Instruction (use only indirect mode)**	0x0	RW
0x04	**BSPI_R_DATA**	32	31:0	**Indirect Access Mode Data (use only indirect mode)**	0x0	RW
0x08	**BSPI_CTRL**	17	16:0	**BSPI Access Mode Control Register**	0x3200	RW
	BSPI_TMD	2	1:0	Indirect or Direct Mode XIP select 2'h0 : single, 2'h2 : dual, 2'h3 : quad		RW
	BSPI_M_TMD	1	2	Direct Mode Fast 및 IO mode select		RW
	-	1	3	*Reserved*		RO
	BSPI_R_HOLD	1	4	Indirect Mode CSn control		RW
	-	3	7:5	*Reserved*		RO
	BSPI_M_DUMMY	2	10:8	Direct Mode QSPI Dummy Control [2] : dummy add enable, [1:0] : 2'h0 = 1byte, 2'h1 = 2byte, 2'h2 = 3byte, 2'h3 = 4byte		RW
	BSPI_M_CTINU	1	11	Direct Mode Continuous mode control and state		RW
	BSPI_P_HOLD	1	12	External HOLDn PIN control		RW
	BSPI_P_WP	1	13	External WPn PIN control		RW
	-	2	15:14	*Reserved*		RO
	BSPI_R_BUSY	1	16	Indirect Mode busy state		RO
0x0C	**BSPI_M_WBSTN**	8	7:0	**Direct Mode Burst Write number (register+1)**	0x0	RW
0x10	**BSPI_M_JEDEC**	24	23:0	**Direct Mode JEDEC ID read**	0x0	RO
0x14	**BSPI_M_STATUS**	8	7:0	**Direct Mode Status**	0x0	RW
0x18	**BSPI_M_CONFIG**	8	7:0	**Direct Mode Configuration**	0x0	RW
0x1C	**BSPI_M_1CMD**	6	5:0	**Direct Mode 1 Byte command**	0x0	WO

	BSPI_M_CERASE	1	0	Direct Mode Chip Erase		
	-	1	1	*Reserved*		
	BSPI_M_WRDIS	1	2	Direct Mode Write disable		
	BSPI_M_WREN	1	3	Direct Mode Write enable		
	BSIP_M_PDDIS	1	4	Direct Mode Power-down disable		
	BSPI_M_PDEN	1	5	Direct Mode Power-down enable		
0x20	**BSPI_M_SERASE**	24	23:0	**Direct Mode Sector Erase Address**	0x0	WO
0x24	**BSPI_M_BERASE**	24	23:0	**Direct Mode Block Erase Address**	0x0	WO
0x28	**BSPI_IST_SET0**	32	9:0	**Direct Mode Instruction set0**	0x6004FFA0	RW
	BSPI_I_CTINU	8	7:0	QSPI Continuous Mode value		
	BSPI_I_RST	8	15:8	Direct Mode Reset		
	BSPI_I_WRDIS	8	23:16	Direct Mode Write disable Instruction		
	BSPI_I_CERASE	8	31:24	Direct Mode Chip Erase Instruction		
0x2C	**BSPI_IST_SET1**	32	9:0	**Direct Mode Instruction set1**	0xEB6BD820	RW
	BSPI_I_SERASE	8	7:0	Direct Mode Sector Erase Instruction		
	BSPI_I_BERASE	8	15:8	Direct Mode Block Erase Instruction		
	BSPI_I_QFOM	8	23:16	Direct Mode Quad Output Mode Instruction		
	BSPI_I_QIOM	8	31:24	Direct Mode Quad IO Mode Instruction		
0x30	**PERI_ID**	24	23:0	**Peripheral ID Register**	0x00000415	RO
	PART_NUM	12	11:0	Identification number for the peripheral		
	CONFIG_NUM	4	15:12	Configuration option of the peripheral		
	MAJ_NUM	4	19:16	Major revision number		
	MIN_NUM	4	23:20	minor revision number		

2. SPI

The MS500 includes three SPIs master/slave interfaces that can communicate at up to 50 Mbit/s in either full-duplex or half-duplex communication modes. During a data transfer, the master always sends 4~16bits of data to the slave, and the slave always sends 4~16 bits of data to the master.

Table 16. SPI Operating Clock Mapping Frequency

SPI Name	Master SCLK Mapping	Slave SCLK Mapping	APB mapping
SPI1	Max 48MHz	Max 8MHz	Max. 96MHz
SPI2	Max 48MHz	Max 8MHz	Max. 96MHz
SPI3	N/A	Max 24MHz$^{\text{NOTE*1}}$	Max. 96MHz

Note 1: This speed is measured with Raspberry Pi3 Model board in typical test environments. That speed is somewhat influenced by whether the board state and temperature environments.

Each SPI Interface includes the following features:

- Maximum data bit rate of one eighth of the APB clock rate
- Compatible with Motorola SPI, Texas Instruments SSI, and National Semiconductor Microwire buses (SPI1 and SPI2).
- Support three chip-selects output, serial-master and serial-slave mode, and software configurable.
- Simplex synchronous transfers on two lines with or without a bidirectional data line (SPI3)
- Master mode baud rate prescaler
- DMA-based or interrupt-based operation.
- Master and slave operation.
- 8x16 FIFOs for transmit and receive.
- Provide direct control of SSEL signal by soft500ware (SPI3)
- Provide programmable loop-back mode for self-test operation.
- Interact with multiple masters and slaves on the bus.
- Support both MSB/LSB first order (SPI3)
- Separate Serial Clock from APB clock

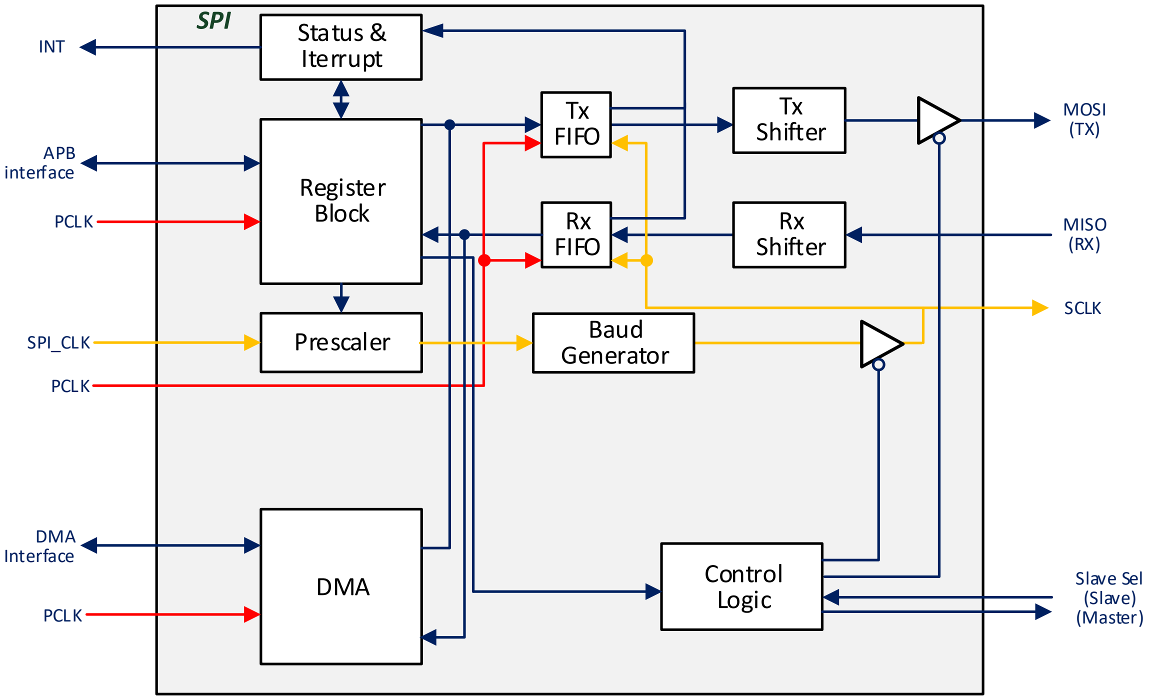

Figure 10 SPI Block Diagram

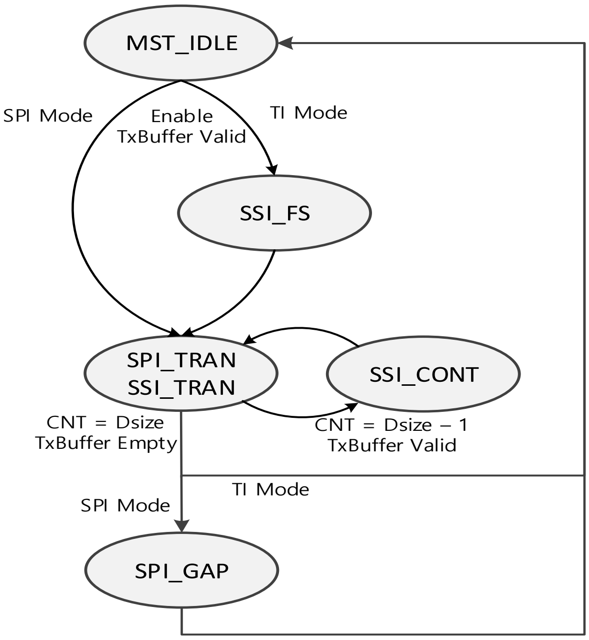

Figure 11 FSM

Table 17 SPI1, SPI2 Register summary

Offset	Symbol	Size	Brief Description	Reset Value	Type
0x000	SSPCR0	16	Control Register 0.	0x0000	RW
0x004	SSPCR1	4	Control Register 1.	0x0	RW
0x008	SSPDR	16	Data Register.	0x---	RW
0x00C	SSPSR	5	Status Register.	0x03	RO
0x010	SSPCPSR	8	Clock Prescale Register.	0x00	RW
0x014	SSPIMSC	4	Interrupt Mask Set of Clear Register.	0x0	RW
0x018	SSPRIS	4	Raw Interrupt Status Register.	0x8	RO
0x01C	SSPMIS	4	Masked Interrupt Status Register.	0x0	RO
0x020	SSPICR	4	Interrupt Clear Register.	0x0	WO
0x024	SSPDMACR	2	DMA Control Register.	0x0	RW
0x028 ~ 0x07C	-		*Reserved.*		-
0x080 ~ 0x08C	-		*Reserved for test.*		-
0x090 ~ 0xFCC	-		*Reserved.*		-
0xFD0 ~ 0xFDC	-	-	*Reserved for future expansion.*		-
0xFE0	CSSPPERIPHID0	8	Peripheral Identification Register 0.	0x22	RO
0xFE4	SSPPERIPHID1	8	Peripheral Identification Register 1.	0x10	RO

0xFE8	SSPPERIPHID2	8	Peripheral Identification Register 2.	0x24	RO
0xFFC	SSPPERIPHID3	8	Peripheral Identification Register 3.	0x00	RO

Table 18 SPI1, SPI2 Register Description

Offset	Symbol	Bit	Brief Description	Reset Value	Type
0x000	**SSPCR0**	**31:0**	**Control Register 0**	**0x00**	**RW**
	SCR	15:8	Serial Clock Rate. The value SCR is used to generate the transmit and receive bit rate of the SSP. The bit rate is: $\dfrac{F_{SSPCLK}}{CPSDVR \ \times (1+SCR)}$ Where CPSDVSR is an even value from 2-254, programmed through the SSPCPSR register and SCR is a value from 0-255.		
	SPH	7	SSPCLKOUT phase. Applicable to Motorola SPI frame format only.		
	SPO	6	SSPCLKOUT polarity. Applicable to Motorola SPI frame.		
	FRF	5:4	Frame Format. 00 = Motorola SPI frame format 01 = TI synchronous serial frame format 10 = National Micro-wire frame format 11 = Reserved, undefined operation.		
	DSS	3:0	Data Size Select. 0000 = Reserved, undefined operation 0001 = Reserved, undefined operation 0010 = Reserved, undefined operation 0011 = 4-bit data 0100 = 5-bit data 0101 = 6-bit data 0110 = 7-bit data 0111 = 8-bit data 1000 = 9-bit data 1001 = 10-bit data 1010 = 11-bit data 1011 = 12-bit data 1100 = 13-bit data 1101 = 14-bit data 1110 = 15-bit data 1111 = 16-bit data.		
0x004	**SSPCR1**	**15:0**	**Control Register 1**	**0x0**	**RW**
	-	15:4	*Reserved, read unpredictable, written as 0.*		
	SOD	3	Slave mode Output Disable. This bit is relevant only in the slave mode, MS=1. In multiple-slave systems, it is possible for an SSP master to broadcast a message to all slaves in the system while ensuring that only one slave drives data onto its serial output line. In such systems that the RXD lines from multiple slaves could be tied together. To operate in such systems, the SOD bit can be set if the SSP slave is not supposed to drive the SSPTXD line: 0 = SSP can drive the SSPTXD output in slave mode. 1 = SSP must not drive the SSPTXD output in slave mode.		
	MS	2	Master or Slave mode Select. This bit can be modified only when the SSP is disabled, SSE=0: 0 = Device configured as master, default. 1 = Device configured as slave.		
	SSE	1	Synchronous Serial port Enable:		

			0= SSP operation disabled. 1 = SSP operation enabled.		
	LBM	0	Loop back mode: 0 = Normal serial port operation enabled. 1 = Output of transmit serial shifter is connected to input of receive serial shifter internally.		
0x008	**SSPDR**	**15:0**	**SSP Data Register**	**0x0**	**RW**
	DATA		Transmit/Receive FIFO: Read = Receive FIFO. Write = Transmit FIFO. Data must be right-justified when the SSP is programmed for a data size that is less than 16 bits. Unused bits at the top are ignored by transmit logic. The received logic automatically right-justified.		
0x00C	**SSPSR**	**15:0**	**SSP Status Register**	**0x0**	**RO**
	-	15:5	*Reserved, read unpredictable, written as 0.*	-	-
	BSY	4	SSP Busy Flag: 0 = SSP is idle. 1 = SSP is currently transmitting and/or receiving a frame or the transmit FIFO is not empty.		
	RFF	3	Receive FIFO Full: 0 = Receive FIFO is not full. 1 = Receive FIFO is full.		
	RNE	2	Receive FIFO Not Empty: 0 = Receive FIFO is empty. 1 = Receive FIFO is not empty.		
	TNF	1	Transmit FIFO Not Full: 0 = Transmit FIFO is full. 1 = Transmit FIFO is not full.		
	TFE	0	Transmit FIFO Empty: 0 = Transmit FIFO is not empty. 1 = Transmit FIFO is empty.		
0x010	**SSPCPSR**	**15:0**	**Clock Prescale Register and Specifies the Division Factor by input SSPCLK**	**0x00**	**RW**
	-	15:8	Reserved, read unpredictable, must be written as 0.		
	CPSDVSR	7:0	Clock Prescale Divisor. Must be an even number from 2-254, depending on the frequency of SSPCLK. The least significant bit always returns zero on reads.		
0x014	**SSPIMSC**	**15:0**	**Interrupt Mask Clear**	**0x00**	**RW**
	-	15:4	*Reserved, read as zero, do not modify.*	-	-
	TXIM	3	Transmit FIFO Interrupt Mask: 0 = Transmit FIFO half-empty or less condition interrupt is masked. 1 = Transmit FIFO half-empty or less condition interrupt is not masked.		
	RXIM	2	Receive FIFO Interrupt Mask: 0 = Receive FIFO half-full or less condition interrupt is masked. 1 = Receive FIFO half-full or less condition interrupt is not masked.		
	RTIM	1	Receive timeout Interrupt Mask: 0 = Receive FIFO not empty and no read prior to timeout period interrupt is masked. 1 = Receive FIFO not empty and no read prior to timeout period interrupt is not masked.		
	RORIM	0	Receive Overrun Interrupt Mask: 0 = Receive FIFO written to while full condition interrupt is masked. 1 = Receive FIFO written to while full condition interrupt is not masked.		
0x018	**SSPMIS**	**15:0**	**Interrupt Mask Status**	**0x00**	**RO**
	-	15:4	*Reserved, read as zero, do not modify.*	-	-

	TXMIS	3	Gives the Transmit FIFO masked interrupt state, after masking, of the SSPTXINTR interrupt.		
	RXMIS	2	Gives the receive FIFO masked interrupt state, after masking, of the SSPRXINTR interrupt.		
	RTMIS	1	Gives the receive timeout masked interrupt state, after masking, of the SSPRTINTR interrupt.		
	RORMIS	0	Gives the receive over run masked interrupt status, after masking, of the SSPRORINTR interrupt.		
0x020	**SSPICR**	**15:0**	**Interrupt Clear Register and write-only**	**0x00**	**WO**
	-	15:2	*Reserved, read as zero, do not modify.*	-	-
	RTIC	1	Clears the SSPRTINTR interrupt.		
	RORIC	0	Clears the SSPRORINTR interrupt.		
0x024	**SSPDMACR**	**15:0**	**DMA Control Register**	**0x00**	**RW**
	-	15:2	*Reserved, read as zero, do not modify.*	-	-
	TXDMAE	1	Transmit DMA Enable. If this bit is set to 1, DMA for the transmit FIFO is enabled.	0x0	RW
	RXDMAE	0	Receive DMA Enable. If this bit is set to 1, DMA for the receive FIFO is enabled.	0x0	RW
0xFE0	SSPPERIPHID0	15:8	*Reserved, read undefined, must read as zero.*	-	-
	PARTNUMBER0	7:0	These bits read back as 0x22.	0x22	RO
0x0FE4	SSPPERIPHID1	15:8	*Reserved, read undefined, must read as zero.*	-	-
	DESIGNER0	7:4	These bits read back as 0x1.	0x1	RO
	PARTNUMBER1	3:0	These bits read back as 0x0.	0x0	RO
0xFE8	SSPPERIPHID2	15:8	*Reserved, read undefined, must read as zero.*	-	-
	REVISION	7:4	These bits return the peripheral revision.	4’h2	RO
	DESIGNER1	3:0	These bits read back as 0x4.	2’h4	RO
0xFEC	SSPPERIPHID3	15:8	*Reserved, read undefined, must read as zero.*	-	-
	CONFIGURATION	7:0	These bits read back as 0x00.	8’h00	RO
0xFF0	SSPCELLID0	15:8	*Reserved, read undefined, must read as zero.*	-	-
	SSPCELLID0	7:0	These bits read back as 0x0D.	8’h0D	RO
0xFF4	SSPCELLID1	15:8	*Reserved, read undefined, must read as zero.*	-	-
	SSPCELLID1	7:0	These bits read back as 0xF0.	8’hF0	RO
0xFF8	SSPCELLID2	15:8	*Reserved, read undefined, must read as zero.*	-	-
	SSPCELLID2	7:0	These bits read back as 0x05.	8’h05	RO
0xFFC	SSPCELLID3	15:8	*Reserved, read undefined, must read as zero.*	-	-
	SSPCELLID3	7:0	These bits read back as 0xB1.	8’hB1	RO

Table 19 SPI3 Controller Register Description

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
Note 1: Register offset is aligned with 32-bits word boundary Note 2: If SPI controller is set to slave mode, it uses SCK (from master) only as serial clock, so internal SPI clock is not used Note 3: The minimum frequency of APB clock is the followings: **Frequency of SCK x minimum APB cycle** (typ.~2) **/ Bits of Data Size** (4~16) Note 4: Mode 3 of SPI flash is sported by CLK_POL=1 and CLK_PHASE=1. Mode 0 of SPI flash is supported by CLK_POL=0, CLK_PHASE=0 and SW-controlled SSEL mode because mode 0 of SPI flash is not exactly compatible with Motorola SPI mode Note 5: Data can be written to transmit FIFO prior to enabling SPI controller and the interrupts						
0x00	**SPI_CTRL**	16	15:0	**SPI Control Register**		RW
	SPI_EN	1	0	SPI Enable 0: Disable (default), 1: Enable		
	MST_SLV	1	1	SPI Master/Slave Selection 0: Master configuration (default) 1: Slave configuration		
	FRM_FORMAT	2	3:2	SPI Frame Format 00: Motorola Mode 01: TI Mode		

				10: *reserved (for National Microwire Mode)* Others: *reserved*		
	CLK_POL	1	4	SPI Clock Polarity 0: Clock goes to LOW when idle 1: Clock goes to HIGH when idle		
	CLK_PHASE	1	5	SPI Clock Phase 0: The first clock transition is the first data capture edge (The transition away from the inter-frame state of the clock) 1: The second clock transition is the first data capture edge (The transition back to the inter-frame state of the clock)		
	DATA_LINE	1	6	Data Line Mode 0: 2-line unidirectional data mode (default) 1: 1-line bidirectional data mode Note: In 1-line bidirectional mode, master use MOSI pin while slave use MISO pin		
	SDO_EN	1	7	SPI Data Output Enable 0: Output enabled in slave mode or 1-line bidirectional mode (default) 1: Output disabled in slave mode or 1-line bidirectional mode Note: 2-line unidirectional master mode is not affected by this bit		
	DATA_SIZE	4	11:8	SPI Data Size 0x0-0x2: Not used 0x3: 4 bits ... N: N+1 bits ... 0xF: 16 bits	0x7	
	BAUD_RATE	4	15:12	SPI Baud Rate 0x0: Freq. of prescaled clock / 2 0x1: Freq. of prescaled clock / 4 ... N: Freq. of prescaled clock / 2^(N+1) ... 0x9: Freq. of prescaled clock / 1024 Others: *reserved*		
0x04	**SPI_EXT_CTRL**	16	15:0	**SPI Extended Control Register**		RW
	SSEL_DIS	1	0	Slave Select Output Disable in master configuration 0: Enable (default), 1: Disable		
	SSEL_CTRL	1	1	Slave Select Control Mode 0: HW, 1: SW		
	SSEL_ASSERT	1	2	Slave Select Assertion for SW control 0: De-assert to HIGH in active low signaling (default) 1: Assert to LOW in active low signaling		
	-	1	3	*Reserved*		Rsvd
	BIT_ORDER	1	4	SPI Bit Order for serialization 0: MSB first (default), 1: LSB first		
	LOOPBK_MODE	1	5	Loop-back Mode 0: Normal operation 1: Output of transmit serial shifter is connected to input of receive serial shifter internally		
	TX_DMA_EN	1	6	Transmit DMA Enable 0: Disable (default), 1: Enable		

	RX_DMA_EN	1	7	Receive DMA Enable 0: Disable (default), 1: Enable		
	CLK_PRESC	7	14:8	Prescale Value for Base Clock 0x00: divided by 1 (not divided) 0x01: divided by 2 … N: divided by N+1 … 0x3F: divided by 64 Others: *reserved*		
	-	1	15	*Reserved*		
0x08 (Read)	**SPI_STATE**	16	15:0	**SPI State Register**		ROC
	TX_FIFO_RDY	1	0	Transmit FIFO Write Ready (i.e. Not Full)		
	RX_FIFO_RDY	1	1	Receive FIFO Read Ready (i.e. Not Empty)		
	TX_FIFO_EMPTY	1	2	Transmit FIFO Empty		
	RX_FIFO_FULL	1	3	Receive FIFO Full		
	SPI_BUSY	1	4	SPI Busy Flag 0: Not busy, 1: Busy in communication or transmit buffer is not empty		
	-	11	15:5	*Reserved*		
0x08 (Write)	**SPI_CTRL_CMD**	16	15:0	**SPI Control Command Register**		WOAC
	CLR_SDO_EN	1	0	Clear **SDO_EN** bit in the **SPI_CTRL** register		
	SET_SDO_EN	1	1	Set **SDO_EN** bit in the **SPI_CTRL** register		
	CLR_SSEL_DIS	1	2	Clear **SSEL_DIS** bit in the **SPI_EXT_CTRL** register		
	SET_SSEL_DIS	1	3	Set **SSEL_DIS** bit in the **SPI_EXT_CTRL** register		
	CLR_SSEL_ASSERT	1	4	Clear **SSEL_ASSERT** bit in the **SPI_EXT_CTRL** register		
	SET_SSEL_ASSERT	1	5	Set **SSEL_ASSERT** bit in the **SPI_EXT_CTRL** register		
	-	10	15:6	*Reserved*		
0x0C	**SPI_DATA**	16	15:0	**SPI Data Port Register** [Write] software can write data to be sent in a future frame to this register whenever the **TX_FIFO_RDY** bit in the **SPI_STATE** register is 1, indicating that the TX FIFO is not full. If the TX FIFO was previously empty and the SPI controller is not busy on the bus, transmission of the data will begin immediately. Otherwise the data written to this register will be sent as soon as all previous data has been sent (and received). If the data size is less than 16 bits, software must right-justify the data written to this register. [Read] software can read data from this register whenever the **RX_FIFO_RDY** bit in the **SPI_STATE** register is 1, indicating that the RX FIFO is not empty. When software reads this register, the SPI controller returns data from the least recent frame in the RX FIFO. If the data size is less than 16 bits, the data is right-justified in this field with higher order bits filled with 0s.		WO/RO
0x10~ 0x1F	***reserved***					
0x20	**RAW_INT_STS**	16	15:0	**Raw Interrupt Status Register**		ROAC
	TX_FIFO_HE	1	0	TX FIFO at least Half-Empty ~ There are 4 or fewer valid entries in the transmit FIFO Note: This interrupt occurs only when the state is changed to be half-empty		

	RX_FIFO_HF	1	1	RX FIFO at least Half-Full ~ There are 4 or more valid entries in the receive FIFO Note: This interrupt occurs only when the state is changed to be half-full		
	-	2	3:2	*Reserved*		
	RX_OVR	1	4	RX FIFO Overrun Error Receive FIFO written to while full condition		Rsvd
	RX_TOUT	1	5	RX Read Time-out Error Receive FIFO not empty and no read prior to timeout period (serial 32 bits cycle)		
	-	1	6	*Reserved (for Mode Fault)*		Rsvd
	-	1	7	*Reserved (for Frame Error)*		Rsvd
	-	8	15:8	*Reserved*		Rsvd
0x24	**INT_EN**	16	15:0	**Interrupt Enable Register** Note: Each bit is corresponding to each bit of **RAW_INT_STS** register. High(1) is enabled state, thus the enabled signal source is used for interrupt generation		RW
0x28 (Read)	**INT_STS**	16	15:0	**Interrupt Status Register** Note: The only bit enabled by **INT_EN** can be set from **RAW_INT_STS**, otherwise it will be read as zero	-	RW1C
0x28 (Write)	**INT_CLEAR**	16	15:0	**Interrupt Clear Register** Note: Each bit is corresponding to each bit of **INT_STS** register. Writing High(1) to the specific bit will clear interrupt status of the bit position		
0x2C~ 0x37	***reserved***					
0x38	**PERI_ID0**	16	15:0	**SPI Peripheral ID0 Register**	HwInit	RO
	PART_NUM	12	11:0	SPI Part Number	0x410	
	CONFIG_NUM	4	15:12	SPI Configuration Number		
0x3C	**PERI_ID1**	16	15:0	**SPI Peripheral ID1 Register**	HwInit	RO
	MAJ_REV	4	3:0	SPI Major Revision		
	MIN_REV	4	7:4	SPI Minor Revision		
	RSVD_NUM	8	15:8	SPI Reserved Number		

3. I²C

The MS500 includes four I²C modules. The I²C-bus is bidirectional interface for inter-IC control communication using only two wires: a Serial Clock line (SCL) and a Serial Data line (SDA). Each device is recognized by a unique address and can operate as either a receiver-only device or as a transmitter with the capability to both receive and send information.

Transmitters and/or receivers can operate in either master or slave mode, depending on whether the device is initiating data transfer or if it is only being addressed. The I²C is a multi-master bus and can be controlled by more than one bus master connected to it.

The I²C modules consists of the following features:

- Standard I²C-compliant bus interfaces may be configured as a Master or a Slave
- Supports both transmitting and receiving data as a master or a slave

- Supports simultaneous master and slave operation
- Arbitration is handled between simultaneously transmitting masters without corruption of serial data on the bus.
- Programmable clock allows adjustment of I^{2}C transfer rates.
- Data transfer is bidirectional between masters and slaves.
- Serial clock synchronization allows devices with different bit rates to communicate via one serial bus.
- Serial clock synchronization is used as a handshake mechanism to suspend and resume serial transfer.
- Supports Fast-mode Plus with data rates of up to 400kbit/s.
- Optional recognition of up to four distinct slave addresses.
- Monitor mode allows observing all I^{2}C-bus traffic, regardless of slave address.
- I^{2}C-bus can be used for test and diagnostic purposes.
- The I^{2}C-bus contains a standard I^{2}C-compliant bus interface with two pins.

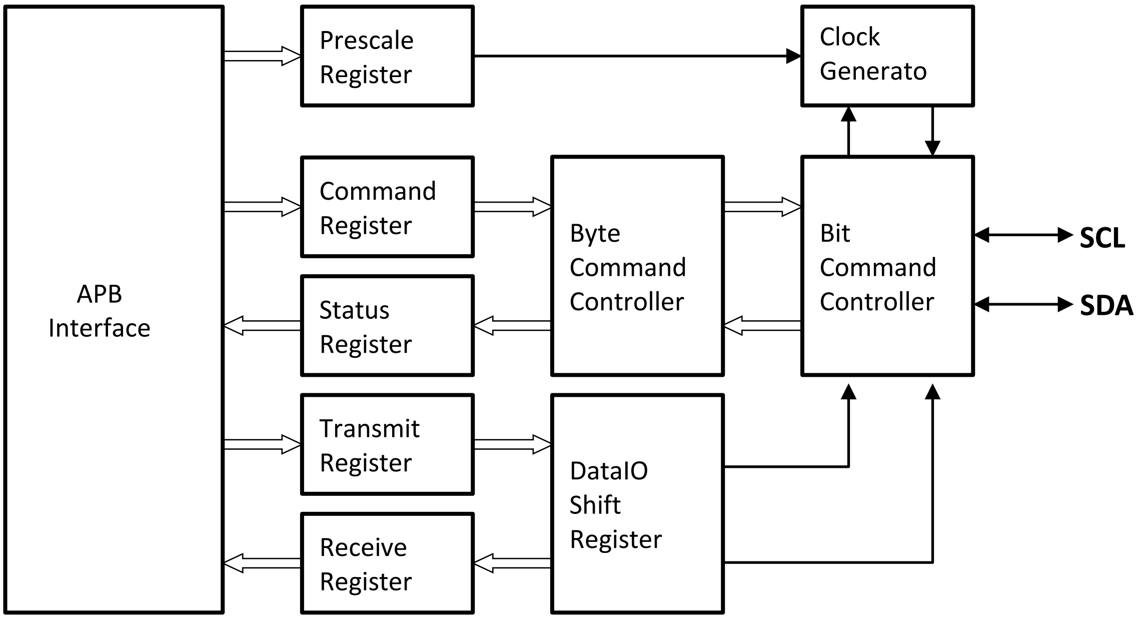

Figure 12 I^{2}C Block Diagram

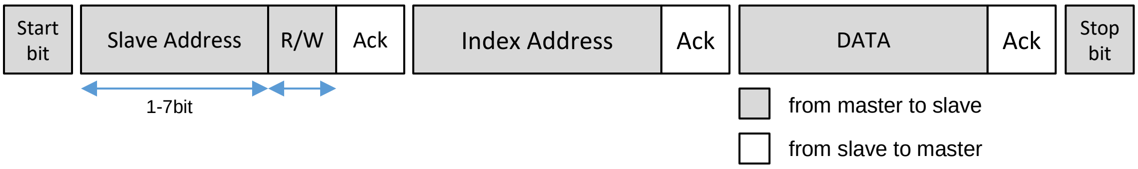

Figure 13 I2C Data transfer

The external I^{2}C mater should send to MS500's I^{2}C slave as below procedure (Figure 14)

1) Method of transmitting data to an I^{2}C slave from an I^{2}C master shall proceed in the following order.
2) The master sends the slave address and waits for an ACK from the slave.
3) After receiving the ACK, the master will send a 1byte of index address and wait for an ACK from the slave. This index address can be set to a temporary value.
4) After receiving the ACK, the master will send a 1byte of data and wait for an ACK from the slave

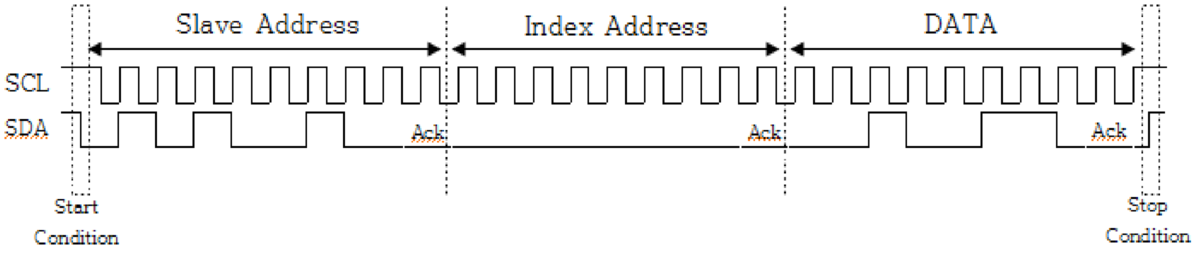

Figure 14 The following waveform shows a typical I2C master transfer

Table 20 I^{2}C Register Description

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
0x00	**I2C_PRES_LO**	8	7:0	I2C Clock Prescaler Register Low-byte [MASTER mode only]	0xFF	RW
0x04	**I2C_PRES_HI**	8	7:0	I2C Clock Prescaler Register High-byte [MASTER mode only]	0xFF	RW
				Note: This register is used to prescale the SCL clock line. Due to the structure of the I2C interface, the core uses a 5*SCL clock internally.		
0x08	**I2C_CTRL**	8	7:0	I2C Control Register	0x00	RW
	I2C_RST	1	0	I2C soft reset bit. When set to '1', soft reset is activated. When set to '0', soft reset is deactivated.		
	SLV_MODE	1	1	I2C core select bit. When set to '1', slave core is enabled. When set to '0', master core is enabled.		

	TRAN_DIR	1	2	I2C DMA Transfer Direction Type select bit. When set to ‘1’, Peripheral-to-Peripheral type. (p2p) When set to ‘0’, Memory-to-Peripheral or Peripheral-to-Memory type. (m2p/p2m)		
	-	*4*	*6:3*	*Reserved*		
	I2C_EN	1	7	I2C core enable bit. When set to ‘1’, the core is enabled. When set to ‘0’, the core is disabled.		
0x0C	**I2C_TX**	8	7:0	I2C Transmit Register	0x00	WO
				Next byte to transmit via I2C		
				In case of a slave address transfer, Bit[0] represents the RW bit. ‘1’ = reading from slave ‘0’ = writing to slave		
0x0C	**I2C_RX**	8	7:0	I2C Receive Register	0x00	RO
				Last byte received via I2C		
0x10	**I2C_CMD**	8	7:0	I2C Command Register [MASTER mode only]	0x00	WO
(Write)	-	3	2:0	Reserved		
	SEND_ACK	1	3	when a receiver, sent ACK (ACK = ‘0’) or NACK (ACK = ‘1’)		
	WRITE	1	4	write to slave		
	READ	1	5	read from slave		
	STOP	1	6	generate stop condition		
	REP_START	1	7	generate (repeated) start condition		
0x10	**I2C_STATE**	8	7:0	I2C State Register [MASTER mode only]	0x00	RO
(Read)	-	1	0	Reserved		
	TRAN_IN_PROG	1	1	Transfer in progress. ‘1’ when transferring data ‘0’ when transfer operation is idle		
	-	4	5:2	Reserved		
	BUS_BUSY	1	6	I2C bus busy ‘1’ after START signal detected ‘0’ after STOP signal detected		
	Rx_ACK	1	7	Received acknowledge from slave. This flag represents acknowledge from the addressed slave. ‘1’ = No acknowledge received ‘0’ = Acknowledge received		RO
0x14	**I2C_DEV_ID**	8	7:0	I2C Device ID Register [SLAVE mode only]	0x00	RW
0x18	**I2C_TX_DMA**	8	7:0	I2C Transmit DMA Control Register	0x00	RWAC
	TX_DMA_SIZE	7	6:0	Transmit DMA Size "0x01"~"0x7F" when master "0x00" when slave		
	TX_DMA_EN	1	7	I2C core Tx DMA enable bit. When set to ‘1’, Tx DMA is enabled. When set to ‘0’, Tx DMA is disabled.		
0x1C	**I2C_RX_DMA**	8	7:0	I2C Receive DMA Control Register	0x00	RWAC
	RX_DMA_SIZE	7	6:0	Receive DMA Size "0x01"~"0x7F" when master "0x00" when slave		

	RX_DMA_EN	1	7	I2C core Rx DMA enable bit. When set to '1', Rx DMA is enabled. When set to '0', Rx DMA is disabled.		
0x20	**I2C_STS**	8	7:0	I2C Status Register	0x00	ROAC
	TRAN_DONE	1	0	Transfer complete (Non-DMA single byte or DMA transfer)		
	ARB_LOST	1	1	Arbitration lost [MASTER mode only] This bit is set when the core lost arbitration. Arbitration is lost when: • a STOP signal is detected, but non requested • The master drives SDA high, but SDA is low.		
	-	6	7:2	Reserved		
0x24	**I2C_INT_EN**	8	7:0	I2C Interrupt Enable Register	0x0	RW
				Note: Each bit is corresponding to each bit of I2C_STS register. 1b is enabled state, thus enabled signal source is used for interrupt generation		
	TRAN_INT_EN	1	0	Transfer Interrupt Enable (Non-DMA single byte or DMA transfer)		
	ARB_INT_EN	1	1	Arbitration lost Interrupt Enable		
0x28 (Read)	**I2C_INT_STS**	8	7:0	I2C Interrupt Status Register	-	RO
				Note: The only bit enabled by I2C_INT_EN can be set from I2C_STS, otherwise it will be read as zero		
	TRAN_INT_STS	1	0	Transfer Interrupt Status (Non-DMA single byte or DMA transfer)		
	ARB_INT_STS	1	1	Arbitration lost Interrupt Status		
	-	7	7:2	Reserved		
0x28 (Write)	**I2C_INT_CLR**	8	7:0	I2C Interrupt Clear Register		WOAC
				Note: Each bit is corresponding to each bit of I2C_INT_STS register. Writing 1b to a bit clear the bit, which of I2C_INT_STS is in same position		
	TRAN_INT_CLR	1	0	Transfer Interrupt Clear (Non-DMA single byte or DMA transfer)		
	ARB_INT_CLR	1	1	Arbitration lost Interrupt Clear		
	-	7	7:2	Reserved		
0x30	**PERI_ID0**	8	7:0	I2C Peripheral ID0 Register	HwInit	RO
				Lower 8 bits of I2C Part Number	0x20	
0x34	**PERI_ID1**	8	7:0	I2C Peripheral ID1 Register	HwInit	RO
	PART_NUM_HI	4	3:0	Upper 4 bits of I2C Part Number	0x4	
	CONFIG_NUM	4	7:4	I2C Configuration Number		
0x38	**PERI_ID2**	8	7:0	I2C Peripheral ID2 Register	HwInit	RO
	MAJ_REV	4	3:0	I2C Major Revision Number		
	MIN_REV	4	7:4	I2C Minor Revision Number		
0x3C	**PERI_ID3**	8	7:0	I2C Peripheral ID3 Register	HwInit	RO

4. UART

The MS500 includes three UARTs that connect to the APB interface providing a communications link between the CPU or host and the UART.

The UARTs include the following features:

- Data sizes of 5, 6, 7, and 8 bits.
- Register locations conform to industry-standard 16C550.
- 1, 1.5, or 2 stop bit generation.
- Parity generation and detection; even, odd, stick or no-parity bit.
- Line break generation and detection.
- Programmable baud rate generator. This enables division of the reference clock by (1x16) to (65535 x16) and generates an internal x16 clock. The divisor can be a fractional number enabling you to use any clock with a frequency >3.6864MHz as the reference clock.
- Provide programmable loop-back mode for self-test operation.
- Separate 16x8 transit and 16x11 receive FIFOs to reduce CPU interrupt.
- Support for DMA.
- Support of the modem control functions CTS, DCD, DSR, RTS, DTR, and RI.

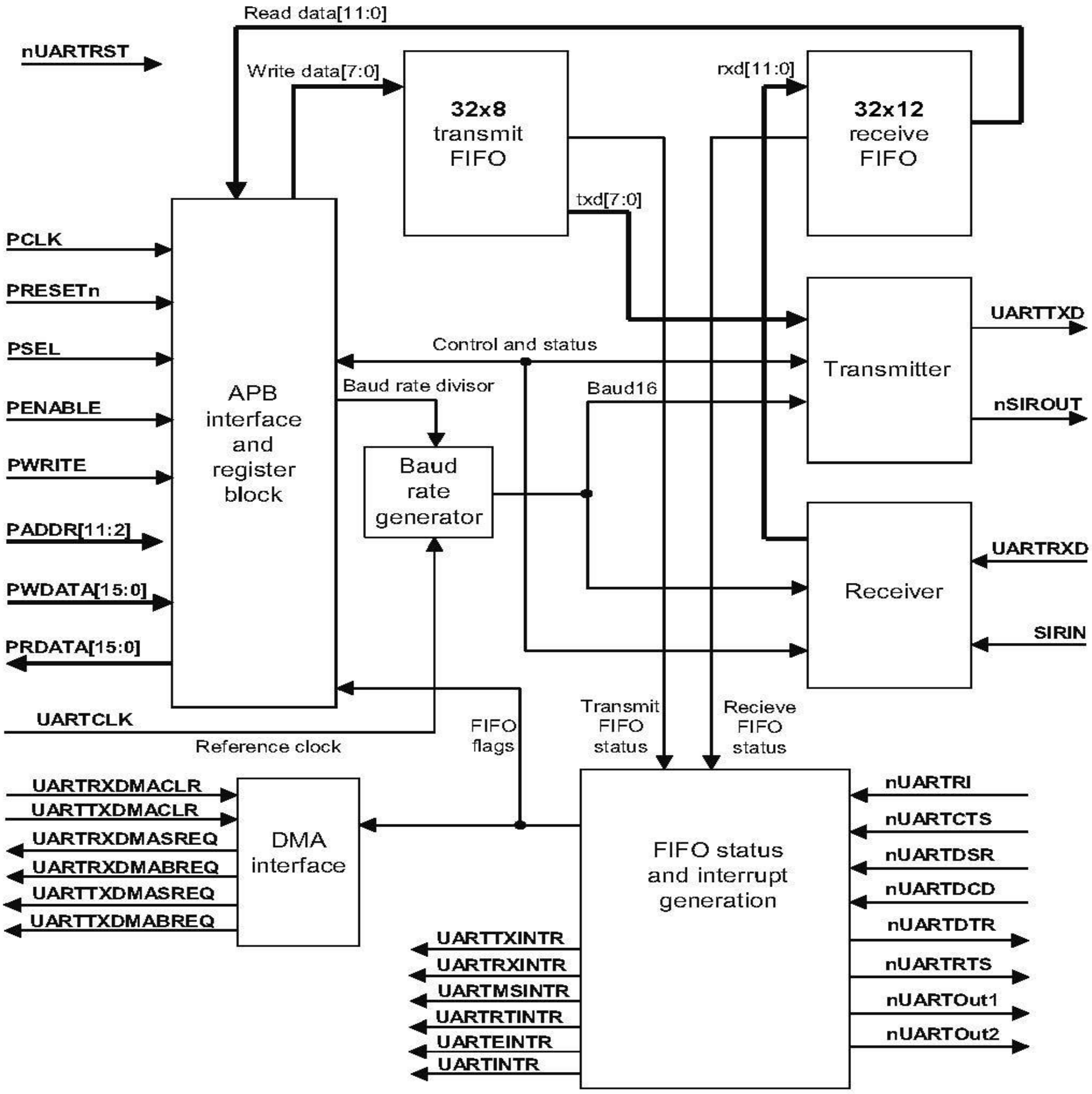

Figure 15 UART Block Diagram

Table 21 UART memory map

Offset	Symbol	Size	Brief Description	Reset Value	Type
0x000	UARTDR	12/8	Data Register.	0x---	RW
0x004	UARTRSR/UARTECR	4/0	Receive Status Register/Error Clear Register.	0x0	RW
0x008 ~ 0x014	-	-	*Reserved.*	-	-
0x018	UARTFR	9	Flag Register.	0b-10010---	RO
0x01C	-	-	*Reserved.*	-	
0x020	-	-	*Reserved.*	-	-
0x024	UARTIBRD	16	Integer Baud Rate Register.	0x0000	RW
0x028	UARTFBRD	6	Fractional Baud Rate Register.	0x00	RW
0x02C	UARTLCR_H	8	Line Control Register.	0x00	RW
0x030	UARTCR	16	Control Register.	0x0300	RW
0x034	UARTIFLS	6	Interrupt FIFO Level Select Register.	0x012	RW

0x038	UARTIMSC	11	Interrupt Mask Set/Clear Register.	0x000	RW
0x03C	UARTRIS	11	Raw Interrupt Status Register.	0x00-	RO
0x040	UARTMIS	11	Masked Interrupt Status Register.	0x00-	RO
0x044	UARTICR	11	Interrupt Clear Register.	-	WO
0x048	UARTDMACR	3	DMA Control Register.	0x00	RW
0x04C~ 0x07C	-	-	*Reserved.*	-	-

Table 22 UART Register Description

Offset	Symbol	Bit	Brief Description	Reset Value	Type
0x000	**UARTDR**	**15:0**	**UART Data Register**	**0x00**	**RW**
	-	15:12	*Reserved.*	-	-
	OE	11	Overrun Error. This bit is set to 1 if data is received and the receive FIFO is already full. This is cleared to 0 once there is an empty space in the FIFO and a new character can be written to it.		
	BE	10	Break Error. This bit is set to 1 if a break condition has been detected. This indicates that the received data input is held LOW for longer than a full-word transmission time (defined as start, data, parity, and stop bits). In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the received data input goes to a 1 (marking stat), and the next valid start bit is received.		
	PE	9	Parity Error. When this bit is set to 1, it indicates that the parity of the received data character does not match the parity selected as defined by bits 2 and 7 of the UARTLCR_H register. In FIFO mode, this error is associated with the character at the top of the FIFO.		
	FE	8	Framing Error. When this bit is set to 1, it indicates that the received character does not have a valid stop bit (a valid stop bit is 1). In FIFO mode, this error is associated with the character at the top of the FIFO.		
	DATA	7:0	Receive (read) data character and Transmit (write) data character.		
0x004	UARTRSR / UARTECR	7:0	A write to this register clears the framing, parity, break, and overrun errors. The data value is not important.	0x0	RW
	-	7:4	*Reserved. Unpredictable when read.*	-	-
	OE	3	Overrun Error. This bit is set to 1 if data is received and the FIFO is already full. This bit is cleared to 0 by a write to UARTECR. *(…continued in the next page.)* *(…)* The FIFO connects remain valid since no further data is written when the FIFO is full. Only the contents shift register are overwritten. The CPU must now read the data in order to empty the FIFO.		

	BE	2	Break Error. This bit is set to 1 if a break condition has been detected. This indicates that the received data input is held LOW for longer than a full-word transmission time (defined as start, data, parity, and stop bits). This bit is cleared to 0 after a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO. When a break occurs, only one 0 character is loaded into the FIFO. The next character is only enabled after the receive data input goes to a 1 (marking state), and the next valid start bit is received.		
	PE	1	Parity Error. When this is set to 1, it indicates that the parity of the received data character does not match the parity selected as defined by bits 2 and 7 of the UARTLCL_H register. This bit is cleared to by a write to UARTECR. In FIFO mode, this error is associated with the character at the top of the FIFO.		
	FE	0	Framing Error. When this bit is set to 1, it indicates that the received character does not have valid stop bit (a valid stop bit is 1). This bit is cleared to 0 by a write to UARTECT. In FIFO mode, this error is associated with the character at the top of the FIFO.		
0x018	**UARTFR**	**15:0**	**UART Flag Register**	**0x00**	**RO**
	-	15:9	*Reserved, do not modify, read as zero.*		
	RI	8	Ring Indicator. This bit is the complement of the UART ring indicator, nUARTRI, modem status input. That is, the bit is 1 when nUARTRI is LOW.		
	TXFE	7	Transmit FIFO Empty. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the transmit holding register is empty. If the FIFO is enabled, the TXFE bit is set when the transmit FIFO is empty. This bit does not indicate if there is data in the transmit shift register.		
	RXFF	6	Receive FIFO Full. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the receive holding register is full. If the FIFO is enabled, the RXFF bit is set when the receive FIFO is full.		
	TXFF	5	Transmit FIFO Full. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the transmit holding register is full. If the FIFO is enabled, the TXFF bit is set when the transmit FIFO is full.		
	RXFE	4	Receive FIFO Empty. Receive FIFO empty. The meaning of this bit depends on the state of the FEN bit in the UARTLCR_H register. If the FIFO is disabled, this bit is set when the receive holding register is empty. If the FIFO is enabled, the RXFE bit is set when the receive FIFO is empty.		

	BUSY	3	UART Busy. If this bit is set to 1, the UART is busy transmitting data. This bit remains set until the complete byte, including all the stop bits, has been sent from the shift register. This bit is set as soon as the transmit FIFO becomes non-empty (regardless of whether the UART is enabled or not).		
	DCD	2	Data Carrier Detect. This bit is the complement of the UART data carrier detect (nUARTDCD) modem status input. That is, the bit is 1 when the modem status input is 0.		
	DSR	1	Data Set Ready. This bit is the complement of the UART data set ready (nUARTDSR) modem status input. That is, the bit is 1 when the modem status input is 0.		
	CTS	0	Clear to Send. This bit is the complement of the UART clear to send (nUARTCTS) modem status input. That is, the bit is 1 when the modem status input is 0.		
0x020	UARTILPR	7:0	8-bit-low-power Divisor Value. These bits are cleared to 0 at reset.	0x00	RW
	ILPDVSR				
0x024	UARTIBRD	15:0	The Integer Baud Rate Divisor. These bits are cleared to 0 on reset.	0x00	RW
	BAUD_DIVINT				
0x028	UARTFBRD	5:0	The Fractional Baud Rate Divisor. These bits are cleared to 0 on reset.	0x00	RW
	BAUD_DIBFRAC				
0x02C	**UARTLCR_H**	**15:0**	**Line Control Register.**	**0x00**	**RW**
	-	15:8	*Reserved, do not modify, read as zero.*		
	SPS	7	Stick Parity Select. When bits 1, 2, and 7 of the UARTLCR_H register are set, the parity bit is transmitted and checked as a 0. When bits 1 and 7 are set, and bit 2 is 0, the parity bit is transmitted and checked as a 1. When this bit is cleared, stick parity is disabled.		
	WLEN	6:5	World Length. The select bits indicate the number of data bits transmitted or received in a frame as follows: 11 = 8-bit, 10 = 7-bit, 01 = 6-bit, 00 = 5-bit.		
	FEN	4	Enable FIFOs. If this bit is set to 1, transmit and receive FIFO buffers are enabled (FIFO mode). When cleared to 0, the FIFOs are disabled (character mode) that is, the FIFOs become 1-byte-deep holding registers.		
	STP2	3	Two Stop Bits Select. If this bit is set to 1, two stop bits are transmitted at the end of the frame. The receive logic does not check for two stop bits being received.		
	EPS	2	Even Parity Select. If this bit is set to 1, even parity generation and checking is performed during transmission and reception, which checks for an even number of 1s in data and parity bits. When cleared to 0, then odd parity is performed which checks for an odd number of 1s.		

			This bit has no effect when parity is disabled by Parity Enable (bit 1) being cleared to 0.		
	PEN	1	Parity Enable. If this bit is set to 1, parity checking and generation is enabled, else parity is disabled and no parity bit added to the data frame.		
	BRK	0	Send Break. If this bit is set to 1, a low-level is continually output on the UARTTXD output, after completing transmission of the current character. For the proper execution of the break command, the software must set this bit for at least two complete frames. For normal use, this bit must be cleared to 0.		
0x030	**UARTCR**	**15:0**	**Control Register**	**0x0300**	**RW**
	CTSEn	15	CTS Hardware Flow Control Enable. If this bit is set to 1, CTS hardware flow control is enabled. Data is only transmitted when the nUARTCTS signal is asserted.		
	RTSEn	14	RTS Hardware Flow Control Enable. If this bit is set to 1, RTS hardware flow control is enabled. Data is only requested when there is space in the receive FIFO for it to be received.		
	OUT2	13	This bit is the complement of the UART Out2 (nUARTOut2) modem status output. That is, when the bit is programmed to a 1, the output is 0. For DTE this can be used as Ring Indicator (RI).		
	OUT1	12	This bit is the complement of the UART Out1 (nUARTOut1) modem status output. That is, when the bit is programmed to a 1 the output is 0. For DTE this can be used as Data Carrier Detect (DCD).		
	RTS	11	Request to Send. This bit is the complement of the UART request to send (nUARTRTS) modem status output. That is, when the bit is programmed to a 1, the output is 0.		
	DTR	10	Data Transmit Ready. This bit is the complement of the UART data transmit ready (nUARTDTR) modem status output. That is, when the bit is programmed to a 1, the output is 0.		
	RXE	9	Receive Enable. If this bit is set to 1, the receive section of the UART is enabled. When the UART is disabled in the middle of reception, it completes the current character before stopping.		
	TXE	8	Transmit Enable. If this bit is set to 1, the transmit section of the UART is enabled. When the UART is disabled in the middle of transmission, it completes the current character before stopping.		
	LBE	7	*Not Supported in MS500*	-	-
	-	6:3	*Reserved, do not modify, read as zero.*	-	-
	SIRLP	2	*Not Supported in MS500.*	-	-
	SIREN	1	*Not Supported in MS500.*	-	-
	UARTEN	0	UART Enable. 0 = UART is disabled, 1 = UART is enabled. If the UART is disabled in the middle of transmission or reception, it completes the current character before stopping.		
0x034	**UARTIFLS**	**15:0**	**Interrupt FIFO Level Select Register**	**0x12**	**RW**

	-	15:6	*Reserved, do not modify, read as zero.*		
	RXIFLSEL	5:3	Receive Interrupt FIFO Level Select. The trigger points for the receive interrupt are as follows: 000 = Receive FIFO becomes ≥ 1/8 full 001 = Receive FIFO becomes ≥ 1/4 full 010 = Receive FIFO becomes ≥ 1/2 full 011 = Receive FIFO becomes ≥ 3/4 full 100 = Receive FIFO becomes ≥ 7/8 full 101-b111 = reserved.		
	TXIFLSEL	2:0	Transmit Interrupt FIFO Level Select. The trigger points for the transmit interrupt are as follows: 000 = Transmit FIFO becomes ≤ 1/8 full 001 = Transmit FIFO becomes ≤ 1/4 full 010 = Transmit FIFO becomes ≤ 1/2 full 011 = Transmit FIFO becomes ≤ 3/4 full 100 = Transmit FIFO becomes ≤ 7/8 full 101-111 = reserved.		
0x038	**UARTIMSC**	**15:0**	**Interrupt Mask set/clear Register**	**0x00**	**RW**
	-	15:11	*Reserved, do not modify, read as zero.*		
	OEIM	10	Overrun Error Interrupt Mask. A read returns the current mask for the UARTOEINTR interrupt. On a write of 1, the mask of the UARTOEINTR interrupt is set. A write of 0 clears the mask.	0x00	RW
	BEIM	9	Break Error Interrupt Mask. A read returns the current mask for the UARTBEINTR interrupt. On a write of 1, the mask of the UARTBEINTR interrupt is set. A write of 0 clears the mask.	0x00	RW
	PEIM	8	Parity Error Interrupt Mask. A read returns the current mask for the UARTPEINTR interrupt. On a write of 1, the mask of the UARTPEINTR interrupt is set. A write of 0 clears the mask.	0x00	RW
	FEIM	7	Framing Error Interrupt Mask. A read returns the current mask for the UARTFEINTR interrupt. On a write of 1, the mask of the UARTFEINTR interrupt is set. A write of 0 clears the mask.	0x00	RW
	RTIM	6	Receive Timeout Interrupt Mask. A read returns the current mask for the UARTRTINTR interrupt. On a write of 1, the mask of the UARTRTINTR interrupt is set. A write of 0 clears the mask.	0x00	RW
	TXIM	5	Transmit Interrupt Mask. A read returns the current mask for the UARTTXINTR interrupt. On a write of 1, the mask of the UARTTXINTR interrupt is set. A write of 0 clears the mask	0x00	RW
	RXIM	4	Receive Interrupt Mask. A read returns the current mask for the UARTRXINTR interrupt. On a write of 1, the mask of the UARTRXINTR interrupt is set. A write of 0 clears the mask.		
	DSRMIM	3	nUARTDSR Modem Interrupt Mask. A read returns the current mask for the UARTDSRINTR interrupt. On a write of 1, the mask of the UARTDSRINTR interrupt is set. A write of 0 clears the mask		
	DCDMIM	2	nUARTDCD Modem Interrupt Mask.		

			A read returns the current mask for the UARTDCDINTR interrupt. On a write of 1, the mask of the UARTDCDINTR interrupt is set. A write of 0 clears the mask.		
	CTSMIM	1	nUARTCTS Modem Interrupt Mask. A read returns the current mask for the UARTCTSINTR interrupt. On a write of 1, the mask of the UARTCTSINTR interrupt is set. A write of 0 clears the mask.		
	RIMIM	0	nUARTRI Modem Interrupt Mask. A read returns the current mask for the UARTRIINTR interrupt. On a write of 1, the mask of the UARTRIINTR interrupt is set. A write of 0 clears the mask.		
0x03C	**UARTRIS**	**15:0**	**Raw Interrupt Status Register**	**0x00**	**RO**
	-	15:11	*Reserved, do not modify, read as zero.*		
	OERIS	10	Overrun Error Interrupt Status. Returns the raw interrupt state of the UARTOEINTR interrupt.		
	BERIS	9	Break Error Interrupt Status. Returns the raw interrupt state of the UARTBEINTR interrupt.		
	PERIS	8	Parity Error Interrupt Status. Returns the raw interrupt state of the UARTPEINTR interrupt.		
	FERIS	7	Framing Error Interrupt Status. Returns the raw interrupt state of the UARTFEINTR interrupt.		
	RTRIS	6	Receive Timeout Interrupt Status. Returns the raw interrupt state of the UARTRTINTR interrupt. a		
	TXRIS	5	Transmit Interrupt Status. Returns the raw interrupt state of the UARTTXINTR interrupt.		
	RXRIS	4	Receive Interrupt Status. Returns the raw interrupt state of the UARTRXINTR interrupt.		
	DSRRMIS	3	nUARTDSR Modem Interrupt Status. Returns the raw interrupt state of the UARTDSRINTR interrupt.		
	DCDRMIS	2	nUARTDCD Modem Interrupt Status. Returns the raw interrupt state of the UARTDCDINTR interrupt.		
	CTSRMIS	1	nUARTCTS Modem Interrupt Status. Returns the raw interrupt state of the UARTCTSINTR interrupt.		
	RIRMIS	0	nUaRTRI Modem Interrupt Status. Returns the raw interrupt state of the UARTRIINTR interrupt.		
0x040	**UARTMIS**	**15:0**	**Masked Interrupt Status Register**	**0x00**	**RO**
	-	15:11	*Reserved, read as zero, do not modify.*		
	OEMIS	10	Overrun Error Masked Interrupt Status. Returns the masked interrupt state of the UARTOEINTR interrupt.	0x00	RO
	BEMIS	9	Break Error Masked Interrupt Status. Returns the masked interrupt state of the UARTBEINTR interrupt.	0x00	RO
	PEMIS	8	Parity Error Masked Interrupt Status. Returns the masked interrupt state of the UARTPEINTR interrupt.		
	FEMIS	7	Framing Error Masked Interrupt Status. Returns the masked interrupt state of the UARTFEINTR interrupt.		
	RTMIS	6	Receive Timeout Masked Interrupt Status. Returns the masked interrupt state of the UARTRTINTR interrupt.		
	TXMIS	5	Transmit Masked Interrupt Status. Returns the masked interrupt state of the UARTTXINTR interrupt.		
	RXMIS	4	Receive Masked Interrupt Status. Returns the masked interrupt state of the UARTRXINTR interrupt.		

	DSRMMIS	3	nUARTDSR Modem Masked Interrupt Status. Returns the masked interrupt state of the UARTDSRINTR interrupt.		
	DCDMMIS	2	nUARTDCD Modem Masked Interrupt Status.		
	CTSMMIS	1	nUARTCTS Modem Masked Interrupt Status.		
	RIMMIS	0	nUARTRI Modem Masked Interrupt Status.		
0x044	**UARTICR**	**15:0**	**Interrupt Clear Register And is Write-Only**	-	WO
	-	15:11	*Reserved, read as zero, do not modify.*		
	OEIC	10	Overrun Error Interrupt Clear. Clears the UARTOEINTR interrupt.	-	
	BEIC	9	Break Error Interrupt Clear. Clears the UARTBEINTR interrupt.	-	
	PEIC	8	Parity Error Interrupt Clear. Clears the UARTPEINTR interrupt.	-	
	FEIC	7	Framing Error Interrupt Clear. Clears the UARTFEINTR interrupt.	-	
	RTIC	6	Receive Timeout Interrupt Clear. Clears the UARTRTINTR interrupt.	-	
	TXIC	5	Transmit Interrupt Clear. Clears the UARTTXINTR interrupt.	-	
	RXIC	4	Receive Interrupt Clear. Clears the UARTRXINTR interrupt.	-	
	DSRMIC	3	nUARTDSR Modem Interrupt Clear. Clears the UARTDSRINTR interrupt.	-	
	DCDMIC	2	nUARTDCD Modem Interrupt Clear. Clears the UARTDCDINTR interrupt.	-	
	CTSMIC	1	nUARTCTS Modem Interrupt Clear. Clears the UARTCTSINTR interrupt.	-	
	RIMIC	0	nUARTRI Modem Interrupt Clear. Clears the UARTRIINTR interrupt	-	
0x048	**UARTDMACR**	**15:0**	**DMA Control Register**	**0x00**	**RW**
	-	15:3	*Reserved, read as zero, do not modify.*		
	DMAONER	2	DMA on Error. If this bit is set to 1, the DMA receive request outputs, UARTRXDMASREQ or UARTRXDMABREQ, are disabled when the UART error interrupt is asserted.		
	TXDMAE	1	Transmit DMA Enable. If this bit is set to 1, DMA for the transmit FIFO is enabled.		
	RXDMAE	0	Receive DMA Enable. If this bit is set to 1, DMA for the receive FIFO is enabled.		
0xFE0	UARTPERIPHID0	15:8	*Reserved, read undefined must read as zeros.*	0x11	RO
	PARTNUMBER0	7:0	These bits read back as 0x11.		
0xFE4	UARTPERIPHID1	15:8	*Reserved, read undefined must read as zeros.*	0x10	RO
	DESIGNER0	7:4	These bits read back as 0x1.		
	PARTNUMBER1	3:0	These bits read back as 0x0.		
0xFE8	UARTPERIPHID2	15:8	*Reserved, read undefined must read as zeros.*	0x14	RO
	REVISION	7:4	These bits read back as 0x3.		
	DESIGNER1	3:0	These bits read back as 0x4.		
0xFFC	UARTPERIPHID3	15:8	*Reserved, read undefined must read as zeros.*		

	CONFIGURATION	7:0	These bits read back as 0x00	0x00	RO
0xFF0	UARTCELLID0	15:8	*Reserved, read undefined must read as zeros.*	-	-
	UARTCELLID0	7:0	These bits read back as 0x0D.	0x0D	RO
0xFF4	UARTCELLID1	15:8	*Reserved, read undefined must read as zeros.*	-	-
	UARTCELLID1	7:0	These bits read back as 0xF0.	0xF0	RO
0xFF8	UARTCELLID2	15:8	*Reserved, read undefined must read as zeros.*	-	-
	UARTCELLID2	7:0	These bits read back as 0x05.	0x05	RO
0xFFC	UARTCELLID3	15:8	*Reserved, read undefined must read as zeros.*	-	-
	UARTCELLID3	7:0	These bits read back as 0xB1.	0xB1	RO

5. SDMMC (HOST CONTROLLER)

The MS500 includes a SD/MMC controller interface. The SD/MMC controller interface supports the following features:

- Compliant with SD Physical Layer Specification Version 3.01: supporting UHS-I mode up to SDR50 and DDR50
- Compliant with SDIO Specification Version 3.00: SDIO host does not have a legacy SPI mode
- Compliant with eMMC/MMC Specification Version 4.41
- Support three different data bus modes: 1-bit (default), 4-bit and 8-bit
- Provide register interface compliant with SD Host Controller Specification Version 3.00
- SDIO interrupts in 1-bit and 4-bit modes.
- SDIO suspend and resume operation.
- SDIO read wait.
- Block size of 1 to 65,535 bytes
- FIFO over-run and under-run prevention by stopping card clock.
- Provide delay line for the tuning of IO interface timing
- Provide SD card detection by debounce filter logic
- Integrated DMA master supporting SDMA (Single Task DMA)
- Support DMA transfer using System DMAC.
- Two FIFOs, TX and RX FIFO (FIFO depth = 32 and FIFO data width = 32 bits).
- Supports HS mode and UHS-1 (SDR50) mode

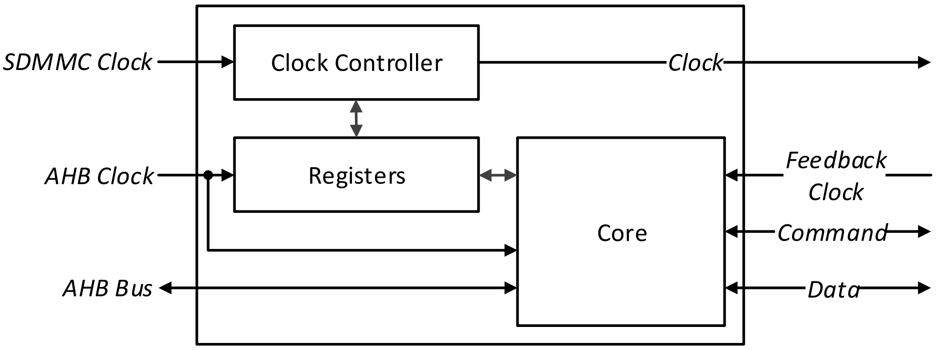

Figure 16 SDMMC Block Diagram

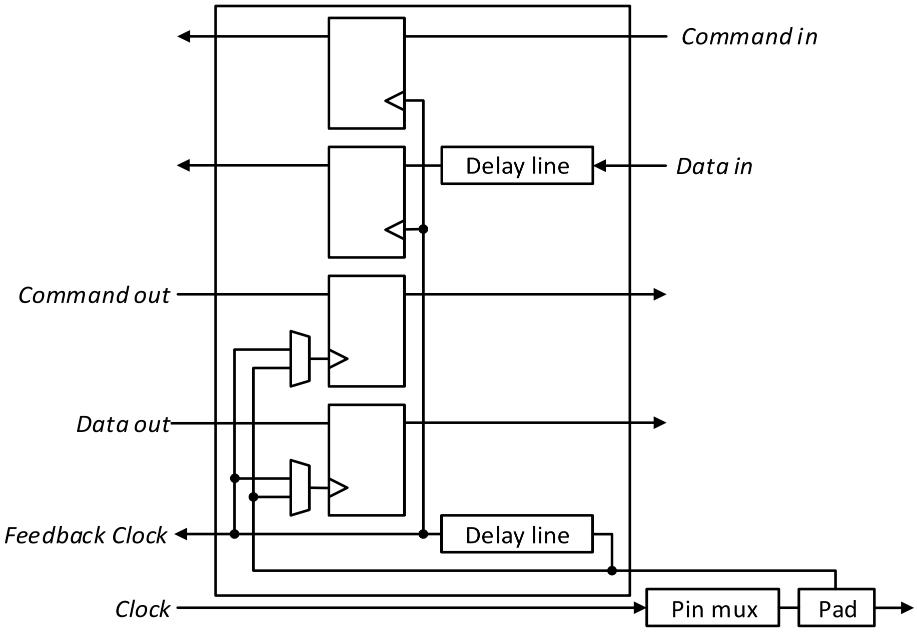

Figure 17 SDMMC I/O Block

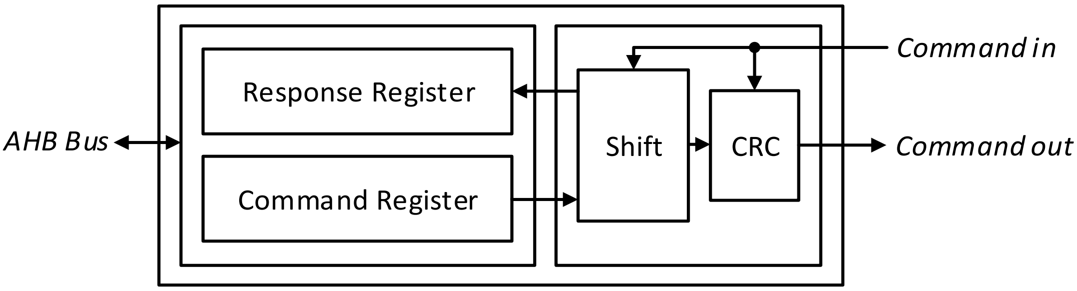

Figure 18 SDMMC Command Path

Table 23 SDMMC Host Controller Register Description

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
Note 1: Register map structure is in compliance with SD Host Controller Specification (SDHCI) Version 3.00. **This does not mean full compatibility with the specification. Incompatible point is described as [Note], otherwise register acts as described in the specification. (Most drivers compatible with SDHCI will work without modification or with slight modifications)** **Note 2: Register interface is 32-bits bus and byte-access able, so it operates like 32-bits memory**						
Parameters to generate SDMMC commands						
0x00	**SDMMC_SDMA_ADDR** (or SDMMC_ARG2)	32	31:0	**SDMA Start Address / Argument 2 (for Auto CMD23) Register** Note: Start address should be aligned with 32-bit word boundary	0x0	WO/RO
0x04	**SDMMC_BLK_SIZE**	16	15:0	**Block Size Register**	0x0	RW
	SDMMC_TRAN_BLKSZ	12	11:0	Transfer Block Size 0x000: No Transfer 0x001: 1 Byte 0x002: 2 Bytes ... 0x800: 2048 Bytes		
	SDMMC_SDMA_BUFSZ	3	14:12	Host SDMA Buffer Boundary Note: SDMA buffer boundary is not supported in this version		
	-	1	15	*Reserved*		Rsvd
0x06	**SDMMC_BLK_CNT**	16	15:0	**Block Count Register**	0x0	WO/RO
				Blocks Count for Current Transfer 0x0000: Stop Transfer 0x0001: 1 Block 0x0002: 2 Blocks ... 0xFFFF: 65535 Blocks		
				When a suspend command is completed, the number of blocks yet to be transferred can be determined by reading this register. Before issuing a resume command, the Host Driver shall restore the previously saved block count		
				Note: Decrements the block count after each block transfer and stops when the count reaches zero Note: This register should be accessed only when no transaction is executing (i.e., after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored		
0x08	**SDMMC_ARG**	32	31:0	**Argument 1 Register**	0x0	RW
0x0C	**SDMMC_TRAN_MODE**	16	15:0	**Transfer Mode Register**	0x0	RW
	SDMMC_DMA_EN	1	0	DMA Enable		
	SDMMC_BLKCNT_EN	1	1	Block Count Enable Note: Multiple transfer with disabling **Block Count Enable** execute Infinite Transfer		
	SDMMC_AUTO_CMD12	1	2	Auto CMD12 Enable		
	SDMMC_AUTO_CMD23	1	3	Auto CMD23 Enable		
	SDMMC_TRAN_DIR	1	4	Data Transfer Direction Select 0: Write (Host -> Card), 1: Read (Card -> Host)		
	SDMMC_MULTI_BLK	1	5	Multi / Single Block Select 0: Single Block, 1: Multi Block		
	-	10	15:6	*Reserved*		Rsvd
0x0E	**SDMMC_CMD**	16	15:0	**Command Register**	0x0	RW

	SMMC_RESP_TYPE	2	1:0	Response Type Select 00: No Response 01: Response Length 136 10: Response Length 48 11: Response Length 48 check Busy after Response		
	-	1	2	*Reserved*		Rsvd
	SDMMC_CMD_CRC	1	3	Command CRC Check Enable		
	SDMMC_CMD_IDX	1	4	Command Index Check Enable		
	SDMMC_CMD_DATA	1	5	Data Present Select 0: No Data Present, 1: Data Present		
	SDMMC_CMD_TYPE	2	7:6	Command Type 00: Normal (other commands) 01: Suspend (CMD52 for writing "Bus Suspend" in CCCR) 10: Resume (CMD52 for writing "Function Select" in CCCR) 11: Abort (CMD12, CMD52 for writing "I/O Abort" in CCCR)		
	SDMMC_CMD_IDXNUM	6	13:8	Command Index (CMD0-63)		
	-	2	15:14	*Reserved*		Rsvd
Response value from the card						
0x10	**SDMMC_RESP**	128	127:0	**Response Register**	0x0	ROC
				R1 / R1b (Normal Response): Bits[31:0] => Card Status R1 / R1b (Auto CMD Response): Bits[127:96] => Card Status for Auto CMD Response R2: Bits[119:0] => CID or CSD Register R3 / R4: Bits[31:0] => OCR Register or Fast I/O etc. R5: Bits[31:0] => Interrupt Request		
Data access port to the internal buffer						
0x20	**SDMMC_DATA**	32	31:0	**Buffer Data Port Register**	-	WO/RO
Present State, controls for the SD Bus, Host reset and so on						
0x24	**SDMMC_PRES_STATE**	32	31:0	**Present State Register**	0x0	ROC
	SDMMC_CMD_INHB	1	0	Command Inhibit (CMD) 0: Can issue command using only CMD line 1: Cannot issue command		
	SDMMC_DATA_INHB	1	1	Command Inhibit (DAT) 0: Can issue command which uses the DAT line 1: Cannot issue command which uses the DAT line		
	SDMMC_DATA_LINE_ACT	1	2	DAT Line Active		
	-	5	7:3	*Reserved*		Rsvd
	SDMMC_DOING_WRITE	1	8	Write Transfer Active		
	SDMMC_DOING_READ	1	9	Read Transfer Active		
	SDMMC_SPACE_AVAIL	1	10	Buffer Write Enable		
	SDMMC_DATA_AVAIL	1	11	Buffer Read Enable		
	-	4	15:12	*Reserved*		Rsvd
	SDMMC_CARD_PRES	1	16	Card Inserted 0: Reset, Debouncing or No Card, 1: Card Inserted		RO
	SDMMC_CARD_STATE_STBL	1	17	Card State Stable 0: Reset or Debouncing, 1: No Card or Inserted		RO
	SDMMC_CARD_DET_LVL	1	18	Card Detect Pin Level 0: No card preset (SDCD# = 1), 1: Card present (SDCD# = 0)		RO
	-	1	19	*Reserved*		Rsvd
	SDMMC_DATA_LVL	4	23:20	DAT[3:0] Line Signal Level		RO
	SDMMC_CMD_LVL	1	24	CMD Line Signal Level		RO
	-	7	31:25	*Reserved*		Rsvd

0x28	**SDMMC_HOST_CTRL**	8	7:0	**Host Control 1 Register**	0x0	RW
	-	1	0	*Reserved*		Rsvd
	SDMMC_CTRL_4BIT	1	1	Data Transfer Width 0: 1-bit mode, 1: 4-bit mode		
	SDMMC_CTRL_HISPD	1	2	High Speed Enable 0: Outputs CMD and DAT line(s) at the falling edge of the SDMMC CLK 1: Outputs CMD and DAT line(s) at the rising edge of the SDMMC CLK		
	-	2	4:3	*Reserved*		Rsvd
	SDMMC_CTRL_8BIT	1	5	Extended Data Transfer Width 0: Bus Width is selected by **Data Transfer Width**, 1: 8-bit Bus Width		
	-	2	7:6	*Reserved*		Rsvd
0x29	**SDMMC_PWR_CTRL**	8	7:0	**Power Control Register**	0x0	RW
	SDMMC_PWR_ON	1	0	SD Bus Power (On/Off) Note: If this bit is cleared, immediately stop driving CMD and DAT and drive SDMMC CLK to low level		
	-	7	7:1	*Reserved*		Rsvd
0x2A	**SDMMC_BLKGAP_CTRL**	8	7:0	**Block Gap Control Register**	0x0	
	SDMMC_STOP_AT_BLKGAP	1	0	Stop at Block Gap Request 0: Transfer, 1: Stop		RW
	SDMMC_CONT_REQ	1	1	Continue Request 0: No affect, 1: Restart		RWAC
	SDMMC_RDWAIT_CTL	1	2	Read Wait Control (En/Disable)		RW
	SDMMC_INT_AT_BLKGAP	1	3	Interrupt at Block Gap Request (En/Disable)		RW
	-	4	7:4	*Reserved*		Rsvd
0x2B	*Reserved for SDHCI Compatibility*					
0x2C	**SDMMC_CLK_CTRL**	16	15:0	**Clock Control Register**	0x0	RW
	SDMMC_CLK_INT_EN	1	0	Internal Clock Enable		
	SDMMC_CLK_INT_STBL	1	1	Internal Clock Stable 0: Not Ready, 1: Ready		ROC
	SDMMC_CLK_CARD_EN	1	2	SD Clock Enable		
	-	3	5:3	*Reserved*		Rsvd
	SDMMC_CLK_DIV_HI	2	7:6	Upper Bits of SD_CLK Frequency Select		
	SDMMC_CLK_DIV	8	15:8	SD_CLK Frequency Select		
				10-bit Divided Clock Mode with **Upper Bits of SD_CLK Frequency Select** (for Host Controller Version 3.00) 0x000: Base Clock (10-255 MHz) ... N: 1/2N Divided Clock (Duty 50%) ... 0x3FF: 1/2046 Divided Clock		
0x2E	**SDMMC_TOUT_CTRL**	8	7:0	**Timeout Control Register**	0x0	RW
	SDMMC_TOUT_CNT_VAL	4	3:0	Data Timeout Counter Value 0000: TMCLK x (2^14) 0001: TMCLK x (2^15) ... 1101: TMCLK x (2^27) Others: *Reserved* Note: TMCLK is SD_CLK		
	-	4	7:4	*Reserved*		Rsvd
0x2F	**SDMMC_SW_RST**	8	7:0	**Software Reset Register**	0x0	RW
	SDMMC_RST_ALL	1	0	Software Reset for All		

	SDMMC_RST_CMD	1	1	Software Reset for CMD Line		
	SDMMC_RST_DATA	1	2	Software Reset for DAT Line		
	-	5	7:3	*Reserved*		Rsvd
Interrupt statuses and enables						
0x30	**SDMMC_INT_STS**	16	15:0	**Normal Interrupt Status Register**	0x0	RW1C
	SDMMC_INT_RESP	1	0	Command Complete Note: **Command Timeout Error** has higher priority than **Command Complete** Note: No occurrence of both **Command Complete** and **Command Timeout Error** means the status of "Interrupted by another factor"		
	SDMMC_INT_DATA_END	1	1	Transfer Complete Note: **Transfer Complete** has higher priority than **Data Timeout Error** Note: No occurrence of both **Transfer Complete** and **Data Timeout Error** means the status of "Interrupted by another factor"		
	SDMMC_INT_BLK_GAP	1	2	Block Gap Event (Transaction stopped at Block Gap)		
	SDMMC_INT_DMA_END	1	3	DMA Interrupt		
	SDMMC_INT_SPACE_AVAIL	1	4	Buffer Write Ready		
	SDMMC_INT_DATA_AVAIL	1	5	Buffer Read Ready		
	SDMMC_INT_CARD_INS	1	6	Card Insertion 0: Card state stable or Debouncing, 1: Card inserted		
	SDMMC_INT_CARD_REM	1	7	Card Removal 0: Card state stable or Debouncing, 1: Card removed		
	SDMMC_INT_CARD_INT	1	8	Card Interrupt 0: No Card Interrupt, 1: Generate Card Interrupt		ROC
				Note: Writing this bit to 1 does not clear this bit. It is cleared by resetting the SD card interrupt factor		
	-	2	10:9	*Reserved*		
	SDMMC_INT_BOOT_ACK	1	11	Boot Acknowledge Received (for MMC/eMMC)		
	-	3	14:12	*Reserved*		Rsvd
	SDMMC_INT_ERR	1	15	Error Interrupt		ROC
0x32	**SDMMC_ERR_STS**	16	15:0	**Error Interrupt Status Register**	0x0	RW
	SDMMC_ERR_CMD_TOUT	1	0	Command Timeout Error Note: If both **Command Timeout Error** and **Command CRC error**, it means the status of "CMD line conflict"		
	SDMMC_ERR_CMD_CRC	1	1	Command CRC Error		
	SDMMC_ERR_CMD_END	1	2	Command End Bit Error		
	SDMMC_ERR_IDX	1	3	Command Index Error		
	SDMMC_ERR_DATA_TOUT	1	4	Data Timeout Error ~ Busy timeout for R1b, Busy timeout after Write CRC status, Write CRC Status timeout, Read Data timeout		
	SDMMC_ERR_DATA_CRC	1	5	Data CRC Error		
	SDMMC_ERR_DATA_END	1	6	Data End Bit Error		
	-	1	7	*Reserved*		Rsvd
	SDMMC_ERR_ACMD	1	8	Auto Command Error		
	-	3	11:9	*Reserved*		Rsvd
	SDMMC_ERR_BUS_RESP	1	12	Bus Transfer Response Error		
	SDMMC_ERR_BUS_TOUT	1	13	Bus Data Ready Timeout Error		
	SDMMC_ERR_FIFO	1	14	Data FIFO overrun/underrun Error		
	-	1	15	*Reserved*		Rsvd
0x34	**SDMMC_INT_EN**	16	15:0	**Normal Interrupt Status Enable Register**		RW

		12	11:0	Each bit corresponds to the same bit position of **Normal Interrupt Status Register**		
		3	14:12	*Reserved*		Rsvd
		1	15	Fixed to 0 Note: Host driver shall control error interrupts using **Error Interrupt Status Enable** register		RO
0x36	**SDMMC_ERR_EN**	16	15:0	**Error Interrupt Status Enable Register**	0x0	RW
		7	6:0	Each bit corresponds to the same bit position of **Error Interrupt Status Register**		
		1	7	*Reserved*		Rsvd
		1	8	This bit corresponds to the same bit position of **Error Interrupt Status Register**		
		3	11:9	*Reserved*		Rsvd
		3	14:12	Each bit corresponds to the same bit position of **Error Interrupt Status Register**		
		1	15	*Reserved*		Rsvd
0x38	**SDMMC_SIG_EN**	16	15:0	**Normal Interrupt Signal Enable Register**	0x0	RW
		12	11:0	Each bit corresponds to the same bit position of **Normal Interrupt Status Register**		
		3	14:12	*Reserved*		
		1	15	Fixed to 0 Note: Host driver shall control error interrupts using **Error Interrupt Signal Enable** register		RO
0x3A	**SDMMC_ERR_SIG_EN**	16	15:0	**Error Interrupt Signal Enable Register**	0x0	RW
		7	6:0	Each bit corresponds to the same bit position of **Error Interrupt Status Register**		
		1	7	*Reserved*		Rsvd
		1	8	This bit corresponds to the same bit position of **Error Interrupt Status Register**		
		3	11:9	*Reserved*		Rsvd
		3	14:12	Each bit corresponds to the same bit position of **Error Interrupt Status Register**		
		1	15	Reserved		
0x3C	**SDMMC_ERR_ACMD**	16	15:0	**Auto CMD Error Status Register**	0x0	ROC
	SDMMC_ERR_ACMD_NOEXE	1	0	Auto CMD12 Not Executed		
	SDMMC_ERR_ACMD_TOUT	1	1	Auto CMD Timeout Error Note: If both **Auto CMD Timeout Error** and **Auto CMD CRC error**, it means the status of "CMD line conflict"		
	SDMMC_ERR_ACMD_CRC	1	2	Auto CMD CRC Error		
	SDMMC_ERR_ACMD_END	1	3	Auto CMD End Bit Error		
	SDMMC_ERR_ACMD_IDX	1	4	Auto CMD Index Error		
	-	11	15:5	*Reserved*		Rsvd
Extension of Host Control Register						
0x3E	**SDMMC_HOST_CTRL2**	16	15:0	**Host Control 2 Register**	0x0	RW
		1	0	*Preserved*		
	SDMMC_CTRL_UHS	1	1	UHS Mode (SDR 50 or Higher Speed) Enable Note: If this bit set, **High Speed Enable** setting is ignored		
	SDMMC_CTRL_DDR	1	2	DDR Mode Enable Note: If this bit set, **High Speed Enable** and **UHS Mode Enable** settings are ignored		
	SDMMC_CTRL_VDD180	1	3	1.8V Signaling Enable		
	SDMMC_CTRL_DRV_TYPE	2	5:4	Driver Strength Select (for 1.8V; Not effective in 3.3V signaling) 00: Driver Type B is Selected (Default) 01: Driver Type A is Selected		

				10: Driver Type C is Selected 11: Driver Type D is Selected		
		10	15:6	*Reserved*		
Vendor specific host controller support information						
0x40	SDHCI_CAP	64	63:0	**Capabilities Register**	0x0	HwInit
0x40~ 0xBF	*Reserved for SDHCI Compatibility*					
0xC0	**SDMMC_HOST_CTRL3**	16	15:0	**Host control 3 Register**		RW
				Note: This register does not exist in SDHCI and mainly used for MMC/eMMC		
	SDMMC_CMD_LINE_MODE	1	0	CMD Line Mode 0: Open-Drain (default), 1: Push-pull		
	SDMMC_CMD_LINE_CTL	1	1	CMD Line Control 0: Auto-controlled (default), 1: Force to Low (for default boot mode)		
	SDMMC_BOOT_EN	1	2	Boot Operation Enable		
	SDMMC_BOOT_ACK	1	3	Boot Acknowledge Mode 0: Don't check Boot ACK (default), 1: Check Boot ACK		
	-	4	7:4	*Reserved*		Rsvd
	SDMMC_DMA_REQ_EN	1	8	DMA Request Enable (for External DMAC) Note: If this bit set, SDMA operation is disabled		
	SDMMC_SDMA_ALIGN	1	9	SDMA Block Address Align 0: Starting address of subsequent block is the next byte address to preceding block 1: Starting address of subsequent block is aligned with 32-bit word boundary	1b	
	-	3	12:10	*Reserved*		Rsvd
	SDMMC_CLK_STOP_DIS	1	13	Clock Stopping (for prevention of FIFO overrun/underrun) Disable ~ Test purpose		
	SDMMC_TEST_GEN	1	14	Test Data Generation Enable		
	SDMMC_RST_LINE	1	15	Reset Line Control 0: Release SDMMC_RSTN to High, 1: Assert SDMMC_RSTN to Low Note: Writing 1 is effective only when writing it twice consecutively	1b	
0xC2	*reserved*	16	15:0	*Reserved*	-	Rsvd
0xC4	**SDMMC_CLK_CTRL2**	32	31:0	**SDMMC Clock Control Register 2**	0x0	RW
				Note: This register does not exist in SDHCI		
	FBCLK_DLY_VAL	6	5:0	Delay Line Control for Feedback clock 0x00: ~0.3 ns ... N: ~ 0.3 ns x (N+1) ... 0x3F: ~19.2 ns		
	-	2	7:6	*Reserved*		Rsvd
	DIN_DLY_VAL	6	13:8	Delay Line Control for Input Data *~ Same as above description*		
	-	2	15:14	*Reserved*		Rsvd
	DOUT_CLK_SEL	1	16	Clock Select for Output Data 0: Feedback clock (default), 1: Delayed feedback clock		
	-	15	31:17	*Reserved*		Rsvd
0xC8	**PERI_ID**	32	31:0	**SDMMC Host Peripheral ID Register**	HwInit	RO
				Note: This register does not exist in SDHCI		
	PART_NUM	12	11:0	SDMMC Part Number	0x480	

	CONFIG_NUM	4	15:12	SDMMC Configuration Number	0x2	
	MAJ_REV	4	19:16	SDMMC Major Revision		
	MIN_REV	4	23:20	SDMMC Minor Revision		
	RSVD_NUM	8	31:24	SDMMC Reserved Number		
0xD0~ 0xFD	*Reserved for SDHCI Compatibility*					
0xFE	**SDMMC_HOST_VER**	16	15:0	**Host Controller Version Register**	HwInit	R
	SDHCI_SPEC_VER	8	7:0	Specification Version Number 0x00: SD Host Spec Version 1.00 0x01: SD Host Spec Version 2.00 0x02: SD Host Spec Version 3.00 Others: *Reserved*	0x02	
	-	8	15:8	*Reserved*		

6. SDIO (SLAVE CONTROLLER)

The MS500 includes a SD/MMC controller interface. The SD/MMC controller interface supports the following features:

- Meets SDIO card specification version 3.0.
- Supports HS mode and UHS-1 (SDR50) mode
- Supports asynchronous interrupt to SD Host controller.
- Enhanced power management using new power state control function.
- Supports Read Wait Control operation, Supports Suspend/Resume operation for superior card performance.
- Up to 7 functions in SPI, SD1 and SD4 mode.
- SD Host clock rate from 0 to 104Mhz.
- Supports SPI, 1-bit and 4-bit SD modes.
- All SDIO form factors supported, standard mini and micro SDIO card.
- Bus Master with Scatter Gather DMA.
- Dual operating voltage range 2.7V - 3.6V and 1.7V - 1.9V.
- Up to 416Mbits per second read and write rates using 4 parallel data lines.
- Cyclic Redundancy Check CRC7 for command and CRC16 for data integrity-CRC checking optional in
- Programmable through a standard AHB slave interface.
- Card responds to Direct read/write (IO52) and Extended read/write(IO53) transactions.

The SDIO Card has a fixed internal register space and a Function Unique Area. The Fixed Area contains information about the Card and certain mandatory and optional registers in fixed locations. The Function Unique Area is a per-function area and different for each SDIO Function. Table18 shows the internal map of an SDIO Card with multiple functions.

Table 24 Memory Mapping of SDIO Registers

Offset	Function 0 (128K)
0x0 ~ 0xFF	CCCR Reg
0x100 ~ 0x1FF	FBR Reg (Function1)
0x200 ~ 0xFFF	Reserved
0x1000 ~ 0x1FFF	CIS0 Area
0x2000 ~ 0x2FFF	CIS1 Area
0x3000 ~ 0x17FFF	Reserved
0x18000	Clock Wakeup Reg
0x18001 ~ 0x1FFFF	Reserved

Offset	Function 1 (128K)
0x0 ~ 0x3C	CCCR Reg
0x100 ~ 0x1FF	FBR Reg (Function1)

Card Common Control Registers (CCCR) - The CCCR allow for quick SD Host checking and control of I/O Card's enable and interrupts on a per card and per function basis. Access to the CCCR is possible even after the initialization when the I/O functions are disabled. Access is performed using the I/O read and write commands.

Function Basic Registers (FBR) - Each function has a 256 byte area used to allow the SD Host to quickly determine the abilities and requirements of each function and to enable software loading. The address of this area is from 0x00n00 to 0x00nFF where n is the function number (0x1 to 0x7).

Card Information Structure (CIS) - The CIS provides complete information about the card and the individual functions. The Card has a common CIS and a CIS for each function. The CIS0, CIS1 and CIS2 are located in ARM system memory. The common area CIS and each function CIS have a pointer to the start of its CIS within ARM system memory.

DIGITAL PERIPHERALS

1. DMA CONTROLLER

The DMA enables peripheral-to-memory, memory-to-peripheral, peripheral-to-peripheral, and memory-to-memory transactions. Each DMA stream provides unidirectional serial DMA transfers for a single source and destination. For example, a bidirectional port requires one stream for transmit and one for receive. The source and destination areas can each be either a memory region or a peripheral, and can be accessed through the AHB master.

The DMA include the following features:

- Two DMA channels: Each channel can support a unidirectional transfer.
- Single DMA and burst DMA request signals: Each peripheral connected to the DMA can assert either a burst DMA request or a single DMA request. The DMA burst size is set by programming the DMA.
- The DMA can transfer data between the SRAM and peripherals such as the SD/MMC, SPI controllers, I2C interfaces.

1.1 Register Summary

The following information applies to the DMA registers.

Table 25 DMA Request Signal Mapping

Request Number	DMA Request select		DMA(2ch) - 16 Requests	
	DREQ_SDMMC_SEL*1	DREQ_ARIA_SEL*2		
REQ0	-		SSP1 Transmit	Single/Burst
REQ1			SSP1 Receive	Single/Burst
REQ2			SSP2 Transmit	Single/Burst
REQ3			SSP2 Receive	Single/Burst
REQ4			UART1 Transmit	Single/Burst
REQ5			UART1 Receive	Single/Burst
REQ6			UART2 Transmit	Single/Burst
REQ7			UART2 Receive	Single/Burst
REQ8			I2C1	Single/Burst
REQ9	0	-	I2C2	Single/Burst
	1		SDMMC	Single/Burst
REQ10	-	0	I2C3	Single/Burst
		1	ARIA Transmit	Burst
REQ11	-	0	I2C4	Single/Burst

		1	ARIA Receive	Burst
REQ12	-		SSP3 Transmit	Single/Burst
REQ13			SSP3 Receive	Single/Burst
REQ14			UART3 Transmit	Single/Burst
REQ15			UART3 Receive	Single/Burst

*1) DREQ_SDMMC_SEL : defined by MISCELLOUNIOUS_IF Register(400000D8h)'s bit[1] as table 20

*2) DREQ_ARIA_SEL : defined by MISCELLOUNIOUS_IF Register(400000D8h)'s bit[2] as table 20

Table 26 DMA Request Select to REQ9/10/11

0xD8	**MISCELLOUNIOUS_IF**	32	31:0	**Miscellaneous Interface Register**		RW
	DMA_SDMMC_SEL	**1**	**1**	**DMA Requests 9 Signal select** **0: I2C2** **1: SDMMC HOST**		
	DMA_ARIA_SEL	**1**	**2**	**DMA Requests 10, 11 Signal select** **0: I2C3,I2C4** **1: ARIA Transmit, ARIA Receive**		
	-	28	31:4	*Reserved*		

Table 27 DMA Register Summary

Offset	Symbol	Bit	Brief Description	Reset Value	Type
0x000	**INT_STATUS**	31:0	Interrupt Status Register.	0x00	RO
0x004	**INT_TC_STATUS**	31:0	Interrupt Terminal Count Status Register.	0x00	RO
0x008	**INT_TC_CLEAR**	31:0	Interrupt Terminal Count Clear Register.	-	WO
0x00C	**INT_ERROR_STATUS**	31:0	Interrupt Error Status Register.	0x00	RO
0x010	**INT_ERR_CLR**	31:0	Interrupt Error Clear Register.	-	WO
0x014	**RAW_INT_TC_STATUS**	31:0	Raw Interrupt Terminal Count Status Register.	-	RO
0x018	**RAW_INT_ERROR_STATUS**	31:0	Raw Error Interrupt Status Register	-	RO
0x01C	**ENBLD_CHNS**	31:0	Enabled Channel Register.	0x00	RO
0x020	**SOFT_B_REQ**	31:0	Software Burst Request Register.	0x0000	RW
0x024	**SOFT_S_REQ**	31:0	Software Single Request Register.	0x0000	RW
0x028	**SOFT_LB_REQ**	31:0	Software Last Burst Request Register.	0x0000	RW
0x02C	**SOFT_LS_REQ**	31:0	Software Last Single Request Register.	0x0000	RW
0x030	**CONFIGUARTION**	31:0	Configuration Register.	0b000	RW
0x034	**SYNC**	31:0	Synchronization Register.	0x0000	RW
0x100	**CH_SRC_ADDR**	31:0	Channel Source Address Registers.	0x0000	RW
0x104	**CH_DEST_ADDR**	31:0	Channel Destination Address Registers.	0x0000	RW
0x108	**CH_LLI**	31:0	Channel Linked List Item Registers.	0x0000	RW
0x10C	**CH_CONTROL**	31:0	Channel Control Registers.	0x0000	RW
0x110	**CH_CONFIGURATION**	31:0	Channel Configuration Registers.	0x00	RW
0xFE0	**PERIPH_ID_0**	31:0	DMACPeriphID0 Register.	0x80	RO
0xFE4	**PERIPH_ID_1**	31:0	DMACPeriphID1 Register.	0x10	RO
0xFE8	**PERIPH_ID_2**	31:0	DMACPeriphID2 Register.	0x04	RO

0xFEC	**PERIPH_ID_3**	31:0	DMACPeriphID3 Register.	0x0A	RO
0xFF0	**PCELL_ID_0**	15:0	DMACPCellID0 Register.	0x0D	RO
0xFF4	**PCELL_ID_1**	15:0	DMACPCellID1 Register.	0xF0	RO
0xFF8	**PCell_ID_2**	15:0	DMACPCellID2 Register.	0x05	RO
0xFFC	**PCell_ID_3**	15:0	DMACPCellID3 Register.	0xB1	RO
0x500	**DMAC_IT_CR**	31:0	Test Control Register	0x00	RW
0x504	**DMAC_IT_OP_1**	31:0	Integration Test Output Register 1.	0x00	RW
0x508	**DMAC_IT_OP_2**	31:0	Integration Test Output Register 2.	0x00	RW
0x50C	**DMAC_IT_OP_3**	31:0	Integration Test Output Register 3.	0x0	RW

1.2 Register Description

The DMA controller supports 2 channels. Each DMA channel has registers specific to the operation of that channel. Other registers control aspects of how source peripherals relate to the DMA controller.

Table 28 DMA Register Description

offset	Symbol	Bit	Brief Description	Reset Value	Type
0x000	INT_STATUS	31:2	*Reserved. Read undefined.*	-	-
		1:0	Status of the DMA interrupt after masking.	0x00	RO
0x004	INT_TC_STATUS	31:2	*Reserved. Read undefined.*	-	-
		1:0	Interrupt Terminal Count request Status.	0x00	RO
0x008	INT_TC_CLEAR	31:2	*Reserved. Read undefined.*	-	-
		1:0	Interrupt Terminal Count request Clear.	0x00	RO
0x00C	INT_ERROR_STATUS	31:2	*Reserved. Read undefined.*	-	-
		1:0	Interrupt Error Status.	0x00	RO
0x010	INT_ERR_CLR	31:2	*Reserved. Read undefined.*	-	-
		1:0	Interrupt Error Clear.	0x00	WO
0x014	RAW_INT_TC_STATUS	31:2	*Reserved. Read undefined.*	-	-
		1:0	Status of the terminal count interrupt prior to masking.	0x0	RO
0x018	RAW_INT_ERROR_STATUS	31:2	*Reserved. Read undefined.*	-	-
		1:0	Status of the error interrupt prior to masking.	0x0	RO
0x01C	ENBLD_CHNS	31:2	*Reserved. Read undefined.*	-	-
		1:0	Enable status for DMA channels.	0x0	RO
0x020	SOFT_B_REQ	31:16	*Reserved. Read undefined.*	-	-
		15:0	Software burst request for each of possible sources.	0x00	RW
0x024	SOFT_S_REQ	31:16	*Reserved. Read undefined.*	-	-
		15:0	Software single request for each of possible sources.	0x00	RW
0x028	SOFT_LB_REQ	31:16	*Reserved. Read undefined.*	-	-
		15:0	Software last burst request for each of possible sources.	0x00	RW
0x02C	SOFT_LS_REQ	31:16	*Reserved. Read undefined.*	-	-
		15:0	Software last single request for each of possible sources.	0x00	RW
0x030	**CONFIGUARTION**	**31:0**	**DMAC Configuration Register**	**0x0**	**RW**
	-	31:2	*Reserved. Read undefined.*		
	M	1	AHB master endianness configuration: 0 = little-endian mode 1 = big-endian mode.		

			This bit is reset to 0.		
	E	0	DMA controller enable: 0 = disabled 1 = enabled. This bit is reset to 0. Disabling the DMA controller reduces power consumption.		
0x034	**SYNC**	**31:0**	**DMAC Sync Reigster**	**0x00**	**RW**
	-	31:16	*Reserved. Read undefined.*		
	DMA_SYNC	15:0	DMA synchronization logic for DMA request signals enabled or disabled. Each bit represents one set of DMA request lines as described in the preceding text: 0 = synchronization logic for DMACBREQ[15:0], DMACSREQ[15:0], DMACLBREQ[15:0], and DMACLSREQ[15:0] request signals is enabled, 1 = synchronization logic is disabled.		
0x100	**CH_SRC_ADDR**	**31:0**	**DMAC source address.**	**0x00**	**RW**
	SRC_ADDR		Reading this register will return the current source address. Note that the source address must be aligned to the source width.		
0x104	**CH_DEST_ADDR**	**31:0**	**DMAC destination address.**	**0x00**	**RW**
	DEST_ADDR		Reading this register will return the current destination address. Note that the destination address must be aligned to the destination width.		
0x108	**CH_LLI**	**31:0**	**Linked list Register**	**0x00**	**-**
	LLI	31:2	Bits [31:2] of the address for the next LLI. Address bits [1:0] are 0.		
	-	1:0	Reserved. Read undefined. Write as zero.		
0x10C	**CH_CONTROL**	**31:0**	**DMAC Channel Control Register**	**0x00**	**RW**
	I	31	Terminal count interrupt enable bit. It controls whether the current LLI is expected to trigger the terminal count interrupt. 0 = the terminal count interrupt is disabled, 1 = the terminal count interrupt is enabled.		
	PROT	30:28	Protection. This information is provided to the source and/or peripherals when a transfer occurs. The transfer information is provided by programming the DMA channel. These bits are programmed by software and can be used by peripherals.		
	DI	27	Destination Increment. 0 = the destination address is not incremented after each transfer, 1 = the destination address is incremented after each transfer.		
	SI	26	Source Increment. 0 = the source address is not incremented after each transfer, 1 = the source address is incremented after each transfer.		
	-	25:24	Reserved. Read undefined.		
	D_WIDTH	23:21	Destination transfer Width. Transfers wider than the AHB master bus width are illegal. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data when required.		
	S_WIDTH	20:18	Source transfer Width. Transfers wider than the AHB master bus width are illegal. The source and destination widths can be different from each other. The hardware automatically packs and unpacks the data when required.		
	DB_SIZE	17:15	Destination Burst Size.		

			Indicates the number of transfers that make up a destination burst transfer request. This value must be set to the burst size of the destination peripheral or set to the memory when the destination is memory. The burst size is the amount of data that is transferred when the DMACBREQ signal goes active in the destination peripheral.		
	SB_SIZE	14:12	Source Burst Size. Indicates the number of transfers that make up a source burst. This value must be set to the burst size of the source peripheral or set to the memory boundary size when the source is memory. The burst size is the amount of data that is transferred when the DMACBREQ signal goes active in the source peripheral.		
	TRNSFER_SIZE	11:0	Transfer Size. A write to this field sets the size of the transfer when the DMA controller is the flow controller. This value counts down from the original value to zero, and so its value indicates the number of transfers left to complete. (…continued in the next page)		
	TRNSFER_SIZE	11:0	(…) A read from this field provides the number of transfers still to be completed on the destination bus. Reading the register when the channel is active does not give useful information because by the time the software has processed the value read, the channel might have progressed, and then disabled. Program the transfer size value to zero if the DMA controller is not the flow controller. If TRANSFER_SIZE is programmed to a non-zero value, the DMA controller might attempt to use this value instead of ignoring the TRNSFER_SIZE.		
0x110	**CH_CONFIGURATION**	**31:0**	**DMA Channel Configuration Register**	**0x00**	**RW**
	-	31:19	*Reserved. Read undefined. Write to zero.*		
	H	18	Halt. 0 = enable DMA requests 1 = ignore subsequent source DMA requests. The contents of the channel FIFO are drained. This value can be used with the Active and Channel Enable bits to clearly disable a DMA channel.		
	A	17	Active. 0 = there is no data in the FIFO channel 1 = the FIFO channel has data. This value can be used with the Halt and Channel Enable bits to clearly disable a DMA channel. Wiring to this bit has no effect.		
	L	16	Lock. When set, this value enables locked transfers.		
	ITC	15	Terminal Count Interrupt Mask. When cleared, this bit masks out the terminal count interrupt of the relevant channel.		
	IE	14	Interrupt Error Mask. When cleared, this bit masks out the error interrupt of the relevant channel.		
	FLOW_CNTRL	13:11	Flow Control and Transfer Type. This value indicates the flow controller and transfer type. The flow controller can be: · DMA controller · Source peripheral · Destination peripheral The transfer type can be: · Memory-to-Memory		

			・ Memory-to-Peripheral ・ Peripheral-to-Memory ・ Peripheral-to-Peripheral		
	-	10	Reserved. Read undefined. Write to zero.		
	DEST_PERIPHERAL	9:6	Destination Peripheral. This value selects the DMA destination request peripheral. This field is ignored if the destination of the transfer is to memory.		
	-	5	Reserved. Read undefined. Write to zero.		
	SRC_PERIPHERAL	4:1	Source Peripheral. This value selects the DMA source request peripheral. This field is ignored if the source of the transfer is from memory.		
	E	0	Channel Enabled. Reading this bit indicates whether a channel is currently enabled or disabled: 0 = channel disabled 1 = channel enabled. Clear the Enable bit to disable a channel. This causes the current AHB transfer to complete (when one is in progress) and loss of any data in the channel's FIFO. Restarting the channel by setting the Channel Enable bit has unpredictable effects, and the channel must be fully re-initialized. In case when the last LLI is reached or when a channel error is encountered, the channel is disabled, and the Channel Enable bit is cleared too. To disable a channel without losing data in a channel's FIFO, the Halt bit must be set so that subsequent DMA requests are ignored. Then the Active bit must be polled until it reaches 0. This indicates that there is no data left in the channel's FIFO. At last, the Channel Enable bit can be cleared.		
0xFE0	PERIPH_ID_0	31:8	*Reserved. Read undefined.*	-	-
	PART_NUMBER_0	7:0	These bits read back as 0x11.	0x81	RO
0xFE4	PERIPH_ID_1	31:8	*Reserved. Read undefined.*		-
	DESIGNER_0	7:4	These bits read back as 0x1.	0x1	RO
	PART_NUMBER_1	3:0	These bits read back as 0x0.	0x0	RO
0xFE8	PERIPH_ID_2	31:8	*Reserved. Read undefined.*	-	-
	REVISION	7:4	These bits read back as 0x1.	0x1	RO
	DESIGNER_1	3:0	These bits read back as 0x4.	0x0	RO
0xFEC	PERIPH_ID_3	31:8	*Reserved. Read undefined.*	-	-
	CONFIGURATION	7	Indicates the number of DMA source requestors for the DMA controller configuration: 0 =16 DMA requestors, 1 = 32 DMA requestors. This set to 0.	0x00	RO
	CONFIGURATION	6:4	Indicates the AHB master bus width: 000 = 32-bit wide, 001 = 64-bit wide, 010 = 128-bit wide, 011 = 256-bit wide, 100 = 512-bit wide, 101 = 1024-bit wide. This is set to 000.	0x0	RO
	CONFIGURATION	3	Indicates the number of AHB masters: 0 = one AHB master interface, 1 = two AHB master interfaces. This is set to 0.	0x0	RO
	CONFIGURATION	2:0	Indicates the number of channels: 000 = 2 channels, 001 = 4 channels,	0x0	RO

			010 = 8 channels, 011 = 16 channels, 100 = 32 channels. This is set to 000.		
0xFF0	PCELL_ID_0	15:8	*Reserved. Read undefined.*	-	-
	DMAC_PCELL_ID_0	7:0	These bits read back as 0x0D.	0x0D	RO
0xFF4	PCELL_ID_1	15:8	*Reserved. Read undefined.*	-	-
	DMAC_PCell_ID_1	7:0	These bits read back as 0xF0.	0xF0	RO
0xFF8	PCell_ID_2	15:8	*Reserved. Read undefined.*	-	-
	DMAC_PCell_ID_2	7:0	Theses bits read back as 0x05.	0x05	RO
0xFFC	PCell_ID_3	15:8	*Reserved. Read undefined.*	-	-
	DMAC_PCell_ID_3	7:0	These bits read back as 0xB1.	0xB1	RO
0x500	DMAC_IT_CR	31:1	*Reserved. Read undefined.*		
	T	0	Test mode enable. Multiplex the test registers to control the input and output lines: 0 = normal operation, 1 = test registers muxed onto input and output.	0x00	RW
0x504	DMAC_IT_OP_1	31:16	*Reserved. Read undefined.*	-	-
	DMAC_CLR	15:0	DMAC_CLR[15:0] response outputs can be set to a certain value in test mode by writing to the register. A read returns the value on the outputs after the test multiplexor.	0x00	RW
0x508	DMAC_IT_OP_2	31:16	*Reserved. Read undefined.*	-	-
	DMAC_TC	15:0	DMAC_TC[15:0] response outputs can be set to a certain value in test mode by writing to the register. A read returns the value on the outputs after the test multiplexor.	0x0	RW
0x50C	DMAC_IT_OP_3	31:2	*Reserved. Read undefined.*	-	-
	E	1	DMAC_INTERR interrupt request can be set to a certain value in test mode by writing to the register. A read returns the value on the output after the test multiplexor.	0x0	RW
	TC	0	DMAC_INT_TC interrupt request to a certain value in test mode by writing to the register. A read returns the value on the output after the test multiplexor.	0x0	RW

2. SECURITY BLOCK

The Security Block consists of the ACA, SCA and TRNG

2.1 ACA (Asymmetrical Crypto Accelerator)

The ACA is designed to offload large arithmetic operations from the host CPU.   The host only needs to load the parameters in the ACA and specify the appropriate function to initiate the autonomous program. No further communication between the ACA and the system is required until the ACA finishes performing the requested operation.

The large arithmetic operations performed by the ACA provide acceleration of the following PKI operations.

1) RSA (with or without CRT): 512, 768, 1024, 1536, 2048-bit
2) ECC-GF(p): 160, 192, 224, 256, 384, 512-bit

2.2 SCA (Symmetric Crypto Accelerator):

SCA performs parallel hashing and encryption/decryption operations with hardwired acceleration engine. The SCA provides a framework including a programmable sequencer, DMA engine, and cryptographic/hashing resources that may handle a variety of protocols involving a cipher and a hash. The SCA also optimizes cryptographic offload for bulk processing of cipher and hash algorithms, as well as combined mode algorithms (e.g. GCM, CCM).

The features supported by SCA are listed below:

- Cipher algorithms: AES
- Cipher modes: ECB, CBC, CTR, CCM, GCM
- Hash (MAC) algorithms: SHA-1, SHA-256
- Hash modes: Raw HMAC.
- 4 independent cryptographic contexts
- FIFO based command/status interface which allows queuing of multiple packets by the host CPU
- Full DMA master which uses data descriptor tables to automatically read and write fragmented data packets to be processed. Packet data traverses the bus only twice per packet (once for the read operation, once for the write operation)
- Secure key port for block cipher algorithms (AES).

2.3 ARIA

1) The ARIA(Academy, Research Institute, Agency) module is Korea standard security algorithm and support 128/192/256bit key length and encrypt/decrypt input data with key value
    - ARIA Hardwire Engine with key length 128, 192, 256-bit
    - Can Operate with DMA engine to high speed operation
    - Highly optimized hardware engine without internal memory

2.4 TRNG

TRNG(True Random Number Generator) generate random data that is statically equivalent to a uniformly distributed random data stream.

- Continuous random bit stream generation
- Automatic seed/reseed from internal random noise source
- Manual seed/reseed by host CPU
- HIGH Speed 25Mbps at 100MHz

3. WATCHDOG TIMER

The MS500 includes one Watchdog Timer Module. The module consists of a 32-bit down counter with a programmable timeout interval that has the capability to generate an interrupt and a reset signal on timing out. It is used to apply a reset to a system in the event of a software failure.

The Watchdog Timer Module contains the following features:

- Programmable 16-bit timer with internal prescaler.
- Optional windowed operation requires reload to occur between a minimum and maximum timeout period, both are programmable.
- Optional warning interrupt can be generated at a programmable time watchdog timeout.
- Supports locking mechanism.
- Uses clock independent with bus clock.

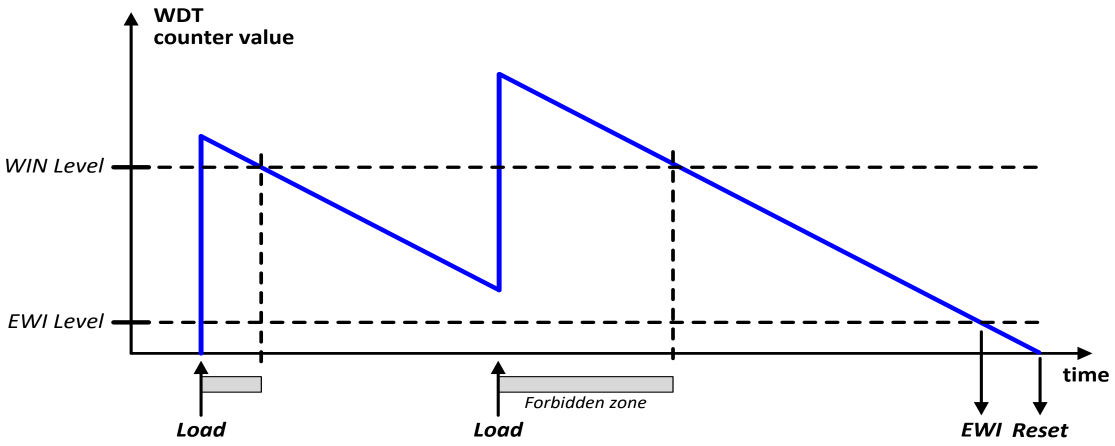

Figure 19 WDT Operation Timing Diagram

The Watchdog module interrupt and reset generation can be enabled or disabled as required by use of the Control Register, WdogControl. When the interrupt generation is disabled then the counter is stopped.

When the interrupt is re-enabled then the counter starts from the value programmed in WdogLoad, and not from the last count value.

Table 29 Watchdog Timer Register Description

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
0x00	**WDT_CTRL**	16	15:0	**Watchdog Timer Control Register**	0x0	RW
	WDT_EN	1	0	Watchdog Timer Enable		
	EWI_EN	1	1	Early Wakeup Interrupt Enable		
	WWDT_EN	1	2	Windowed WDT Enable		
	TEST_EN	1	3	Test Mode Enable Note: If enabled, test interrupt is occurred when watchdog timer is expired		
	TMR_BASE	4	7:4	Watchdog Timer Base 0x0: divided by 1 (not divided) 0x1: divided by 2 ... N: divided by 2^N ...f 0xC: divided by 2^12 (= 4096) Otherwise: *reserved*		
	-	8	15:8	*Reserved*		
0x04	**WDT_STS**	16	15:0	**Watchdog Timer Status Register**		
	EWI_STS	1	0	Early Wakeup Interrupt Status		ROAC
	TEST_INT	1	1	Test Interrupt Status Note: This is working when **TEST_EN** is set		ROAC
	WDT_CNT	14	15:2	Current Bits[15:2] of Watchdog Timer Count		ROC
0x08	**WDT_LOAD**	16	15:0	**Watchdog Timer Load Value Register**		RW
0x0C	**WDT_EWI**	16	15:0	**Watchdog Early Wakeup Interrupt Value Register**		RW
0x10	**WDT_WIN**	16	15:0	**Watchdog Window Value Register**		RW
0x14	**WDT_LOCK**	16	15:0	**Watchdog Timer Lock Register** For registration of Lock CODE (any 15-bits value): [Sequence 1] Write **LOCK_CODE**[14:0] with **LOCK_KEY** = 1b [Sequence 2] Write **LOCK_CODE** (of sequence 1) with **LOCK_KEY** = 0b [Sequence 3] Write **LOCK_CODE** (of sequence 1) with **LOCK_KEY** = 1b [Sequence 4] Read **LOCK_KEY** (optional) For write access to WDT register with Lock Code: [Sequence 1] Write **LOCK_CODE** (of sequence 1) with **LOCK_KEY** = 0b [Sequence 2] Write WDT register Note: Writing to other register is allowed one time right after registered lock code is writed Note: If lock code is not registered write protection doesn't work		RW
	LOCK_CODE	15	14:0	WDT Register Lock Code		
	LOCK_KEY	1	15	Lock Sequence when writing: lock sequence key when reading: lock sequence status - 1b means activated		
0x18	**PERI_ID0**	16	15:0	**WDT Peripheral ID0 Register**	HwInit	RO
	PART_NUM	12	11:0	WDT Part Number	0x020	
	CONFIG_NUM	4	15:12	WDT Configuration Number	0x1	
0x1C	**PERI_ID1**	16	15:0	**WDT Peripheral ID1 Register**	HwInit	RO
	MAJ_REV	4	3:0	WDT Major Revision		
	MIN_REV	4	7:4	WDT Minor Revision		

	RSVD_NUM	8	15:8	WDT Reserved Number		

4. GENERAL-PURPOSE TIMERS

The MS500 contains two General-Purpose Timer Module (GPTM) blocks. The GPTM consists of two programmable 32/16-bit down counters that can generate interrupts or when reaching zero.

The GPTM supports the following features:

- Two 32/16-bit down counters with free-running, periodic and one-shot modes.
- Common clock with separate clock-enables for each timer gives flexible control of the timer intervals.
- Interrupt output generation on timer count reaching zero.

The GPTM consists of two identical programmable Free Running Counters (FRCs) that can be configured for 32-bit or 16-bit operation and one of three timer modes:

1) Free-running:
   The counter operates continuously and wraps around to its maximum value each time that it reaches zero.
2) Periodic:
   The counter operates continuously by reloading from the Load Register each time that the counter reaches zero.
3) One-shot:
   The counter is loaded with a new value by writing to the Load Register. The counter decrements to zero and then halts until it is reprogrammed.

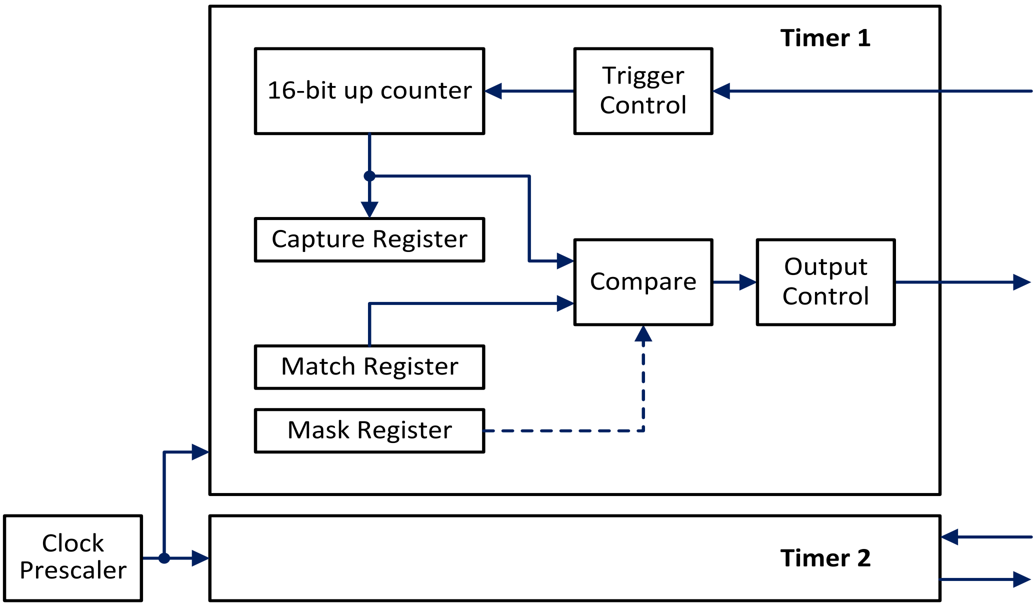

Figure 20 Timer Block Diagram

Table 30 General-Purpose Dual Timer Register Description

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
0x00	**TMR1_CTRL**	16	15:0	**Timer 1 Control Register**		RW
	TMR1_MODE	2	1:0	Timer 1 Mode 00: Periodic (default) 01: One-shot (Stop on match) 10: Free-running 11: *reserved*		
	TMR1_TRIG_MODE	2	3:2	Trigger Mode for Timer 1 00: Not Used (default) 01: Capture (edge-triggered) 10: Timer Clock (edge-triggered) 11: Timer Enable (level-sensitive) Note: If mode 2 or mode 3 is seleted, prescaler is not used		
	TMR1_TRIG_POL	1	4	Trigger Polarity for Timer 1 0: Low / Falling-edge, 1: High / Rising-edge		
	TMR1_TRIG_SEL	1	5	Trigger Select for Timer 1 0: External (default), 1: Internal		
	TMR1_MATCH_OUT	1	6	Match Output Value for Timer 1 Note: Read value is the current state of match output		
	-	1	7	*Reserved*		
	TMR1_OUT_MODE	2	9:8	Match Output Mode for Timer 1 00: Do nothing (default) 01: High 10: Low 11: Toggle		
	-	6	15:10	*Reserved*		
0x04	**TMR1_MATCH**	16	15:0	**Timer 1 Match Value Register** Note: Writing to this also reset counter		RW
0x08	**TMR1_MASK**	16	15:0	**Timer 1 Mask Value Register** Note: A 1 written to any b it will force the compare to be true on the corresponding bit of **TMR1_MASK**		RW
0x0C	**TMR1_CNT**	16	15:0	**Timer 1 Counter Status Register**		RO
0x10	**TMR1_TSTAMP**	16	15:0	**Timer 1 Time Stamp Register**		RO
0x14 ~0x1F	***reserved***	-	-	-	-	Rsvd
0x20	**TMR2_CTRL**	16	15:0	**Timer 2 Control Register** ~ *Same as the description of the corresponding register for Timer 1*		RW
0x24	**TMR2_MATCH**	16	15:0	**Timer 2 Match Value Register** ~ *Same as the description of the corresponding register for Timer 1*		RW
0x28	**TMR2_MASK**	16	15:0	**Timer 2 Mask Value Register** ~ *Same as the description of the corresponding register for Timer 1*		RW
0x2C	**TMR2_CNT**	16	15:0	**Timer 2 Counter Status Register** ~ *Same as the description of the corresponding register for Timer 1*		RO
0x30	**TMR2_TSTAMP**	16	15:0	**Timer 2 Time Stamp Register** ~ *Same as the description of the corresponding register for Timer 1*		RO
0x34 ~0x3F	***reserved***	-	-	-	-	Rsvd
0x40	**TMR_CTRL**	16	15:0	**Timer Control Register**	0x0	RW
	TMR_CNT_RST	1	0	Timer Counter Reset Note: This also reset prescaler counter		RWAC
	MST_TMR_SEL	1	1	Master Timer Select 0: Timer 1, 1: Timer 2 Note: Only **TMRx_CTRL** and its trigger input is used for cascaded or combined mode		

	TMR_LINK_MODE	2	3:2	Timer Link Mode 00: Disabled - Separate Mode (default) 01: Cascaded Mode 10: Combined Mode 11: *Reserved* Note: If combined mode is set, slave timer is stopped at interrupt event of master timer		
	TMR_PRES	4	7:4	Timer Prescaler Value 0x0: divided by 1 (not divided) 0x1: divided by 2 ... N: divided by 2^N ... 0xA: divided by 2^10 (= 1024) Otherwise: *reserved*		
	-	8	15:8	*Reserved*		
0x44	**TMR_EN**	16	15:0	**Timer Enable Register**		RW
	TMR_EN[x]	2	1:0	Timer x (= bit position + 1) Enable Note: If disabled, timer count is stopped without reset Note: If timer is stopped on match (one-shot mode), this is also cleared		
	-	14	15:3	*Reserved*		
0x48	**TMR_INT_EN**	16	15:0	**Timer Interrupt Enable Register** Note: Each bit is corresponding to each bit of **TMR_INT_STS** register. 1b is enabled state, thus enabled signal source is used for interrupt generation		RW
0x4C (Read)	**TMR_INT_STS**	16	15:0	**Timer Interrupt Status Register** Note: The only bit enabled by **TMR_INT_EN** can be set, otherwise it is read as zero Note: Timer interrupt can be occurred only when timer is enabled by TMRx_EN	-	RW1C
	TMR1_MAT_INT	1	0	Timer 1 Match Interrupt		
	TMR2_MAT_INT	1	1	Timer 2 Match Interrupt		
	-	2	3:2	*Reserved*		
	TMR1_CAP_INT	1	4	Timer 1 Capture Interrupt		
	TMR2_CAP_INT	1	5	Timer 2 Capture Interrupt		
	-	10	15:6	*Reserved*		
0x4C (Write)	**TMR_INT_CLR**	16	15:0	**Timer Interrupt Clear Register** Note: Each bit is corresponding to each bit of **TMR_INT_STS** register. Writing 1b to a bit clear the bit, which of **TMR_INT_STS** is in same position		
0x50 ~0x77	***reserved***	-	-	-	-	Rsvd
0x78	**PERI_ID0**	16	15:0	**Timer Peripheral ID0 Register**	HwInit	RO
	PART_NUM	12	11:0	Timer Part Number	0x040	
	CONFIG_NUM	4	15:12	Timer Configuration Number		
0x7C	**PERI_ID1**	16	15:0	**Timer Peripheral ID1 Register**	HwInit	RO
	MAJ_REV	4	3:0	Timer Major Revision		
	MIN_REV	4	7:4	Timer Minor Revision		
	RSVD_NUM	8	15:8	Timer Reserved Number		

5. RTC

The Real-Time Clock (RTC) is a set of counters for measuring time when system power is on. It uses minimal power when the CPU does not access its registers. A separate RTC crystal 32.768 KHz provide to RTC module.

The Real-Time Clock includes:

1) The real-time clock (RTC)
2) Backup SRAM to store system information by CPU before power down mode

The MS500 provides the Real-Time Clock with the following features:

- Measures the passage of time to maintain a calendar and clock. Provides the second, minute, hour (in 12/24 hour), weekday, date, month, year.
- The time and calendar values are coded in binary-coded decimal (BCD) format.
- Ultra-low power design to support always on power domain.
- Supports time stamp events.
- Alarm interrupt can be generated for a specific date/time.
- Programmable alarm with automatic wakeup from Stop and Standby modes.

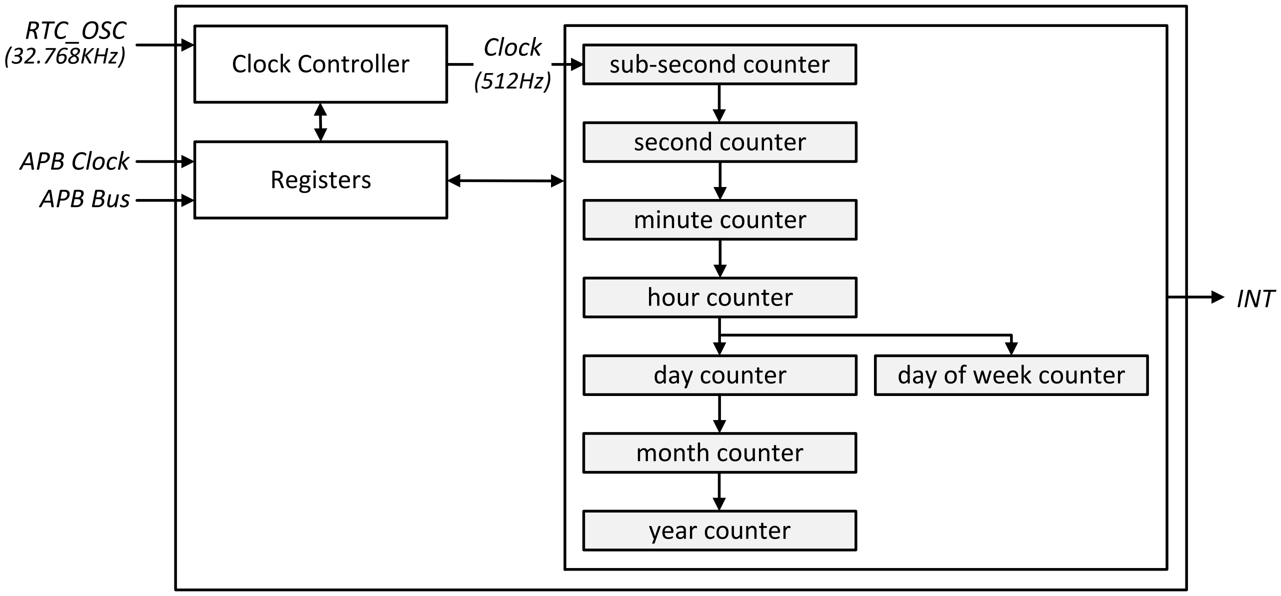

Figure 21 RTC Block Diagram

Table 31 RTC Register Description

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
Note: Read access to **RTC_DATE**, **RTC_TIME** and **RTC_SUB_SEC** has 2~3 wait cycles by PREADY of APB3						
0x00	RTC_CLK_DIV	32	31:0	RTC Clock Divider Value Register	0x20	RW
0x04	**RTC_CTRL**	32	31:0	**RTC Control Register**		RW
	RTC_INIT	1	0	RTC Initialization Note: When this bit is set to 1, RTC is in initialization state, and the time, date counters can be updated with RTC_TIME, RTC_DATE register value.		
	ALARM_EN	1	1	Alarm Enable Note: ONE_SHOT_ALARM, RTC_ALARM_MASK, RTC_ALARM_TIME and RTC_ALARM_DATE must be set before ALARM_EN is set to 1.		
	ONE_SHOT_ALARM	1	2	One-shot Alarm		
	WAKEUP_EN	1	3	Wakeup Timer Enable Note: RTC_WAKEUP_CNT must be set before WAKEUP_EN is set to 1.		
	TSTAMP_EN	1	4	Timestamp Enable		
	-	1	5	*Reserved*		Rsvd
	SUB_SEC_DIS	1	6	Sub Second Counter Disable Note: Sub second counter disable when internal clock frequency is 1Hz.		
	-	1	7	*Reserved*		Rsvd
	-	4	11:8	*Reserved*		Rsvd
	BKSRAM_MODE	2	13:12	RTC-domain Backup SRAM Mode 00: Normal (default) 01: *Reserved* 10: Data Retention Mode 11: *Reserved*		
	-	18	31:14	*Reserved*		Rsvd
0x08	**RTC_INT_EN**	32	31:0	**RTC Interrupt Mask Register** Note: Each bit is corresponding to each bit of **INT_STS** register. High(1) is unmasked state, thus unmasked signal source is used for interrupt generation		
0x0C	**RTC_INT_STS**	32	31:0	**RTC Interrupt Status Register**		RW1C
(Read)	ALARM	1	0	Alarm Interrupt ~ time/date registers(**RTC_TIME**, **RTC_DATE**) match the alarm time/date registers(**RTC_TIME_ALARM**, **RTC_DATE_ALARM**)		
	WAKEUP	1	1	Wakeup Auto-reload Counter Reaches 0		
	TSTAMP_TRIG	1	2	Timestamp Event Occured		
	-	29	31:3	*Reserved*		
0x0C (Write)	**RTC_INT_CLR**	32	31:0	**RTC Interrupt Clear Register** Note: Each bit is corresponding to each bit of **INT_STS** register. Writing High(1) to the specific bit will clear interrupt status of the bit position		
0x10	**RTC_ALARM_MASK**	32	31:0	**RTC Alarm Mask Register**		RW
	SECOND_MASK	1	0	Second value is not compared for the alarm		
	MINUTE_MASK	1	1	Minutes value is not compared for the alarm		
	HOUR_MASK	1	2	Hour value is not compared for the alarm		
	DAY_MASK	1	3	Day value is not compared for the alarm		
	MONTH_MASK	1	4	Month value is not compared for the alarm		
	YEAR_MASK	1	5	Year value is not compared for the alarm		

	DAY_OF_WEEK_MASK	1	6	DAY of Week value is not compared for the alarm		
	-	25	31:7	*Reserved*		Rsvd
0x14	**RTC_WAKEUP_CNT**	32	31:0	**RTC Wakeup Register** Note: Wakeup auto-reload value. Valid bit range is lower 16 bits.		RW
0x18	**RTC_CALIB**	32	31:0	**RTC Calibration Register**		RW
	CALIB_CNT	16	15:0	Calibration counter value		
	CALIB_DIR	1	16	Calibration direction 0: Forward calibration 1: Backward calibration ~ Forward calibration: When calibration counter value is equal **CALIB_CNT**, the RTC timers will jump by 2 seconds. Backward calibration: When calibration counter value is equal **CALIB_CNT**, the RTC timers will stop incrementing for 1 second		
	-	15	31:17	*Reserved*		Rsvd
0x1C	**RTC_TIME**	32	31:0	**RTC Time Register**		RW
	SEC	4	3:0	Seconds in BCD format		
	TEN_SEC	3	6:4	Ten seconds in BCD format		
	-	1	7	*Reserved*		Rsvd
	MIN	4	11:8	Minutes in BCD format		
	TEN_MIN	3	14:12	Ten minutes in BCD format		
	-	1	15	*Reserved*		Rsvd
	HOURS	4	19:16	Hours in BCD format		
	TEN_HOURS	2	21:20	Ten hours in BCD format		
	-	10	31:22	*Reserved*		Rsvd
0x20	**RTC_DATE**	32	31:0	**RTC Date Register**		RW
	DAYS	4	3:0	Days in BCD format		
	TEN_DAYS	2	5:4	Ten days in BCD format		
	-	2	7:6	*Reserved*		Rsvd
	MONTHS	4	11:8	Months in BCD format		
	TEN_MONTHS	1	12	Ten months in BCD format		
	DAY_OF_WEEK	3	15:13	Day of week 000: forbidden 001: Sunday ... 111: Saturday		
	YEARS	4	19:16	Years in BCD format		
	TEN_YEARS	4	23:20	Ten years in BCD format		
	CENT	4	27:24	Centuries in BCD format		
	TEN_CENT	4	31:28	Ten centuries in BCD format		
0x24	**RTC_SUB_SEC**	32	31:0	**RTC Sub Second Register** Note: Valid bit range is lower 9 bits.		RO
0x28	**RTC_ALARM_TIME**	32	31:0	**RTC Alarm Time Register**		RW
	ALARM_SEC	4	3:0	Seconds in BCD format		
	ALARM_TEN_SEC	3	6:4	Ten seconds in BCD format		
	-	1	7	*Reserved*		Rsvd
	ALARM_MIN	4	11:8	Minutes in BCD format		
	ALARM_TEN_MIN	3	14:12	Ten minutes in BCD format		
	-	1	15	*Reserved*		Rsvd
	ALARM_HOURS	4	19:16	Hours in BCD format		
	ALARM_TEN_HOURS	2	21:20	Ten hours in BCD format		
	-	10	31:22	*Reserved*		Rsvd
0x2C	**RTC_ALARM_DATE**	32	31:0	**RTC Alarm Date Register**		RW
	ALARM_DAYS	4	3:0	Days in BCD format		
	ALARM_TEN_DAYS	2	5:4	Ten days in BCD format		
	-	2	7:6	*Reserved*		Rsvd
	ALARM_MONTHS	4	11:8	Months in BCD format		

	ALARM_TEN_MONTHS	1	12	Ten months in BCD format		
	ALARM_DAY_OF_WEEK	3	15:13	Day of week 000: forbidden 001: Sunday ... 111: Saturday		
	ALARM_YEARS	4	19:16	Years in BCD format		
	ALARM_TEN_YEARS	4	23:20	Ten years in BCD format		
	ALARM_CENT	4	27:24	Centuries in BCD format		
	ALARM_TEN_CENT	4	31:28	Ten centuries in BCD format		
0x30	**RTC_TS_TIME**	32	31:0	**RTC Timestamp Time Register**		RO
	TS_SEC	4	3:0	Seconds in BCD format		
	TS_TEN_SEC	3	6:4	Ten seconds in BCD format		
	-	1	7	*Reserved*		Rsvd
	TS_MIN	4	11:8	Minutes in BCD format		
	TS_TEN_MIN	3	14:12	Ten minutes in BCD format		
	-	1	15	*Reserved*		Rsvd
	TS_HOURS	4	19:16	Hours in BCD format		
	TS_TEN_HOURS	2	21:20	Ten hours in BCD format		
	-	10	31:22	*Reserved*		Rsvd
0x34	**RTC_TS_DATE**	32	31:0	**RTC Timestamp Date Register**		RO
	TS_DAYS	4	3:0	Days in BCD format		
	TS_TEN_DAYS	2	5:4	Ten days in BCD format		
	-	2	7:6	*Reserved*		Rsvd
	TS_MONTHS	4	11:8	Months in BCD format		
	TS_TEN_MONTHS	1	12	Ten months in BCD format		
	TS_DAY_OF_WEEK	3	15:13	Day of week 000: forbidden 001: Sunday ... 111: Saturday		
	TS_YEARS	4	19:16	Years in BCD format		
	TS_TEN_YEARS	4	23:20	Ten years in BCD format		
	TS_CENT	4	27:24	Centuries in BCD format		
	TS_TEN_CENT	4	31:28	Ten centuries in BCD format		
0x38	**RTC_TS_SUB_SEC**	32	31:0	**RTC Timestamp Sub-second Register**		RO
				Note: Valid bit range is lower 9 bits.		
0x3C	**RTC_PERI_ID**	32	31:0	**RTC Peripheral ID Register**	HwInit	RO
	PART_NUM	12	11:0	RTC Part Number	0x030	
	CONFIG_NUM	4	15:12	RTC Configuration Number	0x1	
	MAJ_REV	4	19:16	RTC Major Revision		
	MIN_REV	4	23:20	RTC Minor Revision		
	RSVD_NUM	8	31:24	RTC Reserved Number		

6. GENERAL-PURPOSE I/O

The MS500 provides eight GPIO ports with up to 41 GPIO pins for each GPIO port.

The GPIO pins can be used in several ways to set pins as inputs or outputs and use the inputs as combinations of level and edge sensitive interrupts.

6.1 GPIO pin interrupt features

- Up to 16 pins can be selected from all GPIO pins as edge- or level-sensitive interrupt requests. Each request creates a separate interrupt in the NVIC.
- AHB interface to support fast toggle changing every tow clock cycle.
- Bit set/reset and toggle register.
- Edge-sensitive interrupt pins can interrupt on rising or falling edges or both.
- Level-sensitive interrupt pins can be configured as active HIGH or LOW active.
- Provides logically combined interrupt through OR and AND operation.
- Separated bus interface and free running clocks to support CPU sleep.

6.2 GPIO group interrupt features

- The inputs from any number of GPIO pins can be enabled to contribute to a combined group interrupt.
- The polarity of each input enabled for the group interrupt can be configured HIGH or LOW.
- Enabled interrupts can be logically combined through an OR/AND operation.
- Two group interrupts are supported to reflect two distinct interrupt patterns.
- The GPIO group interrupts can wake up from sleep/deep sleep modes.

6.3 GPIO port features

- GPIO pins can be configured as input or output by software.
- All GPIO pins default to inputs with interrupt disabled at reset.
- Pin registers allow pins to be sensed and set individually.

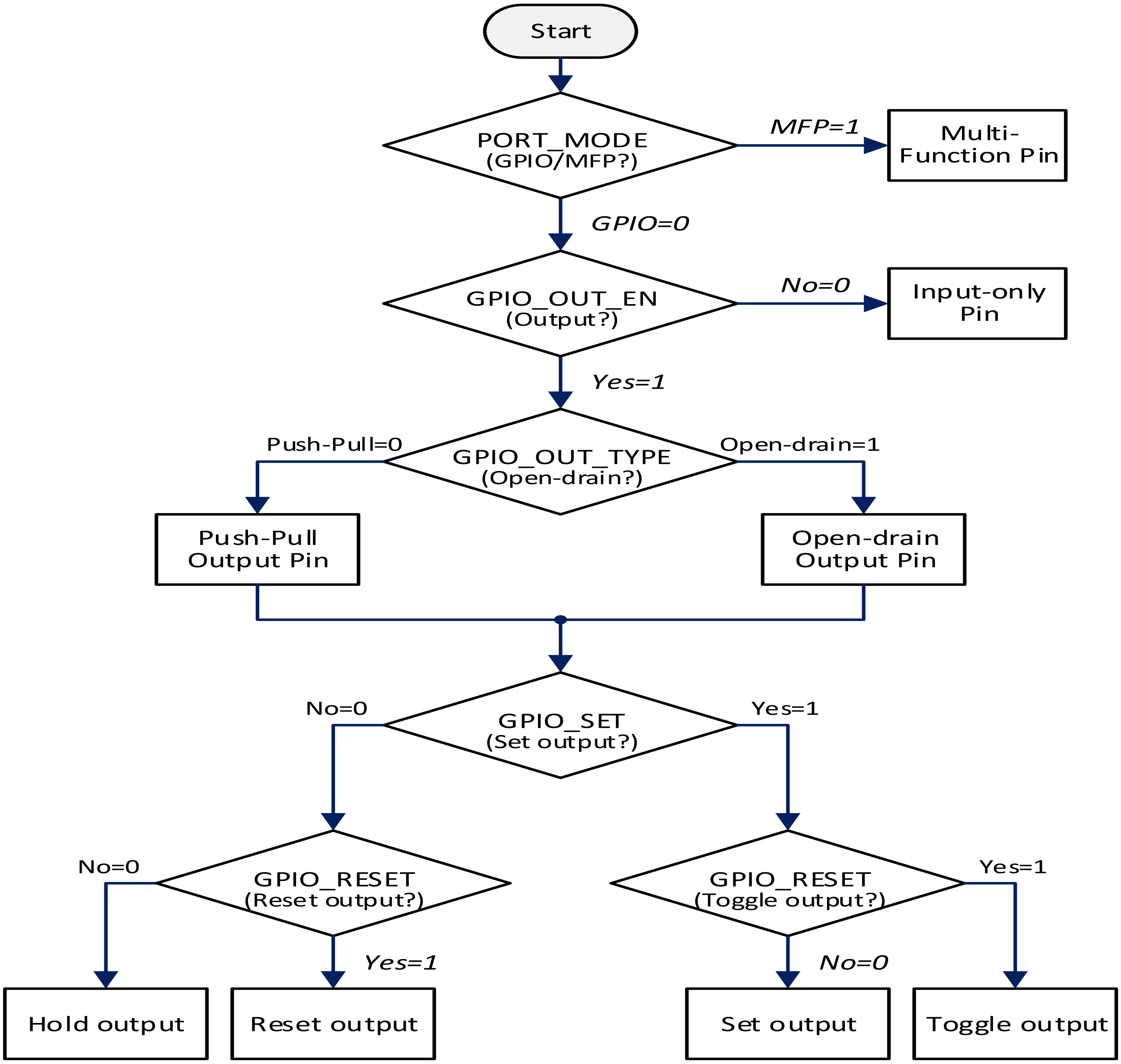

Figure 22 GPIO Pin Control

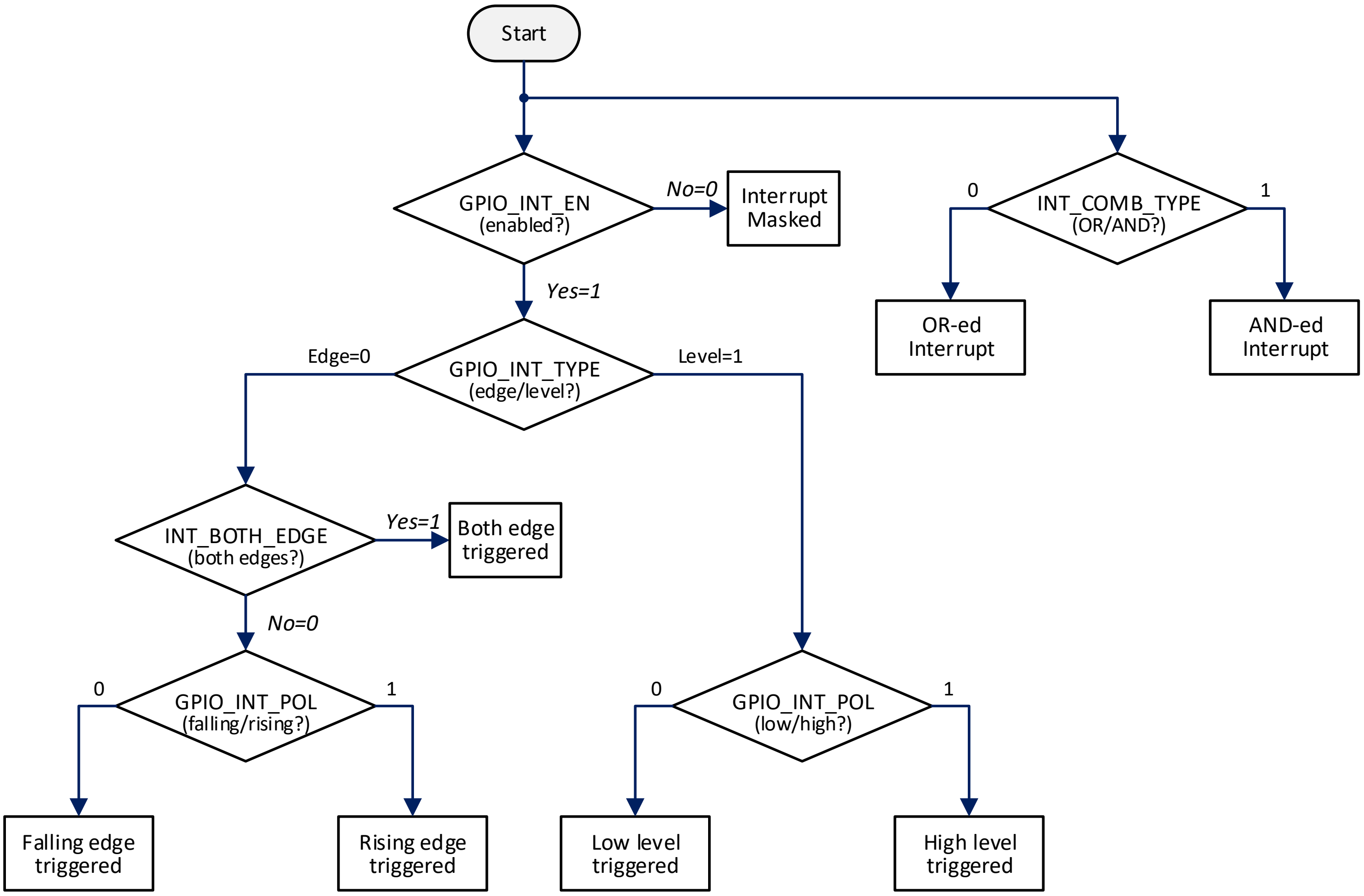

Figure 23 GPIO Interrupt Register

Table 32 GPIO Register Summary

Offset	Symbol	Size	Bit	Brief Description	Reset Value	Type
0x00	**PORT_MODE**	32	31:0	**Port Mode Register** Readback value of Multi-function Port Mode set by SysCon registers 0: GPIO (default), 1: Multi-function mode Note: System reset values are 0x00000038 for port A, 0x00008000 for port B and port C. Otherwise reset value is all zero. Note: Lower 16 bits are valid, other bits are reserved		RO
0x04	**GPIO_OUT_MODE**	32	31:0	**GPIO Port Output Mode Register** Note: These bits are valid when the corresponding bits of **GPIO_PORT_MODE** is 0b		RW
	GPIO_OUT_EN	16	15:0	Output Enable for port x (= bit position) 0: Disable (Input Mode) 1: Enable		
	GPIO_OUT_TYPE	16	31:16	Output Type for port x (= bit position - 16) 0: Push-pull (default) 1: Open-drain Note: These bits are valid when the corresponding bits of **GPIO_OUT_EN** is set		

0x08	**GPIO_DATA**	32	31:0	**GPIO Port Data Register** Note: These bits are read from I/O pad independent of port function and direction setting. But interrupt generation is used by GPIO input mode signal only Note: Lower 16 bits are valid, other bits are reserved		RO
0x0C	**GPIO_DATA_OUT**	32	31:0	**GPIO Port Data Output Register** Note: Read value is the readback of write value Note: Lower 16 bits are valid, other bits are reserved		RW
0x10	**GPIO_SET_RESET**	32	31:0	**GPIO Port Set/Reset Register** This register can simultaneously set and clear for different bits. Note: If both set and reset is requested for the same bits, the corresponding bits are toggled		WOAC
	GPIO_SET	16	15:0	Set output of port x (= bit position)		
	GPIO_RESET	16	31:16	Clear output of port x (= bit position - 16)		
0x14	***reserved***	32	31:0	*Reserved*	-	Rsvd
0x18	**GPIO_INT_TYPE**	32	31:0	**GPIO Port Interrupt Type Register** 0: Edge-triggered 1: Level sensitive Note: Lower 16 bits are valid, other bits are reserved		RW
0x1C	**GPIO_INT_POL**	32	31:0	**GPIO Port Interrupt Polarity Register**		RW
	GPIO_INT_POL[x]	16	15:0	Interrupt polarity for port x (= bit position) These bits have different meanings depending on **GPIO_INT_TYPE** For edge-triggered mode, 0: Falling-edge, 1: Rising-edge For level sensitive mode, 0: Low, 1: High		
	INT_BOTH_EDGE	16	31:16	Both edge enable for port x (= bit position - 16) 0: Disable (default) 1: Enable (Both Edge) Note: These bits are valid when the corresponding bits of **GPIO_INT_TYPE** is zero		
0x20	**GPIO_RAW_STS**	32	31:0	**GPIO Port Interrupt Raw Status Register** Note: each interrupt event is always latched and maintained until cleared by reading Note: Lower 16 bits are valid, other bits are reserved	-	ROAC
0x24	**GPIO_INT_EN**	32	31:0	**GPIO Port Interrupt Enable Register**		RW
	GPIO_INT_EN[x]	16	15:0	Interrupt enable for port x (= bit position) Note: Each bit is corresponding to each bit of **GPIO_RAW_STS** register. 1b is enabled state, thus enabled signal source is used for interrupt generation		
	INT_COMB_MODE	1	16	Interrupt combining mode 0: OR-ed 1: AND-ed Note: In OR-ed mode, each interrupt event is always latched and maintained until cleared by the corresponding bit of **GPIO_INT_CLR** Note: In AND-ed mode, interrupt status register is captured when interrupt combining condition is occurred and maintained until cleared by **GPIO_INT_CLR**		
	-	15	31:17	*Reserved*		
0x28 (Read)	**GPIO_INT_STS**	32	31:0	**GPIO Port Interrupt Status Register** Note: Lower 16 bits are valid, other bits are reserved Note: The only bit enabled by **GPIO_INT_EN** can be set from **GPIO_RAW_STS**, otherwise it will be read as zero	-	RW1C
0x28 (Write)	**GPIO_INT_CLR**	32	31:0	**GPIO Port Interrupt Clear Register** Note: Lower 16 bits are valid, other bits are reserved Note: Each bit is corresponding to each bit of **GPIO_INT_STTS** register. Writing 1b to a bit clear the bit, which of **GPIO_INT_STS** is in same position		

0x2C ~0x3B	*reserved*	-	-	-	-	Rsvd
0x3C	**PERI_ID**	32	31:0	**GPIO Peripheral ID Register**	HwInit	RO
	PART_NUM	12	11:0	GPIO Part Number	0x010	
	CONFIG_NUM	4	15:12	GPIO Configuration Number		
	MAJ_REV	4	19:16	GPIO Major Revision		
	MIN_REV	4	23:20	GPIO Minor Revision		
	RSVD_NUM	8	31:24	GPIO Reserved Number		

ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATES

Table 33 Absolute Maximum Ratings for MS500

Parameter	Symbol	Min	Max	Units
I/O supply voltage range	VDVDD	-0.5	3.62	V
Voltage rating at I/O	VPAD	-0.5	3.62	V
Junction operating temperature range	Tj	-40	125	℃

RECOMMENDED OPERATING CONDITIONS

Table 34 Recommended Operating Conditions for MS500

Parameter	Symbol	Min	Typ.	Max	Units
I/O power supply	VDD_IO	3.0 1.7	3.3 1.8	3.6 1.9	V
Internal AON LDO power supply	VDD_AON	3.0 1.7	3.3 1.8	3.6 1.9	V
Internal MAIN LDO power supply	VDD_LDOM	3.0 1.7	3.3 1.8	3.6 1.9	V
SDIO host IO power supply	VDD_SDIOH	3.0 1.7	3.3 1.8	3.6 1.9	V
SDIO device IO Power supply	VDD_SDIOS	3.0 1.7	3.3 1.8	3.6 1.9	V
mFlash power supply	VDD_FLASH	1.7	1.8	1.9	V
eFuse power supply	VDD_eFUSE	2.25	2.5	2.75	V
Operating temperature	Topr	-30	25	85	℃

POWER CONSUMPTION

1) Active/Run mode current

- The major function is the cryptographic operation (AES-CTR) encryption & decryption
- All I/O ports is WEAK0 pull-down by syscon register
- X'TAL and HS_IRC is enabled

Table 35 Run/Active mode current consumption(1.8V)

1.8V@VDD$_{\text{LDO\_M}}$,VDD$_{\text{LDO\_AON}}$,VDDIO and 1.8V@VDDIO$_{\text{DVDD\_FlashR/DVDD\_FlashL}}$, 25°C, Typical value **Run/Active mode**							
Operation Mode	**Flash**	**Clock Source**	**Frequency**	**Current (unit mA)**			
				LDO_M_DVDD LDO_M	**VDD_AON** LDO_AON	**VDDIO & Flash**	
						VDDIO	**Flash Memory**
Run/Active	Flash On	PLL (X'TAL input)	96MHz	14.13	0.127	0.00003	2.62
			72MHz	12.61	0.097	0.00003	2.60
			48MHz	9.60	0.065	0.00001	2.57
			24MHz	4.44	0.034	0.00001	2.45
		Internal HS_IRC	12MHz	2.66	0.019	0.00001	1.95

Table 36 Run/Active mode current consumption(3.3V)

3.3V@VDD$_{\text{LDO\_M}}$,VDD$_{\text{LDO\_AON}}$,VDDIO and 1.8V@VDDIO $_{\text{DVDD\_FlashR/DVDD\_FlashL}}$, 25°C, Typical value **Run/Active mode**							
Operation Mode	**Flash**	**Clock Source**	**Frequency**	**Current (unit mA)**			
				LDO_M_DVDD **LDO_M**	**VDD_AON** **LDO_AON**	**VDDIO & Flash**	
						VDDIO	**Flash Memory**
Run/Active	Flash On	PLL (X'TAL input)	96MHz	16.72	0.131	0.0067	2.68
			72MHz	12.89	0.100	0.0060	2.60
			48MHz	9.81	0.068	0.0060	2.55
			24MHz	4.56	0.036	0.0060	2.45
		Internal HS_IRC	12MHz	2.75	0.021	0.0060	1.95

- The Active mode power consumption varied depends on which module operate and operate frequency and temperature.

2) LPM0, LPM1, LPM2 mode current

- All I/O ports is WEAK0 state controlled by PAD Pull-Down mode.
- When use internal HS-IRC, X'TAL is disabled.
- When use internal LS_IRC(32KHz), X'TAL and HS_IRC is disabled.

Table 37 LPM0/LPM1/LPM2 mode current consumption(1.8V)

1.8V@VDD_{LDO_M},VDD_{LDO_AON},VDDIO and 1.8V@$VDDIO_{DVDD\_FlashR/DVDD\_FlashL}$, 25^{o}C, Typical value LPM0/LPM1/LPM2 NOTE PD^{*1} : Flash power-down mode							
Operatio n Mode	Flash	Clock Source	Frequency	Current (unit mA)			
				LDO_M_DVDD	VDD_AON	VDDIO & Flash	
				LDO_M	LDO_AON	VDDIO	Flash Memory
Low Power Mode 0	Flash On	PLL (XTAL input)	96MHz	2.541	0.003	0.00002	0.018
			72MHz	2.137	0.003	0.00002	0.018
			24MHz	1.354	0.003	0.00002	0.018
		Internal HS_IRC	12MHz	0.680	0.003	0.00002	0.018
			6MHz	0.452	0.003	0.00002	0.018
			1MHz	0.447	0.003	0.00004	0.018
		Internal LS_IRC	32KHz	0.335	0.003	0.00002	0.018
Low Power Mode 1	Flash On (PD^{*1})	PLL (XTAL input)	96MHz	2.251	0.005	0.00002	0.001
			72MHz	1.847	0.005	0.00002	0.001
			12MHz	0.365	0.005	0.00002	0.001
		Internal HS_IRC	6MHz	0.167	0.005	0.00002	0.001
			1MHz	0.163	0.005	0.00002	0.001
		Internal LS_IRC	32KHz	0.051	0.005	0.00002	0.001
Low Power Mode 2	Flash On (PD^{*1})	PLL (XTAL input)	96MHz	2.249	0.005	0.00002	0.001
			72MHz	1.846	0.005	0.00002	0.001
			12MHz	0.303	0.005	0.00002	0.001
		Internal HS_IRC	6MHz	0.166	0.005	0.00002	0.001
			1MHz	0.161	0.005	0.00002	0.001
		Internal LS_IRC	32KHz	0.048	0.005	0.00002	0.001

Table 38 LPM0/LPM1/LPM2 mode current consumption(3.3V)

3.3V@VDD$_{\text{LDO\_M}}$,VDD$_{\text{LDO\_AON}}$,VDDIO and 1.8V@VDDIO $_{\text{DVDD\_FlashR/DVDD\_FlashL}}$, 25°C, Typical value
LPM0, LPM1, LPM3 mode
NOTE PD^{*1} : Flash power-down mode

Operation Mode	Flash	Clock Source	Frequency	Current (unit mA)			
				LDO_M_DVDD LDO_M	VDD_AON LDO_AON	VDDIO & Flash	
						VDDIO	Flash Memory
Low Power Mode 0	Flash On	PLL (XTAL input)	96MHz	2.598	0.005	0.006	0.019
			72MHz	2.187	0.005	0.006	0.019
			24MHz	1.392	0.005	0.006	0.019
		Internal HS_IRC	12MHz	0.728	0.005	0.006	0.019
			6MHz	0.496	0.005	0.006	0.019
			1MHz	0.495	0.005	0.006	0.019
		Internal LS_IRC	32KHz	0.353	0.005	0.006	0.019
Low Power Mode 1	Flash On (PD^{*1})	PLL (XTAL input)	96MHz	2.320	0.005	0.006	0.001
			72MHz	1.907	0.005	0.006	0.001
			12MHz	0.406	0.005	0.006	0.001
		Internal HS_IRC	6MHz	0.201	0.005	0.006	0.001
			1MHz	0.200	0.005	0.006	0.001
		Internal LS_IRC	32KHz	0.058	0.005	0.006	0.001
Low Power Mode 2	Flash On (PD^{*1})	PLL (XTAL input)	96MHz	2.316	0.005	0.006	0.001
			72MHz	1.907	0.005	0.006	0.001
			12MHz	0.341	0.005	0.006	0.001
		Internal HS_IRC	6MHz	0.199	0.005	0.006	0.001
			1MHz	0.199	0.005	0.006	0.001
		Internal LS_IRC	32KHz	0.056	0.005	0.006	0.001

3) Standby(LPM3) mode current

- LDO_M off and Always on block power on only with LS-IRC (32KHz)
- All I/O ports is WEAK0 state controlled by PAD Pull-Down mode.
- Flash power on/off using external FET switch which controlled by PA1(Standby Signal) output

Table 39 Standby mode (LPM3) mode current consumption(1.8V)

1.8V@VDD$_{\text{LDO\_M}}$,VDD$_{\text{LDO\_AON}}$,VDDIO and 1.8V@VDDIO $_{\text{DVDD\_FlashR/DVDD\_FlashL}}$, 25°C, Typical value
Standby (LPM3) mode

Operation Mode	Flash	Clock Source	Frequency	Current (unit uA)			
				LDO_M_DVDD	VDD_AON	VDDIO & Flash	
				LDO_M	LDO_AON	VDDIO	Flash Memory
Standby	Flash Off	LS_IRC, w/ RTC	32KHz	0.0	3.3	0.003	0.0
		LS_IRC, w/o RTC	32KHz	0.0	2.8	0.002	0.0

Table 40 Standby mode (LPM3) mode current consumption(3.3V)

3.3V@VDD$_{\text{LDO\_M}}$,VDD$_{\text{LDO\_AON}}$,VDDIO and 1.8V@VDDIO $_{\text{DVDD\_FlashR/DVDD\_FlashL}}$, 25°C, Typical value
Standby mode :

Operation Mode	Flash	Clock Source	Frequency	Current (unit uA)			
				LDO_M_DVDD	VDD_AON	VDDIO & Flash	
				LDO_M	LDO_AON	VDDIO	Flash Memory
Standby	Flash Off	LS_IRC, w/ RTC	32KHz	0.51	5.1	1.50	0.0
		LS_IRC, w/o RTC	32KHz	0.49	4.5	1.50	0.0

- The power Flash memory power(DVDD_FlashR/DVDD_FlashL) should be off to reduce power consumption in the flash memory using external switch (FET switch) in standby mode.

Table 41 Standby mode (LPM3) current consumption for LoRa Application

LoRa Application,
1.8V @ VDD$_{\text{LDO\_M}}$, VDD$_{\text{LDO\_AON}}$, DVDD$_{\text{\_FlashR}}$, DVDD$_{\text{\_FlashL}}$, 3.3V @ VDDIO, 25°C Typical value
Standby mode :

Operation Mode	Flash	Clock Source	Frequency	Current (unit uA)			
				LDO_M_DVDD	VDD_AON	VDDIO & Flash	
				LDO_M	LDO_AON	VDDIO	Flash Memory
Standby	Flash Off	LS_IRC, w/ RTC	32KHz	0.0	3.3	1.50	0.0

ELECTRICAL CHARACTERISTICS

Table 42 Electrical Specification

Parameter	Symbol	Min	Typ.	Max	Units
Input high voltage	VIH	0.7*VDDIO	-	-	V
Input low voltage	VIL	-	-	0.3*VDDIO	V
Input high leakage	IIH	-10	-	10	uA
Input low leakage	IIL	-10	-	10	uA
Pull-up current	IPU@3.3V	40	-	104	uA
Pull-down current	IPD@3.3V	-162	-	-39	uA
Output high voltage	VOH@-1mA	VDD-0.4	-	-	V
Output low voltage	VOL@1mA	-	-	0.4	V