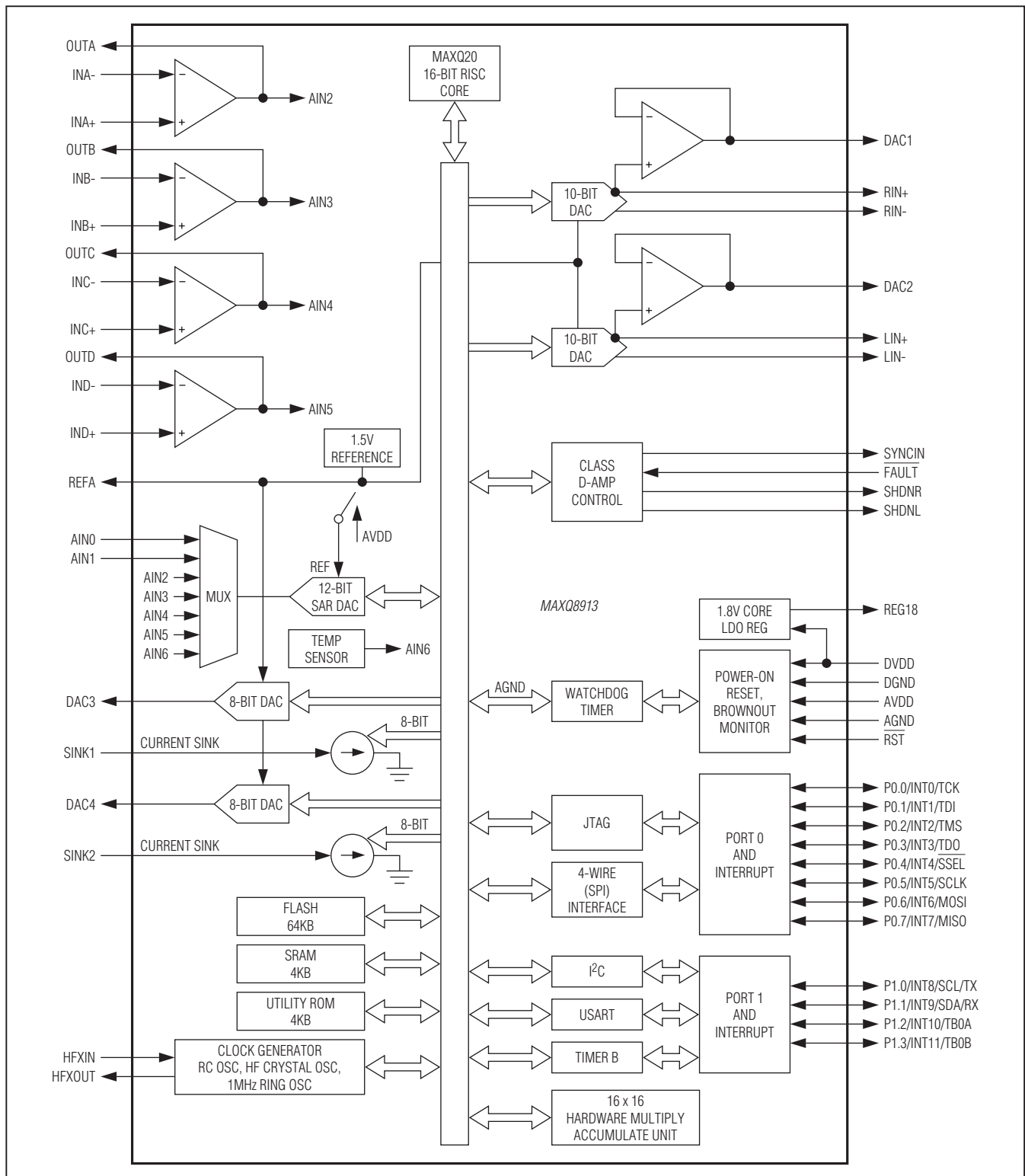


MAXQ FAMILY USER'S GUIDE: MAXQ8913 SUPPLEMENT



For pricing, delivery, and ordering information, please contact Maxim Direct at 1-888-629-4642, or visit Maxim's website at www.maximintegrated.com.

TABLE OF CONTENTS

ADDENDUM TO SECTION 1: OVERVIEW	1-1
1.1 References	1-1
ADDENDUM TO SECTION 2: ARCHITECTURE	2-1
2.1 Instruction Set	2-1
2.2 Harvard Memory Architecture	2-1
2.3 Register Space	2-1
2.4 Memory Organization	2-3
2.4.1 Register Space	2-3
2.4.2 Program Stack	2-3
2.4.3 Data SRAM	2-3
2.4.4 Program Flash	2-3
2.5 Program and Data Memory Mapping	2-3
2.6 Clock Generation	2-5
2.6.1 External High-Frequency Oscillator Circuit	2-5
2.6.2 Ring Oscillator	2-6
2.7 Interrupts	2-7
2.8 Reset Conditions	2-8
2.8.1 Power-On Reset	2-9
2.8.2 Watchdog Timer Reset	2-9
2.8.3 External Reset	2-9
2.9 Power-Management Features	2-10
2.9.1 Divide-by-256 Mode (PMM)	2-11
2.9.2 Switchback Mode	2-11
2.9.3 Stop Mode	2-11
ADDENDUM TO SECTION 3: PROGRAMMING	3-1
ADDENDUM TO SECTION 4: SYSTEM REGISTER DESCRIPTIONS	4-1
4.1 System Register Descriptions	4-3
4.1.1 Processor Status Flags Register (PSF, M8[04h])	4-3
4.1.2 Interrupt Mask Register (IMR, M8[06h])	4-3
4.1.3 System Control Register (SC, M8[08h])	4-4
4.1.4 Interrupt Identification Register (IIR, M8[0Bh])	4-4
4.1.5 System Clock Control Register (CKCN, M8[0Eh])	4-5
ADDENDUM TO SECTION 5: PERIPHERAL REGISTER MODULES	5-1
ADDENDUM TO SECTION 6: GENERAL-PURPOSE I/O MODULE	6-1
6.1 GPIO and External Interrupt Register Descriptions	6-3
6.1.1 Port 0 Direction Register (PDO, M0[10h])	6-3
6.1.2 Port 1 Direction Register (PD1, M0[11h])	6-3

MAXQ Family User's Guide: MAXQ8913 Supplement

TABLE OF CONTENTS (continued)

6.1.3 Port 0 Input Disable Register (PID0, M0[0Dh])	6-3
6.1.4 Port 0 Output Register (PO0, M0[00h])	6-4
6.1.5 Port 1 Output Register (PO1, M0[01h])	6-4
6.1.6 Port 0 Input Register (PI0, M0[08h])	6-4
6.1.7 Port 1 Input Register (PI1, M0[09h])	6-4
6.1.8 External Interrupt Flag 0 Register (EIF0, M0[02h])	6-5
6.1.9 External Interrupt Flag 1 Register (EIF1, M0[04h])	6-5
6.1.10 External Interrupt Enable 0 Register (EIE0, M0[03h])	6-6
6.1.11 External Interrupt Enable 1 Register (EIE1, M0[05h])	6-6
6.1.12 External Interrupt Edge Select 0 Register (EIES0, M0[0Ah]).	6-7
6.1.13 External Interrupt Edge Select 1 Register (EIES1, M0[0Bh]).	6-7
6.2 Port Pin Examples	6-8
6.2.1 Port Pin Example 1: Driving Outputs on Port 0	6-8
6.2.2 Port Pin Example 2: Receiving Inputs on Port 0	6-8
ADDENDUM TO SECTION 7: TIMER/COUNTER 0 MODULE	7-1
ADDENDUM TO SECTION 8: TIMER/COUNTER 1 MODULE	8-1
ADDENDUM TO SECTION 9: TIMER/COUNTER 2 MODULE	9-1
ADDENDUM TO SECTION 10: SERIAL I/O MODULE	10-1
10.1 Serial USART I/O Pins and Control Registers	10-1
10.2 Serial USART Code Examples	10-1
10.2.1 Serial USART Example: Echo Characters in 10-Bit Asynchronous Mode	10-1
ADDENDUM TO SECTION 11: SERIAL PERIPHERAL INTERFACE (SPI) MODULE 11-1	
11.1 SPI Input/Output Pins and Control Registers.	11-1
11.2 SPI Code Examples	11-1
11.2.1 SPI Example 1: Transmitting Data in Master Mode	11-1
11.2.2 SPI Example 2: Receiving Data in Slave Mode	11-2
ADDENDUM TO SECTION 12: HARDWARE MULTIPLIER MODULE	12-1
12.1 Hardware Multiplier Control Registers	12-1
12.2 Hardware Multiplier Code Examples	12-1
12.2.1 Hardware Multiplier Example: Multiply and Square/Accumulate	12-1
ADDENDUM TO SECTION 13: 1-Wire BUS MASTER	13-1
ADDENDUM TO SECTION 14: REAL-TIME CLOCK MODULE	14-1
ADDENDUM TO SECTION 15: TEST ACCESS PORT (TAP)	15-1

TABLE OF CONTENTS (continued)

ADDENDUM TO SECTION 16: IN-CIRCUIT DEBUG MODE		16-1
16.1 Register Read and Write Commands		16-1
16.2 Data Memory Read Command		16-1
16.3 Data Memory Write Command		16-1
16.4 Program Stack Read Command		16-1
16.5 Read Register Map Command		16-1
ADDENDUM TO SECTION 17: IN-SYSTEM PROGRAMMING (JTAG)		17-1
17.1 JTAG Bootloader Protocol		17-1
17.2 Family 0 Commands (Not Password Protected)		17-2
17.3 Family 1 Commands: Load Variable Length (Password Protected)		17-4
17.4 Family 2 Commands: Dump Variable Length (Password Protected)		17-5
17.5 Family 3 Commands: CRC Variable Length (Password Protected)		17-6
17.6 Family 4 Commands: Verify Variable Length (Password Protected)		17-6
17.7 Family 5 Commands: Load and Verify Variable Length (Password Protected)		17-7
17.8 Family E Commands: Erase Fixed Length (Password Protected)		17-7
ADDENDUM TO SECTION 18: MAXQ FAMILY INSTRUCTION SET SUMMARY		18-1
SECTION 19: ANALOG-TO-DIGITAL CONVERTER (SPECIFIC TO MAXQ8913)		19-1
19.1 Analog-to-Digital Converter Features		19-1
19.2 Analog-to-Digital Pins and Control Registers		19-2
19.2.1 Analog-to-Digital Converter Status Register (ADST, M4[06h])		19-3
19.2.2 ADC Conversion Sequence Address Register (ADADDR, M3[01h])		19-5
19.2.3 ADC Control Register (ADCN, M3[08h])		19-5
19.2.4 ADC Data Register (ADDATA, M3[09h])		19-7
19.2.5 ADC Data Buffer Registers (ADBUF[0] to ADBUF[15], ADDATA[00h] to ADDATA[0Fh])		19-7
19.2.6 ADC Conversion Configuration Registers (ADCFG[0] to ADCFG[7], ADDATA[10h] to ADDATA[17h])		19-8
19.2.7 Temperature Sensor Enable Register (TEMPEN, M3[0Ch])		19-9
19.3 Analog-to-Digital Converter Code Examples		19-9
19.3.1 ADC Example 1: Single Conversion		19-9
19.3.2 ADC Example 2: Continuous Conversion		19-9
SECTION 20: DIGITAL-TO-ANALOG CONVERTERS (SPECIFIC TO MAXQ8913)		20-1
20.1 DAC Overview		20-1
20.2 DAC Control Register Descriptions		20-1
20.2.1 DAC 1 Output Register (DAC1OUT, M3[02h])		20-2
20.2.2 DAC 2 Output Register (DAC2OUT, M3[03h])		20-2
20.2.3 DAC 3 Output Register (DAC3OUT, M3[04h])		20-3
20.2.4 DAC 4 Output Register (DAC4OUT, M3[05h])		20-3
20.2.5 Current Sink Control Register (ISINKCN, M3[07h])		20-3

MAXQ Family User's Guide: MAXQ8913 Supplement

TABLE OF CONTENTS (continued)

SECTION 21: TIMER/COUNTER B MODULE (SPECIFIC TO MAXQ8913)	21-1
21.1 Timer/Counter B Register Descriptions	21-1
21.1.1 Timer B Timer/Counter Capture/Reload Register (TBR, M2[07h])	21-1
21.1.2 Timer B Timer/Counter Compare Register (TBC, M2[0Bh])	21-2
21.1.3 Timer B Timer/Counter Control Register (TBCN, M2[06h])	21-2
21.1.4 Timer B Timer Value Register (TBV, M2[0Ah])	21-3
21.2 Timer/Counter B Operation	21-4
21.2.1 Timer B 16-Bit Timer/Counter Mode with Autoreload	21-4
21.2.2 Timer B 16-Bit Capture Mode	21-5
21.2.3 Timer B 16-Bit Up/Down Count with Autoreload Mode	21-6
21.2.4 Timer B Clock Output Mode	21-7
21.2.5 Timer B PWM/Output Control Functionality	21-7
21.2.6 16-Bit Up Count PWM/Output Control Mode	21-8
21.2.7 16-Bit Up/Down Count PWM/Output Control Mode	21-9
21.2.8 EXENB Control During PWM/Output Control Mode	21-11
21.3 Timer B Examples	21-11
21.3.1 Timer B Example: Reloading Timer Mode	21-11
SECTION 22 : I²C BUS INTERFACE (SPECIFIC TO MAXQ8913)	22-1
22.1 I ² C Register Descriptions	22-1
22.1.1 I ² C Data Buffer Register (I2CBUF, M1[06h])	22-2
22.1.1.1 I ² C Data Read and Write	22-2
22.1.1.2 I ² C Address Transmission	22-2
22.1.2 I ² C Status Register (I2CST, M3[01h])	22-2
22.1.3 I ² C Interrupt Enable Register (I2CIE, M1[07h])	22-3
22.1.4 I ² C Control Register (I2CCN, M1[04h])	22-4
22.1.5 I ² C Clock Control Register (I2CCK, M1[0Ch])	22-6
22.1.6 I ² C Timeout Register (I2CTO, M1[0Dh])	22-6
22.1.7 I ² C Slave Address Register (I2CSLA, M1[0Eh])	22-7
22.2 I ² C Code Examples	22-7
22.2.1 I ² C Example 1: Master Mode Transmit	22-7
22.2.2 I ² C Example 2: Master Mode Receive	22-8
22.2.3 I ² C Example 3: Slave Mode Receive	22-8
22.2.4 I ² C Example 4: Slave Mode Receive	22-9
SECTION 23: SUPPLY VOLTAGE MONITOR AND POWER CONTROL (SPECIFIC TO MAXQ8913)	23-1
23.1 SVM and Power Control Register Descriptions	23-1
23.1.1 Power Control Register (PWCN, M0[0Ch])	23-1
23.1.2 Supply Voltage Monitor Register (SVM, M0[06h])	23-2

MAXQ Family User's Guide: MAXQ8913 Supplement

TABLE OF CONTENTS (continued)

SECTION 24: INPUT/OUTPUT AMPLIFIERS (SPECIFIC TO MAXQ8913)	24-1
24.1 Input/Output Amplifier Overview	24-1
24.2 Amplifier Control Register Descriptions	24-1
24.2.1 Amplifier Control Register (AMPCN, M3[06h])	24-1
24.2.2 Op Amp Control Register (OPMCN, M3[0Ah])	24-2
SECTION 25: UTILITY ROM (SPECIFIC TO MAXQ8913)	25-1
25.1 Overview	25-1
25.2 In-Application Programming Functions	25-2
25.2.1 UROM_flashWrite	25-2
25.2.2 UROM_flashErasePage	25-2
25.2.3 UROM_flashEraseAll	25-2
25.3 Data Transfer Functions	25-3
25.3.1 UROM_moveDP0	25-4
25.3.2 UROM_moveDP0inc	25-4
25.3.3 UROM_moveDP0dec	25-4
25.3.4 UROM_moveDP1	25-5
25.3.5 UROM_moveDP1inc	25-5
25.3.6 UROM_moveDP1dec	25-5
25.3.7 UROM_moveBP	25-6
25.3.8 UROM_moveBPinc	25-6
25.3.9 UROM_moveBPdec	25-6
25.3.10 UROM_copyBuffer	25-7
25.4 Utility ROM Examples	25-7
25.4.1 Utility ROM Example 1: Reading Constant Word Data from Flash	25-7
25.4.2 Utility ROM Example 2: Reading Constant Byte Data from Flash (Indirect Function Call)	25-8
APPENDIX 1: MAXQ8913 DEVICE INCLUDE FILE FOR MAX-IDE	A1-1
REVISION HISTORY	R-1

MAXQ Family User's Guide: MAXQ8913 Supplement

LIST OF FIGURES

Figure 2-1. MAXQ8913 System and Peripheral Register Map2-2
Figure 2-2. Memory Map When Executing from Program Flash Memory2-4
Figure 2-3. Memory Map When Executing from Utility ROM.2-4
Figure 2-4. Memory Map When Executing from Data SRAM2-5
Figure 2-5. MAXQ8913 Clock Sources2-6
Figure 2-6. Power-On Reset Timing.2-9
Figure 2-7. External Reset Timing2-10
Figure 19-1. ADC Block Diagram19-1
Figure 21-1. Timer B Autoreload Mode Block Diagram21-5
Figure 21-2. Timer B 16-Bit Capture Mode Block Diagram.21-5
Figure 21-3. Timer B 16-Bit Up/Down Count with Autoreload Mode Block Diagram21-6
Figure 21-4. Timer B Clock Output Mode Block Diagram.21-7
Figure 21-5. Up-Count PWM/Output Control Mode Block Diagram21-8
Figure 21-6. Timer B PWM/Output Control Mode Waveform (Count Up)21-9
Figure 21-7. Timer B Up/Down-Count PWM/Output Control Mode Block Diagram21-10
Figure 21-8. Timer B PWM/Output Control Mode Waveform (Up/Down Count)21-10
Figure 25-1. Memory Map When Executing from Utility ROM.25-3

LIST OF TABLES

Table 2-1. System Clock Generation and Control Registers	2-7
Table 2-2. Interrupt Sources and Control Bits	2-8
Table 2-3. System Power-Management Registers	2-10
Table 4-1. System Register Map	4-1
Table 4-2. System Register Bit Functions	4-1
Table 4-3. System Register Reset Values	4-2
Table 4-4. System Clock Modes	4-6
Table 5-1. Peripheral Register Map	5-1
Table 5-2. Peripheral Register Bit Functions	5-2
Table 5-3. Peripheral Register Bit Reset Values	5-3
Table 6-1. Port Pin Special and Alternate Functions	6-1
Table 6-2. P1[3:0] Input/Output States (in Standard Mode)	6-2
Table 6-3. P0[7:0] Input/Output States (in Standard Mode)	6-2
Table 10-1. Serial USART Input and Output Pins	10-1
Table 10-2. Serial USART Control Registers	10-1
Table 11-1. SPI Input and Output Pins	11-1
Table 11-2. SPI Control Registers	11-1
Table 12-1. Hardware Multiplier Control Registers	12-1
Table 16-1. Output from DebugReadMap Command	16-2
Table 17-1. Bootloader Status Codes	17-1
Table 17-2. Bootloader Status Flags	17-3
Table 18-1. Instruction Set Summary	18-1
Table 19-1. ADC Input and Power-Supply Pins	19-2
Table 19-2. ADC Control Registers	19-2
Table 19-3. ADC Sample Rates Using a 10MHz Crystal	19-6
Table 20-1. DAC Control Registers	20-1
Table 21-1. Type B Timer/Counter Input and Output Pins	21-1
Table 21-2. Type B Timer/Counter Control Registers	21-1
Table 21-3. Timer/Counter B Mode Summary	21-4
Table 21-4. Timer B PWM/Output Control Function	21-7
Table 22-1. I ² C Input and Output Pins	22-1
Table 22-2. I ² C Interface Control Registers	22-1
Table 24-1. Amplifier Control Registers	24-1
Table 25-1. Functions for MAXQ8913 Utility ROM	25-1

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 1: OVERVIEW

This document is provided as a supplement to the *MAXQ Family User's Guide*, covering new or modified features specific to the MAXQ8913. This document must be used in conjunction with the *MAXQ Family User's Guide*, available on our website at www.maxim-ic.com/MAXQUG. Addenda are arranged by section number, which correspond to sections in the *MAXQ Family User's Guide*. Additions and changes, with respect to the *MAXQ Family User's Guide*, are contained in this document, and updates/additions are added when available.

The MAXQ8913 is a low-power, high-performance, 16-bit, RISC microcontroller based on the MAXQ® architecture design. It is targeted specifically for dual-axis optical image stabilization (OIS) applications and includes a wide range of peripherals including a 7-channel, 12-bit successive-approximation analog-to-digital converter (SAR ADC), two 10-bit and two 8-bit digital-to-analog converters (DACs), four op amps for ADC input conditioning, two programmable current sinks, and support for an external D-class audio amplifier. The MAXQ8913 is uniquely suited for any application that requires high performance and low-power operation.

1.1 References

Refer to the *MAXQ Family User's Guide* for the following information:

- Description of the core architecture, instruction set, and memory mapping common to all MAXQ microcontrollers.
- Definitions and functions of the common system register set, including accumulators, data pointers, loop counters, and general-purpose registers.
- Descriptions of common clock generation, interrupt handling, and reset/power-management modes.
- Descriptions and programming examples for common MAXQ peripherals found on the MAXQ8913 including the serial universal synchronous/asynchronous receiver-transmitter (USART), SPI™ interface, and hardware multiplier.
- Description of the test access port (TAP) and in-circuit debug interface.
- Description of the in-system programming mode.

The MAXQ8913 data sheet, which contains electrical/timing specifications and pin descriptions, is available at www.maxim-ic.com/MAXQ8913.

Errata sheets for the MAXQ8913 and other MAXQ microcontrollers are available at www.maxim-ic.com/errata.

For more information on other MAXQ microcontrollers, development hardware and software, frequently asked questions, and software examples, visit the MAXQ page at www.maxim-ic.com/MAXQ.

MAXQ is a registered trademark of Maxim Integrated Products, Inc.

SPI is a trademark of Motorola, Inc.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 2: ARCHITECTURE

The MAXQ8913 shares the common architecture features with other members of the MAXQ microcontroller family. Details are discussed in the following sections.

2.1 Instruction Set

This device uses the standard 16-bit MAXQ20 core instruction set as described in the *MAXQ Family User's Guide*.

2.2 Harvard Memory Architecture

Program memory, data memory, and register space follow the Harvard architecture model. Each type of memory is kept separate and is accessed by a separate bus, allowing different word lengths for different types of memory. Registers can be either 8 bits or 16 bits in width. Program memory is 16 bits in width to accommodate the standard MAXQ20 16-bit instruction set. Data memory is also 16 bits in width, but can be accessed in 8-bit or 16-bit modes for maximum flexibility.

The MAXQ8913 includes a flexible memory-management unit (MMU) that allows code to be executed from either the program flash, the utility ROM, or the internal data SRAM. Any of these three memory spaces can also be accessed in data space at any time, with the single restriction that the physical memory area that is currently being used as program space cannot be simultaneously read from in data space.

2.3 Register Space

The MAXQ8913 contains the standard set of MAXQ20 system registers as described in the *MAXQ Family User's Guide*, but with differences noted in this guide where they exist.

Peripheral register space (modules 0 to 3) contains registers that are used to access the following peripherals:

- 12-bit SAR ADC converter with up to seven single-ended or three differential input channels
- Four DAC output channels (two 10-bit, two 8-bit)
- Two programmable current sink outputs
- External D-amplifier support
- Internal temperature sensor (read through ADC channel 6)
- General-purpose 8-bit I/O ports (P0 and P1)
- External interrupts (up to 11)
- Programmable Type B timer/counter
- Serial USART interface
- I²C interface
- SPI interfaces (master/slave)
- Hardware multiplier/accumulator

The lower 8 bits of all registers in modules 0 to 3 (as well as the AP module M8) are bit addressable.

MAXQ Family User's Guide: MAXQ8913 Supplement

		REGISTER MODULE												
		M0	M1	M2	M3	M4	M8	M9	M11	M12	M13	M14	M15	
REGISTER INDEX	00h	P00	SCON	MCNT	ADST		AP	A[0]	PFX	IP				
	01h	PO1	SBUF	MA	ADADDR		APC	A[1]			SP			
	02h	EIFO	SPICN0	MB	DAC10UT			A[2]			IV			
	03h	EIE0	SPIB0	MC2	DAC20UT			A[3]					OFFS	DP[0]
	04h	EIF1	I2CCN	MC1	DAC30UT		PSF	A[4]					DPC	
	05h	EIE1	I2CST	MC0	DAC40UT		IC	A[5]					GR	
	06h	SVM	I2CBUF	TBOCN	AMPCN		IMR	A[6]				LC[0]	GRL	
	07h		I2CIE	TBOR	ISINKCN			A[7]				LC[1]	BP	DP[1]
	08h	PIO	SMD	MC1R	ADCN		SC	A[8]					GRS	
	09h	PI1	PR	MC0R	ADDATA			A[9]					GRH	
	0Ah	EIES0	SPICF0	TBOV	OPMCN			A[10]					GRXL	
	0Bh	EIES1	SPICK0	TBOC	DACEN		IIR	A[11]					BPI[OFFS]	
	0Ch	PWCN	I2CCK		TEMPEN			A[12]						
	0Dh	PID0	I2CT0					A[13]						
	0Eh		I2CSLA				CKCN	A[14]						
	0Fh						WDCN	A[15]						
	10h	PD0												
	11h	PD1												
	12h													
	13h													
14h														
15h														
16h														
17h														
18h														
19h														
1Ah														
1Bh														
1Ch														
1Dh														
1Eh														
1Fh														

RESERVED
OR
OP CODE

8-CHANNEL,
12-BIT
SAR ADC

PORT PINS
(GPIO)

DACs,
ANALOG
FUNCTIONS

INTERRUPT
CONTROL

HARDWARE
MULTIPLIER

SERIAL,
SPI, I²C

TIMERS

ACC
ARRAY,
CONTROL

OTHER
FUNCTIONS

Figure 2-1. MAXQ8913 System and Peripheral Register Map

MAXQ Family User's Guide: MAXQ8913 Supplement

2.4 Memory Organization

As with all MAXQ microcontrollers, the MAXQ8913 contains logically separate program and data memory spaces. All memory is internal, and physical memory segments (other than the stack and register memories) can be accessed as either program memory or as data memory, but not both at once.

The MAXQ8913 contains the following physical memory segments.

2.4.1 Register Space

As described in the *MAXQ Family User's Guide*, register space on MAXQ microcontrollers consists of 16 register modules, each of which could contain up to 32 registers. Of these possible 16 register modules, only 11 are used on the MAXQ8913: seven for system registers and four for peripheral registers.

2.4.2 Program Stack

The MAXQ8913 provides a 16 x 16 hardware stack to support subroutine calls and system interrupts. This stack is used automatically by CALL/RET and PUSH/POP instructions, and can also be accessed directly through the SP register as described in the *MAXQ Family User's Guide*.

When using the in-circuit debugging features, one word of the stack must be reserved to store the return location when execution branches into the debugging routines in the utility ROM. If in-circuit debug is not used, the entire stack is available for application use.

2.4.3 Data SRAM

The MAXQ8913 contains up to 2048 words (4KB) of on-chip data SRAM, which can be mapped into either program or data space. The contents of this SRAM are indeterminate after power-on reset, but are maintained during stop mode and across non-POR resets.

When using the in-circuit debugging features, the highest 19 bytes of the SRAM must be reserved for saved state storage and working space for the debugging routines in the utility ROM. If in-circuit debug is not used, the entire SRAM is available for application use.

2.4.4 Program Flash

The MAXQ8913 contains 32KWords (32K x 16) of flash memory, which normally serves as program memory. When executing from the data SRAM or utility ROM, this memory is mapped to data space (as 32KWords or 64KB) and can be used for lookup tables and similar functions.

Since program memory is mapped into data space starting at address 8000h, only half the available program memory can be mapped into data space at one time when operating in byte-access mode. The CDA0 (code data access) bit is used to control which half of program memory is available in data space as shown in Figure 2-3 and Figure 2-4, and as described in the *MAXQ Family User's Guide*. When operating in word-access mode, the entire 32KWord program memory can be mapped into data space at once.

Flash memory mapped into data space can be read from directly, like any other type of data memory. However, writing to flash memory must be done indirectly by calling the in-application functions provided by the utility ROM. See *Section 25: Utility ROM* for more details.

2.5 Program and Data Memory Mapping

Figures 2-2, 2-3, and 2-4 show the mapping of physical memory segments into the program and data memory space. The mapping of memory segments into program space is always the same. The mapping of memory segments into data space varies depending on which memory segment is currently being executed from.

In all cases, whichever memory segment is currently being executed from in program space may not be accessed in data space.

MAXQ Family User's Guide: MAXQ8913 Supplement

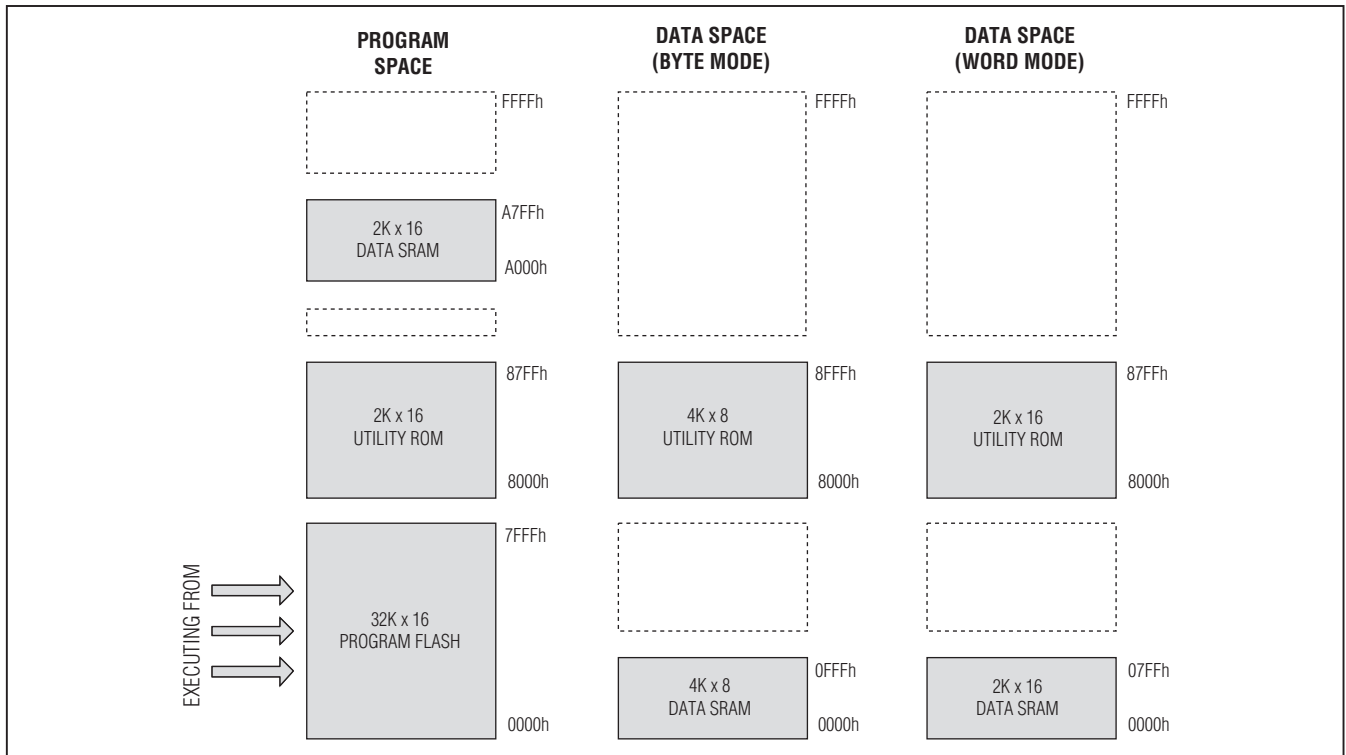


Figure 2-2. Memory Map When Executing from Program Flash Memory

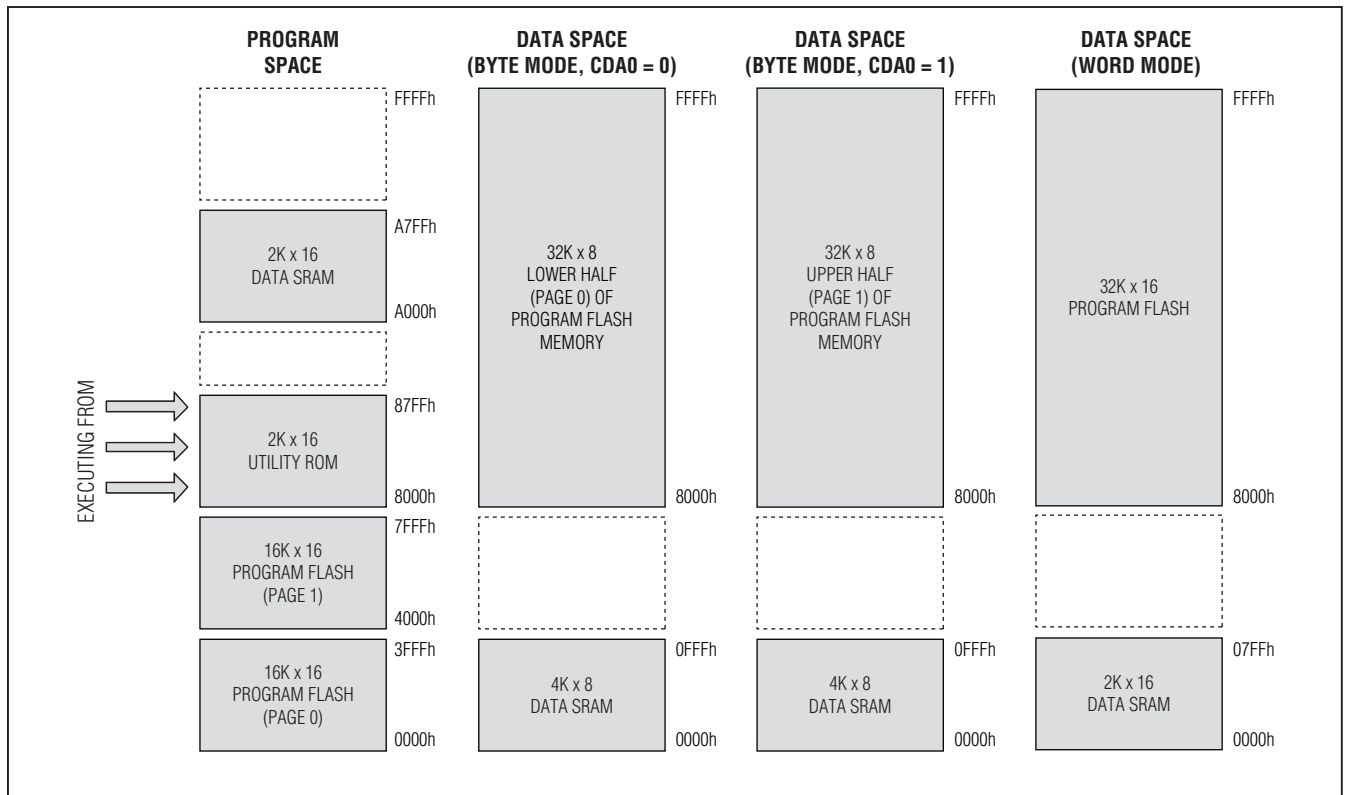


Figure 2-3. Memory Map When Executing from Utility ROM

MAXQ Family User's Guide: MAXQ8913 Supplement

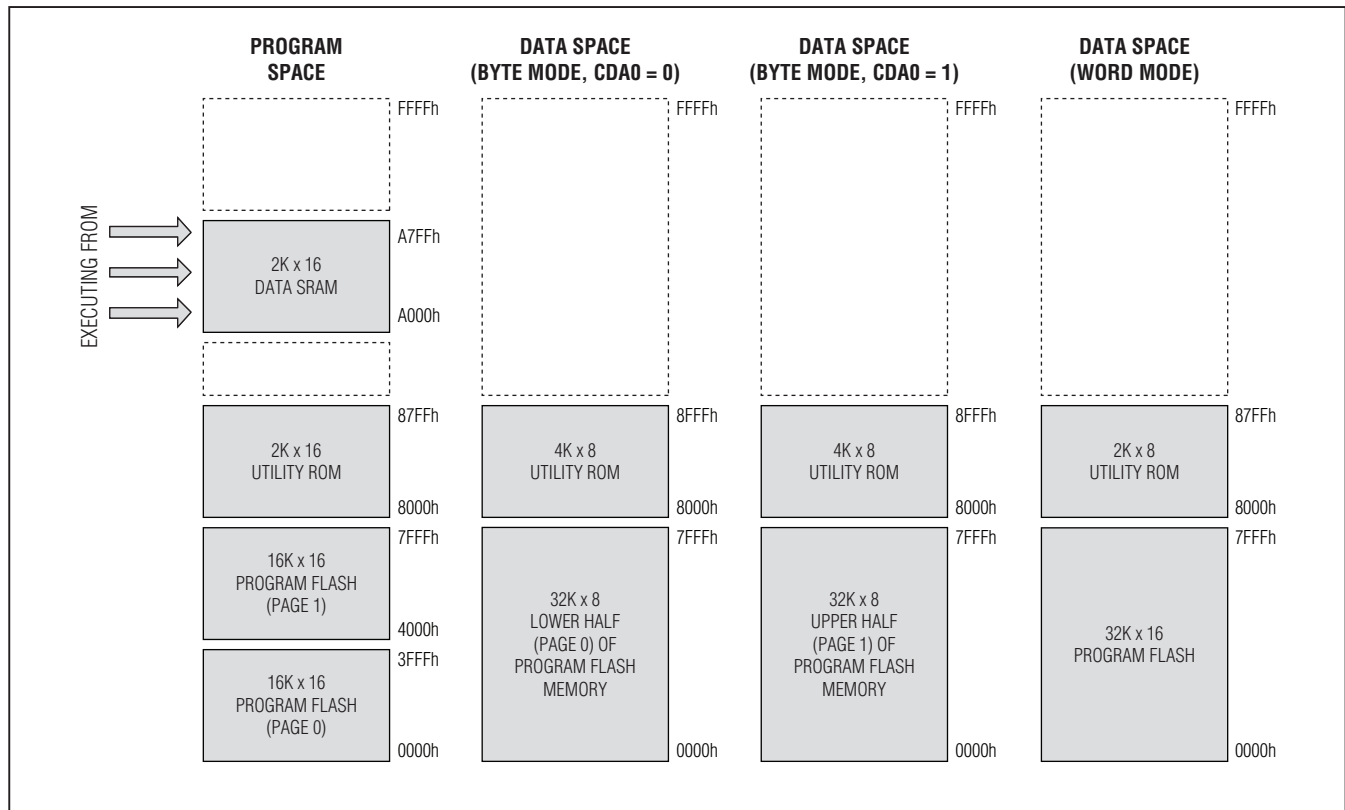


Figure 2-4. Memory Map When Executing from Data SRAM

2.6 Clock Generation

All functional modules in the MAXQ8913 are synchronized to a single system clock. This system clock can be generated from one of the following clock sources:

- External high-frequency clock
- Internal high-frequency oscillator using external crystal or resonator circuit
- Internal 1MHz ring oscillator

The MAXQ8913 does not support an external RC relaxation oscillator circuit or the 32kHz crystal input described in the *MAXQ Family User's Guide*.

The following registers and bits are used to control clock generation and selection. For more information, see the register descriptions in this guide and in the *MAXQ Family User's Guide*.

2.6.1 External High-Frequency Oscillator Circuit

The high-frequency oscillator operates as described in *Section 2.7: Clock Generation* of the *MAXQ Family User's Guide*. If used, the external crystal or resonator circuit for this oscillator should be connected between the HFXIN and HFXOUT pins.

The high-frequency oscillator can be disabled by setting HFXD (PWCN.0) to 1; this is only allowed if the high-frequency oscillator is not currently being used as the clock source (RGMD and RGSL must both equal 1). In this configuration, an external clock can be used to directly drive HFXIN; refer to the MAXQ8913 data sheet for more details.

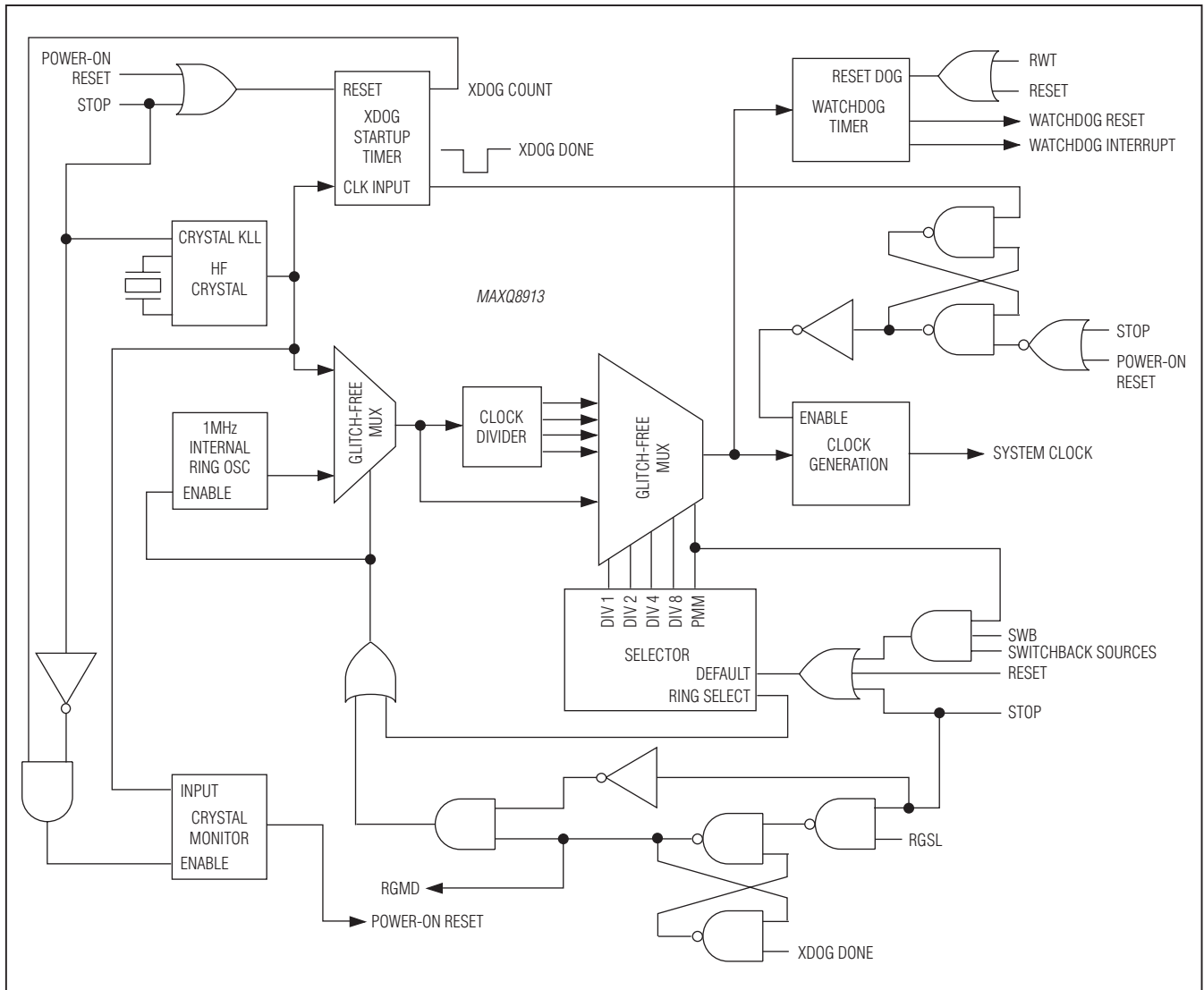


Figure 2-5. MAXQ8913 Clock Sources

2.6.2 Ring Oscillator

The MAXQ8913 contains an internal ring oscillator that can optionally be used as a system clock. The ring oscillator operates at a frequency of approximately 1MHz (refer to the MAXQ8913 data sheet for details).

On power-on reset, the ring oscillator is automatically enabled as the system clock source while the high-frequency oscillator warms up. Once the warmup count for the high-frequency oscillator has completed, the clock source switches to the high-frequency oscillator automatically. If no external crystal or resonator circuit is provided at HFXIN, the switchover never occurs, and the clock runs from the ring oscillator indefinitely.

To explicitly select the ring oscillator as the system clock source, the RGSL bit (CKCN.6) must be set to 1. Setting this bit switches over the system clock source to the ring oscillator following a 10-cycle delay of the 1MHz ring clock. The RGMD (CKCN.5) bit indicates the current system clock source. If the ring oscillator is currently providing the system clock, RGMD reads as 1; otherwise, RGMD reads as 0.

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 2-1. System Clock Generation and Control Registers

REGISTER	ADDRESS	BIT(S)	FUNCTION
CKCN	M8[0Eh]	[2:0]—PMME, CD[1:0]	000: System clock = high-frequency clock divided by 1. 001: System clock = high-frequency clock divided by 2. 010: System clock = high-frequency clock divided by 4. 011: System clock = high-frequency clock divided by 8. 1xx: System clock = high-frequency clock/256.
CKCN	M8[0Eh]	5—RGMD	0: System clock is being provided by an external source. 1: System clock is being provided by the ring oscillator.
CKCN	M8[0Eh]	6—RGSL	0: Selects an external source for system clock generation. 1: Selects the ring oscillator for system clock generation.
PWCN	M0[0Ch]	0—HFXD	0: High-frequency oscillator operates normally (default). 1: Disables the high-frequency oscillator, allowing an external clock to be provided at HFXIN.

Because the RGSL bit is cleared by power-on reset only, if this bit is set before entering stop mode, the ring oscillator is still used as the system clock source when stop mode is exited. In this case, a 10-ring oscillator cycle warmup delay is required when exiting stop mode before execution resumes using the ring oscillator as the system clock source.

When the system clock source is switched back from the ring oscillator to the high-frequency oscillator by clearing RGSL to 0, the ring oscillator is still used as the system clock source until the warmup period has completed for the high-frequency oscillator. This is reflected by the value of the RGMD bit, which remains at 1 until the warmup for the high-frequency oscillator has completed and the clock switches over, at which point RGMD switches to 0.

2.7 Interrupts

In general, interrupt handling on the MAXQ8913 operates as described in the *MAXQ Family User's Guide*. All interrupt sources have the same priority, and all interrupts cause program execution to branch to the location specified by the Interrupt Vector (IV) register, which defaults to 0000h.

Table 2-2 lists all possible interrupt sources for the MAXQ8913, along with their corresponding module interrupt enable bits, local interrupt enable bits, and interrupt flags.

- Each module interrupt enable bit, when cleared to 0, blocks interrupts originating in that module from being acknowledged. When the module interrupt enable bit is set to 1, interrupts from that module are acknowledged (unless all interrupts have been disabled globally).
- Each local interrupt enable bit, when cleared to 0, disables the corresponding interrupt. When the local interrupt-enable bit is set to 1, the interrupt is triggered whenever its interrupt flag is set to 1 by hardware or by software.
- Each interrupt flag bit, when set to 1, causes its corresponding interrupt to trigger. Interrupt flag bits are typically set by hardware and must be cleared by software (generally in the interrupt handler routine).

Note that for an interrupt to fire, the following five conditions must exist:

- Interrupts must be enabled globally by setting IGE (IC.0) to 1.
- The module interrupt enable bit for the interrupt source's module must be set to 1.
- The local interrupt enable bit for the specific interrupt source must be set to 1.
- The interrupt flag for the interrupt source must be set to 1. Typically, this is done by hardware when the condition that requires interrupt service occurs.
- The interrupt-in-service (INS) bit must be cleared to 0. This bit is set automatically upon vectoring to the interrupt handler (IV) address and cleared automatically upon exit (RETI/POPI), so the only reason to clear this bit manually (inside the interrupt handler routine) is to allow nested interrupt handling.

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 2-2. Interrupt Sources and Control Bits

INTERRUPT	MODULE ENABLE BIT	LOCAL ENABLE BIT	INTERRUPT FLAG
Watchdog Interrupt	IMS (IMR.7)	EWDI (WDCN.6)	WDIF (WDCN.3)
External Interrupt 0 (P0.0)	IM0 (IMR.0)	EX0 (EIE0.0)	IE0 (EIF0.0)
External Interrupt 1 (P0.1)	IM0 (IMR.0)	EX1 (EIE0.1)	IE1 (EIF0.1)
External Interrupt 2 (P0.2)	IM0 (IMR.0)	EX2 (EIE0.2)	IE2 (EIF0.2)
External Interrupt 3 (P0.3)	IM0 (IMR.0)	EX3 (EIE0.3)	IE3 (EIF0.3)
External Interrupt 4 (P0.4)	IM0 (IMR.0)	EX4 (EIE0.4)	IE4 (EIF0.4)
External Interrupt 5 (P0.5)	IM0 (IMR.0)	EX5 (EIE0.5)	IE5 (EIF0.5)
External Interrupt 6 (P0.6)	IM0 (IMR.0)	EX6 (EIE0.6)	IE6 (EIF0.6)
External Interrupt 7 (P0.7)	IM0 (IMR.0)	EX7 (EIE0.7)	IE7 (EIF0.7)
External Interrupt 8 (P1.0)	IM0 (IMR.0)	EX8 (EIE1.0)	IE8 (EIF1.0)
External Interrupt 9 (P1.1)	IM0 (IMR.0)	EX9 (EIE1.1)	IE9 (EIF1.1)
External Interrupt 10 (P1.2)	IM0 (IMR.0)	EX10 (EIE1.2)	IE10 (EIF1.2)
External Interrupt 11 (P1.3)	IM0 (IMR.0)	EX11 (EIE1.3)	IE11 (EIF1.3)
Supply Voltage Monitor Interrupt	IM0 (IMR.0)	SVMIE (SVM.2)	SVMI (SVM.3)
Serial Port Receive	IM1 (IMR.1)	ESI (SMD.2)	RI (SCON.0)
Serial Port Transmit	IM1 (IMR.1)	ESI (SMD.2)	TI (SCON.1)
SPI Mode Fault Interrupt	IM1 (IMR.1)	ESPII (SPICF.7)	MODF (SPICN.3)
SPI Write Collision Interrupt	IM1 (IMR.1)	ESPII (SPICF.7)	WCOL (SPICN.4)
SPI Receive Overrun Interrupt	IM1 (IMR.1)	ESPII (SPICF.7)	ROVR (SPICN.5)
SPI Transfer Complete Interrupt	IM1 (IMR.1)	ESPII (SPICF.7)	SPIC (SPICN.6)
I ² C START Condition Interrupt	IM1 (IMR.1)	I2CSRI (I2CST.0)	I2CSRIE (I2CIE.0)
I ² C Transmit Complete Interrupt	IM1 (IMR.1)	I2CTXI (I2CST.1)	I2CTXIE (I2CIE.1)
I ² C Receive Ready Interrupt	IM1 (IMR.1)	I2CRXI (I2CST.2)	I2CRXIE (I2CIE.2)
I ² C Clock Stretch Interrupt	IM1 (IMR.1)	I2CSTRI (I2CST.3)	I2CSTRIE (I2CIE.3)
I ² C Timeout Interrupt	IM1 (IMR.1)	I2CTOI (I2CST.4)	I2CTOIE (I2CIE.4)
I ² C Slave Address Match Interrupt	IM1 (IMR.1)	I2CAMI (I2CST.5)	I2CAMIE (I2CIE.5)
I ² C Arbitration Loss Interrupt	IM1 (IMR.1)	I2CALI (I2CST.6)	I2CALIE (I2CIE.6)
I ² C NACK Interrupt	IM1 (IMR.1)	I2CNACKI (I2CST.7)	I2CNACKIE (I2CIE.7)
I ² C General Call Address Interrupt	IM1 (IMR.1)	I2CGCI (I2CST.8)	I2CGCIE (I2CIE.8)
I ² C Receiver Overrun Interrupt	IM1 (IMR.1)	I2CROI (I2CST.9)	I2CROIE (I2CIE.9)
I ² C STOP Condition Interrupt	IM1 (IMR.1)	I2CSPI (I2CST.11)	I2CSPIE (I2CIE.11)
Type B Timer—External Trigger	IM2 (IMR.2)	EXFB (TBCN.6)	ETB (TBCN.1)
Type B Timer—Overflow	IM2 (IMR.2)	TFB (TBCN.7)	ETB (TBCN.1)
ADC Data Available Interrupt	IM3 (IMR.3)	ADDAIE (ADCN.5)	ADDAI (ADST.5)
Amplifier Interrupt	IM3 (IMR.3)	AMPIE (AMPCN.4)	AMPIF (AMPCN.5)

2.8 Reset Conditions

There are four possible reset sources for the MAXQ8913. While in the reset state, the enabled system clock oscillator continues running, but no code execution occurs. Once the reset condition has been removed or has completed, code execution resumes at address 8000h for all reset types.

MAXQ Family User's Guide: MAXQ8913 Supplement

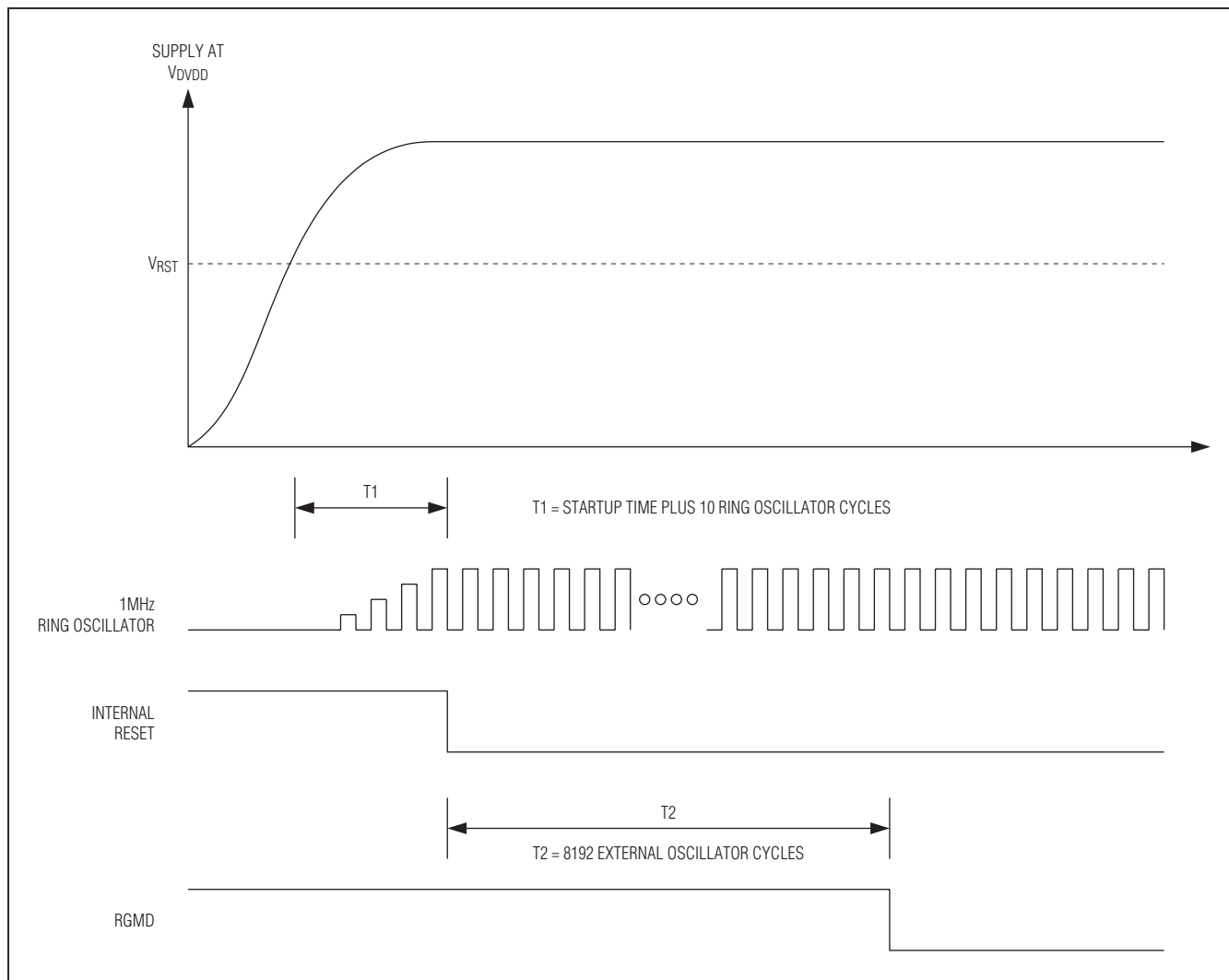


Figure 2-6. Power-On Reset Timing

2.8.1 Power-On Reset

When power is first applied to the MAXQ8913, or when the supply voltage at DVDD drops below the V_{RST} level, the processor is held in a power-on reset state. See Figure 2-6. For the MAXQ8913 to exit power-on reset, the following two conditions must apply:

- The supply voltage at DVDD is above the power-on reset level V_{RST}.
- The ring oscillator has completed a 10-cycle delay.

2.8.2 Watchdog Timer Reset

The watchdog timer on the MAXQ8913 functions as described in the *MAXQ Family User's Guide*.

2.8.3 External Reset

External reset through the $\overline{\text{RST}}$ input is a synchronous reset source. After the external reset low has been removed and sampled, execution resumes following a delay of four clock cycles, as shown in Figure 2-7.

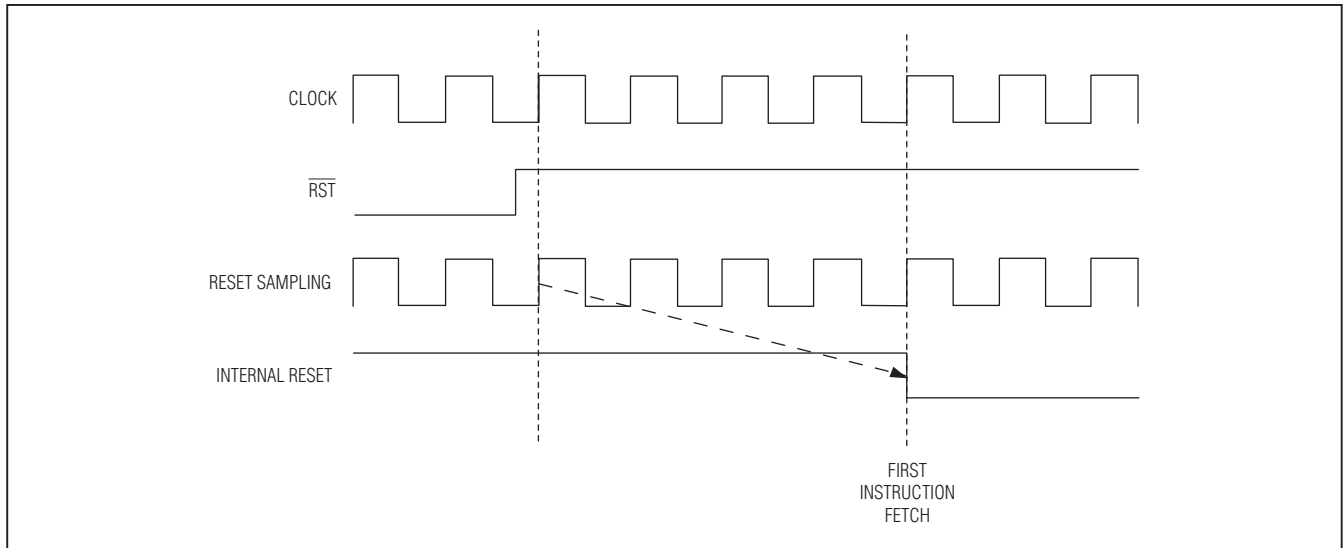


Figure 2-7. External Reset Timing

2.9 Power-Management Features

The MAXQ8913 provides the following features to assist in power management:

- Divide-by-256 (PMM) mode to reduce current consumption.
- Switchback mode to exit PMM mode automatically when rapid processing is required.
- Ultra-low-power stop mode.
- Selective regulator and brownout detection disable during stop mode.

Table 2-3 shows the system registers and bits used to control power-management features. For more information, see the register descriptions in this document and in the *MAXQ Family User's Guide*.

Table 2-3. System Power-Management Registers

REGISTER	ADDRESS	BIT	FUNCTION
CKCN	M8[0Eh]	[1:0]—CD[1:0]	00: System clock = selected clock source divided by 1. 01: System clock = selected clock source divided by 2. 10: System clock = selected clock source divided by 4. 11: System clock = selected clock source divided by 8.
CKCN	M8[0Eh]	2—PMME	0: System clock is determined by the settings of CD[1:0]. 1: System clock = selected clock source divided by 256.
CKCN	M8[0Eh]	3—SWB	When set to 1, enables automatic switchback from PMM (divide-by-256 mode) to normal clock-divide mode under certain conditions.
CKCN	M8[0Eh]	4—STOP	When set to 1, causes the processor to enter stop mode.
PWCN	M0[0Ch]	0—HFXD	0: Enables the high-frequency oscillator. 1: Disables the high-frequency oscillator, allowing an external clock to be provided at HFXIN.
PWCN	M0[0Ch]	6—REGEN	0: Internal regulator is shut down during stop mode. 1: Internal regulator remains powered on during stop mode.
PWCN	M0[0Ch]	7—BOD	0: Brownout detection remains enabled during stop mode. 1: Brownout detection is enabled during stop mode.

MAXQ Family User's Guide: MAXQ8913 Supplement

2.9.1 Divide-by-256 Mode (PMM)

In this power-management mode, all operations continue as normal, but at a reduced clock rate (the selected clock source divided by 256).

This power-management mode is entered by setting the PMME bit (CKCN.2) to 1 and CD[1:0] to 0. When PMM mode is exited (either by clearing the PMME bit or as a result of a switchback trigger), system operation reverts to divide-by-1 mode.

2.9.2 Switchback Mode

As described in the *MAXQ Family User's Guide*, switchback mode is used to provide an automatic exit from power-management mode when a higher clock rate is required to respond to I/O, such as USART activity, SPI activity, or an external interrupt.

Switchback mode is enabled when the SWB (CKCN.3) bit is set to 1 and the PMME (CKCN.2) bit is set to 1 (the system is in the PMM mode). If switchback is enabled, the PMME bit is cleared (causing the system to exit power-management mode) when any of the following conditions occur:

- An external interrupt condition occurs on an INTn pin and the corresponding external interrupt is enabled.
- An active-low transition occurs on the RX pin and the USART is enabled to receive data.
- The SBUF register is written to transmit a byte over the USART.
- The SPIB register is written to transmit a byte with the SPI interface enabled in master mode.
- The \overline{SSEL} signal is asserted low with the SPI interface enabled in slave mode.
- A START condition occurs on the I²C bus and the I²C START interrupt is triggered.
- The supply voltage drops below the supply voltage monitor (SVM) threshold, and the SVM interrupt is triggered.
- An ADC conversion is initiated by setting ADCONV to 1.
- Active debug mode is entered either by a breakpoint match or direct issuance of the debug command from background mode.

As described in the *MAXQ Family User's Guide*, if any of these conditions is true (a switchback source is active) and the SWB bit has been set, the PMME bit cannot be set to enter power-management mode.

2.9.3 Stop Mode

Stop mode disables all clocked circuits within the MAXQ8913 and halts the processor completely. All on-chip clocks, timers, serial ports, and other peripherals are stopped, and no code execution occurs. Once in stop mode, the MAXQ8913 is in a near-static state, with power consumption determined largely by leakage currents.

Stop mode is invoked by setting the STOP bit to 1. The MAXQ8913 enters stop mode immediately when the STOP bit is set. Entering stop mode does not affect the setting of the clock control bits; this allows the system to return to its original operating frequency following stop mode removal.

The processor exits stop mode if any of the following conditions occur. In order to exit stop mode by means of an interrupt, the interrupt must be enabled globally, by module, and locally prior to entering stop mode.

- External reset (from the \overline{RST} pin)
- Power-on/brownout reset
- External interrupt
- I²C START interrupt
- Supply voltage monitor interrupt (SVMSTOP must be set to 1)

Note that exiting stop mode through external reset or power-on reset causes the processor to undergo a normal reset cycle, as opposed to resuming execution at the point at which it entered stop mode. Exiting stop mode by means of an interrupt causes the processor to vector to the interrupt handler routine at IV. Following the completion of the interrupt handler, execution resumes at the instruction following the one that caused the entry into stop mode.

MAXQ Family User's Guide: MAXQ8913 Supplement

When the processor exits stop mode, program execution resumes using the previously selected clock source following a 10-ring oscillator cycle delay plus any additional delay time required to enable the internal regulator and other circuitry (refer to the IC data sheet for details).

- If RGSL = 1, the processor continues running from the ring oscillator indefinitely.
- If RGSL = 0, the processor continues running from the ring oscillator until the high-frequency clock source completes its warmup count (8192 cycles for external crystal oscillator, 10 cycles for external clock input), at which point it switches over to the high-frequency clock automatically.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 3: PROGRAMMING

Refer to *Section 3: Programming* of the *MAXQ Family User's Guide* for examples of general program operations involving the MAXQ core. The MAXQ8913 contains the MAXQ20 (16-bit accumulator version) of the MAXQ core.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 4: SYSTEM REGISTER DESCRIPTIONS

Refer to *Section 4: System Register Descriptions* of the *MAXQ Family User's Guide* for functional descriptions of the registers and bits in Table 4-1.

Table 4-1. System Register Map

CYCLES TO READ	CYCLES TO WRITE	REGISTER INDEX	AP (M8)	A (M9)	PFX (M11)	IP (M12)	SP (M13)	DPC (M14)	DP (M15)
1	1	00h	AP	A[0]	PFX[0]	IP	—	—	—
1	1	01h	APC	A[1]	PFX[1]	—	SP	—	—
1	1	02h	—	A[2]	PFX[2]	—	IV	—	—
1	1	03h	—	A[3]	PFX[3]	—	—	OFFS	DP[0]
1	1	04h	PSF	A[4]	PFX[4]	—	—	DPC	—
1	1	05h	IC	A[5]	PFX[5]	—	—	GR	—
1	1	06h	IMR	A[6]	PFX[6]	—	LC[0]	GRL	—
1	1	07h	—	A[7]	PFX[7]	—	LC[1]	BP	DP[1]
1	2	08h	SC	A[8]	—	—	—	GRS	—
1	2	09h	—	A[9]	—	—	—	GRH	—
1	2	0Ah	—	A[10]	—	—	—	GRXL	—
1	2	0Bh	<i>IIR</i>	A[11]	—	—	—	BP[OFFS]	—
1	2	0Ch	—	A[12]	—	—	—	—	—
1	2	0Dh	—	A[13]	—	—	—	—	—
1	2	0Eh	CKCN	A[14]	—	—	—	—	—
1	2	0Fh	WDCN	A[15]	—	—	—	—	—

Note: Register names that appear in *italics* indicate read-only registers. Register names that appear in **bold** indicate 16-bit registers. All other registers are 8 bits in width.

Table 4-2. System Register Bit Functions

REG	BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AP									—	—	—	—	AP (4 bits)			
APC									CLR	IDS	—	—	—	MOD2	MOD1	MOD0
PSF									Z	S	—	GPF1	GPF0	OV	C	E
IC									—	—	—	—	—	—	INS	IGE
IMR									IMS	—	—	—	IM3	IM2	IM1	IM0
SC									TAP	—	—	CDA0	—	—	PWL	—
IIR									IIS	—	—	—	II3	II2	II1	II0
CKCN									—	RGSL	RGMD	STOP	SWB	PMME	CD1	CD0
WDCN									POR	EWDI	WD1	WD0	WDIF	WTRF	EWT	RWT
A[0:15]	A[0:15] (16 bits)															
PFX	PFX (16 bits)															
IP	IP (16 bits)															
SP	—	—	—	—	—	—	—	—	—	—	—	—	SP (4 bits)			
IV	IV (16 bits)															
LC[0]	LC[0] (16 bits)															
LC[1]	LC[1] (16 bits)															

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 4-2. System Register Bit Functions (continued)

REG	BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFS	OFFS (8 bits)															
DPC	—	—	—	—	—	—	—	—	—	—	—	WBS2	WBS1	WBS0	SDPS1	SDPS0
GR	GR (16 bits)															
GRL									GR.7	GR.6	GR.5	GR.4	GR.3	GR.2	GR.1	GR.0
BP	BP (16 bits)															
GRS	GR.7	GR.6	GR.5	GR.4	GR.3	GR.2	GR.1	GR.0	GR.15	GR.14	GR.13	GR.12	GR.11	GR.10	GR.9	GR.8
GRH									GR.15	GR.14	GR.13	GR.12	GR.11	GR.10	GR.9	GR.8
GRXL	GR.7	GR.7	GR.7	GR.7	GR.7	GR.7	GR.7	GR.7	GR.7	GR.6	GR.5	GR.4	GR.3	GR.2	GR.1	GR.0
BP[OFFS]	BP[OFFS] (16 bits)															
DP[0]	DP[0] (16 bits)															
DP[1]	DP[1] (16 bits)															

Table 4-3. System Register Reset Values

REG	BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AP									0	0	0	0	0	0	0	0
APC									0	0	0	0	0	0	0	0
PSF									0	0	0	0	0	0	0	0
IC									0	0	0	0	0	0	0	0
IMR									0	0	0	0	0	0	0	0
SC									1	0	0	0	0	0	s	0
IIR									0	0	0	0	0	0	0	0
CKCN									1	s	s	0	0	0	0	0
WDCN									s	s	0	0	0	s	s	0
A[0:15]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PFX	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
IP	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SP	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
IV	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LC[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
LC[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OFFS									0	0	0	0	0	0	0	0
DPC	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0
GR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRL									0	0	0	0	0	0	0	0
BP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRS	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GRH									0	0	0	0	0	0	0	0
GRXL	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
BP[OFFS]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DP[0]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DP[1]	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: Bits marked as "s" have special behavior upon reset; see the register descriptions for details.

MAXQ Family User's Guide: MAXQ8913 Supplement

4.1 System Register Descriptions

This section details the functionality of any system register contained in the MAXQ8913 that operates differently from its description in the *MAXQ Family User's Guide*. Addresses for all system and peripheral registers are given as "Mx[yy]," where x is the module number (from 0 to 15 decimal) and yy is the register index (from 00h to 1Fh hexadecimal). Fields in the bit definition tables are defined as follows:

- **Name:** Symbolic names of bits or bit fields in this register.
- **Reset:** The value of each bit in this register following a standard reset. If this field reads "unchanged," the given bit is unaffected by standard reset. If this field reads "s," the given bit does not have a fixed 0 or 1 reset value because its value is determined by another internal state or external condition.
- **POR:** If present this field defines the value of each bit in this register following a power-on reset (as opposed to a standard reset). Some bits are unaffected by standard resets and are set/cleared by POR only.
- **Access:** Bits can be read-only (r) or read/write (rw). Any special restrictions or conditions that could apply when reading or writing this bit are detailed in the bit description.

4.1.1 Processor Status Flags Register (PSF, M8[04h])

Bit #	7	6	5	4	3	2	1	0
Name	Z	S	—	GPF1	GPF0	OV	C	E
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	rw	rw	rw	rw	rw

This register operates as described in the *MAXQ Family User's Guide*, with the exception that the overflow bit (OV) can be written by software.

4.1.2 Interrupt Mask Register (IMR, M8[06h])

Bit #	7	6	5	4	3	2	1	0
Name	IMS	—	—	—	IM3	IM2	IM1	IM0
Reset	0	0	0	0	0	0	0	0
Access	rw	r	r	r	rw	rw	rw	rw

The first four bits in this register are interrupt mask bits for modules 0 to 3, one bit per module. The eighth bit, IMS, serves as a mask for any system module interrupt sources. Setting a mask bit allows the enabled interrupt sources for the associated module or system (with IMS) to generate interrupt requests. Clearing the mask bit effectively disables all interrupt sources associated with that module or, in the case of IMS, all system interrupt sources. The IMR register is intended to facilitate user-definable interrupt prioritization.

Bit 7: System Module Interrupt Mask (IMS)

Bits 6:4: Reserved

Bit 3: Module 3 Interrupt Mask (IM3)

Bit 2: Module 2 Interrupt Mask (IM2)

Bit 1: Module 1 Interrupt Mask (IM1)

Bit 0: Module 0 Interrupt Mask (IM0)

4.1.3 System Control Register (SC, M8[08h])

Bit #	7	6	5	4	3	2	1	0
Name	TAP	—	—	CDA0	—	—	PWL	—
Reset	1	0	0	0	0	0	Unchanged	0
POR	1	0	0	0	0	0	1	0
Access	rw	r	r	r	r	r	rw	r

Bit 7: Test Access (Debug) Port Enable (TAP)

0 = Debug port functions are disabled, and P0.0 to P0.3 can be used as general-purpose I/O pins.

1 = Port pins P0.0 to P0.3 are enabled to act as debug port (JTAG) inputs and outputs.

Bits 6:5, 3:2, 0: Reserved

Bit 4: Code Data Access (CDA0). Setting this bit to 0 or 1 enables access to either the low or high page of program memory in data space when accessing data in byte mode, as shown in Figure 2-3 and Figure 2-4. When accessing data space in word mode, the setting of this bit has no effect.

Bit 1: Password Lock (PWL). This bit defaults to 1 on power-on reset only. When this bit is 1, it requires a 32-byte password to be matched with the password in the program space (words 10h to 1Fh) before allowing access to the ROM loader's utilities for read/write of program memory and debug functions. Clearing this bit to 0 disables the password protection to the ROM loader.

4.1.4 Interrupt Identification Register (IIR, M8[0Bh])

Bit #	7	6	5	4	3	2	1	0
Name	IIS	—	—	—	I13	I12	I11	I10
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	r	r	r	r

The first three bits in this register indicate interrupts pending in modules 0 to 3, one bit per module. The eighth bit, IIS, indicates a pending system interrupt (from the watchdog timer or other system function). The interrupt pending flags are set only for enabled interrupt sources waiting for service. The interrupt pending flag is cleared when the pending interrupt source(s) within that module are disabled or when the interrupt flag(s) are cleared by software.

Bit 7: Interrupt Pending Flag for System Modules (IIS)

Bits 6:4: Reserved

Bit 3: Interrupt Pending Flag for Module 3 (I13)

Bit 2: Interrupt Pending Flag for Module 2 (I12)

Bit 1: Interrupt Pending Flag for Module 1 (I11)

Bit 0: Interrupt Pending Flag for Module 0 (I10)

MAXQ Family User's Guide: MAXQ8913 Supplement

4.1.5 System Clock Control Register (CKCN, M8[0Eh])

Bit #	7	6	5	4	3	2	1	0
Name	—	RGSL	RGMD	STOP	SWB	PMME	CD1	CD0
Reset	1	0	0	0	0	0	0	0
Access	r	rw	r	rw	rw	rw	rw*	rw*

*Unrestricted read access. This bit can only be modified when PMME = 0.

The CKCN register bit settings determine the system clock source and clock divider as described in Table 4-4.

Bit 7: Reserved

Bit 6: Ring Oscillator Select (RGSL)

0 = Selects the high-frequency clock source (external crystal/resonator or external clock input) as the system clock source.

1 = Selects the ring oscillator as the system clock source.

Bit 5: Ring Oscillator Mode (RGMD). This read-only status bit indicates the clock source that is currently being used.

0 = The high-frequency clock (external crystal or external clock input) is currently being used as the system clock source, because the ring oscillator is not selected (RGSL = 0).

1 = The ring oscillator is currently being used as the system clock source. This is either because it is selected as the clock source (RGSL = 1), or because the high-frequency clock source is in the process of warming up.

Bit 4: Stop-Mode Select (STOP). Setting this bit to 1 causes the processor to enter stop mode. This does not change the currently selected clock-divide ratio.

Bit 3: Switchback Enable (SWB). Setting this bit to 1 enables switchback mode. If power-management mode (divide by 256) is active and switchback is enabled, the PMME bit is cleared to 0 when any of the following conditions occur.

- An external interrupt is generated based on an edge detect.
- The serial port is enabled to receive data and detects a low condition on its data receive pin.
- The serial port has a byte written to its buffer register by software.
- The SPI interface is enabled in master mode, and the SPIB register is written by software.
- The SPI interface is enabled in slave mode, and an external master drives the \overline{SSEL} line low.
- A START condition occurs on the I²C bus, causing an I²C START interrupt to be generated.
- The power supply drops below the SVM threshold, causing an SVM interrupt to be generated.
- An ADC conversion is initiated by software by setting the ADCONV bit to 1.
- Debug mode is entered through command entry or a breakpoint match.

Triggering a switchback condition only clears the PMME bit; the settings of CD0 and CD1 remain the same. When power-management mode is active, the SWB bit cannot be set to 1 as long as any of the above conditions are true.

Bit 2: Power-Management Mode Enable (PMME)

Bits 1:0: Clock Divide 1:0 (CD[1:0]). These three bits control the divide ratio or enable power-management mode for the system clock as shown in Table 4-4. CD0 and CD1 can always be read, and they can be written as long as PMME = 0.

Setting the PMME bit to 1 activates PMM mode. While PMME is set to 1, CD0 and CD1 cannot be changed; their values determine the clock-divide ratio that is used when the processor exits power-management mode.

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 4-4. System Clock Modes

RGMD	SWB	PMME	CD1	CD0	SYSTEM CLOCK	SWITCHBACK
0	0	0	0	0	High-frequency clock/1	—
0	0	0	0	1	High-frequency clock/2	—
0	0	0	1	0	High-frequency clock/4	—
0	0	0	1	1	High-frequency clock/8	—
0	0	1	X	X	High-frequency clock/256	Not Active
0	1	1	X	X	High-frequency clock/256	Active
1	0	0	0	0	Ring oscillator clock/1	—
1	0	0	0	1	Ring oscillator clock/2	—
1	0	0	1	0	Ring oscillator clock/4	—
1	0	0	1	1	Ring oscillator clock/8	—
1	0	1	X	X	Ring oscillator clock/256	Not Active
1	1	1	X	X	Ring oscillator clock/256	Active

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 5: PERIPHERAL REGISTER MODULES

Table 5-1. Peripheral Register Map

CYCLES TO READ	CYCLES TO WRITE	REGISTER INDEX	M0	M1	M2	M3	M4	M5
1	1	00h	PO0	SCON	MCNT	ADST	—	—
1	1	01h	PO1	SBUF	MA	ADADDR	—	—
1	1	02h	EIF0	SPICN	MB	DAC1OUT	—	—
1	1	03h	EIE0	SPIB	MC2	DAC2OUT	—	—
1	1	04h	EIF1	I2CCN	MC1	DAC3OUT	—	—
1	1	05h	EIE1	I2CST	MC0	DAC4OUT	—	—
1	1	06h	SVM	I2CBUF	TBCN	AMPCN	—	—
1	1	07h	—	I2CIE	TBR	ISINKCN	—	—
1	2	08h	<i>PI0</i>	SMD	MC1R	ADCN	—	—
1	2	09h	<i>PI1</i>	PR	MC0R	ADDATA	—	—
1	2	0Ah	EIES0	SPICF	TBV	OPMCN	—	—
1	2	0Bh	EIES1	SPICK	TBC	DACEN	—	—
1	2	0Ch	PWCN	I2CCK	—	TEMPEN	—	—
1	2	0Dh	PID0	I2CTO	—	—	—	—
1	2	0Eh	—	I2CSLA	—	—	—	—
1	2	0Fh	—	—	—	—	—	—
2	2	10h	PD0	—	—	—	—	—
2	2	11h	PD1	—	—	—	—	—
2	2	12h	—	—	—	—	—	—
2	2	13h	—	—	—	—	—	—
2	2	14h	—	—	—	—	—	—
2	2	15h	—	—	—	—	—	—
2	2	16h	—	—	—	—	—	—
2	2	17h	—	—	—	—	—	—
2	2	18h	—	—	—	—	—	—
2	2	19h	—	—	—	—	—	—
2	2	1Ah	—	—	—	—	—	—
2	2	1Bh	—	—	—	—	—	—
2	2	1Ch	—	—	—	—	—	—
2	2	1Dh	—	—	—	—	—	—
2	2	1Eh	—	—	—	—	—	—
2	2	1Fh	—	—	—	—	—	—

Note: Register names that appear in italics indicate registers in which all bits are read-only. Register names that appear in bold indicate 16-bit registers. All other registers are 8 bits in width.

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 5-2. Peripheral Register Bit Functions

REG	BIT																	
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
PO0	PO0 (8 bits)																	
PO1	PO1 (4 bits)																	
EIF0									IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0		
EIE0									EX7	EX6	EX5	EX4	EX3	EX2	EX1	EX0		
EIF1											IE11	IE10	IE9	IE8				
EIE1											EX11	EX10	EX9	EX8				
SVM	—	—	—	—	SVTH (4 bits)				—	—	—	SVM STOP	SVM I	SVM IE	SVM RDY	SVM EN		
PIO	PIO (8 bits)																	
PI1	PI1 (4 bits)																	
EIES0									IT7	IT6	IT5	IT4	IT3	IT2	IT1	IT0		
EIES1											IT11	IT10	IT9	IT8				
PWCN									BOD	REGEN							HFXD	
PID0	PID0 (8 bits)																	
PD0	PD0 (8 bits)																	
PD1	PD1 (5 bits)																	
SCON									SM0/FE	SM1	SM2	REN	TB8	RB8	TI	RI		
SBUF	SBUF (8 bits)																	
SPICN									STBY	SPIC	ROVR	WCOL	MODF	MODFE	MSTM	SPIEN		
SPIB	SPIB (16 bits)																	
I2CCN	I2C RST	—	—	—	—	—	I2C STREN	I2C GCEN	I2C STOP	I2C START	I2C ACK	I2C STRS	I2C EA	I2C MODE	I2C MST	I2C EN		
I2CST	I2C BUS	I2C BUSY	—	—	I2C SPI	I2C SCL	I2C ROI	I2C GC I	I2C NACK I	I2C ALI	I2C AMI	I2C TOI	I2C STR I	I2C RX I	I2C TX I	I2C SRI		
I2CBUF	I2CBUF (10 bits)																	
I2CIE	—	—	—	—	I2C SPIE	—	I2C ROIE	I2C GCIE	I2C NACKIE	I2C ALIE	I2C AMIE	I2C TOIE	I2C STRIE	I2C RXIE	I2C TXIE	I2C SRIE		
SMD									—	—	—	—	—	—	ESI	SMOD	FEDE	
PR	PR (16 bits)																	
SPICF									ESPII	SAS			—	—	—	CHR	CKPHA	CKPOL
SPICK	SPICK (8 bits)																	
I2CCK	I2CCKH (8 bits)								I2CCKL (8 bits)									
I2CTO	I2CTO (8 bits)																	
I2CSLA	I2CSLA (10 bits)																	
MCNT									OF	MCW	CLD	SQU	OPCS	MSUB	MMAC	SUS		
MA	MA (16 bits)																	
MB	MB (16 bits)																	
MC2	MC2 (16 bits)																	
MC1	MC1 (16 bits)																	
MC0	MC0 (16 bits)																	
TBCN	C/TB	—	—	TBCS	TBCR	TBPS2	TBPS1	TBPS0	TFB	EXFB	TBOE	DCEN	EXENB	TRB	ETB	CP/RLB		
TBR	TBR (16 bits)																	
MC1R	MC1R (16 bits)																	

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 5-2. Peripheral Register Bit Functions (continued)

REG	BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MC0R	MC0R (16 bits)															
TBV	TBV (16 bits)															
TBC	TBC (16 bits)															
ADST	—	—	—	—	ADDAT (4 bits)				REFOK	ADC ONV	ADDAI	ADC FG	ADIDX (4 bits)			
ADADDR	—	—	—	—	SEQSTORE (4 bits)				—	SEQSTART (3 bits)			—	SEQEND (3 bits)		
DAC1OUT	—	—	—	—	—	—	DAC1OUT (10 bits)									
DAC2OUT	—	—	—	—	—	—	DAC2OUT (10 bits)									
DAC3OUT	DAC3OUT (8 bits)															
DAC4OUT	DAC4OUT (8 bits)															
AMPCN									—	AMP FLT	AMPIF	AMPIE	AMP CK1	AMP CK0	AMP EN1	AMP EN0
ISINKCN	ISINK2 (8 bits)								ISINK1 (8 bits)							
ADCN	—	—	—	—	ADINT 1	ADINT 0	ADCLK 1	ADCLK 0	IREF EN	AD CONT	AD DAIE	AD PMO	ADACQ (4 bits)			
ADDATA	ADDATA (16 bits)															
OPMCN									—	—	—	—	OPM END	OPM ENC	OPM ENB	OPM ENA
DACEN									—	—	DAC OUT2	DAC OUT1	DAC EN4	DAC EN3	DAC EN2	DAC EN1
TEMPEN									—	—	—	—	—	—	—	TSEN

Table 5-3. Peripheral Register Bit Reset Values

REG	BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PO0									1	1	1	1	1	1	1	1
PO1									0	0	0	1	1	1	1	1
EIF0									0	0	0	0	0	0	0	0
EIE0									0	0	0	0	0	0	0	0
EIF1									0	0	0	0	0	0	0	0
EIE1									0	0	0	0	0	0	0	0
SVM	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
PI0									s	s	s	s	s	s	s	s
PI1									0	0	0	s	s	s	s	s
EIES0									0	0	0	0	0	0	0	0
EIES1									0	0	0	0	0	0	0	0
PWCN	0	0	0	0	0	0	0	0	0	0	0	0	s	s	0	0
PID0									0	0	0	0	0	0	0	0
PD0									0	0	0	0	0	0	0	0
PD1									0	0	0	0	0	0	0	0
SCON									0	0	0	0	0	0	0	0
SBUF									0	0	0	0	0	0	0	0
SPICN									0	0	0	0	0	0	0	0

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 5-3. Peripheral Register Bit Reset Values (continued)

REG	BIT															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPIB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I2CCN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I2CST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I2CBUF	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I2CIE	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SMD									0	0	0	0	0	0	0	0
PR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SPICF									0	0	0	0	0	0	0	0
SPICK									0	0	0	0	0	0	0	0
I2CCK	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0
I2CTO									0	0	0	0	0	0	0	0
I2CSLA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MCNT									0	0	0	0	0	0	0	0
MA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TBCN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TBR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC1R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
MC0R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TBV	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
TBC	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ADST	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ADADDR	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DAC1OUT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DAC2OUT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
DAC3OUT									0	0	0	0	0	0	0	0
DAC4OUT									0	0	0	0	0	0	0	0
AMPCN									0	1	0	0	0	0	0	0
ISINKCN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ADCN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ADDATA	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
OPMCN									0	0	0	0	0	0	0	0
DACEN									0	0	0	0	0	0	0	0
TEMPEN									0	0	0	0	0	0	0	0

Note: Bits marked as "s" have special behavior upon reset; see the register descriptions for details.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 6: GENERAL-PURPOSE I/O MODULE

The MAXQ8913 provides up to 12 port pins for general-purpose I/O that are grouped into logical ports P0 and P1. Each of these port pins has the following features:

- CMOS output drivers
- Schmitt trigger inputs
- Optional weak pullup to DVDD when operating in input mode

For the eight GPIO pins located in port 0 (P0.0 to P0.7) the Schmitt trigger inputs can be explicitly disabled on a pin-by-pin basis by setting bits in the PID0 register.

Many of the port pins on the MAXQ8913 are also multiplexed with special and alternate functions as listed below. All these functions are disabled by default with the exception of the debug port interface pins, which are enabled by default following any reset. The behavior of these functions breaks down into two overall categories.

- **Special functions** override the PD and PO settings for the port pin when they are enabled. Once the special function takes control, normal control of the port pin is lost until the special function completes its task or is disabled. Examples of special functions include serial port transmit and I²C clock and data.
- **Alternate functions** operate in parallel with the PD and PO settings for the port pin, and generally consist of input-only functions such as external interrupts. When an alternate function is enabled for a port pin, the port pin's output state is still controlled in the usual manner.

Table 6-1. Port Pin Special and Alternate Functions

PORT PIN	FUNCTION TYPE	FUNCTION	ENABLED WHEN
P0.0	Alternate	External Interrupt 0	(EIE0.0) EX0 = 1
P0.0	Alternate	JTAG Interface—TAP Clock (TCK)	(SC.7) TAP = 1
P0.1	Alternate	External Interrupt 1	(EIE0.1) EX1 = 1
P0.1	Alternate	JTAG Interface—TAP Data Input (TDI)	(SC.7) TAP = 1
P0.2	Alternate	External Interrupt 2	(EIE0.2) EX2 = 1
P0.2	Alternate	JTAG Interface—TAP Mode Select (TMS)	(SC.7) TAP = 1
P0.3	Alternate	External Interrupt 3	(EIE0.3) EX3 = 1
P0.3	Special	JTAG Interface—TAP Data Output (TDO)	(SC.7) TAP = 1
P0.4	Alternate	External Interrupt 4	(EIE0.4) EX4 = 1
P0.4	Alternate	SPI Slave Select Input ($\overline{\text{SSEL}}$)	SPIEN = 1 and MSTM = 0
P0.5	Alternate	External Interrupt 5	(EIE0.5) EX5 = 1
P0.5	Alternate	SCLK—SPI Clock Input (Slave Mode)	SPIEN = 1 and MSTM = 0
P0.5	Special	SCLK—SPI Clock Output (Master Mode)	SPIEN = 1 and MSTM = 1
P0.6	Alternate	External Interrupt 6	(EIE0.6) EX6 = 1
P0.6	Alternate	MOSI—SPI Slave Input (Slave Mode)	SPIEN = 1 and MSTM = 0
P0.6	Special	MOSI—SPI Master Output (Master Mode)	SPIEN = 1 and MSTM = 1
P0.7	Alternate	External Interrupt 7	(EIE0.7) EX7 = 1
P0.7	Special	MISO—SPI Slave Output (Slave Mode)	SPIEN = 1 and MSTM = 0
P0.7	Alternate	MISO—SPI Master Input (Master Mode)	SPIEN = 1 and MSTM = 1
P1.0	Alternate	External Interrupt 8	(EIE1.0) EX8 = 1
P1.0	Special	Serial Port Data (Mode 0)	Mode 0: SBUF is written or REN is set to 1 (until serial transmission completes)
P1.0	Alternate	Serial Port Receive (Modes 1/2/3)	Modes 1/2/3: REN = 1
P1.0	Special	SCL—I ² C Clock Line	I ² CEN=1 (overrides serial port Rx)
P1.1	Alternate	External Interrupt 9	(EIE1.1) EX9 = 1

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 6-1. Port Pin Special and Alternate Functions (continued)

PORT PIN	FUNCTION TYPE	FUNCTION	ENABLED WHEN
P1.1	Special	Serial Port 0 Transmit/Clock	Mode 0: REN set to 1 (until serial transmission completes) Mode 1/2/3: SBUF is written (until serial transmission completes)
P1.1	Special	SDA—I ² C Data Line	I2CEN = 1 (overrides serial port Tx)
P1.2	Alternate	External Interrupt 10	(EIE1.2) EX10 = 1
P1.2	Alternate	Type B Timer 0 Input A (Counter Input)	C/TB = 1
P1.2	Special	Type B Timer 0 Output A (Clock Output)	C/TB = 0 and TB0E = 1
P1.3	Alternate	External Interrupt 11	(EIE1.3) EX11 = 1
P1.3	Alternate	Type B Timer 0 Input B	EXENB = 1
P1.3	Special	Type B Timer 0 Output B (Compare Output)	TBCR:TBCS<>00b

The port pins on the MAXQ8913 operate the same as standard MAXQ port pins, with input/output states defined in Table 6-2. Refer to the IC data sheet for specific voltage and current characteristics of port pins in input or output mode.

Table 6-2. P1[3:0] Input/Output States (in Standard Mode)

PD1.y	PO1.y	PORT PIN MODE	PORT PIN (Px.y) STATE	PI1.x READ VALUE
0	0	Input	Three-State	Depends on external signal
0	1	Input	Weak pullup high	1 (unless overdriven externally)
1	0	Output	Strong drive low	0 (unless overdriven externally)
1	1	Output	Strong drive high	1 (unless overdriven externally)

The port pins P0[7:0] on the MAXQ8913 have an additional operating mode available that allows the Schmitt input triggers on individual pins to be enabled or disabled. When PID0.x is set to 1, the corresponding input bit PIO.x always reads 0, regardless of the logic level that is currently present at the pin. When PID0.x is set to 0, the input bit PIO.x operates normally.

Table 6-3. P0[7:0] Input/Output States (in Standard Mode)

PD0.x	PO0.x	PID0.x	PORT PIN MODE	PORT PIN (Px.y) STATE	PIO.x READ VALUE
0	0	0	Input	Three-State	Depends on external signal
0	0	1	No function	Three-State	0
0	1	0	Input	Weak pullup high	1 (unless overdriven externally)
0	1	1	Weak pullup	Weak pullup High	0
1	0	0	Output	Strong drive low	0 (unless overdriven externally)
1	0	1	Output	Strong drive low	0
1	1	0	Output	Strong drive high	1 (unless overdriven externally)
1	1	1	Output	Strong drive high	0

MAXQ Family User's Guide: MAXQ8913 Supplement

6.1 GPIO and External Interrupt Register Descriptions

The following peripheral registers are used to control the general-purpose I/O and external interrupt features specific to the MAXQ8913. Addresses of registers are given as “Mx[yy],” where x is the module number (from 0 to 15 decimal) and yy is the register index (from 00h to 1Fh hexadecimal). Fields in the bit definition tables are defined as follows:

- **Name:** Symbolic names of bits or bit fields in this register.
- **Reset:** The value of each bit in this register following a standard reset. If this field reads “unchanged,” the given bit is unaffected by standard reset. If this field reads “s,” the given bit does not have a fixed 0 or 1 reset value because its value is determined by another internal state or external condition.
- **POR:** If present this field defines the value of each bit in this register following a power-on reset (as opposed to a standard reset). Some bits are unaffected by standard resets and are set/cleared by POR only.
- **Access:** Bits can be read-only (r) or read/write (rw). Any special restrictions or conditions that could apply when reading or writing this bit are detailed in the bit description.

6.1.1 Port 0 Direction Register (PD0, M0[10h])

Bit #	7	6	5	4	3	2	1	0
Name	PD0							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Each of the bits in this register controls the input/output direction of a port pin (P0.0 to P0.7) as follows:

0 = The port pin is in input mode, either with a weak pullup (if PO = 1) or three-stated (if PO = 0).

1 = The port pin is in output mode, with the output level to drive given by PO.

6.1.2 Port 1 Direction Register (PD1, M0[11h])

Bit #	7	6	5	4	3	2	1	0
Name	—	—	—	—	PD1			
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Each of the bits in this register controls the input/output direction of a port pin (P1.0 to P1.3) as follows:

0 = The port pin is in input mode, either with a weak pullup (if PO = 1) or three-stated (if PO = 0).

1 = The port pin is in output mode, with the output level to drive given by PO.

6.1.3 Port 0 Input Disable Register (PID0, M0[0Dh])

Bit #	7	6	5	4	3	2	1	0
Name	PID0							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Each of the bits in this register controls the Schmitt trigger enable of a port pin (P0.0 to P0.7) as follows:

0 = The input Schmitt trigger is enabled; the port pin operates normally.

1 = The input Schmitt trigger is disabled; the port pin cannot be used for input.

MAXQ Family User's Guide: MAXQ8913 Supplement

6.1.4 Port 0 Output Register (PO0, M0[00h])

Bit #	7	6	5	4	3	2	1	0
Name	PO0							
Reset	1	1	1	1	1	1	1	1
Access	rw	rw	rw	rw	rw	rw	rw	rw

This register stores the data that is output on any of the pins of port 0 that have been defined as output pins. If the port pins are in input mode, this register controls the weak pullup for each pin. Changing the data direction of any pins for this port (through register PD0) does not affect the value in this register.

6.1.5 Port 1 Output Register (PO1, M0[01h])

Bit #	7	6	5	4	3	2	1	0
Name	—	—	—	—	PO1			
Reset	0	0	0	0	1	1	1	1
Access	r	r	r	r	rw	rw	rw	rw

This register stores the data that is output on any of the pins of port 1 that have been defined as output pins. If the port pins are in input mode, this register controls the weak pullup for each pin. Changing the data direction of any pins for this port (through register PD1) does not affect the value in this register.

6.1.6 Port 0 Input Register (PIO, M0[08h])

Bit #	7	6	5	4	3	2	1	0
Name	PIO							
Reset	s	s	s	s	s	s	s	s
Access	r	r	r	r	r	r	r	r

Each of the read-only bits in this register reflects the logic state present at the corresponding port pin, unless the Schmitt trigger for that pin has been disabled by setting PID0.x to 1.

6.1.7 Port 1 Input Register (PI1, M0[09h])

Bit #	7	6	5	4	3	2	1	0
Name	—	—	—	—	PI1			
Reset	0	0	0	0	s	s	s	s
Access	r	r	r	r	r	r	r	r

Each of the read-only bits in this register reflects the logic state present at the corresponding port pin.

MAXQ Family User's Guide: MAXQ8913 Supplement

6.1.8 External Interrupt Flag 0 Register (EIF0, M0[02h])

Bit #	7	6	5	4	3	2	1	0
Name	IE7	IE6	IE5	IE4	IE3	IE2	IE1	IE0
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Each bit in this register is set when a negative or positive edge (depending on the ITn bit setting) is detected on the corresponding interrupt pin. Once an external interrupt has been detected, the interrupt flag bit remains set until cleared by software or a reset. Setting any of these bits causes the corresponding interrupt to trigger if it is enabled to do so.

Bit 7: External Interrupt 7 Edge Detect (IE7)

Bit 6: External Interrupt 6 Edge Detect (IE6)

Bit 5: External Interrupt 5 Edge Detect (IE5)

Bit 4: External Interrupt 4 Edge Detect (IE4)

Bit 3: External Interrupt 3 Edge Detect (IE3)

Bit 2: External Interrupt 2 Edge Detect (IE2)

Bit 1: External Interrupt 1 Edge Detect (IE1)

Bit 0: External Interrupt 0 Edge Detect (IE0)

6.1.9 External Interrupt Flag 1 Register (EIF1, M0[04h])

Bit #	7	6	5	4	3	2	1	0
Name	—	—	—	—	IE11	IE10	IE9	IE8
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	rw	rw	rw	rw

Each bit in this register is set when a negative or positive edge (depending on the ITn bit setting) is detected on the corresponding interrupt pin. Once an external interrupt has been detected, the interrupt flag bit remains set until cleared by software or a reset. Setting any of these bits causes the corresponding interrupt to trigger if it is enabled to do so.

Bits 7:4: Reserved

Bit 3: External Interrupt 11 Edge Detect (IE11)

Bit 2: External Interrupt 10 Edge Detect (IE10)

Bit 1: External Interrupt 9 Edge Detect (IE9)

Bit 0: External Interrupt 8 Edge Detect (IE8)

MAXQ Family User's Guide: MAXQ8913 Supplement

6.1.10 External Interrupt Enable 0 Register (EIE0, M0[03h])

Bit #	7	6	5	4	3	2	1	0
Name	EX7	EX6	EX5	EX4	EX3	EX2	EX1	EX0
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Each bit in this register controls the enable for one external interrupt. If the bit is set to 1, the interrupt is enabled (if it is not otherwise masked). If the bit is set to 0, its corresponding interrupt is disabled.

Bit 7: External Interrupt 7 Enable (EX7)

Bit 6: External Interrupt 6 Enable (EX6)

Bit 5: External Interrupt 5 Enable (EX5)

Bit 4: External Interrupt 4 Enable (EX4)

Bit 3: External Interrupt 3 Enable (EX3)

Bit 2: External Interrupt 2 Enable (EX2)

Bit 1: External Interrupt 1 Enable (EX1)

Bit 0: External Interrupt 0 Enable (EX0)

6.1.11 External Interrupt Enable 1 Register (EIE1, M0[05h])

Bit #	7	6	5	4	3	2	1	0
Name	—	—	—	—	EX11	EX10	EX9	EX8
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	rw	rw	rw	rw

Each bit in this register controls the enable for one external interrupt. If the bit is set to 1, the interrupt is enabled (if it is not otherwise masked). If the bit is set to 0, its corresponding interrupt is disabled.

Bits 7:4: Reserved

Bit 3: External Interrupt 11 Enable (EX11)

Bit 2: External Interrupt 10 Enable (EX10)

Bit 1: External Interrupt 9 Enable (EX9)

Bit 0: External Interrupt 8 Enable (EX8)

MAXQ Family User's Guide: MAXQ8913 Supplement

6.1.12 External Interrupt Edge Select 0 Register (EIES0, M0[0Ah])

Bit #	7	6	5	4	3	2	1	0
Name	IT7	IT6	IT5	IT4	IT3	IT2	IT1	IT0
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Each bit in this register controls the edge select mode for an external interrupt as follows:

0 = The external interrupt triggers on a rising (positive) edge.

1 = The external interrupt triggers on a negative (falling) edge.

Bit 7: Edge Select for External Interrupt 7 (IT7)

Bit 6: Edge Select for External Interrupt 6 (IT6)

Bit 5: Edge Select for External Interrupt 5 (IT5)

Bit 4: Edge Select for External Interrupt 4 (IT4)

Bit 3: Edge Select for External Interrupt 3 (IT3)

Bit 2: Edge Select for External Interrupt 2 (IT2)

Bit 1: Edge Select for External Interrupt 1 (IT1)

Bit 0: Edge Select for External Interrupt 0 (IT0)

6.1.13 External Interrupt Edge Select 1 Register (EIES1, M0[0Bh])

Bit #	7	6	5	4	3	2	1	0
Name	—	—	—	—	IT11	IT10	IT9	IT8
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	rw	rw	rw	rw

Each bit in this register controls the edge select mode for an external interrupt as follows:

0 = The external interrupt triggers on a rising (positive) edge.

1 = The external interrupt triggers on a negative (falling) edge.

Bits 7:4: Reserved

Bit 3: Edge Select for External Interrupt 11 (IT11)

Bit 2: Edge Select for External Interrupt 10 (IT10)

Bit 1: Edge Select for External Interrupt 9 (IT9)

Bit 0: Edge Select for External Interrupt 8 (IT8)

6.2 Port Pin Examples

6.2.1 Port Pin Example 1: Driving Outputs on Port 0

```
move P00, #000h      ; Set all outputs low
move PD0, #0FFh      ; Set all P0 pins to output mode
```

6.2.2 Port Pin Example 2: Receiving Inputs on Port 0

```
move P00, #0FFh      ; Set weak pullups ON on all pins
move PD0, #000h      ; Set all P0 pins to input mode

nop                   ; Wait for external source to drive P0 pins

move Acc, P10         ; Get input values from P0 (will return FF if
                       ; no other source drives the pins low)
```


MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 7: TIMER/COUNTER 0 MODULE

The MAXQ8913 does not provide this peripheral module.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 8: TIMER/COUNTER 1 MODULE

The MAXQ8913 does not provide this peripheral module.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 9: TIMER/COUNTER 2 MODULE

The MAXQ8913 does not provide this peripheral module.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 10: SERIAL I/O MODULE

The MAXQ8913 provides one serial universal synchronous/asynchronous receiver-transmitter (USART) interface that operates as described in the *MAXQ Family User's Guide*.

10.1 Serial USART I/O Pins and Control Registers

Table 10-1. Serial USART Input and Output Pins

SERIAL USART FUNCTION	PIN	MULTIPLEXED WITH GPIO
RX: Serial Receive	M3	P1.1
TX: Serial Transmit	N4	P1.0

Table 10-2. Serial USART Control Registers

REGISTER	ADDRESS	FUNCTION
SCON	M1[00h]	Serial Port Control Register. Serial port mode, receive enable, 9th bit control, and interrupt flags.
SBUF	M1[01h]	Serial Port Data Buffer. Input and output data buffer.
SMD	M1[08h]	Serial Port Mode Register. Controls baud rate, interrupt enable, and framing error detection.
PR	M1[09h]	Serial Port Phase Register. Contains counter reload value for baud-rate generation.

10.2 Serial USART Code Examples

10.2.1 Serial USART Example: Echo Characters in 10-Bit Asynchronous Mode

```
move SCON.6, #1      ; Set to mode 1 (10-bit asynchronous)
move SCON.4, #1      ; Enable receiver
move SMD.1, #1       ; Baud rate = 16 x baud clock
move PR, #007DDh    ; P = 2^21 * 9600 / 10.000MHz

move SCON.0, #0      ; Clear received character flag
move SCON.1, #0      ; Clear transmit character flag
```

```
move Acc, #0Dh
call TxChar
move Acc, #0Ah
call TxChar
move Acc, #'>'
call TxChar
move Acc, #' `
call TxChar
```

```
mainLoop:
call RxChar
call TxChar
jump mainLoop
```

MAXQ Family User's Guide: MAXQ8913 Supplement

```
=====
;=
;= TxChar - Outputs a character to serial port.
;=
;= Inputs : Acc - Character to send.
;=

TxChar:
    move  SBUF, Acc          ; Send character
TxChar_Loop:
    move  C, SCON.1         ; Check transmit flag
    sjump NC, TxChar_Loop  ; Stall until last transmit has completed
    move  SCON.1, #0        ; Clear the transmit flag
    ret

;=====
;=
;= RxChar - Receives a character from serial port.
;=
;= Outputs : Acc - Character received.
;=

RxChar:
    move  C, SCON.0         ; Wait for receive flag to be set to 1
    sjump NC, RxChar
    move  Acc, SBUF         ; Get received character
    move  SCON.0, #0       ; Clear receive interrupt flag
    ret
```

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 11: SERIAL PERIPHERAL INTERFACE (SPI) MODULE

The MAXQ8913 provides a serial peripheral interface (SPI) module, which operates as described in the *MAXQ Family User's Guide* with the following additions:

- The maximum clock rate when operating in slave mode is the system clock divided by 4.
- The SPI configuration register (SPICF) contains an additional bit, slave active select (SAS, SPICF.6). This bit allows the sense of the slave-select input to be changed from active low (SAS = 0, default) to active high (SAS = 1).

11.1 SPI Input/Output Pins and Control Registers

Table 11-1. SPI Input and Output Pins

SPI INTERFACE FUNCTION	PIN	MULTIPLEXED WITH GPIO
$\overline{\text{SSEL}}$: Slave Select	N8	P0.4
SCLK: Slave Clock	M7	P0.5
MOSI: Master Out-Slave In	L6	P0.6
MISO: Master In-Slave Out	K5	P0,7

Table 11-2. SPI Control Registers

REGISTER	ADDRESS	FUNCTION
SPICN	M1[02h]	SPI Control Register. Enable, master/slave-mode select, and status and interrupt flags.
SPICF	M1[0Ah]	SPI Configuration Register. Clock polarity/phase, character length, interrupt enable, $\overline{\text{SSEL}}$ active-high/low mode select (SAS).
SPICK	M1[0Bh]	SPI Clock Register. Master baud rate = $0.5 \times \text{Sysclk}/(\text{SPICK} + 1)$.
SPIB	M1[03h]	SPI Data Buffer. Writes go to the SPI write buffer; reads come from the SPI read buffer.

11.2 SPI Code Examples

11.2.1 SPI Example 1: Transmitting Data in Master Mode

```
move PD5.2, #1          ; Chip select for slave device
move PO5.2, #1          ; Start high

move SPICN, #03h       ; Enable SPI in master mode
move SPICF, #00h       ; Sample data at clock rising edge, 8 bit character
move SPICK, #63        ; SPI clock = sysclk/128

move PO5.2, #0         ; Drive chip select low
move SPIB, #12h        ; Transmit byte
call waitXfer
move SPIB, #34h        ; Transmit byte
call waitXfer
move PO5.2, #1         ; Release chip select

....

waitXfer:
move C, SPICN.6        ; Wait for transfer to complete
jump NC, waitXfer
move SPICN.6, #0       ; Clear transfer flag
ret
```

11.2.2 SPI Example 2: Receiving Data in Slave Mode

```
move SPICN, #01h      ; Enable SPI in slave mode
move SPICF, #00h     ; Sample data at clock rising edge, 8 bit character
```

```
call getByte
move A[0], GR
call getByte
move A[1], GR
call getByte
move A[2], GR
call getByte
move A[3], GR
```

...

getByte:

```
move C, SPICN.6      ; Wait for transfer to complete
jump NC, getByte
move SPICN.6, #0     ; Clear transfer flag
move GR, SPIB        ; Get character
ret
```

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 12: HARDWARE MULTIPLIER MODULE

The MAXQ8913 provides a hardware multiplier module that provides the following features (detailed in the *MAXQ Family User's Guide*):

- Completes a 16-bit x 16-bit multiply-accumulate or multiply-subtract operation in a single cycle
- Includes 48-bit accumulator
- Supports seven different multiplication operations:
 - Unsigned 16-bit multiply
 - Unsigned 16-bit multiply and accumulate
 - Unsigned 16-bit multiply and subtract
 - Signed 16-bit multiply
 - Signed 16-bit multiply and negate
 - Signed 16-bit multiply and accumulate
 - Signed 16-bit multiply and subtract

12.1 Hardware Multiplier Control Registers

The associated registers for this module are listed in Table 12-1.

Table 12-1. Hardware Multiplier Control Registers

REGISTER	ADDRESS	FUNCTION
MCNT	M2[00h]	Multiplier Control Register. Controls the operation and mode selection for the multiplier.
MA	M2[01h]	Multiplier Operand A Register. Input register for the multiplier operations.
MB	M2[02h]	Multiplier Operand B Register. Input register for the multiplier operations.
MC2	M2[03h]	Multiplier Accumulate Register 2. Contains bits 32 to 47 of the accumulator.
MC1	M2[04h]	Multiplier Accumulate Register 1. Contains bits 16 to 31 of the accumulator.
MC0	M2[05h]	Multiplier Accumulate Register 0. Contains bits 0 to 15 of the accumulator.
MC1R	M2[08h]	Multiplier Read Register 1. Contains bits 16 to 31 of the last multiply operation result.
MC0R	M2[09h]	Multiplier Read Register 0. Contains bits 0 to 15 of the last multiply operation result.

12.2 Hardware Multiplier Code Examples

12.2.1 Hardware Multiplier Example: Multiply and Square/Accumulate

```
move  MCNT,#021h      ; Unsigned multiply, no accumulate
move  MA,  #00155h
move  MB,  #000AAh    ; MC[2:0] = 00_0000_E272h

move  MCNT,#012h      ; Square single operand and accumulate
move  MA,  #000FFh    ; MC[2:0] = 00_0001_E073h
move  MB,  #00099h    ; MC[2:0] = 00_0002_3BE4h
```


MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 13: 1-Wire BUS MASTER

The MAXQ8913 does not provide this peripheral module.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 14: REAL-TIME CLOCK MODULE

The MAXQ8913 does not provide this peripheral module.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 15: TEST ACCESS PORT (TAP)

The JTAG/TAP port on the MAXQ8913 is multiplexed with port pins P0.0, P0.1, P0.2, and P0.3. These pins default to the JTAG/TAP function on reset, which means that the part is always ready for in-circuit debugging or in-circuit programming operations following any reset.

Once an application has been loaded and starts running, the JTAG/TAP port can still be used for in-circuit debugging operations. If in-circuit debugging functionality is not needed, the associated port pins can be reclaimed for application use by setting the TAP (SC.7) bit to 0. This disables the JTAG/TAP interface and allows the four pins to operate as normal port pins.

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 16: IN-CIRCUIT DEBUG MODE

The MAXQ8913 provides an in-circuit debugging interface through the debug port as described in the *MAXQ Family User's Guide*. This interface provides the following functions for use in debugging application software:

- Single-step (trace) execution
- Four program address breakpoints
- Two breakpoints configurable as data address or register address breakpoints
- Register read and write
- Program stack read
- Data memory read and write
- Optional password protection

The following sections provide specific notes on the operation of the MAXQ8913 in debugging mode.

16.1 Register Read and Write Commands

Any register location can be read or written using these commands, including reserved locations and those used for op code support. No protection is provided by the debugging interface, and avoiding side effects is the responsibility of the host system communicating with the MAXQ8913.

Writing to the IP register alters the address that execution resumes at once the debugging engine exits.

In general, reading a register through the debug interface returns the value that was in that register **before** the debugging engine was invoked. An exception to this rule is the SP register; reading the SP register through the debug interface actually returns the value (SP+1).

16.2 Data Memory Read Command

When invoking this command, ICDA should be set to the word address of the starting location to read from, and ICDD should be set to the number of words. The input address must be based on the utility ROM memory map, as shown in Figure 2-3.

Data memory words returned by this command are output LSB first.

16.3 Data Memory Write Command

When invoking this command, ICDA should be set to the word address of the location to write to, and ICDD should be set to the data word to write. The input address must be based on the utility ROM memory map, as shown in Figure 2-3.

16.4 Program Stack Read Command

When invoking this command, ICDA should be set to the address of the starting stack location (value of SP) to read from, and ICDD should be set to the number of words. The address given in ICDA is the highest value that is used, as words are popped off the stack and returned in descending order.

Stack words returned by this command are output LSB first.

16.5 Read Register Map Command

This command outputs all peripheral registers in the range M0[00h] to M4[1Fh], along with a fixed set of system registers. The following formatting rules apply to the returned data:

- System registers are output as 8 bits or 16 bits, least significant byte first.
- All peripheral registers are output as 16 bits, least significant byte first. The top byte of 8-bit registers are returned as 00h.
- Nonimplemented and reserved peripheral registers in the range M0[00h] to M4[1Fh] are represented as empty word values in Table 16-1. These values should be ignored.
- Registers SBUF, SPIB, I2CBUF, and ADDATA are not read, and their values are returned as 0000h.

MAXQ Family User's Guide: MAXQ8913 Supplement

The first byte output by this command is the value **160** (0A0h), which represents the number of peripheral register words output. Table 16-1 lists the remaining 352 bytes output by this command. Values shown as “—” are don't care values that should be ignored by the debugging host.

Table 16-1. Output from DebugReadMap Command

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xA	xB	xC	xD	xE	xF
0x	PO0		PO1		EIF0		EIE0		EIF1		EIE1		SVM		—	
1x	PI0		PI1		EIES0		EIES1		PWCN		PID0		—		—	
2x	PDO		PD1		—		—		—		—		—		—	
3x	—		—		—		—		—		—		—		—	
4x	SCON		—		SPICN		—		I2CCN		I2CST		—		I2CIE	
5x	SMD		PR		SPICF		SPICK		I2CCK		I2CTO		I2CSLA		—	
6x	—		—		—		—		—		—		—		—	
7x	—		—		—		—		—		—		—		—	
8x	MCNT		MA		MB		MC2		MC1		MC0		TBCN		TBR	
9x	MC1R		MC0R		TBV		TBC		—		—		—		—	
Ax	—		—		—		—		—		—		—		—	
Bx	—		—		—		—		—		—		—		—	
Cx	ADST		ADADDR		DAC1OUT		DAC2OUT		DAC3OUT		DAC4OUT		AMPCN		ISINKCN	
Dx	ADCN		—		OPMCN		DACEN		TEMPEN		—		—		—	
Ex	—		—		—		—		—		—		—		—	
Fx	—		—		—		—		—		—		—		—	
10x	—		—		—		—		—		—		—		—	
11x	—		—		—		—		—		—		—		—	
12x	AP	APC	PSF	IC	IMR	SC	IIR	CKCN	WDCN	00	A[0]	A[1]	A[2]			
13x	A[3]		A[4]		A[5]		A[6]		A[7]		A[8]		A[9]		A[10]	
14x	A[11]		A[12]		A[13]		A[14]		A[15]		IP		SP+1		IV	
15x	LC[0]		LC[1]		OFFS		DPC		GR		BP		DP[0]		DP[1]	

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 17: IN-SYSTEM PROGRAMMING (JTAG)

The MAXQ8913 provides a JTAG-compatible debug port interface for in-system programming (bootloader) operations. In order to use this interface for in-system programming, the SPE bit must be set through the debug port. This is done while the device is held in reset, using the system programming instruction as described in the *MAXQ Family User's Guide*.

17.1 JTAG Bootloader Protocol

The only bootloader interface supported for the MAXQ8913 is the debug port. When using the debug port, the clock rate (TCK) must be kept below 1/8 the system clock rate.

All bootloader commands begin with a single command byte. The high four bits of this command byte define the command family (from 0 to 15), while the low four bits define the specific command within that family.

All commands (except for those in Family 0) follow this format:

BYTE 1	BYTE 2	BYTE 3	BYTE 4	(LENGTH) BYTES/WORDS
Command	Length	Param 1	Param 2	Data

After each command has completed, the loader outputs a "prompt" byte to indicate that it has finished the operation. The prompt byte is the single character ">" (byte value 03Eh).

Bootloader commands that fail for any reason set the bootloader status byte to an error code value describing the reason for the failure. This status byte can be read by means of the Get Status command (04h).

Table 17-1. Bootloader Status Codes

STATUS VALUE	FUNCTION
00	No Error. The last command completed successfully.
01	Family Not Supported. An attempt was made to use a command from a family the bootloader does not support.
02	Invalid Command. An attempt was made to use a nonexistent command within a supported command family.
03	No Password Match. An attempt was made to use a password-protected command without first matching a valid password. Or, the Match Password command was called with an incorrect password value.
04	Bad Parameter. The parameter (address or otherwise) passed to the command was out of range or otherwise invalid.
05	Verify Failed. The verification step failed on a Load/Verify or Verify command.
06	Unknown Register. An attempt was made to read from or write to a nonexistent register.
07	Word Mode Not Supported. An attempt was made to set word-mode access, but the bootloader supports byte-mode access only.
08	Master Erase Failed. The bootloader was unable to perform master erase.

All commands in Family 0 can be executed without first matching the password. All other commands (in families 1x through Fx) are password protected; the password must first be matched before these commands can be executed.

A special case exists when the program memory has not been initialized (following master erase). If the password (stored in word locations 0010h to 001Fh in program memory) is all 0000h words or all FFFFh words, the bootloader treats the password as having been matched and automatically unlocks the password bit. This allows access to password-protected commands following master erase (when no password has been set in program memory).

When providing addresses for code or data read or write to bootloader commands, all addresses run from 0000h to (memory size – 1).

17.2 Family 0 Commands (Not Password Protected)

Command 00h—No Operation

I/O	BYTE 1
Input	00h
Output	

Command 01h—Exit Loader

This command causes the bootloader command loop to exit. Upon exit, the MAXQ8913 clears the SPE bit and resets itself internally. Following the internal reset, execution jumps to the beginning of application code at address 0000h.

I/O	BYTE 1
Input	01h
Output	

Command 02h—Master Erase

This command erases (programs to FFFFh) all words in the program flash memory and writes all words in the data SRAM to zero. After this command completes, the password lock bit is automatically cleared, allowing access to all bootloader commands.

I/O	BYTE 1
Input	02h
Output	

Command 03h—Password Match

This command accepts a 32-byte password value, which is matched against the password in program memory (in byte mode) from addresses 0020h to 003Fh. If the value matches, the password lock is cleared.

I/O	BYTE 1	BYTES 2 TO 33	BYTE 34	BYTE 35
Input	03h	32-Byte Password Value	00h	00h
Output				03Eh

Command 04h—Get Status

The status code returned by this command is defined in Table 17-1. The flags byte contains the bit status flags as shown in Table 17-2.

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
Input	04h	00h	00h	00h	00h
Output			Flags	Status Code	03Eh

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 17-2. Bootloader Status Flags

FLAG BIT	FUNCTION
0	Password Lock 0 = The password is unlocked or had a default value; password-protected commands can be used. 1 = The password is locked. Password-protected commands cannot be used.
1	Word/Byte Mode 0 = The bootloader is currently in byte mode for memory reads/writes. 1 = The bootloader is currently in word mode for memory reads/writes. (Note: The MAXQ8913 supports byte mode only.)
2	Word/Byte Mode Supported 0 = The bootloader supports byte mode only. 1 = The bootloader supports word mode as well as byte mode. (Note: The MAXQ8913 supports byte mode only.)
3 to 8	Reserved

Command 05h—Get Supported Commands

The SupportL (LSB) and SupportH (MSB) bytes form a 16-bit value that indicates which command families this bootloader supports. If bit 0 is set to 1, it indicates that Family 0 is supported. If bit 1 is set to 1, it indicates that Family 1 is supported, and so on. For the MAXQ8913, the value returned is 403Fh, indicating that command families 0, 1, 2, 3, 4, 5, and E are supported.

The CodeLen and DataLen bytes return the fixed block lengths used by the Load/Dump/Verify Fixed Length commands for code and data space, respectively. Because fixed block load is not supported on the MAXQ8913, both these values are returned as 00h.

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7
Input	05h	00h	00h	00h	00h	00h	00h
Output			SupportL	SupportH	CodeLen	DataLen	03Eh

Command 06h—Get Code Size

This command returns SizeH:SizeL, which represents the size of available code memory in words minus 1. If this command is unsupported, the return value is 0000h, meaning “unknown amount of memory.”

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
Input	06h	00h	00h	00h	00h
Output			SizeL	SizeH	03Eh

Command 07h—Get Data Size

This command returns SizeH:SizeL, which represents the size of available data memory in words minus 1. If this command is unsupported, the return value is 0000h, meaning “unknown amount of memory.”

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
Input	07h	00h	00h	00h	00h
Output			SizeL	SizeH	03Eh

Command 08h—Get Loader Version

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
Input	08h	00h	00h	00h	00h
Output			VersionL	VersionH	03Eh

Command 09h—Get Utility ROM Version

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5
Input	09h	00h	00h	00h	00h
Output			VersionL	VersionH	03Eh

Command 0Ah—Set Word/Byte-Mode Access

The mode byte should be 0 to set byte-access mode or 1 to set word-access mode. The current access mode is returned in the status flag byte by command 04h, as well as a flag to indicate whether word-access mode is supported by this particular bootloader. **Note: The MAXQ8913 supports byte mode only.**

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4
Input	0Ah	Mode	00h	00h
Output				03Eh

Command 0Dh—Get ID Information

For the MAXQ8913, the information returned by this command is a zero-terminated ROM banner string.

I/O	BYTE 1	(VARIABLE)	LAST BYTE
Input	0Dh	00h, 00h, 00h . . .	00h
Output		Device-Dependent Information	03Eh

17.3 Family 1 Commands: Load Variable Length (Password Protected)

Command 10h—Load Code Variable Length

This command programs (Length) bytes of data into the program flash starting at byte address (AddressH:AddressL), with the following restrictions.

- The low bit of the address is always forced to zero, since instructions in program flash must be word aligned.
- In byte mode, if an odd number of bytes is input, the final word written to the program flash has its most significant byte set to 00h by default.
- Memory locations in flash that have previously been loaded must be erased (using the Master Erase command, the Erase Code Fixed Length command, or the **flashErasePage** or **flashEraseAll** utility ROM routines) before they can be loaded with a different value.
- In keeping with standard MAXQ little-endian memory architecture, the least significant byte of each word is loaded first. For example, if one loads bytes (11h, 22h, 33h, 44h) starting at address 0000h, the first two words of program space are written to (2211h, 4433h).

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	(LENGTH) BYTES	BYTE LENGTH+5	BYTE LENGTH+6
Input	10h	Length	AddressL	AddressH	Data to load	00h	00h
Output							03Eh

MAXQ Family User's Guide: MAXQ8913 Supplement

Command 11h—Load Data Variable Length

This command writes (Length) bytes of data into the data SRAM starting at byte address (AddressH:AddressL).

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	(LENGTH) BYTES	BYTE LENGTH+5	BYTE LENGTH+6
Input	11h	Length	AddressL	AddressH	Data to load	00h	00h
Output							03Eh

17.4 Family 2 Commands: Dump Variable Length (Password Protected)

Command 20h—Dump Code Variable Length

This command has a slightly different format depending on the length of the dump requested. It returns the contents of the program flash memory: (Length) or (LengthH:LengthL) bytes starting at byte address (AddrH:AddrL).

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	(LENGTH) BYTES	BYTE LENGTH+7
Input (to dump < 256 bytes)	20h	1	AddrL	AddrH	Length	00h	00h...	00h
Output							Memory contents	03Eh

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	(LENGTH) BYTES	BYTE LENGTH+8
Input (to dump 256+ bytes)	20h	2	AddrL	AddrH	LengthL	LengthH	00h	00h...	00h
Output								Memory contents	03Eh

Command 21h—Dump Data Variable Length

This command has a slightly different format depending on the length of the dump requested. It returns the contents of the data SRAM: (Length) or (LengthH:LengthL) bytes starting at byte address (AddrH:AddrL).

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	(LENGTH) BYTES	BYTE LENGTH+7
Input (to dump < 256 bytes)	21h	1	AddrL	AddrH	Length	00h	00h...	00h
Output							Memory contents	03Eh

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	(LENGTH) BYTES	BYTE LENGTH+8
Input (to dump 256+ bytes)	21h	2	AddrL	AddrH	LengthL	LengthH	00h	00h...	00h
Output								Memory contents	03Eh

17.5 Family 3 Commands: CRC Variable Length (Password Protected)

Command 30h—CRC Code Variable Length

This command has a slightly different format depending on the length of the CRC requested. It returns the CRC-16 value (CrcH:CrcL) of the program flash: (Length) or (LengthH:LengthL) bytes/words starting at (AddrH:AddrL).

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8	BYTE 9
Input (to CRC < 256 bytes)	30h	1	AddrL	AddrH	Length	00h	00h	00h	00h
Output							CrcH	CrcL	03Eh

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8	BYTE 9	BYTE 10
Input (to CRC 256+ bytes)	30h	2	AddrL	AddrH	LengthL	LengthH	00h	00h	00h	00h
Output								CrcH	CrcL	03Eh

Command 31h—CRC Data Variable Length

This command has a slightly different format depending on the length of the CRC requested. It returns the CRC-16 value (CrcH:CrcL) of the data SRAM: (Length) or (LengthH:LengthL) bytes/words starting at (AddrH:AddrL).

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8	BYTE 9
Input (to CRC < 256 bytes)	31h	1	AddrL	AddrH	Length	00h	00h	00h	00h
Output							CrcH	CrcL	03Eh

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	BYTE 5	BYTE 6	BYTE 7	BYTE 8	BYTE 9	BYTE 10
Input (to CRC 256+ bytes)	31h	2	AddrL	AddrH	LengthL	LengthH	00h	00h	00h	00h
Output								CrcH	CrcL	03Eh

17.6 Family 4 Commands: Verify Variable Length (Password Protected)

Command 40h—Verify Code Variable Length

This command operates in the same manner as the “Load Code Variable Length” command, except that instead of programming the input data into flash memory, it verifies that the input data matches the data already in code space. If the data does not match, the status code is set to reflect this failure.

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	(LENGTH) BYTES	BYTE LENGTH+5	BYTE LENGTH+6
Input	40h	Length	AddressL	AddressH	Data to verify	00h	00h
Output							03Eh

Command 41h—Verify Data Variable Length

This command operates in the same manner as the “Load Data Variable Length” command, except that instead of writing the input data into data SRAM, it verifies that the input data matches the data already in data space. If the data does not match, the status code is set to reflect this failure.

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	(LENGTH) BYTES	BYTE LENGTH+5	BYTE LENGTH+6
Input	41h	Length	AddressL	AddressH	Data to verify	00h	00h
Output							03Eh

MAXQ Family User's Guide: MAXQ8913 Supplement

17.7 Family 5 Commands: Load and Verify Variable Length (Password Protected)

Command 50h—Load and Verify Code Variable Length

This command combines the functionality of the “Load Code Variable Length” and “Verify Code Variable Length” commands.

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	(LENGTH) BYTES	BYTE LENGTH+5	BYTE LENGTH+6
Input	50h	Length	AddressL	AddressH	Data to load and verify	00h	00h
Output							03Eh

Command 51h—Load and Verify Data Variable Length

This command combines the functionality of the “Load Data Variable Length” and “Verify Data Variable Length” commands.

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4	(LENGTH) BYTES	BYTE LENGTH+5	BYTE LENGTH+6
Input	51h	Length	AddressL	AddressH	Data to load and verify	00h	00h
Output							03Eh

17.8 Family E Commands: Erase Fixed Length (Password Protected)

Command E0h—Erase Code Fixed Length

This command erases (programs to FFFFh) all words in a 512-word page of the program flash memory. The address given should be located in the 512-word page to be erased. For example, providing address 0000h (in byte mode) to this command erases the first 512-word page, address 0400h erases the second page, and so on. There are 64 flash pages total, from 0000h to 7C00h.

I/O	BYTE 1	BYTE 2	BYTE 3	BYTE 4
Input	E0h	0	AddressL	AddressH
Output				

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDENDUM TO SECTION 18: MAXQ FAMILY INSTRUCTION SET SUMMARY

Refer to the *MAXQ Family User's Guide*. Table 18-1 from the *MAXQ Family User's Guide* is reproduced here.

Table 18-1. Instruction Set Summary

	MNEMONIC	DESCRIPTION	16-BIT INSTRUCTION WORD	STATUS BITS AFFECTED	AP INC/DEC	NOTES
LOGICAL OPERATIONS	AND src	Acc ← Acc AND src	f001 1010 ssss ssss	S, Z	Y	1
	OR src	Acc ← Acc OR src	f010 1010 ssss ssss	S, Z	Y	1
	XOR src	Acc ← Acc XOR src	f011 1010 ssss ssss	S, Z	Y	1
	CPL	Acc ← ~Acc	1000 1010 0001 1010	S, Z	Y	
	NEG	Acc ← ~Acc + 1	1000 1010 1001 1010	S, Z	Y	
	SLA	Shift Acc left arithmetically	1000 1010 0010 1010	C, S, Z	Y	
	SLA2	Shift Acc left arithmetically twice	1000 1010 0011 1010	C, S, Z	Y	
	SLA4	Shift Acc left arithmetically four times	1000 1010 0110 1010	C, S, Z	Y	
	RL	Rotate Acc left (w/o C)	1000 1010 0100 1010	S	Y	
	RLC	Rotate Acc left (through C)	1000 1010 0101 1010	C, S, Z	Y	
	SRA	Shift Acc right arithmetically	1000 1010 1111 1010	C, Z	Y	
	SRA2	Shift Acc right arithmetically twice	1000 1010 1110 1010	C, Z	Y	
	SRA4	Shift Acc right arithmetically four times	1000 1010 1011 1010	C, Z	Y	
	SR	Shift Acc right (0 → msbit)	1000 1010 1010 1010	C, S, Z	Y	
	RR	Rotate Acc right (w/o C)	1000 1010 1100 1010	S	Y	
RRC	Rotate Acc right (through C)	1000 1010 1101 1010	C, S, Z	Y		
BIT OPERATIONS	MOVE C, Acc.	C ← Acc.	1110 1010 bbbb 1010	C		
	MOVE C, #0	C ← 0	1101 1010 0000 1010	C		
	MOVE C, #1	C ← 1	1101 1010 0001 1010	C		
	CPL C	C ← ~C	1101 1010 0010 1010	C		
	MOVE Acc., C	Acc. ← C	1111 1010 bbbb 1010	S, Z		
	AND Acc.	C ← C AND Acc.	1001 1010 bbbb 1010	C		
	OR Acc.	C ← C OR Acc.	1010 1010 bbbb 1010	C		
	XOR Acc.	C ← C XOR Acc.	1011 1010 bbbb 1010	C		
	MOVE dst., #1	dst. ← 1	1ddd dddd 1bbb 0111	C, S, E, Z		2
	MOVE dst., #0	dst. ← 0	1ddd dddd 0bbb 0111	C, S, E, Z		2
MOVE C, src.	C ← src.	fbbb 0111 ssss ssss	C			
MATH	ADD src	Acc ← Acc + src	f100 1010 ssss ssss	C, S, Z, OV	Y	1
	ADDC src	Acc ← Acc + (src + C)	f110 1010 ssss ssss	C, S, Z, OV	Y	1
	SUB src	Acc ← Acc - src	f101 1010 ssss ssss	C, S, Z, OV	Y	1
	SUBB src	Acc ← Acc - (src + C)	f111 1010 ssss ssss	C, S, Z, OV	Y	1

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 18-1. Instruction Set Summary (continued)

	MNEMONIC	DESCRIPTION	16-BIT INSTRUCTION WORD	STATUS BITS AFFECTED	AP INC/DEC	NOTES
BRANCHING	{L/S}JUMP src	IP ← IP + src or src	f000 1100 ssss ssss			6
	{L/S}JUMP C, src	If C=1, IP ← (IP + src) or src	f010 1100 ssss ssss			6
	{L/S}JUMP NC, src	If C=0, IP ← (IP + src) or src	f110 1100 ssss ssss			6
	{L/S}JUMP Z, src	If Z=1, IP ← (IP + src) or src	f001 1100 ssss ssss			6
	{L/S}JUMP NZ, src	If Z=0, IP ← (IP + src) or src	f101 1100 ssss ssss			6
	{L/S}JUMP E, src	If E=1, IP ← (IP + src) or src	0011 1100 ssss ssss			6
	{L/S}JUMP NE, src	If E=0, IP ← (IP + src) or src	0111 1100 ssss ssss			6
	{L/S}JUMP S, src	If S=1, IP ← (IP + src) or src	f100 1100 ssss ssss			6
	{L/S}DJNZ LC[n], src	If --LC[n] <> 0, IP ← (IP + src) or src	f10n 1101 ssss ssss			6
	{L/S}CALL src	@++SP ← IP+1; IP ← (IP+src) or src	f011 1101 ssss ssss			6, 7
	RET	IP ← @SP--	1000 1100 0000 1101			
	RET C	If C=1, IP ← @SP--	1010 1100 0000 1101			
	RET NC	If C=0, IP ← @SP--	1110 1100 0000 1101			
	RET Z	If Z=1, IP ← @SP--	1001 1100 0000 1101			
	RET NZ	If Z=0, IP ← @SP--	1101 1100 0000 1101			
	RET S	If S=1, IP ← @SP--	1100 1100 0000 1101			
	RETI	IP ← @SP-- ; INS ← 0	1000 1100 1000 1101			
	RETI C	If C=1, IP ← @SP-- ; INS ← 0	1010 1100 1000 1101			
	RETI NC	If C=0, IP ← @SP-- ; INS ← 0	1110 1100 1000 1101			
	RETI Z	If Z=1, IP ← @SP-- ; INS ← 0	1001 1100 1000 1101			
RETI NZ	If Z=0, IP ← @SP-- ; INS ← 0	1101 1100 1000 1101				
RETI S	If S=1, IP ← @SP-- ; INS ← 0	1100 1100 1000 1101				
DATA TRANSFER	XCH (MAXQ20 only)	Swap Acc bytes	1000 1010 1000 1010	S	Y	
	XCHN	Swap nibbles in each Acc byte	1000 1010 0111 1010	S	Y	
	MOVE dst, src	dst ← src	fddd dddd ssss ssss	C, S, Z, E	(Note 8)	7, 8
	PUSH src	@++SP ← src	f000 1101 ssss ssss			7
	POP dst	dst ← @SP--	1ddd dddd 0000 1101	C, S, Z, E		7
	POPI dst	dst ← @SP-- ; INS ← 0	1ddd dddd 1000 1101	C, S, Z, E		7
	CMP src	E ← (Acc = src)	f111 1000 ssss ssss	E		
	NOP	No operation	1101 1010 0011 1010			

Note 1: The active accumulator (Acc) is not allowed as the src in operations where it is the implicit destination.

Note 2: Only module 8 and modules 0 to 5 (when implemented for a given product) are supported by these single-cycle bit operations. Potentially affects C or E if PSF register is the destination. Potentially affects S and/or Z if AP or APC is the destination.

Note 3: The terms Acc and A[AP] can be used interchangeably to denote the active accumulator.

Note 4: Any index represented by or found inside [] brackets is considered variable, but required.

Note 5: The active accumulator (Acc) is not allowed as the dst if A[AP] is specified as the src.

Note 6: The '{L/S}' prefix is optional.

Note 7: Instructions that attempt to simultaneously push/pop the stack (e.g. PUSH @SP--, PUSH @SPI--, POP @++SP, POPI @++SP) or modify SP in a conflicting manner (e.g., MOVE SP, @SP--) are invalid.

Note 8: Special cases: If 'MOVE APC, Acc' sets the APC.CLR bit, AP is cleared, overriding any autoinc/dec/modulo operation specified for AP. If 'MOVE AP, Acc' causes an autoinc/dec/modulo operation on AP, this overrides the specified data transfer (i.e., Acc is not transferred to AP).

MAXQ Family User's Guide: MAXQ8913 Supplement

SECTION 19: ANALOG-TO-DIGITAL CONVERTER (SPECIFIC TO MAXQ8913)

The MAXQ8913 provides a 12-bit, successive approximation analog-to-digital converter (ADC) with an integrated analog multiplexer. The ADC can perform either single-ended conversions from one of six external input channels (and an additional channel reserved for an internal temperature sensor) or differential conversions from one of three external channel pairs. The voltage reference used for each conversion can be selected from an internal precision bandgap reference, an external reference, or the AVDD analog power supply.

19.1 Analog-to-Digital Converter Features

- 12-bit single-ended conversion with up to seven analog channel inputs (six external inputs and one temp sensor)
- 12-bit differential conversion with up to three analog channel pair inputs (each differential pair takes the place of two single-ended channel inputs)
- Autoscan feature performs up to eight conversions in sequence automatically without CPU intervention
- Conversion sequence can be performed once (single conversion mode) or repeatedly (continuous conversion mode)
- Up to 16 sample words can be stored in a dedicated data buffer until the processor is ready to retrieve them
- Selectable clock divider runs the ADC from a divide by 1, divide by 2, divide by 4, or divide by 8 of the system clock
- Sample acquisition time can optionally be extended on a per-conversion basis
- Data results can be left-aligned or right-aligned on a per-conversion basis
- Converter reference is switchable among AVDD, internal reference, and external reference
- Optional power-management mode shuts the ADC off between conversions to save power
- Configurable data available interrupt signals the CPU following each conversion, each sequence, or after every 12 or 16 samples
- Four uncommitted op amps whose outputs can optionally be used as inputs for channels AIN2, AIN3, AIN4, and AIN5

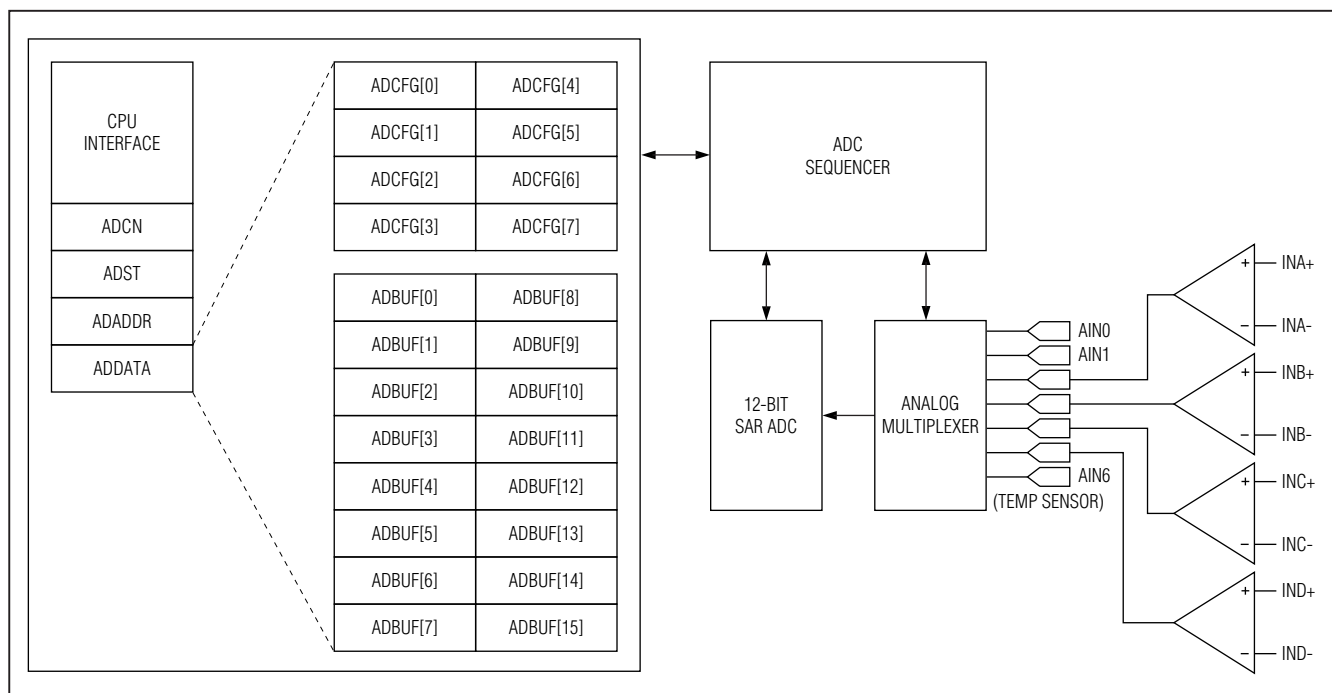


Figure 19-1. ADC Block Diagram

19.2 Analog-to-Digital Pins and Control Registers

Tables 19-1 and 19-2 list the pins and control registers dedicated to the ADC. Note that all ADC pins are dedicated, so none of them is multiplexed with GPIO port pins. Addresses for all registers are given as “Mx[yy],” where x is the module number (from 0 to 15 decimal) and yy is the register index (from 00h to 1Fh hexadecimal). Fields in the bit definition tables are defined as follows:

- **Name:** Symbolic names of bits or bit fields in this register.
- **Reset:** The value of each bit in this register following a standard reset. If this field reads “unchanged,” the given bit is unaffected by standard reset. If this field reads “s,” the given bit does not have a fixed 0 or 1 reset value because its value is determined by another internal state or external condition.
- **POR:** If present this field defines the value of each bit in this register following a power-on reset (as opposed to a standard reset). Some bits are unaffected by standard resets and are set/cleared by POR only.
- **Access:** Bits can be read-only (r) or read/write (rw). Any special restrictions or conditions that could apply when reading or writing this bit are detailed in the bit description.

Table 19-1. ADC Input and Power-Supply Pins

PIN	NAME	ADC INTERFACE FUNCTION
E4	AVDD	Analog Supply Voltage
B5	AGND	Analog Ground
G2	REFA	External ADC Voltage Reference
H1	AIN0	Single-Ended Analog Input Channel 0
H3	AIN1	Single-Ended Analog Input Channel 1
B9	AIN2 (OUTA)	Single-Ended Analog Input Channel 2
B3	AIN3 (OUTB)	Single-Ended Analog Input Channel 3
B1	AIN4 (OUTC)	Single-Ended Analog Input Channel 4
D1	AIN5 (OUTD)	Single-Ended Analog Input Channel 5
H1, H3	(AIN0, AIN1)	Differential Input Channel 0
B9, B3	(AIN2, AIN3)	Differential Input Channel 1
B1, D1	(AIN4, AIN5)	Differential Input Channel 2

Table 19-2. ADC Control Registers

REGISTER	ADDRESS	FUNCTION
ADST	M3[00h]	ADC Status Register. Contains the ADCFG and ADBUF register index selection bits, conversion start bit, and other status bits for the ADC.
ADADDR	M3[01h]	ADC Address Register. Defines the first and last ADCFG registers used in a conversion sequence as well as the first ADBUF register written in a conversion sequence.
ADCN	M3[08h]	ADC Control Register. Controls sample acquisition extend, power-management mode, single/continuous sequence conversion, interrupt modes, and clock division for the ADC.
ADDATA	M3[09h]	ADC Data Register. Acts as a read/write access point to registers ADCFG[0] to ADCFG[7] and ADBUF[0] to ADBUF[15].
ADCFG[0]	ADDATA[10h]	ADC Sequence Configuration Register 0
ADCFG[1]	ADDATA[11h]	ADC Sequence Configuration Register 1
ADCFG[2]	ADDATA[12h]	ADC Sequence Configuration Register 2
ADCFG[3]	ADDATA[13h]	ADC Sequence Configuration Register 3
ADCFG[4]	ADDATA[14h]	ADC Sequence Configuration Register 4
ADCFG[5]	ADDATA[15h]	ADC Sequence Configuration Register 5

MAXQ Family User's Guide: MAXQ8913 Supplement

Table 19-2. ADC Control Registers (continued)

REGISTER	ADDRESS	FUNCTION
ADCFG[6]	ADDATA[16h]	ADC Sequence Configuration Register 6
ADCFG[7]	ADDATA[17h]	ADC Sequence Configuration Register 7
ADBUF[0]	ADDATA[00h]	ADC Sample Buffer Register 0. Read-only register containing ADC conversion result.
ADBUF[1]	ADDATA[01h]	ADC Sample Buffer Register 1. Read-only register containing ADC conversion result.
ADBUF[2]	ADDATA[02h]	ADC Sample Buffer Register 2. Read-only register containing ADC conversion result.
ADBUF[3]	ADDATA[03h]	ADC Sample Buffer Register 3. Read-only register containing ADC conversion result.
ADBUF[4]	ADDATA[04h]	ADC Sample Buffer Register 4. Read-only register containing ADC conversion result.
ADBUF[5]	ADDATA[05h]	ADC Sample Buffer Register 5. Read-only register containing ADC conversion result.
ADBUF[6]	ADDATA[06h]	ADC Sample Buffer Register 6. Read-only register containing ADC conversion result.
ADBUF[7]	ADDATA[07h]	ADC Sample Buffer Register 7. Read-only register containing ADC conversion result.
ADBUF[8]	ADDATA[08h]	ADC Sample Buffer Register 8. Read-only register containing ADC conversion result.
ADBUF[9]	ADDATA[09h]	ADC Sample Buffer Register 9. Read-only register containing ADC conversion result.
ADBUF[10]	ADDATA[0Ah]	ADC Sample Buffer Register 10. Read-only register containing ADC conversion result.
ADBUF[11]	ADDATA[0Bh]	ADC Sample Buffer Register 11. Read-only register containing ADC conversion result.
ADBUF[12]	ADDATA[0Ch]	ADC Sample Buffer Register 12. Read-only register containing ADC conversion result.
ADBUF[13]	ADDATA[0Dh]	ADC Sample Buffer Register 13. Read-only register containing ADC conversion result.
ADBUF[14]	ADDATA[0Eh]	ADC Sample Buffer Register 14. Read-only register containing ADC conversion result.
ADBUF[15]	ADDATA[0Fh]	ADC Sample Buffer Register 15. Read-only register containing ADC conversion result.
TEMPEN	M3[0Ch]	Temperature Sensor Enable Register. Enables/disables the on-board temperature sensor whose value can be read through ADC channel 6.

Note: Address *ADDATA[xxh]* refers to reading/writing the *ADDATA* register with *ADCFG:ADIDX[3:0]* (*ADST[4:0]*) set to *xxh*. For example, *ADBUF[7]* is read by setting *ADST[4:0]* to 00111b and reading the *ADDATA* register.

The following peripheral registers are used to control the analog-to-digital converter functions.

19.2.1 Analog-to-Digital Converter Status Register (ADST, M4[06h])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	ADDAT[3:0]			
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	r	r	r	r

Bit #	7	6	5	4	3	2	1	0
Name	REFOK	ADCONV	ADDAI	ADCFG	ADIDX[3:0]			
Reset	0	0	0	0	0	0	0	0
Access	r	rw*	rw	rw	rw	rw	rw	rw

**ADCONV* cannot be written when *PMME = 1* and *SWB = 0*.

Bits 15:12: Reserved

Bits 11:8: ADC Data Available Address Bits (ADDAT[3:0]). These read-only status bits indicate the *ADBUF* data buffer location that was last written by the ADC hardware. During a conversion sequence, these bits are updated by hardware as each conversion is completed and written to the data buffer.

MAXQ Family User's Guide: MAXQ8913 Supplement

Bit 7: Internal Reference OK (REFOK). This read-only status bit indicates whether the internal reference is ready for use by the ADC.

0 = The internal reference is either disabled (IREFEN = 0) or is still warming up.

1 = The internal reference is ready for use.

Bit 6: ADC Start Conversion (ADCONV). Writing this bit to 1 starts the ADC conversion sequence. In single-conversion mode, this bit is cleared automatically by hardware at the end of the conversion sequence. In continuous-conversion mode, this bit remains set (and conversion continues) until it is reset to 0 by software. Setting this bit to 0 causes the current conversion sequence to stop. If the ADC is in the middle of a conversion, it stops after that conversion has completed. If the ADC is in the middle of an extended acquisition time period, it stops immediately.

Entering stop or PMM mode causes the current conversion to stop and the ADCONV bit to clear to 0. This bit cannot be written to 1 (to start a conversion) in PMM mode unless switchback is enabled (SWB = 1).

Bit 5: ADC Data Available Interrupt Flag (ADDAI). This bit is set to 1 by hardware when the conditions defined by ADINT[1:0] (ADCN[11:10]) are met. Setting this bit triggers an interrupt if ADDAIE = 1 and the interrupt is not otherwise masked. This bit is cleared by hardware automatically when an ADC conversion is started (ADCONV is written to 1); it can also be cleared to 0 by software.

Bit 4: ADC Conversion Configuration Register Select (ADCFG); Bits 3:0: ADC Configuration/Data Buffer Register Index (ADIDX[3:0]). These register bits select the ADC configuration or ADC data buffer register that is accessed when ADDATA is read or written. Note that the ADC data buffer registers are read-only.

Reading from or writing to ADDATA causes the value ADIDX[3:0] to autoincrement, but does not affect the value of ADCFG, even if the ADIDX value rolls over from 1111b to 0000b. For example, setting ADCFG to 1 and ADIDX[3:0] to 1101b selects ADCFG[5] for read/write access. Reading ADDATA successively then returns the values ADCFG[5], ADCFG[6], ADCFG[7], ADCFG[0], ADCFG[1], and so on.

ADCFG	ADIDX3	ADIDX2	ADIDX1	ADIDX0	READING ADDATA	WRITING ADDATA
0	0	0	0	0	Reads ADBUF[0]	No effect
0	0	0	0	1	Reads ADBUF[1]	No effect
0	0	0	1	0	Reads ADBUF[2]	No effect
0	0	0	1	1	Reads ADBUF[3]	No effect
0	0	1	0	0	Reads ADBUF[4]	No effect
0	0	1	0	1	Reads ADBUF[5]	No effect
0	0	1	1	0	Reads ADBUF[6]	No effect
0	0	1	1	1	Reads ADBUF[7]	No effect
0	1	0	0	0	Reads ADBUF[8]	No effect
0	1	0	0	1	Reads ADBUF[9]	No effect
0	1	0	1	0	Reads ADBUF[10]	No effect
0	1	0	1	1	Reads ADBUF[11]	No effect
0	1	1	0	0	Reads ADBUF[12]	No effect
0	1	1	0	1	Reads ADBUF[13]	No effect
0	1	1	1	0	Reads ADBUF[14]	No effect
0	1	1	1	1	Reads ADBUF[15]	No effect
1	X	0	0	0	Reads ADCFG[0]	Writes to ADCFG[0]
1	X	0	0	1	Reads ADCFG[1]	Writes to ADCFG[1]
1	X	0	1	0	Reads ADCFG[2]	Writes to ADCFG[2]
1	X	0	1	1	Reads ADCFG[3]	Writes to ADCFG[3]
1	X	1	0	0	Reads ADCFG[4]	Writes to ADCFG[4]
1	X	1	0	1	Reads ADCFG[5]	Writes to ADCFG[5]
1	X	1	1	0	Reads ADCFG[6]	Writes to ADCFG[6]
1	X	1	1	1	Reads ADCFG[7]	Writes to ADCFG[7]

MAXQ Family User's Guide: MAXQ8913 Supplement

19.2.2 ADC Conversion Sequence Address Register (ADADDR, M3[01h])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	SEQSTORE[3:0]			
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	rw*	rw*	rw*	rw*

Bit #	7	6	5	4	3	2	1	0
Name	—	SEQSTART[2:0]			—	SEQEND[2:0]		
Reset	0	0	0	0	0	0	0	0
Access	r	rw*	rw*	rw*	r	rw*	rw*	rw*

*Can only be written when $ADCONV = 0$.

Bits 15:12, 7, 3: Reserved

Bits 11:8: ADC Sequence Sample Storage Address (SEQSTORE[3:0]). These bits contain the index of the first ADBUF register (inclusive) that is used to store samples from the ADC conversion sequence.

Bits 6:4: ADC Sequence Start Address (SEQSTART[2:0]). These bits contain the index of the first ADCFG register (inclusive) that is used to define the ADC conversion sequence.

Bits 2:0: ADC Sequence End Address (SEQEND[2:0]). These bits contain the index of the last ADCFG register (inclusive) that is used to define the ADC conversion sequence.

19.2.3 ADC Control Register (ADCN, M3[08h])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	ADINT1	ADINT0	ADCLK1	ADCLK0
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	rw*	rw*	rw*	rw*

Bit #	7	6	5	4	3	2	1	0
Name	IREFEN	ADCONT	ADDAIE	ADPMO	ADACQ[3:0]			
Reset	0	0	0	0	0	0	0	0
Access	rw*	rw*	rw*	rw*	rw*	rw*	rw*	rw*

*Can only be written when $ADCONV = 0$.

Bits 15:12: Reserved

Bits 11:10: ADC Data Available Interrupt Interval (ADINT[1:0]). These bits select the condition for generating an ADC data available interrupt (setting ADDAI to 1).

ADINT1	ADINT0	SET ADDAI TO 1 . . .
0	0	After each ADC conversion (every sample)
0	1	After the conversion sequence has completed (works for single-conversion mode only)
1	0	Every 12 ADC samples
1	1	Every 16 ADC samples

MAXQ Family User's Guide: MAXQ8913 Supplement

Bits 9:8: ADC Clock Divider (ADCLK[1:0]). These bits control the generation of the ADC clock from the system clock as follows:

ADCLK1	ADCLK0	ADC CLOCK
0	0	System Clock/1 (default setting)
0	1	System Clock/2
1	0	System Clock/4
1	1	System Clock/8

However, since there is an upper limit on the sample rate of the ADC (approximately 300ksps; refer to the IC data sheet), not all clock division settings could be valid, depending on the system clock frequency. A single ADC conversion requires 16 ADC clocks, which allows us to calculate possible ADC sample rates as shown in Table 19-3.

Bit 7: ADC Internal Reference Enable (IREFEN). This bit controls the ADC internal reference.

0 = The internal reference is disabled. The ADREF bit in each configuration register (ADCFG) selects between AVDD and the external reference.

1 = The internal reference is enabled. Once REFOK = 1, the ADREF bit in each configuration register (ADCFG) selects between AVDD and the internal reference.

Bit 6: ADC Continuous Sequence Mode (ADCONT). This bit selects single- or continuous-sequence mode.

0 = Single-conversion sequence mode. In this mode, setting ADCONV = 1 starts a single-conversion sequence, with starting and ending configuration registers as defined in the ADADDR register. Once the conversion sequence completes, ADCONV automatically clears to 0, and the ADC powers down (if PMO = 0).

1 = Continuous-conversion sequence mode. In this mode, setting ADCONV = 1 also starts a conversion sequence, but once the sequence has completed, it simply repeats again. To stop the conversions, ADCONV must explicitly be cleared to 0 by software.

Bit 5: ADC Data Available Interrupt Enable (ADDAIE). This bit controls the ADC data available interrupt.

0 = The ADC interrupt is disabled.

1 = An interrupt is triggered (if not otherwise masked) when ADDAI = 1.

Bit 4: ADC Power-Management Override (PMO). This bit controls power management for the ADC.

0 = The ADC automatically powers up at the beginning of a conversion sequence and powers down when the sequence has finished (or when ADCONV is set to 0). This adds a delay of approximately 20 ADC clocks to the conversion sequence time.

1 = ADC power management is disabled. After setting PMO to 1, the software should wait long enough for the ADC to power up before initiating a conversion (refer to the IC data sheet for timing). Once the ADC has powered up, it remains powered on as long as PMO is set to 1, unless stop mode is entered.

Table 19-3. ADC Sample Rates Using a 10MHz Crystal

ADCLK[1:0]	SAMPLE RATE AT 10MHz (CLOCK/1) (ksps)	SAMPLE RATE AT 5MHz (CLOCK/2) (ksps)	SAMPLE RATE AT 2.5MHz (CLOCK/4) (ksps)	SAMPLE RATE AT 1.25MHz (CLOCK/8) (ksps)	SAMPLE RATE AT 8.4MHz (FLL) (ksps)
00	625 (invalid)	312.5	156.25	78.13	525 (invalid)
01	312.5	156.25	78.13	39	262.5 (invalid)
10	156.25	78.13	39	19.5	131
11	78.13	39	19.5	9.76	65.6

MAXQ Family User's Guide: MAXQ8913 Supplement

Bits 3:0: ADC Sample Acquisition Time Extend (ADACQ[3:0]). These bits set the extended sample acquisition time period. For a given conversion, sample acquisition time is extended if the ADACQEN bit is set in the conversion configuration (ADCFG) register. If this bit is set, the acquisition time is extended by:

$$16 \times (\text{ADACQ}[3:0] + 1) \times \text{ADC clock period}$$

19.2.4 ADC Data Register (ADDATA, M3[09h])

Bit #	15	14	13	12	11	10	9	8
Name	ADDATA							
Reset	s	s	s	s	s	s	s	s
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	ADDATA							
Reset	s	s	s	s	s	s	s	s
Access	rw	rw	rw	rw	rw	rw	rw	rw

Note: The effect of read or write operation depends on ADIDX and ADCFG bit settings.

This register is an access point for the eight ADC configuration registers (ADCFG[0] to ADCFG[7]) and 16 ADC data buffer registers (ADBUF[0] to ADBUF[15]). Reading or writing ADDATA actually reads or writes the selected register, as determined by ADST[4:0].

19.2.5 ADC Data Buffer Registers (ADBUF[0] to ADBUF[15], ADDATA[00h] to ADDATA[0Fh])

Bit #	15	14	13	12	11	10	9	8
Name	ADBUF							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	ADBUF							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

The 16 ADC data buffer registers ADBUF[0] to ADBUF[15] serve as temporary holding storage for ADC conversion samples until the samples can be read by the processor. They can be read or written at any time, whether or not an ADC conversion is in progress.

As each ADC conversion completes, the resulting sample is written to one of the ADBUF registers, starting with the index given by SEQSTORE (ADADDR[11:8]) and incrementing from there with each new sample written. The index of the most recent ADBUF register written to by the ADC controller is always available in the ADDAT bit field (ADST[11:8]).

The ADC samples are written into the ADBUF registers in either left-aligned or right-aligned format as selected by the ADALGN bit for that conversion configuration register. Once a sample is written into an ADBUF register, the data remains there until it is erased by software or until it is overwritten by another ADC sample, 16 conversions later. When continuous-conversion mode is used, it is the responsibility of the user software to monitor the data available interrupt and read ADC samples from the ADBUF registers before they are overwritten by subsequent samples.

19.2.6 ADC Conversion Configuration Registers (ADCFG[0] to ADCFG[7], ADDATA[10h] to ADDATA[17h])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	—	—	—	—
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	r	r	r	r

Bit #	7	6	5	4	3	2	1	0
Name	—	ADREF	ADACQEN	ADALGN	ADDIFF	ADCH2	ADCH1	ADCH0
Reset	0	0	0	0	0	0	0	0
Access	r	rw*	rw*	rw*	rw*	rw*	rw*	rw*

*Can only be written when $ADCONV = 0$.

The eight conversion configuration registers ADCFG[0] to ADCFG[7] provide settings for each individual conversion in an ADC conversion sequence. As the ADC autoscans, it reads each configuration register in the sequence in turn and performs a conversion using the settings from that register. The starting and ending configuration registers (inclusive) in the sequence are given by the SEQSTART and SEQEND bit fields in the ADADDR register.

The number of configuration registers selected by ADADDR also determines the number of conversions performed in the sequence (one conversion per register selected). The ADCFG registers cannot be written to while a conversion sequence is in progress ($ADCONV = 1$).

Bits 15:7: Reserved

Bit 6: ADC Reference Select (ADREF). This bit determines (in conjunction with IREFEN) which reference is used for this ADC conversion.

0 = AVDD (default) is used as the reference for this conversion.

1 = If IREFEN = 1, the internal reference is used for this conversion; otherwise, the external reference is used.

Bit 5: ADC Sample Acquisition Extension Enable (ADACQEN). This bit determines whether the acquisition time for this conversion is extended by the number of ADC clock cycles given by ADACQ[3:0].

Bit 4: ADC Data Alignment Select (ADALGN). This bit determines how the ADC sample for this conversion is stored in the ADBUF register.

0 = The ADC data is stored right-adjusted in bits [11:0] of the ADBUF register. For single-ended conversions, bits [15:12] are filled with zeros, while for differential conversions bits [15:12] are sign extended from bit 11.

1 = The ADC data is stored left-adjusted in bits [15:4] of the ADBUF register with bits [3:0] zero padded.

Bit 3: ADC Differential Mode Select (ADDIFF); Bits 2:0: ADC Channel Select (ADCH[2:0]). These four bits control which channel or pair of channels is used for a given conversion, and whether the conversion is performed in single-ended or differential mode. In differential mode, since there are only three channel pairs, bit ADCH[2] is ignored.

ADDIFF	ADCH2	ADCH1	ADCH0	ADC CONVERSION TYPE
0	0	0	0	Single-ended conversion: AIN0 (to AGND)
0	0	0	1	Single-ended conversion: AIN1 (to AGND)
0	0	1	0	Single-ended conversion: AIN2 (to AGND)
0	0	1	1	Single-ended conversion: AIN3 (to AGND)
0	1	0	0	Single-ended conversion: AIN4 (to AGND)
0	1	0	1	Single-ended conversion: AIN5 (to AGND)
0	1	1	0	Single-ended conversion: AIN6 (temperature sensor)
0	1	1	1	Reserved

MAXQ Family User's Guide: MAXQ8913 Supplement

ADDIFF	ADCH2	ADCH1	ADCH0	ADC CONVERSION TYPE
1	X	0	0	Differential conversion: (AIN0, AIN1)
1	X	0	1	Differential conversion: (AIN2, AIN3)
1	X	1	0	Differential conversion: (AIN4, AIN5)
1	X	1	1	Reserved

19.2.7 Temperature Sensor Enable Register (TEMPEN, M3[0Ch])

Bit #	7	6	5	4	3	2	1	0
Name	—	—	—	—	—	—	—	TSEN
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	r	r	r	rw

Bits 7:1: Reserved

Bit 0: Temperature Sensor Enable (TSEN). This bit enables/disables the on-board temperature sensor whose value can be read through ADC channel 6 (single-ended mode only).

0 = Temperature sensor is disabled (default value).

1 = Temperature sensor is enabled.

19.3 Analog-to-Digital Converter Code Examples

19.3.1 ADC Example 1: Single Conversion

```

move ADCN,    #0300h    ; Set ADC clock to sysclk/8 (78ksps at 10MHz)

move ADST,    #0010h    ; Points ADDATA to config register 0
move ADDATA,  #06h      ; Single-ended conversion on channel AN6, AVDD ref
move ADST.6,  #1        ; Start conversion

waitConvert:
move C, ADST.6
jump C, waitConvert    ; Conversion has completed when ADST.6 clears to 0

move ADST,    #0000h    ; Points ADDATA to data register 0
move Acc,     ADDATA     ; Get conversion result

```

19.3.2 ADC Example 2: Continuous Conversion

```

move ADCN,    #0F00h    ; Set ADC clock to sysclk/8 (78ksps at 10MHz),
                        ; also set Data Available interrupt to trigger
                        ; following every 16 samples
move ADCN.6,  #1        ; Enable continuous conversion mode

move ADST,    #0010h    ; Points ADDATA to config register 0
move ADDATA,  #06h      ; ACFG[0]: Single-ended conversion on AN6, AVDD ref
move ADDATA,  #07h      ; ACFG[1]: Single-ended conversion on AN7, AVDD ref
move ADADDR,  #0001h    ; Sequence runs from ACFG[0] to ACFG[1] inclusive

move ADST.6,  #1        ; Start conversion (continuous)

```

MAXQ Family User's Guide: MAXQ8913 Supplement

```
waitConvert:
  move C, ADST.5
  jump NC, waitConvert ; Wait for 16 samples to be captured (ADDAI=1)

  move ADST.6, #0 ; Stop conversion
  move ADST.5, #0 ; Clear data available flag

  move ADST, #0000h ; Points ADDATA to data register 0
  move A[0], ADDATA ; Get conversion data
  move A[1], ADDATA
  move A[2], ADDATA
  move A[3], ADDATA
  move A[4], ADDATA
  move A[5], ADDATA
  move A[6], ADDATA
  move A[7], ADDATA
  move A[8], ADDATA
  move A[9], ADDATA
  move A[10], ADDATA
  move A[11], ADDATA
  move A[12], ADDATA
  move A[13], ADDATA
  move A[14], ADDATA
  move A[15], ADDATA
```


MAXQ Family User's Guide: MAXQ8913 Supplement

SECTION 20: DIGITAL-TO-ANALOG CONVERTERS (SPECIFIC TO MAXQ8913)

20.1 DAC Overview

The MAXQ8913 provides six independent digital-to-analog converter (DAC) channels. Four of these DAC channels produce voltage outputs, while the other two channels act as programmable current sinks.

The DAC module on the MAXQ8913 provides the following features:

- Two 10-bit (DAC1 and DAC2) voltage output channels that can be used to generate either single-ended (DAC1/DAC2) or differential (RIN/LIN) voltages.
- Two 8-bit (DAC3 and DAC4) voltage output channels that generate single-ended voltage outputs only.
- Two 8-bit (SINK1 and SINK2) programmable current sinks.

Table 20-1 lists the control registers dedicated to the DAC channels. Note that all DAC pins are dedicated, so none of them are multiplexed with GPIO port pins.

Table 20-1. DAC Control Registers

REGISTER	ADDRESS	FUNCTION
DAC1OUT	M3[02h]	DAC 1 Output Register. Sets the 10-bit voltage output value for DAC channel 1.
DAC2OUT	M3[03h]	DAC 2 Output Register. Sets the 10-bit voltage output value for DAC channel 2.
DAC3OUT	M3[04h]	DAC 3 Output Register. Sets the 8-bit voltage output value for DAC channel 3.
DAC4OUT	M3[05h]	DAC 4 Output Register. Sets the 8-bit voltage output value for DAC channel 4.
ISINKCN	M3[07h]	Current Sink Control Register. Enables/disables and sets the 8-bit current sink capability for both current sink channels (SINK1 and SINK2).
DACEN	M3[0Bh]	DAC Output Enable Register. Enables/disables the outputs for the four voltage DAC channels.

20.2 DAC Control Register Descriptions

The following peripheral registers are used to control the DAC functions. Addresses of registers are given as “Mx[yy],” where x is the module number (from 0 to 15 decimal) and yy is the register index (from 00h to 1Fh hexadecimal). Fields in the bit definition tables are defined as follows:

- **Name:** Symbolic names of bits or bit fields in this register.
- **Reset:** The value of each bit in this register following a standard reset. If this field reads “unchanged,” the given bit is unaffected by standard reset. If this field reads “s,” the given bit does not have a fixed 0 or 1 reset value because its value is determined by another internal state or external condition.
- **POR:** If present this field defines the value of each bit in this register following a power-on reset (as opposed to a standard reset). Some bits are unaffected by standard resets and are set/cleared by POR only.
- **Access:** Bits can be read-only (r) or read/write (rw). Any special restrictions or conditions that could apply when reading or writing this bit are detailed in the bit description.

MAXQ Family User's Guide: MAXQ8913 Supplement

20.2.1 DAC 1 Output Register (DAC1OUT, M3[02h])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	—	—	DAC1OUT	
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	r	r	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	DAC1OUT							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:10: Reserved

Bits 9:0: DAC 1 Output Value. Bits 9:0 in this register set the voltage output value for the DAC1 channel between zero (DAC1OUT[9:0] = 000000000b) and full scale (DAC1OUT[9:0] = 111111111b). For this register value to be effective, one or both of the following two register bits in DACEN must be set.

- If DACEN.0 (DACEN1) is set to 1, the DAC1OUT value is output to differential pins RIN+/RIN-.
- If DACEN.4 (DACOUT1) is set to 1, the DAC1OUT value is output to pin DAC1.

20.2.2 DAC 2 Output Register (DAC2OUT, M3[03h])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	—	—	DAC2OUT	
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	r	r	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	DAC2OUT							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:10: Reserved

Bits 9:0: DAC 2 Output Value. Bits 9:0 in this register set the voltage output value for the DAC2 channel between zero (if DAC2OUT[9:0] = 000000000b) and full scale (DAC2OUT[9:0] = 111111111b). For this register value to be effective, one or both of the following two register bits in DACEN must be set.

- If DACEN.1 (DACEN2) is set to 1, the DAC2OUT value is output to differential pins LIN+/LIN-.
- If DACEN.5 (DACOUT2) is set to 1, the DAC2OUT value is output to pin DAC2.

MAXQ Family User's Guide: MAXQ8913 Supplement

20.2.3 DAC 3 Output Register (DAC3OUT, M3[04h])

Bit #	7	6	5	4	3	2	1	0
Name	DAC3OUT							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 7:0: DAC 3 Output Value. This register sets the voltage output value for the DAC3 channel between zero (DAC3OUT = 0000000b) and full scale (DAC3OUT = 1111111b). For this register value to be effective, DACEN.2 (DACEN3) must be set to 1 to enable the voltage output at pin DAC3. If DACEN3 = 0, the value stored in this register has no effect on the voltage level of pin DAC3.

20.2.4 DAC 4 Output Register (DAC4OUT, M3[05h])

Bit #	7	6	5	4	3	2	1	0
Name	DAC4OUT							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 7:0: DAC 4 Output Value. This register sets the voltage output value for the DAC4 channel between zero (DAC4OUT = 0000000b) and full scale (DAC4OUT = 1111111b). For this register value to be effective, DACEN.3 (DACEN4) must be set to 1 to enable the voltage output at pin DAC4. If DACEN4 = 0, the value stored in this register has no effect on the voltage level of pin DAC4.

20.2.5 Current Sink Control Register (ISINKCN, M3[07h])

Bit #	15	14	13	12	11	10	9	8
Name	ISINK2							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	ISINK1							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:8: Current Sink 2 Output Value (ISINK2.[7:0]). Bits 15 to 8 in this register set the operating mode for current sink output 2 (SINK2) as follows.

- ISINK2 = 00h. Current output SINK2 is disabled.
- ISINK2 = (01h to FFh). Current output SINK2 is enabled, with a current sink capability given by

$$I_{SINK2} = ISINKCN[15:8] \times 62.5\mu A$$

Bits 7:0: Current Sink 1 Output Value (ISINK1.[7:0]). Bits 7:0 in this register set the operating mode for current sink output 1 (SINK1) as follows.

- ISINK1 = 00h. Current output SINK1 is disabled.
- ISINK1 = (01h to FFh). Current output SINK1 is enabled, with a current sink capability given by

$$I_{SINK1} = ISINKCN[7:0] \times 62.5\mu A$$

MAXQ Family User's Guide: MAXQ8913 Supplement

SECTION 21: TIMER/COUNTER B MODULE (SPECIFIC TO MAXQ8913)

The MAXQ8913 provides one Type B timer/counter module that operates as described in this section. Table 21-1 and Table 21-2 list the associated pins and registers for these timer/counter modules.

Table 21-1. Type B Timer/Counter Input and Output Pins

TIMER/COUNTER FUNCTION	PIN	MULTIPLEXED WITH GPIO
TB0A: Timer B I/O Pin A	K3	P1.2
TB0B: Timer B I/O Pin B	L2	P1.3

Table 21-2. Type B Timer/Counter Control Registers

REGISTER	ADDRESS	FUNCTION
TBR	M2[07h]	Type B Timer/Counter Capture/Reload Register. Holds the reload value in autoreload mode; used to store capture values in capture mode.
TBC	M2[0Bh]	Type B Timer/Counter Compare Register. This register is used for comparison against the TBV value when compare mode is enabled.
TBCN	M2[06h]	Type B Timer/Counter Control Register. Contains the control, mode, and interrupt bits.
TBV	M2[0Ah]	Type B Timer/Counter Value Register. Contains the current timer/counter value.

21.1 Timer/Counter B Register Descriptions

The following peripheral registers are used to control the LCD display controller. Addresses for all registers are given as "Mx[yy]," where x is the module number (from 0 to 15 decimal) and yy is the register index (from 00h to 1Fh hexadecimal). Fields in the bit definition tables are defined as follows:

- **Name:** Symbolic names of bits or bit fields in this register.
- **Reset:** The value of each bit in this register following a standard reset. If this field reads "unchanged," the given bit is unaffected by standard reset. If this field reads "s," the given bit does not have a fixed 0 or 1 reset value because its value is determined by another internal state or external condition.
- **POR:** If present this field defines the value of each bit in this register following a power-on reset (as opposed to a standard reset). Some bits are unaffected by standard resets and are set/cleared by POR only.
- **Access:** Bits can be read-only (r) or read/write (rw). Any special restrictions or conditions that could apply when reading or writing this bit are detailed in the bit description.

21.1.1 Timer B Timer/Counter Capture/Reload Register (TBR, M2[07h])

Bit #	15	14	13	12	11	10	9	8
Name	TBR							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw
Bit #	7	6	5	4	3	2	1	0
Name	TBR							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0: Timer B Capture/Reload Register. This register is used to capture the TBV value when Timer B is configured in capture mode. This register is also used as the 16-bit reload value when Timer B is configured in autoreload mode.

MAXQ Family User's Guide: MAXQ8913 Supplement

21.1.2 Timer B Timer/Counter Compare Register (TBC, M2[0Bh])

Bit #	15	14	13	12	11	10	9	8
Name	TBC							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	TBC							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0: Timer B Compare Register. This register is used for comparison vs. the TBV value when Timer B is operated in compare mode.

21.1.3 Timer B Timer/Counter Control Register (TBCN, M2[06h])

Bit #	15	14	13	12	11	10	9	8
Name	C/TB	—	—	TBCS	TBCR	TBPS2	TBPS1	TBPS0
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	TFB	EXFB	TBOE	DCEN	EXENB	TRB	ETB	CP/RLB
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit 15: Counter/Timer Select (C/TB). This bit determines whether Timer B functions as a timer or counter. Setting this bit to 1 causes Timer B to count negative transitions on the TBOA pin. Clearing this bit to 0 causes Timer B to function as a timer. The speed of Timer B is determined by the TBPS[2:0] bits of TBCN.

Bits 14:13: Reserved. Read returns zero.

Bits 12:11: TB0B Pin Output Reset/Set Mode Bits (TBCS, TBCR). These mode bits define whether the PWM mode output function is enabled on the TB0B pin, the initial output starting state, and what compare mode output function is in effect. Note that the TB0B pin still has certain input functionality when the PWM output function is enabled.

Bits 10:8: Timer B Clock Prescaler Bits (TBPS[2:0]). These bits select the clock prescaler applied to the system clock input to Timer B. The TBPS[2:0] bits should be configured by the user when the timer is stopped (TRB = 0). While hardware does not prevent changing the TBPS[2:0] bits when the timer is running, the resultant behavior is indeterminate.

$$\text{Timer B Clock} = \text{System Clock} / 2^{(2 \times \text{TBPS}[2:0])}$$

TBPS[2:0]	TIMER B INPUT CLOCK
000	Sysclk/1
001	Sysclk/4
010	Sysclk/16
011	Sysclk/64
100	Sysclk/256
101	Sysclk/1024
11x	Sysclk/1

MAXQ Family User's Guide: MAXQ8913 Supplement

Bit 7: Timer B Overflow Flag (TFB). This bit is set when Timer B overflows from TBR or the count is equal to 0000h in down-count mode. It must be cleared by software.

Bit 6: External Timer B Trigger Flag (EXFB). When configured as a timer (C/TB = 0), a negative transition on the TB0B pin causes this flag to be set if (CP/RLB = EXENB = 1) or (CP/RLB = DCEN = 0 and EXENB = 1) or (CP/RLB = 0 and EXENB = 1 and TBCS:TBCR <> 00b). When configured in any of these ways, this flag can be set independent of the state of the TRB bit (e.g., EXFB can still be set on detection of a negative edge when TRB = 0).

When CP/RLB = 0 and DCEN = 1 and TBCS:TBCR = 00b, EXFB toggles whenever Timer B underflows or overflows. Overflow/underflow condition is the same as described in the TFB bit description. In this mode, EXFB can be used as the 17th timer bit and does not cause an interrupt. If set by a negative transition, this flag must be cleared by software. Setting this bit to 1 forces a timer interrupt if enabled.

Bit 5: Timer B Output Enable (TBOE). Setting this bit to 1 enables the clock output function on the TB0A pin if C/TB = 0. Timer B rollovers do not cause interrupts. Clearing this bit to 0 allows the TB0A pin to function as either a standard port pin or a counter input for Timer B.

Bit 4: Down-Count Enable (DCEN). This bit, in conjunction with the TB0B pin, controls the direction that Timer B counts in 16-bit autoreload mode. Clearing this bit to 0 causes Timer B to count up only. Setting this bit to 1 enables the up/down-counting mode (i.e., it causes Timer B to count up if the TB0B pin is 1 and to count down if the TB0B pin is 0). When Timer B PWM-output mode functionality is enabled along with up/down counting (DCEN = 1), the up/down count control of Timer B is controlled internally based upon the count in relation to the register settings. In the compare modes, the DCEN bit controls whether the timer counts up and resets (DCEN = 0), or counts up and down (DCEN = 1).

Bit 3: Timer B External Enable (EXENB). Setting this bit to 1 enables the capture/reload function on the TB0B pin for a negative transition (in up-counting mode). A reload results in TBV being reset to 0000h. Clearing this bit to 0 causes Timer B to ignore all external events on TB0B pin. When operating in autoreload mode (CP/RLB = 0) with the PWM output functionality enabled, enabling the TB0B input function (EXENB = 1) allows PWM output negative transitions to set the EXFB flag, however, no reload occurs as a result of the external negative-edge detection.

Bit 2: Timer B Run Control (TRB). This bit enables Timer B operation when set to 1. Clearing this bit to 0 halts Timer B operation and preserves the current count in TBV.

Bit 1: Enable Timer B Interrupt (ETB). Setting this bit to 1 enables the interrupt from the Timer B TFB and EXFB flags in TBCN. In Timer B clock-output mode (TBOE = 1), the timer overflow flag (TFB) is still set on an overflow, however, the TBOE = 1 condition prevents this flag from causing an interrupt when ETB = 1.

Bit 0: Capture/Reload Select (CP/RLB). This bit determines whether the capture or reload function is used for Timer B. Timer B functions in an autoreload mode following each overflow/underflow. See the TFB bit description for overflow/underflow condition. Setting this bit to 1 causes a Timer B capture to occur when a falling edge is detected on the TB0B pin if EXENB is 1. Clearing this bit to 0 causes an autoreload to occur when Timer B overflow or a falling edge is detected on TB0B pin if EXENB is 1. It is not intended that the Timer B compare functionality should be used when operating in capture mode.

21.1.4 Timer B Timer Value Register (TBV, M2[0Ah])

Bit #	15	14	13	12	11	10	9	8
Name	TBV							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	TBV							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:0: Timer B Value Register. This register is used to load and read the 16-bit Timer B value.

21.2 Timer/Counter B Operation

Timer/Counter B is a 16-bit programmable device that supports clock input prescaling and set/reset/toggle PWM/output control functionality not found on other MAXQ timer implementations. A new register, TBC, supports certain PWM/output control functions in some implementations. Another distinguishing characteristic of Timer/Counter B is that its count ranges from 0000h to the value stored in the 16-bit capture/reload register (TBR) whereas in other implementations (e.g., Timer 1), the count ranges from the value in the reload register to FFFFh.

The possible Timer B operating modes and related control bits are shown in Table 21-3. A complete description of each mode is contained in the subsequent sections. In all modes of operation, the timer is enabled by setting the Timer B run control bit (TRB) of the Timer B control register (TBCN.2) to 1. If this bit is cleared to 0 (reset default condition), no timer activity is possible. When Timer B is operated as a timer (i.e., it counts scaled system clocks), the three bits TBPS2, TBPS1, and TBPS0 in the timer control register (TBCN[10:8]) determine the factor by which the active system clock is divided (prescaled) before being counted by the timer. Other relevant control bits are described in the following mode descriptions. A complete listing of the Timer B registers and bits with their effects on timer operation are given in *Section 21.1: Timer/Counter B Register Descriptions*.

Table 21-3. Timer/Counter B Mode Summary

TIMER B OPERATIONAL MODE	TBCN REGISTER BIT SETTINGS						
	TBCS:TBCR	TBOE	DCEN	EXENB	C/TB	CP/RLB	OPTIONAL CONTROL
Autoreload	00	0	0	0	X	0	—
Autoreload Using TB0B Pin	00	0	0	1	X	0	—
Capture Using TB0B Pin	00	0	0	1	X	1	—
Up/Down Count Using TB0B Pin	00	0	1	0	X	0	—
Up-Count PWM/Output Control	<>00	X	0	X	X	0	—
Up/Down PWM/Output Control	<>00	X	1	X	X	0	—
—	—	0	X	X	1	X	Input Clock = TB0A Pin
Clock Output on TB0B Pin	—	1	X	X	0	0	—

21.2.1 Timer B 16-Bit Timer/Counter Mode with Autoreload

The 16-bit autoreload mode of Timer B is established by clearing the CP/RLB bit of the control register (TBCN.0) to 0. In this mode, the timer performs a simple 16-bit timer or counter function that is reset to 0000h when a match between the Timer B count value register (TBV) and the Timer B capture/reload register (TBR) occurs. A functional diagram of autoreload mode is illustrated in Figure 21-1. If the C/TB bit of the timer's control register (TBCN.15) is a logic 0, the timer's input clock is a prescaled system clock. When C/TB is a logic 1, pulses on the TBOA pin are counted. As in all modes, counting or timing is enabled or disabled with the Timer B run control bit TRB (TBCN.2).

When enabled (i.e., TRB = 1) in this mode, Timer B begins counting up from the current value contained in the TBV register. The TBV register contents can be read or written any time by software and contains the current value of the timer. When the value in the TBV register reaches the value contained in capture/reload register TBR, the TBF flag is set to 1. This flag can generate an interrupt if enabled. In addition, upon this match, the timer reloads the TBV register with 0000h, and continues timing (or counting) up from that value. The reload value contained in the TBR register is preloaded by software. This register cannot be used for the capture function while also performing autoreload, so these modes are mutually exclusive.

While in autoreload mode, Timer B can also be forced to reload the TBV register with 0000h using the TB0B pin. If the control bit EXENB (TBCN.3) is set to 1, a 1-to-0 transition on the TB0B pin causes a reload. If the EXENB bit is cleared to 0, the TB0B pin is ignored.

MAXQ Family User's Guide: MAXQ8913 Supplement

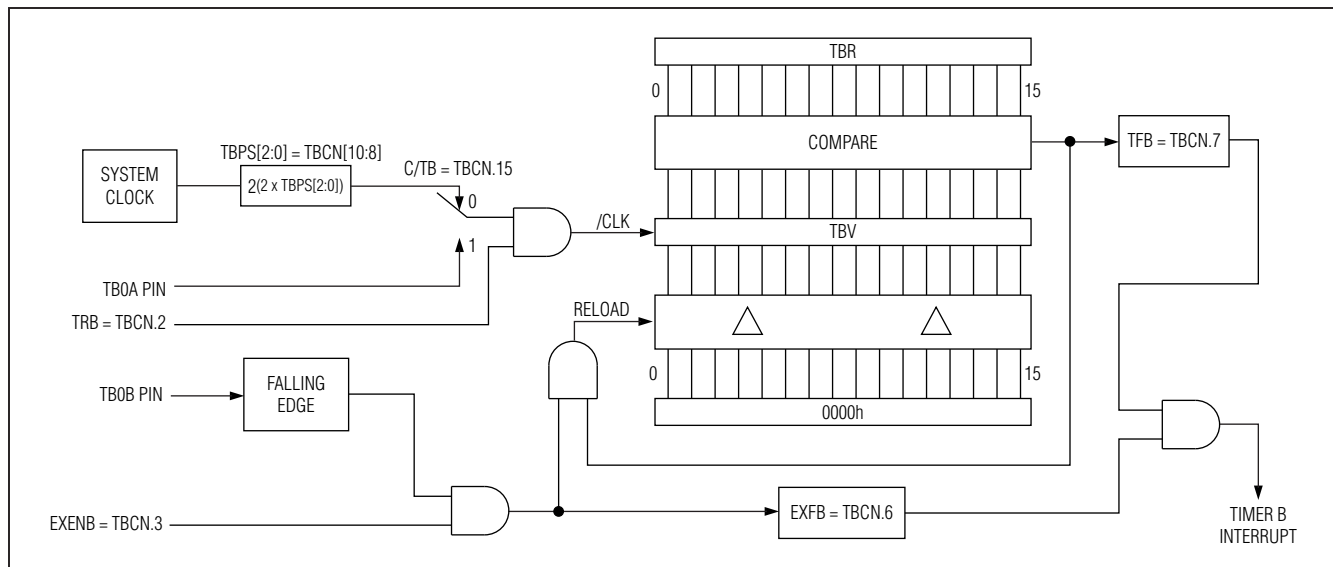


Figure 21-1. Timer B Autoreload Mode Block Diagram

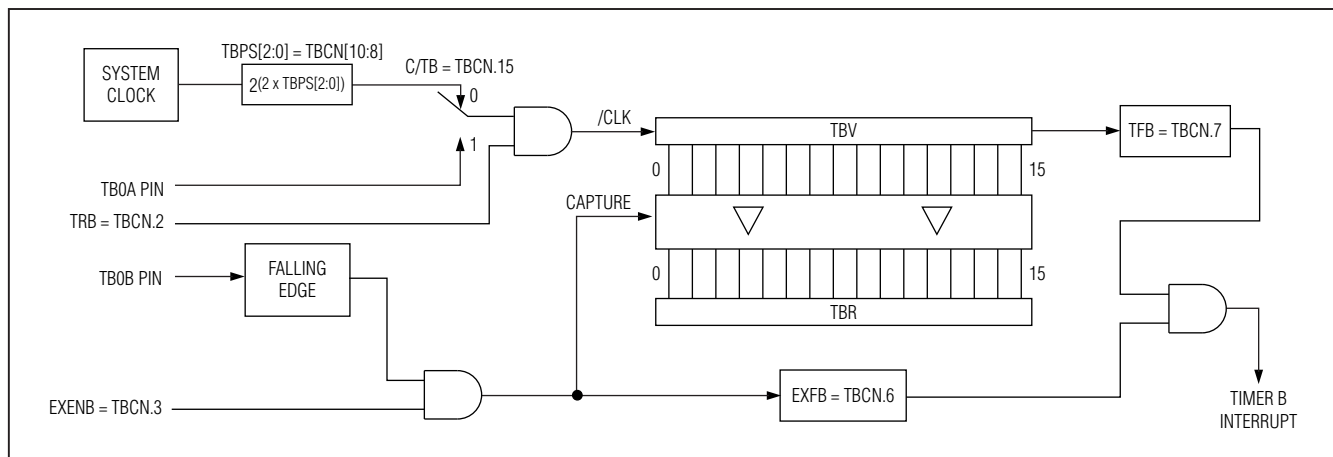


Figure 21-2. Timer B 16-Bit Capture Mode Block Diagram

21.2.2 Timer B 16-Bit Capture Mode

The 16-bit capture mode of Timer B is configured by setting the CP/RLB bit of the control register (TBCN.0) to 1. A functional diagram of this mode is shown in Figure 21-2. When the timer is enabled in this mode, it begins counting up from the value contained in the TBV register until reaching an overflow state, i.e., FFFFh \rightarrow 0000h; at which point it sets the TBF flag (TBCN.7) and continues counting upward. When the TBF flag is set, it can generate an interrupt if enabled. This count cycle is repeated without processor intervention as long as the timer is enabled. As the counting proceeds, the value in the TBV register is captured in the capture/reload register (TBR) if and when a high-to-low transition occurs on the TBOB pin and the EXENB bit of the control register (TBCN.3) is set to 1. The EXFB flag (TBCN.6) is also set when the capture occurs, and this flag can generate an interrupt if enabled. If the EXENB bit is cleared to 0, transitions on the TBOB pin do not cause a capture event.

21.2.3 Timer B 16-Bit Up/Down Count with Autoreload Mode

The 16-bit up/down-count autoreload mode is enabled by clearing the capture/reload bit CP/RLB of the control register (TBCN.0) to 0 and setting the down-count enable bit, DCEN (TBCN.4), to 1. This mode is illustrated in Figure 21-3. When DCEN is set to 1, Timer B counts up from 0000h or down from the value contained in the TBR register as controlled by the state of the TBOB pin. When the TBOB pin is 1, the timer counts up, and counts down when the pin is 0. When DCEN is 0, Timer B can only count up.

When counting up and a match occurs between the value in the TBV register and value in the TBR register, the value of 0000h is loaded into the TBV register. When counting down and the value in the TBV register reaches 0000h, the value in the TBR register is loaded into the TBV register and downward counting continues.

Note that in this mode of operation, the overflow/underflow output of the timer is provided to an edge-detection circuit as well as to the TBF bit of the control register (TBCN.7). This edge-detection circuit toggles the EXFB bit (TBCN.6) on every overflow or underflow. Therefore, the EXFB bit behaves as a 17th bit of the counter, and can be used as such.

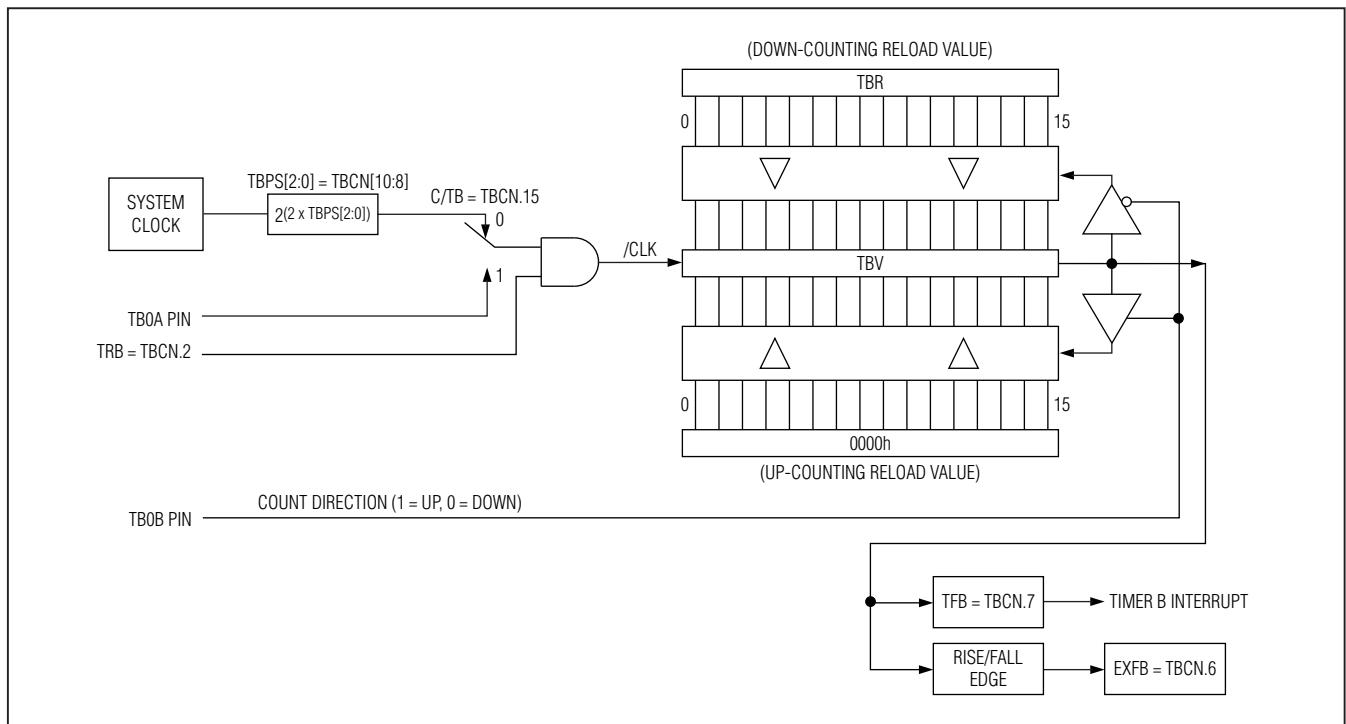


Figure 21-3. Timer B 16-Bit Up/Down Count with Autoreload Mode Block Diagram

MAXQ Family User's Guide: MAXQ8913 Supplement

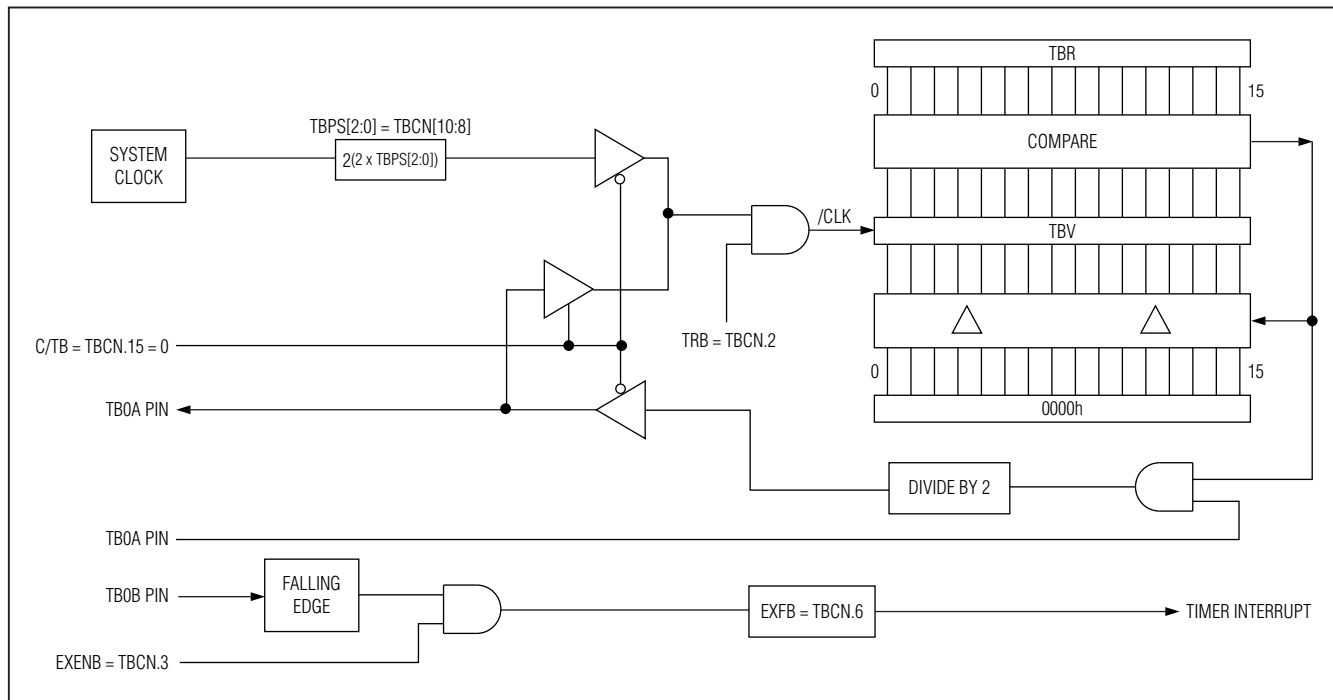


Figure 21-4. Timer B Clock Output Mode Block Diagram

21.2.4 Timer B Clock Output Mode

Timer B can be configured to drive a clock output on the TBOA pin as shown in Figure 21-4. For the timer to operate in this mode, the capture/reload select bit (CP/RLB = TBCN.0) and the counter/timer select bit (C/TB = TBCN.15) must be cleared to 0, and the Timer B output-enable bit (TBOE = TBCN.5) must be set to 1. In this mode, the DCEN bit has no effect. The clock signal output is a 50% duty-cycle square wave with a frequency given by the equation:

$$\text{Clock Output Frequency} = \text{System Clock} / (2 \times \text{TBR})$$

Therefore, for a system clock of 1MHz and a TBR register value of 0005h (arbitrary example), the clock output frequency is 100kHz.

21.2.5 Timer B PWM/Output Control Functionality

The PWM/output control function is enabled whenever either of the TCBS or TBCR bits (TBCN[12:11]) is set to 1. Table 21-4 shows how these bits determine the specific operation.

Table 21-4. Timer B PWM/Output Control Function

TBCS:TBCR	FUNCTION	INITIAL STATE (IF TRB = 0)
00	None (Disabled)	No change
01	Reset on TBC Match, Set on 0000h	Low
10	Set on TBC Match, Reset on TBR Match	High
11	Toggle on TBC Match	No change

When the timer is not running (i.e., TRB = 0), the initial output state of the TB0B pin is established as low or high, respectively, if the reset function (TBCR = 1, TBCS = 0) or set function (TBCR = 0, TBCS = 1) is configured. Invoking the toggle function does not change the already defined starting state for TB0B, thus a fixed high or low starting state can be defined for the toggle mode by first passing through the set or reset mode.

When the PWM/output control function is configured to the reset mode (TBCS = 0, TBCR = 1), clearing the TBC register to 0000h or writing its value to something greater than the counting range (i.e., greater than the value in the TBR register) effectively disables the reset operation, and produces a single pin set operation when an overflow occurs. When the PWM/output control function is configured to the set mode (TBCS = 1, TBCR = 0), making TBC = TBR or writing TBC's value to something greater than the counting range disables the set operation, and produces a single reset on TBR match. When the PWM/output control function is configured to toggle, loading the TBC register with a value outside the counting range, to 0000h, or to the value in TBR disables the toggle function.

21.2.6 16-Bit Up Count PWM/Output Control Mode

Figure 21-5 shows a functional diagram of the up count with PWM/output control mode, and Figure 21-6 shows example TB0B pin output waveforms. The set and reset modes provide similar functionality. They support up to 16-bit resolution PWM with the ability to change the frequency using the TBR reload value. The toggle mode allows a 50% duty-cycle waveform to be created (when the TBC register remains fixed with Timer B running). With the TBC register value outside of the count range, the set and reset functions allow a timed clear or set of the TB0B pin without need of polling or interrupting the CPU for manual port pin control.

Up-count set, reset PWM duty cycle is calculated as follows (where period = TBR + 1 Timer B clocks):

$$\text{Set mode} = (TBR - TBC)/(TBR + 1)$$

$$\text{Reset mode} = TBC/(TBR + 1)$$

The period for the 50% duty-cycle signal generated in toggle mode is:

$$\text{Toggle mode} = 2 \times (TBR + 1) \text{ Timer B Clock Periods}$$

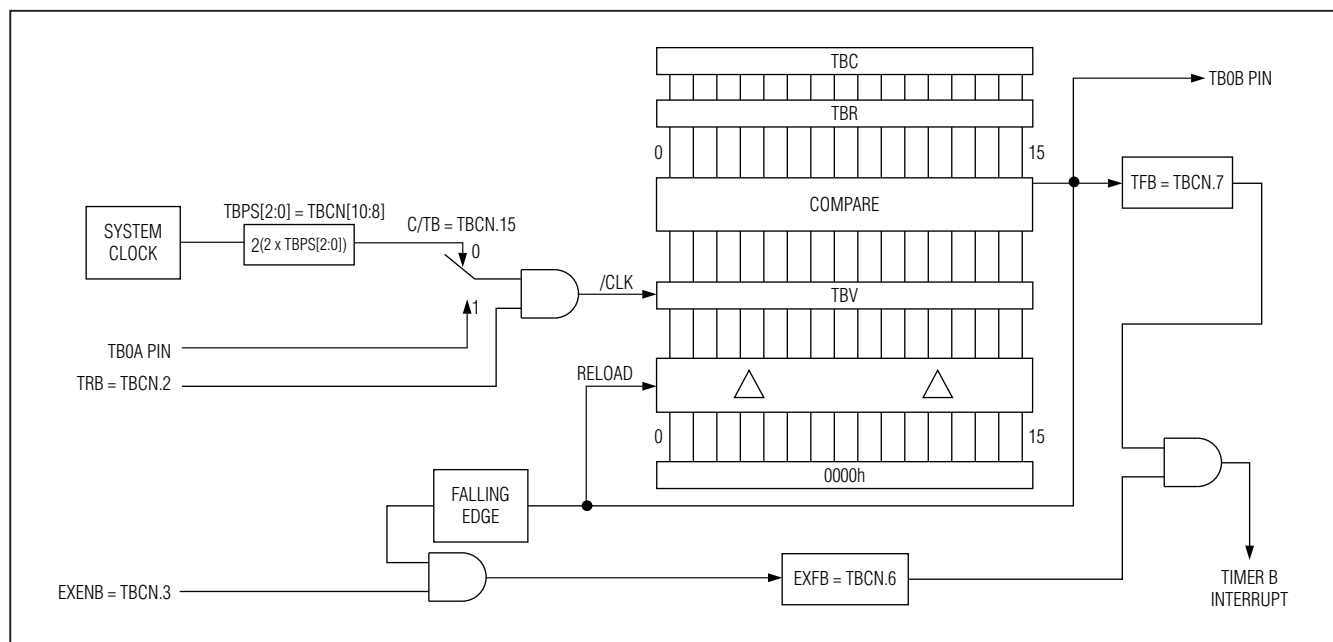


Figure 21-5. Up-Count PWM/Output Control Mode Block Diagram

MAXQ Family User's Guide: MAXQ8913 Supplement

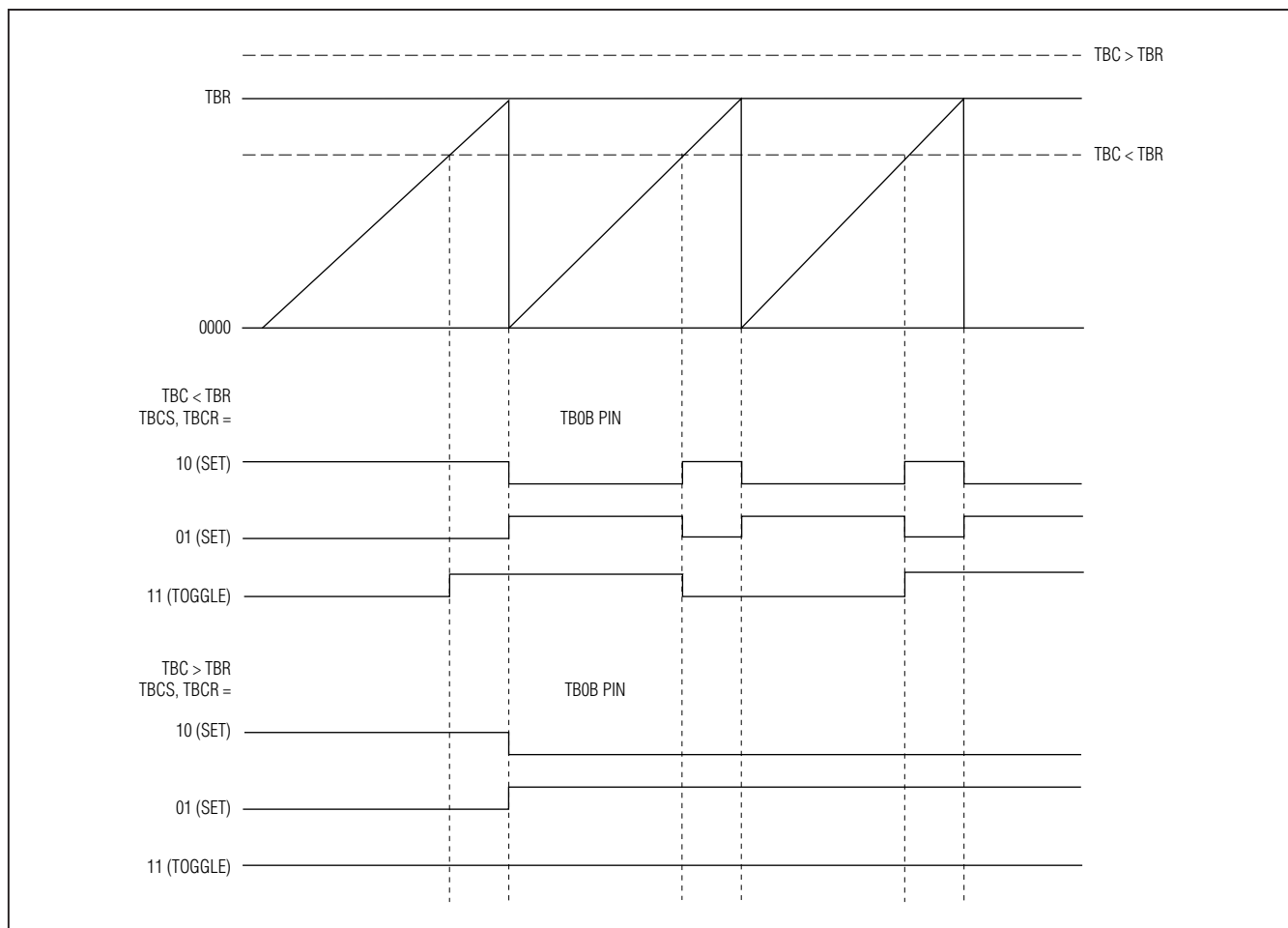


Figure 21-6. Timer B PWM/Output Control Mode Waveform (Count Up)

21.2.7 16-Bit Up/Down Count PWM/Output Control Mode

Figure 21-7 shows a functional diagram of the up/down-count PWM/output control mode. When the Timer B PWM/output control functionality is enabled at the same time as the up/down-count autoreload mode, the TBOB pin no longer controls the direction of counting. Instead, the up/down counting is controlled by logic inside the timer and is determined by the value of the TBV count value register. When the timer is counting upward and reaches the value in the TBR register, it reverses its direction of counting in the next cycle. When the timer is down counting and reaches 0000h, it reverses direction and begins counting up. This behavior and the results of the TBCS and TBCR bit setting is shown in Figure 21-8.

The up/down-count PWM duty cycle is calculated as follows (where period = 2 x TBR Timer B clocks):

$$\text{Set mode} = (TBR + TBC)/(2 \times TBR)$$

$$\text{Reset mode} = TBC/(2 \times TBR)$$

$$\text{Toggle mode} = TBC/TBR \text{ or } (TBR - TBC)/TBR$$

The set and reset up/down-count PWM/output control modes effectively allow 17-bit resolution since set allows duty-cycle variation $\geq 50\%$ with 50% of the period always being high, and reset allows duty-cycle variation $\leq 50\%$ with 50% of the period always being low. The toggle mode provides a center-aligned 16-bit PWM with twice the period of the pure up-counting autoreload mode.

MAXQ Family User's Guide: MAXQ8913 Supplement

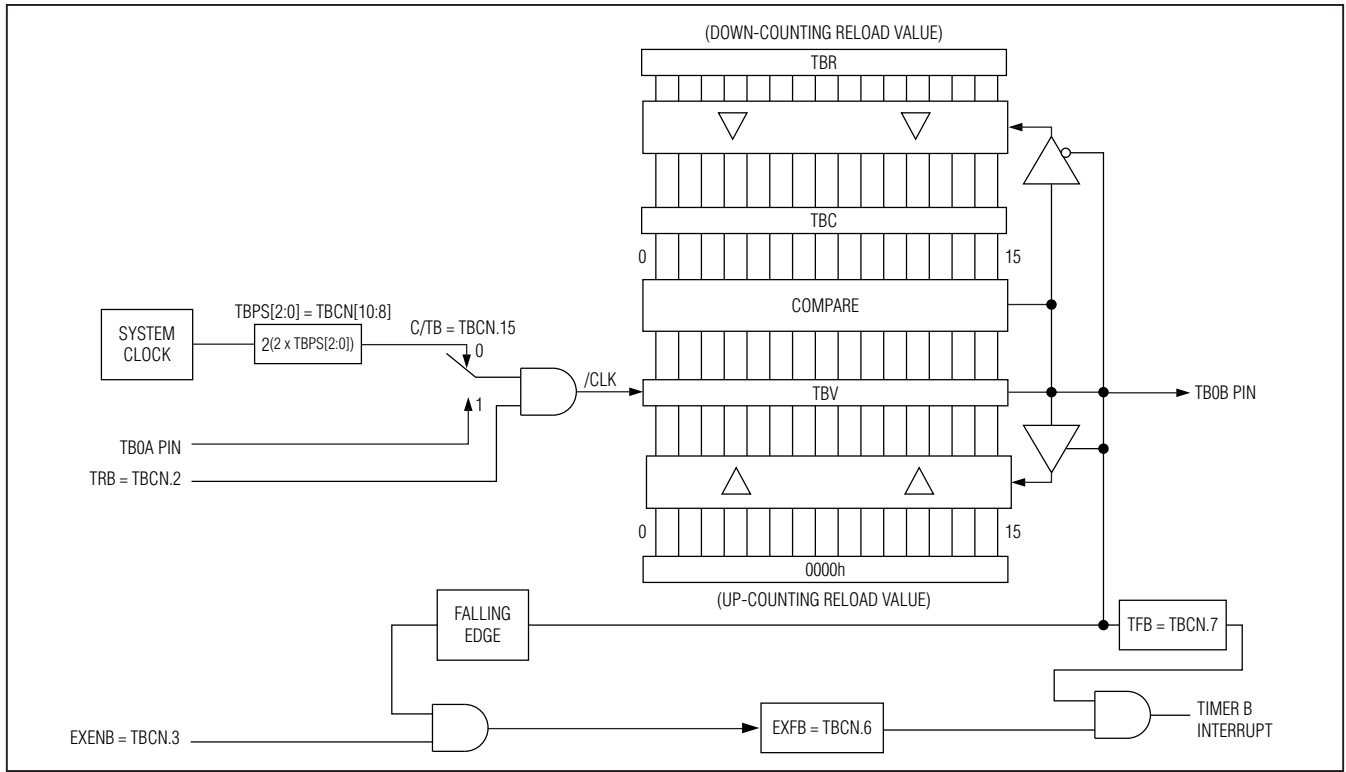


Figure 21-7. Timer B Up/Down-Count PWM/Output Control Mode Block Diagram

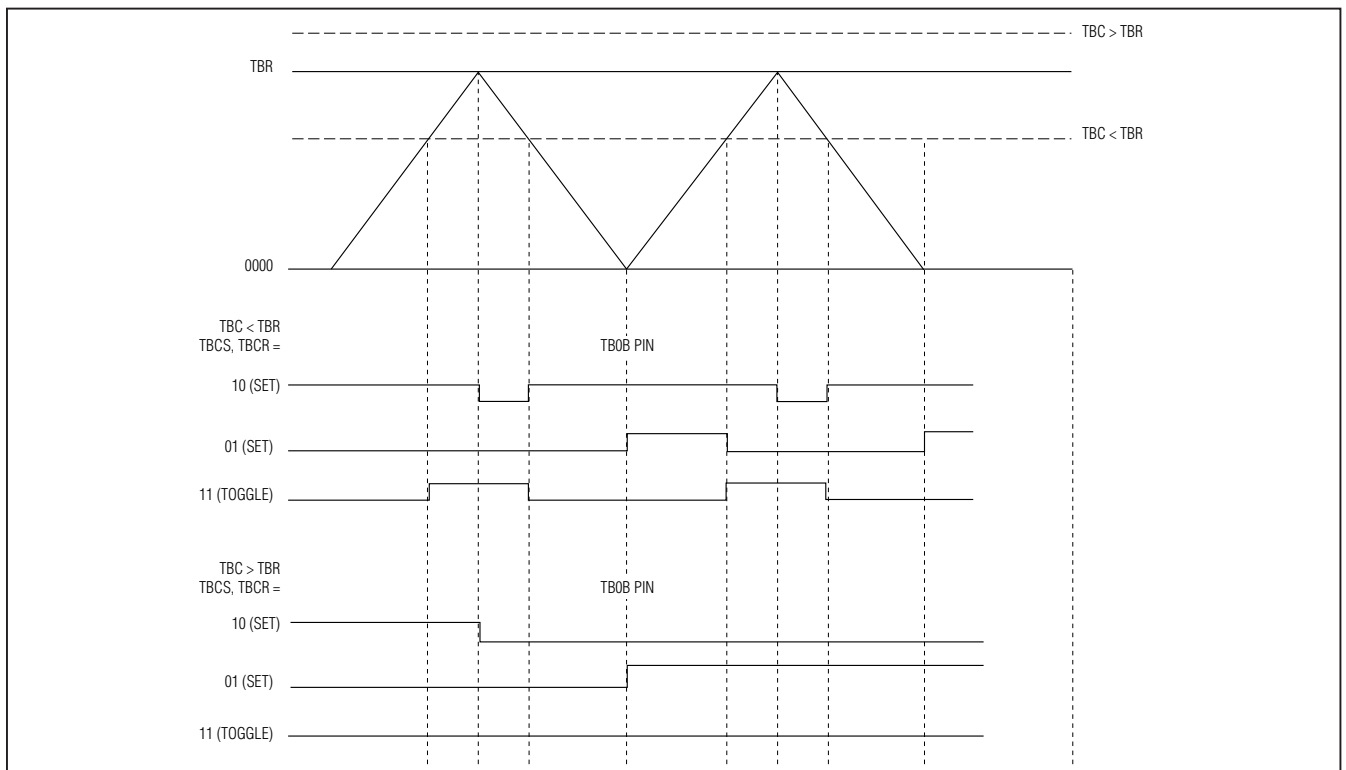


Figure 21-8. Timer B PWM/Output Control Mode Waveform (Up/Down Count)

MAXQ Family User's Guide: MAXQ8913 Supplement

21.2.8 EXENB Control During PWM/Output Control Mode

The TBOB input function (EXENB = 1) and the PWM/output control function (TBCS:TBCR<>00b) can be enabled at the same time. However, the input function changes slightly when this is done. In this configuration, the detection of a falling edge on the TBOB pin results in setting of the EXFB interrupt flag, but does **not** force an autoreload.

21.3 Timer B Examples

21.3.1 Timer B Example: Reloading Timer Mode

```
move PD5.2, #1          ; Set P5.2 to output mode
move PO5.2, #1          ; Drive high (LED off)

move TBCN, #000001000000000000b ; Timer mode, reload mode, clock/256
move TBR, #8000h        ; Reload every 8000h timer cycles
move TBCN.2, #1         ; Start timer

;; Reload time is 1/sysclk * 256 * 8000h == 838ms

timerLoop:
move C, TBCN.7          ; Check reload flag
jump NC, timerLoop

move TBCN.7, #0         ; Clear reload flag
move Acc, PO5
xor #0100b              ; Toggle bit 2
move PO5, Acc
jump timerLoop
```

MAXQ Family User's Guide: MAXQ8913 Supplement

SECTION 22 : I²C BUS INTERFACE (SPECIFIC TO MAXQ8913)

The MAXQ8913 provides an inter-IC (I²C) communications module that includes master and slave modes. The associated pins and registers for this interface are listed in Table 22-1 and Table 22-2.

Table 22-1. I²C Input and Output Pins

I ² C INTERFACE FUNCTION	PIN	MULTIPLEXED WITH GPIO
SCL: Clock	N4	P1.0
SDA: Data	M3	P1.1

Table 22-2. I²C Interface Control Registers

REGISTER	ADDRESS	FUNCTION
I2CBUF	M1[06h]	I ² C Data Buffer Register. Interface register for the input and output I ² C buffers.
I2CST	M1[05h]	I ² C Status Register. Contains the interrupt and status flags for the I ² C interface.
I2CIE	M1[07h]	I ² C Interrupt Enable Register. Contains the interrupt-enable bits and control bits for general call address matching and clock stretching.
I2CCN	M1[04h]	I ² C Control Register. Contains the control and configuration bits for the I ² C interface.
I2CCK	M1[0Ch]	I ² C Clock Control Register. Defines the SCL high and low clock periods for I ² C master mode.
I2CTO	M1[0Dh]	I ² C Timeout Register. Enables/disables the master mode timeout when the bus is busy or SCL is held low by another device longer than the maximum allowed time, and defines the timeout period.
I2CSLA	M1[0Eh]	I ² C Slave Address Register. Defines the 7-bit/10-bit address that is recognized by the slave portion of the I ² C interface.

22.1 I²C Register Descriptions

The following peripheral registers are used to control the integrated I²C peripheral on the MAXQ8913. Addresses for all registers are given as "Mx[yy]," where x is the module number (from 0 to 15 decimal) and yy is the register index (from 00h to 1Fh hexadecimal). Fields in the bit definition tables are defined as follows:

- **Name:** Symbolic names of bits or bit fields in this register.
- **Reset:** The value of each bit in this register following a standard reset. If this field reads "unchanged," the given bit is unaffected by standard reset. If this field reads "s," the given bit does not have a fixed 0 or 1 reset value because its value is determined by another internal state or external condition.
- **POR:** If present this field defines the value of each bit in this register following a power-on reset (as opposed to a standard reset). Some bits are unaffected by standard resets and are set/cleared by POR only.
- **Access:** Bits can be read-only (r) or read/write (rw). Any special restrictions or conditions that could apply when reading or writing this bit are detailed in the bit description.

22.1.1 I²C Data Buffer Register (I2CBUF, M1[06h])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	—	—	I2CBUF	
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	r	r	w*	w*

Bit #	7	6	5	4	3	2	1	0
Name	I2CBUF							
Reset	0	0	0	0	0	0	0	0
Access	rw*	rw*	rw*	rw*	rw*	rw*	rw*	rw*

Note: ADCONV cannot be written when PMME = 1 and SWB = 0.
*Can be written only when I2CBUSY = 0.

Bits 9:0: This register is used as the read and write data buffer for I²C data and address transmissions.

22.1.1.1 I²C Data Read and Write

Data for I²C transfer is read and written to this location. The I²C transmit and receive buffers are internally stored separately; however, both are accessed through this buffer. For data transfers, only the low eight bits (I2CBUF[7:0]) are significant.

22.1.1.2 I²C Address Transmission

When transmitting an I²C address, the address should be loaded into I2CBUF[6:0] for a 7-bit address (if I2CEA = 0) and into I2CBUF[9:0] for a 10-bit address (if I2CEA = 1). Bits 8 and 9 are effectively write-only since there is no circumstance under which reading them returns useful data.

22.1.2 I²C Status Register (I2CST, M3[01h])

Bit #	15	14	13	12	11	10	9	8
Name	I2CBUS	I2CBUSY	—	—	I2CSPI	I2CSCL	I2CROI	I2CGCI
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	rw	r	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	I2CNACKI	I2CALI	I2CAMI	I2CTOI	I2CSTRI	I2CRXI	I2CTXI	I2CSRI
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit 15: I²C Bus Busy (I2CBUS). This bit is set to 1 when a START/repeated START condition is detected and cleared to 0 when the STOP condition is detected. This bit is reset to 0 on all forms of reset and when I2CEN = 0. This bit is controlled by hardware and is read-only.

Bit 14: I²C Busy (I2CBUSY). This bit is used to indicate the current status of the I²C module. The I2CBUSY is set to 1 when the I²C controller is actively participating in a transaction or when it does not have control of the bus. This bit is controlled by hardware and is read-only.

Bits 13:12: Reserved. Read returns 0.

Bit 11: I²C STOP Interrupt Flag (I2CSPI). This bit is set to 1 when a STOP condition (P) is detected. This bit must be cleared to 0 by software once set. Setting this bit to 1 by software causes an interrupt if enabled.

Bit 10: I²C SCL Status (I2CSCL). This bit reflects the logic state of the SCL signal. This bit is set to 1 when SCL is at a logic-high (1), and cleared to 0 when SCL is at a logic-low (0). This bit is controlled by hardware and is read-only.

MAXQ Family User's Guide: MAXQ8913 Supplement

Bit 9: I²C Receiver Overrun Flag (I2CROI). This bit indicates a receive overrun when set to 1. This bit is set to 1 if the receiver has already received two bytes since the last CPU read. This bit is cleared to 0 by software reading the I2CBUF. Setting this bit to 1 by software causes an interrupt if enabled. Writing 0 to this bit does not clear the interrupt.

Bit 8: I²C General Call Interrupt Flag (I2CGCI). This bit is set to 1 when the general call is enabled (I2CGCEN = 1) and the general call address is received. This bit must be cleared to 0 by software once set. Setting this bit to 1 by software causes an interrupt if enabled.

Bit 7: I²C NACK Interrupt Flag (I2CNACKI). This bit is set to 1 if the I²C transmitter receives a NACK from the receiver. Setting this bit to 1 by hardware causes an interrupt if enabled. This bit must be cleared to 0 by software once set. This bit is set by hardware only.

Bit 6: I²C Arbitration Loss Flag (I2CALI). This bit is set to 1 when the I²C is configured as a master and loses in the arbitration. When the master loses arbitration, the I2CMST bit is cleared to 0. Setting this bit to 1 by hardware causes an interrupt if enabled. This bit must be cleared to 0 by software once set. This bit is set by hardware only.

Bit 5: I²C Slave Address Match Interrupt Flag (I2CAMI). This bit is set to 1 when the I²C controller receives an address that matches the contents in its slave address register (I2CSLA) during the address stage. This bit must be cleared to 0 by software once set. Setting this bit to 1 by software causes an interrupt if enabled.

Bit 4: I²C Timeout Interrupt Flag (I2CTOI). This bit is set to 1 if either the I²C controller cannot generate a START condition or the I²C SCL low time has expired the timeout value specified in the I2CTO register. This happens when the I²C controller is operating in master mode and some other device on the bus is using the bus or holding SCL low for an extended period of time. This bit must be cleared to 0 by software once set. Setting this bit to 1 by software causes an interrupt if enabled.

Bit 3: I²C Clock Stretch Interrupt Flag (I2CSTRI). This bit indicates that the I²C controller is operating with clock stretching enabled and is holding the SCL clock signal low. The I²C controller releases SCL after this bit has been cleared to 0. Setting this bit to 1 by hardware causes an interrupt if enabled. This bit must be cleared to 0 by software once set. This bit is set by hardware only.

Bit 2: I²C Receive Ready Interrupt Flag (I2CRXI). This bit indicates that a data byte has been received in the I²C buffer. This bit must be cleared by software once set. Setting this bit to 1 by hardware causes an interrupt if enabled. This bit is set by hardware only.

Bit 1: I²C Transmit Complete Interrupt Flag (I2CTXI). This bit indicates that an address or a data byte has been successfully shifted out and the I²C controller has received an acknowledgment from the receiver (NACK or ACK). This bit must be cleared by software once set. Setting this bit to 1 by software causes an interrupt if enabled.

Bit 0: I²C START Interrupt Flag (I2CSRI). This bit is set to 1 when a START condition (S or Sr) is detected. This bit must be cleared to 0 by software once set. Setting this bit to 1 by software causes an interrupt if enabled.

22.1.3 I²C Interrupt Enable Register (I2CIE, M1[07h])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	I2CSPIE	—	I2CROIIE	I2CGCIE
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	r	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	I2CNACKIE	I2CALIE	I2CAMIE	I2CTOIE	I2CSTRIE	I2CRXIE	I2CTXIE	I2CSRIE
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:12, 10: Reserved. Read returns 0.

Bit 11: I²C STOP Interrupt Enable (I2CSPIE). Setting this bit to 1 causes an interrupt to the CPU when a STOP condition is detected (I2CSPI = 1). Clearing this bit to 0 disables the STOP detection interrupt from generating.

MAXQ Family User's Guide: MAXQ8913 Supplement

Bit 9: I²C Receiver Overrun Interrupt Enable (I2CROIE). Setting this bit to 1 causes an interrupt to the CPU when a receiver overrun condition is detected (I2ROI = 1). Clearing this bit to 0 disables receiver overrun detection interrupt from generating.

Bit 8: I²C General Call Interrupt Enable (I2CGCIE). Setting this bit to 1 generates an I2CGCI (general call interrupt) to the CPU when general call is enabled (I2CGCEN = 1). Clearing this bit to 0 disables general call interrupt from generating.

Bit 7: I²C NACK Interrupt Enable (I2CNACKIE). Setting this bit to 1 causes an interrupt to the CPU when a NACK is detected (I2CNACKI = 1). Clearing this bit to 0 disables NACK detection interrupt from generating.

Bit 6: I²C Arbitration Loss Interrupt Enable (I2CALIE). Setting this bit to 1 causes an interrupt to the CPU when the I²C master loses in an arbitration (I2CALI = 1). Clearing this bit to 0 disables arbitration loss interrupt from generating.

Bit 5: I²C Slave Address Match Interrupt Enable (I2CAMIE). Setting this bit to 1 causes an interrupt to the CPU when the I²C controller detects an address that matches the I2CSLA value (I2CAMI = 1). Clearing this bit to 0 disables address match interrupt from generating.

Bit 4: I²C Timeout Interrupt Enable (I2CTOIE). Setting this bit to 1 causes an interrupt to the CPU when a timeout condition is detected (I2CTOI = 1). Clearing this bit to 0 disables timeout interrupt from generating.

Bit 3: I²C Clock Stretch Interrupt Enable (I2CSTRIE). Setting this bit to 1 generates an interrupt to the CPU when the clock stretch interrupt flag is set (I2CSTRI = 1). Clearing this bit disables the clock stretch interrupt from generating.

Bit 2: I²C Receive Ready Interrupt Enable (I2CRXIE). Setting this bit to 1 causes an interrupt to the CPU when the receive interrupt flag is set (I2CRXI = 1). Clearing this bit to 0 disables the receive interrupt from generating.

Bit 1: I²C Transmit Complete Interrupt Enable (I2CTXIE). Setting this bit to 1 causes an interrupt to the CPU when the transmit interrupt flag is set (I2CTXI = 1). Clearing this bit to 0 disables the transmit interrupt from generating.

Bit 0: I²C START Interrupt Enable (I2CSRIE). Setting this bit to 1 causes an interrupt to the CPU when a START condition is detected (I2CSRI = 1). Clearing this bit to 0 disables the START detection interrupt from generating.

22.1.4 I²C Control Register (I2CCN, M1[04h])

Bit #	15	14	13	12	11	10	9	8
Name	I2CRST	—	—	—	—	—	I2CSTREN	I2CGCEN
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	I2CSTOP	I2CSTART	I2CACK	I2CSTRS	I2CEA	I2CMODE	I2CMST	I2CEN
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Note 1: I2CSTART and I2CSTOP are mutually exclusive and reset to 0 when I2CMST = 0 or I2CEN = 0.

Note 2: I2CRST is reset to 0 when I2CEN = 0.

Note 3: Writes to I2CMST, I2CMODE, and I2CEN are ignored when I2CBUSY = 1.

Note 4: If I2CRST = 1, I2CEN can be written when I2CBUSY = 1.

Note 5: Write to I2CACK is ignored if I2RST = 1.

Bit 15: I²C Reset (I2CRST). Setting this bit to 1 aborts the current transaction and resets the I²C controller. This bit is set to 1 by software and is only cleared to 0 by hardware after the reset or when I2CEN = 0.

Bits 14:10: Reserved. Read returns 0.

Bit 9: I²C Clock Stretch Enable (I2CSTREN). Setting this bit to 1 stretches the clock (holds SCL low) at the end of the clock cycle specified in I2CSTRS. Clearing this bit disables clock stretching.

MAXQ Family User's Guide: MAXQ8913 Supplement

Bit 8: I²C General Call Enable (I2CGCEN). Setting this bit to 1 enables the I²C to respond to a general call address (address = 0000 0000). Clearing this bit to 0 prevents the I²C from responding to the general call address.

Bit 7: I²C STOP Enable (I2CSTOP). Setting this bit to 1 generates a STOP condition. This bit automatically is self-cleared to 0 after the STOP condition has been generated.

In master mode, setting this bit can also start the timeout timer if enabled. If the timeout timer expires before the STOP condition can be generated, a timeout interrupt is generated to the CPU if enabled. The I2CSTOP bit is also cleared to 0 by the timeout event.

Note that this bit has no effect when the I²C is operating in slave mode (I2CMST = 0), and is reset to 0 when I2CMST = 0 or I2CEN = 0. Setting the I2CSTOP bit to 1 while I2CSTART = 1 is an invalid operation and is ignored, leaving I2CSTOP bit cleared to 0.

Bit 6: I²C START (I2CSTART). Setting this bit automatically generates a START condition when the bus is free or generates a repeated START condition during a transfer where the I²C module is operating as the master. This bit is automatically self-cleared to 0 after the START condition has been generated. If the I²C START interrupt is enabled, a START condition generates an interrupt to the CPU.

In master mode, setting this bit can also start the timeout timer if enabled. If the timeout timer expires before the START condition can be generated, a timeout interrupt is generated to the CPU if enabled. The I2CSTART bit is also cleared to 0 by the timeout event.

Note that this bit has no effect when the I²C is operating in slave mode (I2CMST = 0) and is reset to 0 when I2CMST = 0 or I2CEN = 0. Also, the I2CSTART and I2CSTOP bits are mutually exclusive. If both bits are set at the same time, it is considered as an invalid operation and the I²C controller ignores the request and resets both bits to 0. Setting the I2CSTART bit to 1 while I2CSTOP = 1 is an invalid operation and is ignored, leaving the I2CSTART bit cleared to 0.

Bit 5: I²C Data Acknowledge Bit (I2CACK). This bit selects the acknowledge bit returned by the I²C controller while acting as a receiver. Setting this bit to 1 generates a NACK (leaving SDA high). Clearing the I2CACK bit to 0 generates an ACK (pulling SDA low) during the acknowledgement cycle. This bit retains its value unless changed by software or hardware. When an I²C abort is in progress (I2CRST = 1), this bit is set to 1 by hardware, and software writes to this bit are ignored when I2CRST = 1.

Bit 4: I²C Clock Stretch Select (I2CSTRS). Setting this bit to 1 enables clock stretching after the falling edge of the 8th clock cycle. Clearing this bit to 0 enables clock stretching after the falling edge of the 9th clock cycle. This bit has no effect when clock stretching is disabled (I2CSTREN = 0).

Bit 3: I²C Extended Address Mode Enable (I2CEA). When this bit is set to 0 (the default), all address transmissions in master mode operate using a 7-bit address, and the slave mode functions match against a 7-bit address. Setting this bit to 1 increases the address range to 10 bits for both master mode transmissions and slave mode address matches.

Bit 2: I²C Transfer Mode (I2CMODE). The transfer mode bit selects the direction of data transfer with respect to the master. When the I2CMODE bit is set to 1, the master is operating in receiver mode (reading from slave). When the I2CMODE bit is cleared to 0, the master is operating in transmitter mode (writing to slave).

Note that software writing to this bit is prohibited in slave mode. When operating in master mode, software configures this bit to the desired direction of data transfer. When operating in slave mode, the direction of data transfer is determined by the $\overline{R/\overline{W}}$ bit received during the address stage, and this bit reflects the actual $\overline{R/\overline{W}}$ bit value in the current transfer and is set by hardware. Software writing to this bit in slave mode is ignored.

Bit 1: I²C Master-Mode Enable (I2CMST). The I2CMST bit functions as a master-mode enable bit for the I²C module. When the I2CMST bit is set to 1, the I²C operates as a master. When the I2CMST is cleared to 0, the I²C module operates in slave mode. This bit is automatically cleared whenever the I²C controller receives a slave address match (I2CAML = 1), loses arbitration (I2CALI = 1), or matches the general call address (I2CGCI = 1).

Bit 0: I²C Enable (I2CEN). This bit enables the I²C function. When set to 1, the I²C communication unit is enabled. When cleared to 0, the I²C function is disabled.

22.1.5 I²C Clock Control Register (I2CCK, M1[0Ch])

Bit #	15	14	13	12	11	10	9	8
Name	I2CCKH							
Reset	0	0	0	0	0	0	1	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	I2CCKL							
Reset	0	0	0	0	0	1	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Note 1: Write to this register is ignored when I2CBUSY = 0.

Note 2: This register has no function in slave mode.

Bits 15:8: I²C Clock High (I2CCKH[7:0]). These bits define the I²C SCL high period in number of system clocks, with bit 7 as the most significant bit. The duration of SCL high time is calculated using the following equation:

$$\text{I}^2\text{C High Time Period} = \text{System Clock} \times (\text{I2CCKH}[7:0] + 1)$$

When operating in master mode, I2CCKH must be set to a minimum value of 2 to ensure proper operation. Any value less than 2 is set to 2.

Bits 7:0: I²C Clock Low (I2CCKL[7:0]). These bits define the I²C SCL low period in number of system clocks, with bit 7 as the most significant bit. The duration of SCL low time is calculated using the following equation:

$$\text{I}^2\text{C Low Time Period} = \text{System Clock} \times (\text{I2CCKL}[7:0] + 1)$$

When operating in master mode, I2CCKL must be set to a minimum value of 4 to ensure proper operation. Any value less than 4 is set to 4.

22.1.6 I²C Timeout Register (I2CTO, M1[0Dh])

Bit #	7	6	5	4	3	2	1	0
Name	I2CTO							
Reset	0	0	0	0	0	0	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 7:0: I²C Timeout Register. This register is used only in master mode. This register determines the number of the I²C bit period (SCL high + SCL low) the I²C master waits for SCL to go high. The timeout timer resets to 0 and starts to count after the I2CSTART bit is set, or every time SCL goes low. When cleared to 00h, the timeout function is disabled and the I²C waits for SCL to go high indefinitely during a transmission. When set to any other values, the I²C waits until the timeout expires and sets the I2CTOI flag.

$$\text{I}^2\text{C Timeout} = \text{I}^2\text{C Bit Rate} \times (\text{I2CTO}[7:0] + 1)$$

Note that these bits have no effect when the I²C module is operating in slave mode (I2CMST = 0). When operating in slave mode, SCL is controlled by an external master.

MAXQ Family User's Guide: MAXQ8913 Supplement

22.1.7 I²C Slave Address Register (I2CSLA, M1[0Eh])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	—	—	I2CSLA	
Reset	0	0	0	0	0	0	1	0
Access	r	r	r	r	r	r	rw	rw

Bit #	7	6	5	4	3	2	1	0
Name	I2CSLA							
Reset	0	0	0	0	0	1	0	0
Access	rw	rw	rw	rw	rw	rw	rw	rw

Bits 15:10: Reserved. Read returns zero.

Bits 9:7: I²C Slave Address Register Extended Bits. When 10-bit addressing mode is enabled, these bits contain the additional address bits (along with bits 6:0) that make up the 10-bit address the I²C engine responds to in slave mode. If 10-bit addressing mode is not enabled, these bits have no function.

Bits 6:0: I²C Slave Address Register. These address bits contain the address of the I²C device. When a match to this address is detected, the I²C controller automatically acknowledges the transmitter with the I2CACK bit value if the I²C module is enabled (I2CEN = 1). The I2CAMI flag is set to 1 and the I2CMST bit is cleared to 0. An interrupt is generated to the CPU if enabled.

22.2 I²C Code Examples

22.2.1 I²C Example 1: Master Mode Transmit

```
; I2C configured as master, transmit to slave address 08h

; Setup for Master Mode Transmit

move I2CCN, #003h      ; I2CEN = 1, I2CMST = 1
call wait_busy         ; Polling routine to wait for I2CBUSY to clear

move I2CCN, #043h      ; I2CEN = 1, I2CMST = 1, I2CMODE = 0, I2CSTART = 1
call wait_start        ; Polling routine to wait for I2CSTART to clear
call wait_busy         ; Polling routine to wait for I2CBUSY to clear

move I2CIE.1, #01h     ; Enable Transmit Complete Interrupt
move I2CBUF, #008h     ; Slave address set to 08h
call wait_tx_complete  ; Wait for transmit interrupt

;; Verify ACK from slave

move ACC, I2CST        ; Move I2C Status Register to accumulator
and #080h              ; Check for NACK bit set in status register
cmp #000h
jump ne, FAIL          ; If NACK bit set, handle retransmission, else continue

move I2CBUF, #0aah     ; Byte to transmit
call wait_tx_complete  ; Wait for transmit interrupt
```

22.2.2 I²C Example 2: Master Mode Receive

```
; I2C configured as master, receive from slave address 08h:

; Setup for Master Mode Receive
move I2CCN, #047h      ; I2CEN = 1, I2CMST = 1, I2CMODE = 1, I2CSTART = 1
call wait_start        ; Polling routine to wait for I2CSTART to clear
call wait_busy         ; Polling routine to wait for I2CBUSY to clear

move I2CIE.2, #01h    ; Enable Receive Ready Interrupt
move I2CBUF, #008h    ; Slave address set to 08h
call wait_tx_complete ; Wait for transmit interrupt
call wait_rxbuf       ; Wait for receive interrupt

;; Byte received in I2CBUF, clear I2C interrupt flag and wait for next interrupt
```

22.2.3 I²C Example 3: Slave Mode Receive

```
; I2C configured as slave with address 1ah

; Setup for Slave Mode Receive
move I2CSLA, #01ah    ; I2C Slave Address = 01ah
move I2CCN, #0001h    ; I2CEN = 1, I2CMST = 0, I2CMODE = 0, I2CSTART = 0
call wait_start        ; Polling routine to wait for I2CSTART to be set,
                       ; indicating a received START

;; Check for address match
move ACC, I2CST
and #0020h            ; Check for Address Match flag set
cmp #0020h
jump ne, no_match     ; If address match bit not set, not for us, else:

move I2CIE.2, #01h    ; Enable Receive Ready Interrupt
call wait_rxbuf       ; Wait for a receive interrupt

;; Byte received in I2CBUF, clear I2C interrupt flag and wait for next interrupt
```

MAXQ Family User's Guide: MAXQ8913 Supplement

22.2.4 I²C Example 4: Slave Mode Receive

```
; I2C configured as slave with address 1ah

; Setup for Slave Mode Receive

move I2CSLA, #01ah      ; I2C Slave Address = 01ah
move I2CCN, #0001h     ; I2CEN = 1, I2CMST = 0, I2CMODE = 0, I2CSTART = 0
call wait_start        ; Polling routine to wait for I2CSTART to be set,
                       ; indicating a received START

;; Check for address match

move ACC, I2CST
and #0020h             ; Check for Address Match flag set
cmp #0020h
jump ne, no_match     ; Not an address match

move ACC, I2CCN        ; Check transfer mode is set
and #004h
cmp #004h
jump ne, not_sl_xmit  ; If transfer mode is low, not a slave transmit, else:

move I2CBUF, #0aah    ; Data byte to be transmitted
call wait_xmit        ; Poll for transmit done

;; Verify ACK received from master

move ACC, I2CST
and #080h
cmp #000h
jump ne, FAIL         ; If NACK bit set, handle retransmission, else continue
```

MAXQ Family User's Guide: MAXQ8913 Supplement

SECTION 23: SUPPLY VOLTAGE MONITOR AND POWER CONTROL (SPECIFIC TO MAXQ8913)

The MAXQ8913 provides a number of features to allow monitoring and control of its on-board power supplies. The supply voltage monitor register (SVM) monitors the DVDD power supply and can alert the processor through an interrupt if DVDD falls below a programmable threshold. Other control bits allow the processor to reduce power consumption and configure various aspects of clock operation.

The MAXQ8913 provides the following power monitoring and control features:

- SVM compares DVDD against a programmable threshold from approximately 2.7V to 3.5V.
- Optional SVM register interrupt can be triggered when DVDD drops below the programmed threshold.
- SVM interrupt can be used to trigger switchback or exit from stop mode.
- Internal 1.8V regulator can optionally be disabled in stop mode to conserve power.
- Brownout detection can optionally be disabled in stop mode to conserve power.

23.1 SVM and Power Control Register Descriptions

The following peripheral registers are used to control the supply voltage monitoring and power control functions. Addresses for all registers are given as “Mx[yy],” where x is the module number (from 0 to 15 decimal) and yy is the register index (from 00h to 1Fh hexadecimal). Fields in the bit definition tables are defined as follows:

- **Name:** Symbolic names of bits or bit fields in this register.
- **Reset:** The value of each bit in this register following a standard reset. If this field reads “unchanged,” the given bit is unaffected by standard reset. If this field reads “s,” the given bit does not have a fixed 0 or 1 reset value because its value is determined by another internal state or external condition.
- **POR:** If present this field defines the value of each bit in this register following a power-on reset (as opposed to a standard reset). Some bits are unaffected by standard resets and are set/cleared by POR only.
- **Access:** Bits can be read-only (r) or read/write (rw). Any special restrictions or conditions that could apply when reading or writing this bit are detailed in the bit description.

23.1.1 Power Control Register (PWCN, M0[0Ch])

Bit #	7	6	5	4	3	2	1	0
Name	BOD	REGEN	—	—	—	—	—	HFXD
Reset	No effect	No effect	No effect	No effect	No effect	No effect	No effect	No effect
POR	0	0	0	0	0	0	0	0
Access	rw	rw	r	r	r	r	r	rw*

*HFXD can only be written to 1 when both RGSL = 1 and RGMD = 1 (running from ring oscillator).

Bit 7: Brownout-Detection Disable (BOD). This bit controls whether the brownout detection is enabled during stop mode only and only when the internal 1.8V regulator is also disabled (REGEN = 0).

0 = Brownout detection is disabled during stop mode, if REGEN = 0.

1 = Brownout detection is enabled during stop mode.

Bit 6: Regulator Enable (REGEN). This bit controls whether the internal 1.8V regulator is enabled during stop mode only. (The internal regulator is always enabled outside of stop mode.)

0 = The internal regulator is disabled during stop mode to conserve power.

1 = The internal regulator is enabled during stop mode.

Bits 5:1: Reserved. Read returns 0.

MAXQ Family User's Guide: MAXQ8913 Supplement

Bit 0: High-Frequency Crystal Oscillator Disable (HFXD). Setting this bit to 1 disables the high-frequency crystal oscillator on the MAXQ8913. However, a high-frequency external clock can still be provided at HFXIN. Clearing this bit to 0 enables the high-frequency crystal oscillator.

23.1.2 Supply Voltage Monitor Register (SVM, M0[06h])

Bit #	15	14	13	12	11	10	9	8
Name	—	—	—	—	SVTH3	SVTH2	SVTH1	SVTH0
Reset	0	0	0	0	0	1	1	1
Access	r	r	r	r	rw*	rw*	rw*	rw*

Bit #	7	6	5	4	3	2	1	0
Name	—	—	—	SVMSTOP	SVMI	SVMIE	SVMRDY	SVMEN
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	rw	rw	rw	r	rw

*SVTH[3:0] can only be written when the SVM is not running (SVMEN = 0).

Bits 15:12, 7:5: Reserved

Bits 11:8: Supply Voltage Monitor Threshold (SVTH[3:0]). These bits select the programmable DVDD threshold setting for the SVM. The level can be adjusted from approximately 2.0V to 3.5V in steps of 0.1V, and is given by:

$$\text{SVM threshold} = 2.0\text{V} + (\text{SVTH}[3:0] \times 0.1\text{V})$$

Note: Settings below 2.7V have no effect and should not be used.

The default setting (SVTH[3:0] = 0111b) is 2.7V. Note that these bits can only be changed when SVMEN = 0.

Bit 4: Supply Voltage Monitor Stop-Mode Enable (SVMSTOP). This bit controls the operation of the SVM in stop mode.

0 = The SVM is disabled during stop mode.

1 = The SVM is enabled during stop mode (if SVMEN = 1).

Bit 3: Supply Voltage Monitor Interrupt Flag (SVMI). This bit is set to 1 by hardware when the DVDD supply is below the threshold voltage set by SVTH[3:0]. If SVMIE = 1, setting this bit to 1 by either hardware or software triggers an interrupt. This bit must be cleared by software, but if DVDD is still below the threshold, the bit immediately is set by hardware again.

Bit 2: Supply Voltage Monitor Interrupt Enable (SVMIE). Setting this bit to 1 allows an interrupt to be generated (if not otherwise masked) when SVMI is set to 1. Clearing this bit to 0 disables the SVM interrupt.

Bit 1: Supply Voltage Monitor Ready (SVMRDY). This read-only status bit indicates whether the SVM is ready for use.

0 = The SVM is disabled (SVMEN = 0), stop mode was entered with SVMSTOP = 0, or the SVM is in the process of warming up.

1 = The SVM is enabled and ready for use.

Bit 0: Supply Voltage Monitor Enable (SVMEN). Setting this bit to 1 enables the SVM and begins monitoring DVDD against the programmed (SVTH[3:0]) threshold. Clearing this bit to 0 disables the SVM.

MAXQ Family User's Guide: MAXQ8913 Supplement

SECTION 24: INPUT/OUTPUT AMPLIFIERS (SPECIFIC TO MAXQ8913)

24.1 Input/Output Amplifier Overview

The MAXQ8913 controls two amplifier stages as described in this section. The input amplifier stage consists of four internal, uncommitted op amps that can be used to amplify and/or filter the inputs to the ADC. The output stage consists of an external Class D amplifier that can be used to postprocess the outputs from DAC1 and DAC2.

Table 24-1. Amplifier Control Registers

REGISTER	ADDRESS	FUNCTION
AMPCN	M3[06h]	Amplifier Control Register. Controls the external Class-D amplifier.
OPMCN	M3[0Ah]	Op Amp Control Register. Enables/disables the internal uncommitted op amps.

24.2 Amplifier Control Register Descriptions

Addresses of registers are given as “Mx[yy],” where x is the module number (from 0 to 15 decimal) and yy is the register index (from 00h to 1Fh hexadecimal). Fields in the bit definition tables are defined as follows:

- **Name:** Symbolic names of bits or bit fields in this register.
- **Reset:** The value of each bit in this register following a standard reset. If this field reads “unchanged,” the given bit is unaffected by standard reset. If this field reads “s,” the given bit does not have a fixed 0 or 1 reset value because its value is determined by another internal state or external condition.
- **POR:** If present this field defines the value of each bit in this register following a power-on reset (as opposed to a standard reset). Some bits are unaffected by standard resets and are set/cleared by POR only.
- **Access:** Bits can be read-only (r) or read/write (rw). Any special restrictions or conditions that could apply when reading or writing this bit are detailed in the bit description.

24.2.1 Amplifier Control Register (AMPCN, M3[06h])

Bit #	7	6	5	4	3	2	1	0
Name	—	AMPFLT	AMPIF	AMPIE	AMPCK1	AMPCK0	AMPEN1	AMPEN0
Reset	0	1	0	0	0	0	0	0
Access	r	r	rw	rw	rw	rw	rw	rw

Bit 7: Reserved. All reads return 0.

Bit 6: Amplifier Fault Flag (AMPFLT). This bit is set to 0 by hardware when the $\overline{\text{FAULT}}$ input from the external D-amplifier goes low. It is set to 1 by hardware whenever $\overline{\text{FAULT}}$ is high, or when both amplifiers are disabled (AMPEN0 = 0 and AMPEN1 = 0), or when the amplifier clock is disabled (AMPCK[1:0] == 00b).

Bit 5: Amplifier Interrupt Flag (AMPIF). If this bit is set to 1, an interrupt is generated if AMPIE = 1. This bit is set to 1 by hardware when the AMPFLT flag changes from 1 to 0. This bit must be cleared by software once set.

Bit 4: Amplifier Interrupt Enable (AMPIE). If this bit is set to 1, an interrupt is generated when AMPIF = 1. If this bit is set to 0, amplifier interrupts are disabled.

Bits 3:2: Amplifier Clock Divide (AMPCK[1:0]). These two bits control the D-amplifier clock output that is generated on the SYNCIN pin. This clock is generated from a divided-down version of the system clock as follows:

AMPCK1 = 0, AMPCK0 = 0: No output clock (disabled).

AMPCK1 = 0, AMPCK0 = 1: SYNCIN clock = System clock divided by 2.

AMPCK1 = 1, AMPCK0 = 0: SYNCIN clock = System clock divided by 3.

AMPCK1 = 1, AMPCK0 = 1: SYNCIN clock = System clock divided by 4.

MAXQ Family User's Guide: MAXQ8913 Supplement

Bit 1: Amplifier 1 Enable (AMPEN1). This bit controls the active-high SHDNR output signal for the right-side motor driver as follows:

0 = Disabled. SHDNR output is driven high.

1 = Enabled. SHDNR output is driven low to enable motor driver.

Bit 0: Amplifier 0 Enable (AMPEN0). This bit controls the active-high SHDNL output signal for the left-side motor driver as follows:

0 = Disabled. SHDNL output is driven high.

1 = Enabled. SHDNL output is driven low to enable motor driver.

24.2.2 Op Amp Control Register (OPMCN, M3[0Ah])

Bit #	7	6	5	4	3	2	1	0
Name	—	—	—	—	OPMEND	OPMENC	OPMENB	OPMENA
Reset	0	0	0	0	0	0	0	0
Access	r	r	r	r	rw	rw	rw	rw

Bits 7:4: Reserved. All reads return 0.

Bit 3: Op Amp D Enable (OPMEND). Setting this bit to 1 enables op amp D; setting this bit to 0 disables it. The OUTD pin can still be used as an input to AIN5 even when the op amp is turned off.

Bit 2: Op Amp C Enable (OPMENC). Setting this bit to 1 enables op amp C; setting this bit to 0 disables it. The OUTC pin can still be used as an input to AIN4 even when the op amp is turned off.

Bit 1: Op Amp B Enable (OPMENB). Setting this bit to 1 enables op amp B; setting this bit to 0 disables it. The OUTB pin can still be used as an input to AIN3 even when the op amp is turned off.

Bit 0: Op Amp A Enable (OPMENA). Setting this bit to 1 enables op amp A; setting this bit to 0 disables it. The OUTA pin can still be used as an input to AIN2 even when the op amp is turned off.

MAXQ Family User's Guide: MAXQ8913 Supplement

SECTION 25: UTILITY ROM (SPECIFIC TO MAXQ8913)

25.1 Overview

The MAXQ8913 utility ROM includes routines that provide the following functions to application software:

- In-application programming routines for flash memory (program, erase, mass erase)
- Single word/byte copy and buffer copy routines for use with lookup tables

To provide backwards compatibility among different versions of the utility ROM, a function address table is included that contains the entry points for all user-callable functions. With this table, user code can determine the entry point for a given function as follows:

1. Read the location of the function address table from address 0800Dh in the utility ROM.
2. The entry points for each function listed below are contained in the function address table, one word per function, in the order given by their function numbers.

For example, the entry point for the **UROM_flashEraseAll** function can be determined by the following procedure.

1. `functionTable = dataMemory[0800Dh]`
2. `flashWriteEntry = dataMemory[functionTable + 2]`

It is also possible to call utility ROM functions directly, using the entry points given above. Standard include files are provided for this purpose with the MAXQ development toolset; also see Appendix 1. This method calls functions more quickly, but the application may need to be recompiled in order to run properly with a different version of the utility ROM.

Table 25-1. Functions for MAXQ8913 Utility ROM

INDEX	FUNCTION NAME	ENTRY POINT	SUMMARY
0	UROM_flashWrite	8788h	Programs a single word of flash memory.
1	UROM_flashErasePage	87ABh	Erases (programs to FFFFh) a 512-word sector of flash memory.
2	UROM_flashEraseAll	87C1h	Erases (programs to FFFFh) all flash memory.
3	UROM_moveDP0	87D0h	Reads a byte/word at DP[0].
4	UROM_moveDP0inc	87D3h	Reads a byte/word at DP[0], then increments DP[0].
5	UROM_moveDP0dec	87D6h	Reads a byte/word at DP[0], then decrements DP[0].
6	UROM_moveDP1	87D9h	Reads a byte/word at DP[1].
7	UROM_moveDP1inc	87DCh	Reads a byte/word at DP[1], then increments DP[0].
8	UROM_moveDP1dec	87DFh	Reads a byte/word at DP[1], then decrements DP[0].
9	UROM_moveBP	87E2h	Reads a byte/word at BP[OFFS].
10	UROM_moveBPinc	87E5h	Reads a byte/word at BP[OFFS], then increments OFFS.
11	UROM_moveBPdec	87E8h	Reads a byte/word at BP[OFFS], then decrements OFFS.
12	UROM_copyBuffer	87EBh	Copies LC[0] values (up to 256) from DP[0] to BP[OFFS].

25.2 In-Application Programming Functions

25.2.1 UROM_flashWrite

Function: UROM_flashWrite
Summary: Programs a single word of flash memory.
Inputs: A[0]: Word address in program flash memory to write to.
A[1]: Word value to write to flash memory.
Outputs: Carry: Set on error and cleared on success.
Destroys: PSF, LC[1]

Notes:

- This function uses one stack level to save and restore values.
- If the watchdog reset function is active, it should be disabled before calling this function.
- If the flash location has already been programmed to a non-FFFF value, this function returns with an error (Carry set). In order to reprogram a flash location, it must first be erased by calling **UROM_flashErasePage** or **UROM_flashEraseAll**.

25.2.2 UROM_flashErasePage

Function: UROM_flashErasePage
Summary: Erases (programs to 0FFFFh) a 512-word page of flash memory.
Inputs: A[0]: Word address located in the page to be erased. (The page number is the high 9 bits of A[0].)
Outputs: Carry: Set on error and cleared on success.
Destroys: LC[1], A[0]

Notes:

- If the watchdog reset function is active, it should be disabled before calling this function.
- When calling this function from flash, care should be taken that the return address is not in the page which is being erased.

25.2.3 UROM_flashEraseAll

Function: UROM_flashEraseAll
Summary: Erases (programs to 0FFFFh) all locations in flash memory.
Inputs: None.
Outputs: Carry: Set on error and cleared on success.
Destroys: LC[1], A[0]

Notes:

- If the watchdog reset function is active, it should be disabled before calling this function.
- This function can only be called by code running from the RAM. Attempting to call this function while running from the flash results in an error.

MAXQ Family User's Guide: MAXQ8913 Supplement

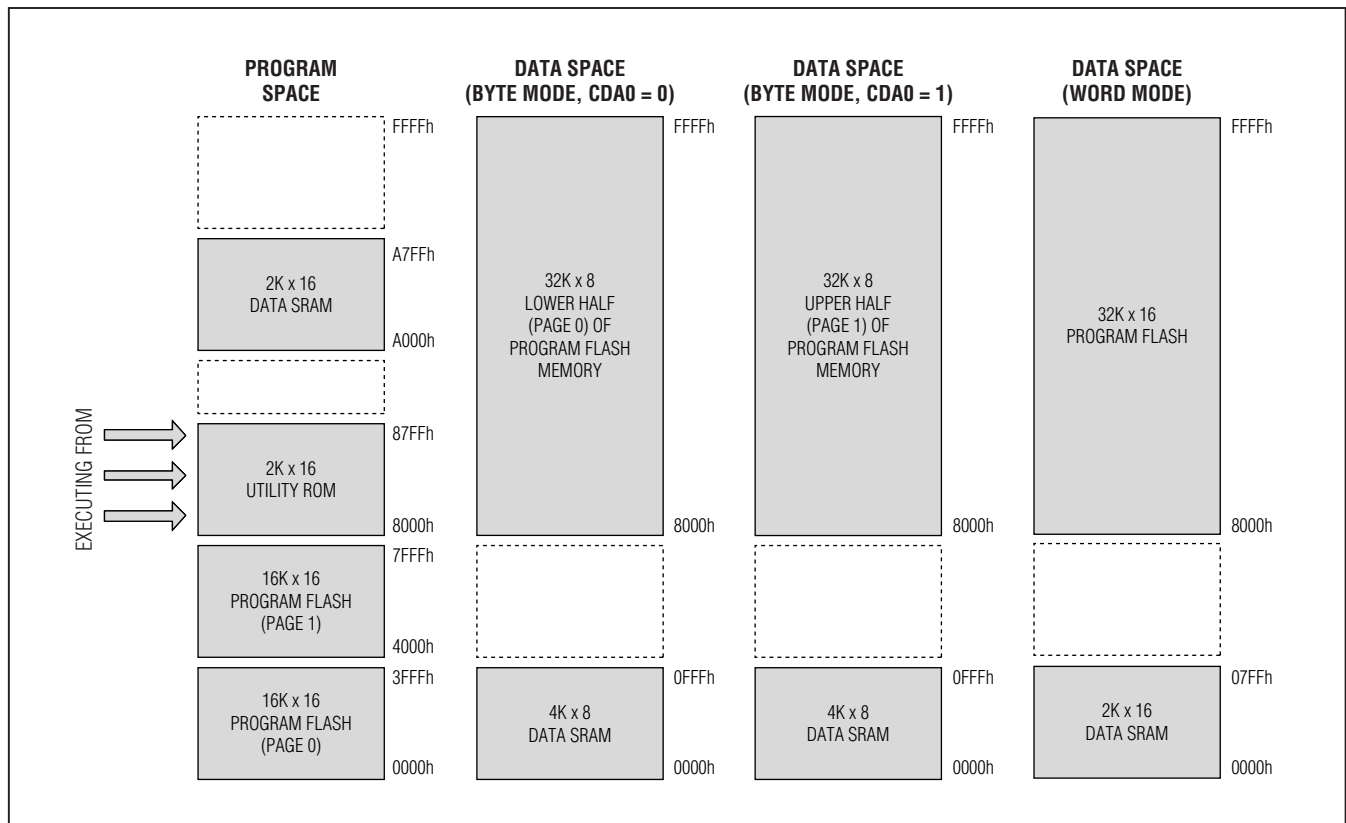


Figure 25-1. Memory Map When Executing from Utility ROM

25.3 Data Transfer Functions

The following utility ROM functions are used to transfer data from one memory segment to another. They fall into two categories:

- The **UROM_moveDP0<inc/dec>**, **UROM_moveDP1<inc/dec>**, and **UROM_moveBP<inc/dec>** functions are used to read data that exists in the currently executing segment of program memory. For example, code running from the program flash can use these functions to read data (such as constant string or array data or lookup tables) that is also stored in the program flash. These functions can also be used by code running from the RAM to read data stored in the RAM.
- The **UROM_copyBuffer** function is used to copy a block of data from the program flash or data RAM to a specified location in the data RAM. This is often used to copy code into the data RAM before executing it.

Since these functions are executed from the utility ROM, addresses must be specified correctly to point to memory spaces as seen when executing from the utility ROM. This is illustrated in Figure 25-1.

For example, data located at word address 0100h in the program flash must be accessed at word address 8100h (or byte address 8200h) when using any of the functions listed in the following sections.

25.3.1 UROM_moveDP0

Function: UROM_moveDP0
Summary: Reads the byte/word value pointed to by DP[0].
Inputs: DP[0]: Address to read from.
Outputs: GR: Data byte/word read.
Destroys: None.

Notes:

- Before calling this function, DPC should be set appropriately to configure DP[0] for byte or word mode.
- The address passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointer before reading the byte/word value.

25.3.2 UROM_moveDP0inc

Function: UROM_moveDP0inc
Summary: Reads the byte/word value pointed to by DP[0], then increments DP[0].
Inputs: DP[0]: Address to read from.
Outputs: GR: Data byte/word read.
DP[0] is incremented.
Destroys: None.

Notes:

- Before calling this function, DPC should be set appropriately to configure DP[0] for byte or word mode.
- The address passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointer before reading the byte/word value.

25.3.3 UROM_moveDP0dec

Function: UROM_moveDP0dec
Summary: Reads the byte/word value pointed to by DP[0], then decrements DP[0].
Inputs: DP[0]: Address to read from.
Outputs: GR: Data byte/word read.
DP[0] is decremented.
Destroys: None.

Notes:

- Before calling this function, DPC should be set appropriately to configure DP[0] for byte or word mode.
- The address passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointer before reading the byte/word value.

MAXQ Family User's Guide: MAXQ8913 Supplement

25.3.4 UROM_moveDP1

Function: UROM_moveDP1
Summary: Reads the byte/word value pointed to by DP[1].
Inputs: DP[1]: Address to read from.
Outputs: GR: Data byte/word read.
Destroys: None.

Notes:

- Before calling this function, DPC should be set appropriately to configure DP[1] for byte or word mode.
- The address passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointer before reading the byte/word value.

25.3.5 UROM_moveDP1inc

Function: UROM_moveDP1inc
Summary: Reads the byte/word value pointed to by DP[1], then increments DP[1].
Inputs: DP[1]: Address to read from.
Outputs: GR: Data byte/word read.
DP[1] is incremented.
Destroys: None.

Notes:

- Before calling this function, DPC should be set appropriately to configure DP[1] for byte or word mode.
- The address passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointer before reading the byte/word value.

25.3.6 UROM_moveDP1dec

Function: UROM_moveDP1dec
Summary: Reads the byte/word value pointed to by DP[1], then decrements DP[1].
Inputs: DP[1]: Address to read from.
Outputs: GR: Data byte/word read.
DP[1] is decremented.
Destroys: None.

Notes:

- Before calling this function, DPC should be set appropriately to configure DP[1] for byte or word mode.
- The address passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointer before reading the byte/word value.

25.3.7 UROM_moveBP

Function: UROM_moveBP
Summary: Reads the byte/word value pointed to by BP[OFFS].
Inputs: BP[OFFS]: Address to read from.
Outputs: GR: Data byte/word read.
Destroys: None.

Notes:

- Before calling this function, DPC should be set appropriately to configure BP[OFFS] for byte or word mode.
- The address passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointer before reading the byte/word value.

25.3.8 UROM_moveBPinc

Function: UROM_moveBPinc
Summary: Reads the byte/word value pointed to by BP[OFFS], then increments OFFS.
Inputs: DP[1]: Address to read from.
Outputs: GR: Data byte/word read.
OFFS is incremented.
Destroys: None.

Notes:

- Before calling this function, DPC should be set appropriately to configure BP[OFFS] for byte or word mode.
- The address passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointer before reading the byte/word value.

25.3.9 UROM_moveBPdec

Function: UROM_moveBPdec
Summary: Reads the byte/word value pointed to by BP[OFFS], then decrements OFFS.
Inputs: DP[1]: Address to read from.
Outputs: GR: Data byte/word read.
OFFS is decremented.
Destroys: None.

Notes:

- Before calling this function, DPC should be set appropriately to configure BP[OFFS] for byte or word mode.
- The address passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointer before reading the byte/word value.

MAXQ Family User's Guide: MAXQ8913 Supplement

25.3.10 UROM_copyBuffer

Function: UROM_copyBuffer

Summary: Copies LC[0] bytes/words (up to 256) from DP[0] to BP[OFFS].

Inputs:

DP[0]: Address to copy from.
BP[OFFS]: Address to copy to.
LC[0]: Number of bytes or words to copy.

Outputs:

OFFS is incremented by LC[0].
DP[0] is incremented by LC[0].

Destroys:

LC[0]

Notes:

- This function can be used to copy from program flash to data RAM, or from one part of data RAM to another. It cannot be used to copy data into flash memory, however; the **UROM_writeFlash** function should be used for this purpose.
- Before calling this function, DPC should be set appropriately to configure DP[0] and BP[OFFS] for byte or word mode. Both DP[0] and BP[OFFS] should be configured to the same mode (byte or word) for correct buffer copying.
- The addresses passed to this function should be based on the data memory mapping for the utility ROM, as shown in Figure 25-1. When a byte mode address is used, CDA0 must be set appropriately to access either the upper or lower half of program flash/ROM memory.
- This function automatically refreshes the data pointers before reading the byte/word values.

25.4 Utility ROM Examples

25.4.1 Utility ROM Example 1: Reading Constant Word Data from Flash

```
move    DPC, #1Ch                ; Set all pointers to word mode
move    DP[0], #(table + 8000h)   ; Point to address of data as viewed
                                           ; in the Utility ROM memory map

lcall   #UROM_moveDP0inc
move    A[0], GR                  ; A[0] = 1111h
lcall   #UROM_moveDP0inc
move    A[1], GR                  ; A[1] = 2222h
lcall   #UROM_moveDP0inc
move    A[2], GR                  ; A[2] = 3333h
lcall   #UROM_moveDP0inc
move    A[3], GR                  ; A[3] = 4444h

sjump  $

org 0100h

table:
dw      1111h, 2222h, 3333h, 4444h
```

25.4.2 Utility ROM Example 2: Reading Constant Byte Data from Flash (Indirect Function Call)

```
move   DPC, #1Ch                ; Set all pointers to word mode
move   DP[0], #800Dh            ; Fetch location of function table from Utility ROM
move   BP, @DP[0]              ; Set base pointer to function table location
move   Offs, #INDX_moveDP0inc   ; Set offset to moveDP0inc entry in table
move   A[7], @BP[Offs]         ; Get address of moveDP0inc function

move   DPC, #00h               ; Set all pointers to byte mode

move   DP[0], #((table * 2) + 8000h); Point to address of data as viewed
                                           ; in the Utility ROM memory map
                                           ; and convert to byte mode pointer

lcall  A[7]                    ; moveDP0inc
move   A[0], GR                ; A[0] = 34h
lcall  A[7]                    ; moveDP0inc
move   A[1], GR                ; A[1] = 12h
lcall  A[7]                    ; moveDP0inc
move   A[2], GR                ; A[2] = 78h
lcall  A[7]                    ; moveDP0inc
move   A[3], GR                ; A[3] = 56h

sjump  $

org 0100h

table:
dw     1234h, 5678h
```

MAXQ Family User's Guide: MAXQ8913 Supplement

APPENDIX 1: MAXQ8913 DEVICE INCLUDE FILE FOR MAX-IDE

```
=====
;=
;= Copyright (C) 2009 Maxim Integrated Products.
;= All rights Reserved. Printed in U.S.A.
;=
;= Permission is hereby granted, free of charge, to any person obtaining a
;= copy of this software and associated documentation files (the
;= "Software"), to deal in the Software without restriction, including
;= without limitation the rights to use, copy, modify, merge, publish,
;= distribute, sublicense, and/or sell copies of the Software, and to
;= permit persons to whom the Software is furnished to do so, subject to
;= the following conditions:
;=
;= The above copyright notice and this permission notice shall be included
;= in all copies or substantial portions of the Software source code.
;=
;= THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
;= OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
;= MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT.
;= IN NO EVENT SHALL MAXIM INTEGRATED PRODUCTS BE LIABLE FOR ANY CLAIM,
;= DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
;= OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR
;= THE USE OR OTHER DEALINGS IN THE SOFTWARE.
;=
;= Except as contained in this notice, the name of Maxim Integrated Products
;= shall not be used except as stated in the Maxim Branding Policy.
;=
;= Description: Peripheral register definitions and Utility ROM function
;= entry points for the MAXQ8913 microcontroller.
;=
;= Filename: maxQ8913.inc
;=
;= Compiler: Macro / MaxQasm
;=
;= Hardware: Any
;=
=====
```

MAXQ Family User's Guide: MAXQ8913 Supplement

```
=====
;= Module 0 =
=====
#define PO0    M0[0]    ; Port 0 Output
#define PO1    M0[1]    ; Port 1 Output
#define EIF0   M0[2]    ; External Interrupt Flag 0
#define EIE0   M0[3]    ; External Interrupt Enable 0
#define EIF1   M0[4]    ; External Interrupt Flag 1
#define EIE1   M0[5]    ; External Interrupt Enable 1
#define SVM    M0[6]    ; Supply Voltage Monitor
; Reserved    *M0[7]
#define PI0    M0[8]    ; Port 0 Input
#define PI1    M0[9]    ; Port 1 Input
#define EIES0  M0[10]   ; External Interrupt Edge Select 0
#define EIES1  M0[11]   ; External Interrupt Edge Select 1
#define PWCN   M0[12]   ; Power Control Register
#define PID0   M0[13]   ; Port 0 Input Disable
; Reserved    *M0[14]
; Reserved    *M0[15]
#define PD0    M0[16]   ; Port 0 Direction
#define PD1    M0[17]   ; Port 1 Direction
; Reserved    *M0[18]
; Reserved    *M0[19]
; Reserved    *M0[20]
; Reserved    *M0[21]
; Reserved    *M0[22]
; Reserved    *M0[23]
; Reserved    *M0[24]
; Reserved    *M0[25]
; Reserved    *M0[26]
; Reserved    *M0[27]
; Reserved    *M0[28]
; Reserved    *M0[29]
; Reserved    *M0[30]
; Reserved    *M0[31]
=====
;= Module 1 =
=====
#define SCON   M1[0]    ; Serial Port Control
#define SBUF   M1[1]    ; Serial Port Buffer
```

MAXQ Family User's Guide: MAXQ8913 Supplement

```
#define SPICN M1[2] ; SPI Control
#define SPIB M1[3] ; SPI Data Buffer
#define I2CCN M1[4] ; I2C Control
#define I2CST M1[5] ; I2C Status
#define I2CBUF M1[6] ; I2C Buffer
#define I2CIE M1[7] ; I2C Interrupt Enable
#define SMD M1[8] ; Serial Port Mode
#define PR M1[9] ; Serial Port Phase (Baud Rate)
#define SPICF M1[10] ; SPI Configuration
#define SPICK M1[11] ; SPI Clock
#define I2CCK M1[12] ; I2C Clock
#define I2CTO M1[13] ; I2C Timeout
#define I2CSLA M1[14] ; I2C Slave Address
; Reserved *M1[15]
; Reserved *M1[16]
; Reserved *M1[17]
; Reserved *M1[18]
; Reserved *M1[19]
; Reserved *M1[20]
; Reserved *M1[21]
; Reserved *M1[22]
; Reserved *M1[23]
; Reserved *M1[24]
; Reserved *M1[25]
; Reserved *M1[26]
; Reserved *M1[27]
; Reserved *M1[28]
; Reserved *M1[29]
; Reserved *M1[30]
; Reserved *M1[31]
;=====
;= Module 2 =
;=====
#define MCNT M2[0] ; Multiplier Control
#define MA M2[1] ; Multiplier Operand A
#define MB M2[2] ; Multiplier Operand B
#define MC2 M2[3] ; Multiply-Accumulate Register 2
#define MC1 M2[4] ; Multiply-Accumulate Register 2
#define MC0 M2[5] ; Multiply-Accumulate Register 2
#define TBCN M2[6] ; Timer B Control
```

MAXQ Family User's Guide: MAXQ8913 Supplement

```
#define TBR M2[7] ; Timer B Capture/Reload
#define MC1R M2[8] ; Multiplier Read Register 1
#define MC0R M2[9] ; Multiplier Read Register 0
#define TBV M2[10] ; Timer B Value
#define TBC M2[11] ; Timer B Compare
; Reserved *M2[12]
; Reserved *M2[13]
; Reserved *M2[14]
; Reserved *M2[15]
; Reserved *M2[16]
; Reserved *M2[17]
; Reserved *M2[18]
; Reserved *M2[19]
; Reserved *M2[20]
; Reserved *M2[21]
; Reserved *M2[22]
; Reserved *M2[23]
; Reserved *M2[24]
; Reserved *M2[25]
; Reserved *M2[26]
; Reserved *M2[27]
; Reserved *M2[28]
; Reserved *M2[29]
; Reserved *M2[30]
; Reserved *M2[31]

;=====
;= Module 3 =
;=====

#define ADST M3[0] ; ADC Status
#define ADADDR M3[1] ; ADC Address
#define DAC1OUT M3[2] ; DAC 1 Output Data
#define DAC2OUT M3[3] ; DAC 2 Output Data
#define DAC3OUT M3[4] ; DAC 3 Output Data
#define DAC4OUT M3[5] ; DAC 4 Output Data
#define AMPCN M3[6] ; Amplifier Control
#define ISINKCN M3[7] ; Current Sink Control
#define ADCN M3[8] ; ADC Control
#define ADDATA M3[9] ; ADC Data Read/Write
#define OPMCN M3[10] ; Op Amp Control
#define DACEN M3[11] ; DAC Enable
#define TEMPEN M3[12] ; Temperature Sensor Enable
```

MAXQ Family User's Guide: MAXQ8913 Supplement

```
; Reserved      *M3[13]
; Reserved      *M3[14]
; Reserved      *M3[15]
; Reserved      *M3[16]
; Reserved      *M3[17]
; Reserved      *M3[18]
; Reserved      *M3[19]
; Reserved      *M3[20]
; Reserved      *M3[21]
; Reserved      *M3[22]
; Reserved      *M3[23]
; Reserved      *M3[24]
; Reserved      *M3[25]
; Reserved      *M3[26]
; Reserved      *M3[27]
; Reserved      *M3[28]
; Reserved      *M3[29]
; Reserved      *M3[30]
; Reserved      *M3[31]

;=====
;= Utility ROM Functions =
;=====
; MAXQ891X A1/A2 Utility ROM
;
; Banner: "MAXQ891X Loader 1.10 02-26-2008"
#define  INDX_flashWrite 0
#define  INDX_flashErasePage 1
#define  INDX_flashEraseAll 2
#define  INDX_moveDP0 3
#define  INDX_moveDP0inc 4
#define  INDX_moveDP0dec 5
#define  INDX_moveDP1 6
#define  INDX_moveDP1inc 7
#define  INDX_moveDP1dec 8
#define  INDX_moveBP 9
#define  INDX_moveBPinc 10
#define  INDX_moveBPdec 11
#define  INDX_copyBuffer 12

#define  UROM_flashWrite #08788h
#define  UROM_flashErasePage #087ABh
```


MAXQ Family User's Guide: MAXQ8913 Supplement

```
#define UROM_flashEraseAll #087C1h
#define UROM_moveDP0 #087D0h
#define UROM_moveDP0inc #087D3h
#define UROM_moveDP0dec #087D6h
#define UROM_moveDP1 #087D9h
#define UROM_moveDP1inc #087DCh
#define UROM_moveDP1dec #087DFh
#define UROM_moveBP #087E2h
#define UROM_moveBPinc #087E5h
#define UROM_moveBPdec #087E8h
#define UROM_copyBuffer #087EBh
```

MAXQ Family User's Guide: MAXQ8913 Supplement

REVISION HISTORY

REVISION NUMBER	REVISION DATE	SECTION NUMBER	DESCRIPTION	PAGES CHANGED
0	8/09	—	Initial release.	—



Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time. The parametric values (min and max limits) shown in the Electrical Characteristics table are guaranteed. Other parametric values quoted in this data sheet are provided for guidance.