ON Semiconductor®

AND9349/D

AX8052 Programming Manual

Revision 2

Table of Contents

1. Device Overview

AX8052 is a fully integrated embedded microcontroller optimized for short range radio applications, designed to be paired with Axsem RF technology.

1.1. Features

- AX8052
    - Industry standard 8052 instruction set
    - High performance core, most instructions require only 1 clock per instruction byte
    - 20MIPS
    - Dual DPTR for high speed external memory copies
    - 22 interrupt vectors
- Debugger
    - 3 Wire (1 dedicated, 2 shared with GPIO Pins) debugger interface
    - True hardware debugger with breakpoints and single stepping support
    - User programmable 64bit key to restrict debugging to authorized personnel
    - DebugLink interface allows "printf" style debugging without utilizing a UART or GPIO pins
- Memory
    - In-Circuit programmable FLASH
    - Large RAM
    - High performance memory crossbar
    - 4 Word ×16 Bits fully associative cache and prefetch unit to hide FLASH access latency
- Clocking
    - Flexible clocking options thanks to an on-chip 20MHz fast RC oscillator, 10kHz/640Hz low power RC oscillator, fast crystal oscillator, low power tuning fork crystal oscillator
    - Fully automatic calibration of on-chip RC oscillators to a reference clock
    - Clock monitor can detect failures of the main clock and switch to the on-chip fast RC oscillator
    - Watchdog
- Power Modes
    - Standby, sleep and deep sleep power modes for very low idle power consumption
    - On-chip power-on reset and brown out detection
- 16Bit Wakeup Timer
    - 2 Counting registers
    - 4 Event registers allow flexible wakeup and software schedules

- Dedicated Interface to Axsem Transceiver IC
    - Easy access to transceiver registers by mapping transceiver registers into X address space
    - Transceiver crystal may clock MCU
- GPIO
    - Up to 24 GPIO pins, depending on package
    - PB0-PB7 and PC0-PC7 5V tolerant inputs
    - All GPIO pins support individually programmable pull-ups and interrupt on change
    - Flexible allocation of GPIO pins to peripherals
- 16Bit General Purpose Timer (3x)
    - Sawtooth and triangle modes
    - Sigma-Delta mode converts timer into a DAC
    - Optional double buffering of the PERIOD register allows controlled frequency changes
    - Optional high-byte buffering allows atomic 16bit accesses
    - Flexible clocking options, can use any internal or an external clock source, prescaler included
- 16Bit Output Compare Unit (2x)
    - Used together with a General Purpose Timer to create PWM waveforms
    - Optional double buffering
- 16Bit Input Capture Unit (2x)
    - Used together with a General Purpose Timer to time events on an external or internal signal
- UART (2x)
    - 5-9bit word length, 1-2 stop bits
    - Uses one of the General Purpose Timers as baud rate generator
- Master/Slave SPI
    - 4 or 3 line mode (with or without slave select line)
    - Programmable clock phases
- ADC
    - 10 Bit 500kSamples/s ADC
    - Up to 8 channels
    - Single ended and differential sampling
    - x0.1, x1 and x10 gain amplifier
    - Internal 1V reference
    - Flexibly programmable conversion schedule
    - Built-in temperature sensor
- Analog Comparators
    - Internal or external reference
    - Output signal may be routed to GPIO, read by software, or used as Input Capture trigger
- DMA controller
    - 2 independent DMA channels
    - Moves data between X-RAM and most on-chip peripherals

    - Cycle-steal and round-robin memory arbitration ensure minimal impact on the AX8052 core
    - Chained buffer descriptors allow arbitrarily elaborate buffering schemes and flexible interrupt generation
- Advanced Encryption Standard (AES)[1]
    - Dedicated AES crypto controller
    - Dedicated DMA engine to fetch input data and keystream from X RAM and store output data into X RAM
    - Multi Megabit/s data rates
    - Supports AES-128, AES-192 and AES-256 international standards
    - Programmable round number and software key schedule generation allow longer key lengths for higher security applications
    - ECB, CFB and OFB chaining modes
- True Random Number Generator (RNG)[1]
    - Crypographic random numbers
    - Passes the NIST Statistical Test Suite for Random Number Generators

1 For further information, contact exportcontrol@axsem.com

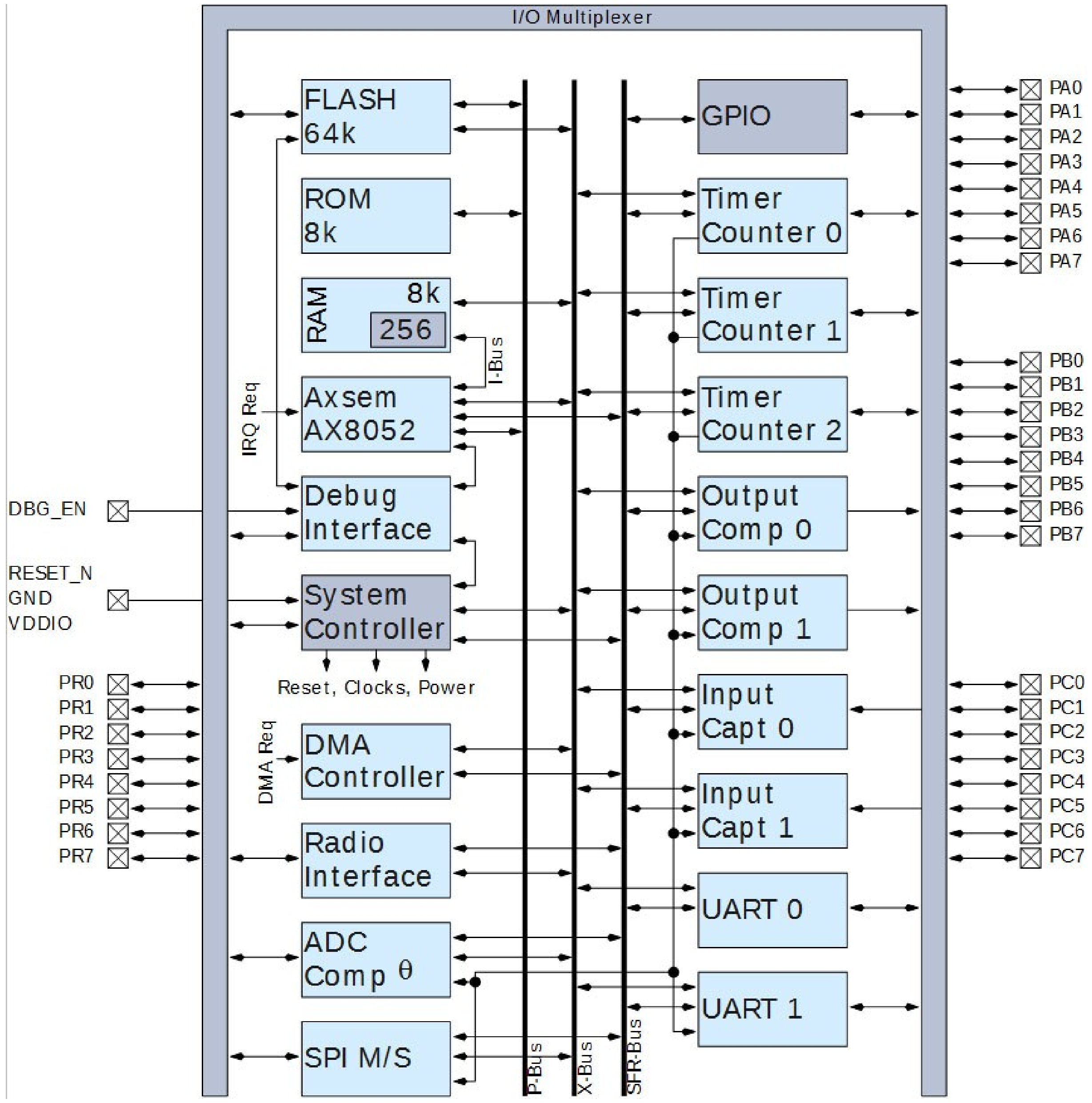


Figure 1: Microfoot Block Diagram

Figure 1 shows the block diagram of the Microfoot device. All blocks are interconnected via three busses, the P, X and SFR bus.

2. AX8052

The AX8052 core is fully compatible with the MCS-51 instruction set. Standard 803x/805x assembler and compilers can be used to develop software. The peripherals however are vastly improved and therefore not compatible with other 8051/8052 implementations.

2.1. Performance

The AX8052 core employs a pipelined architecture that greatly increases its instruction throughput over the standard 8052 architecture. Instead of using 12, 24 or 48 clock cycles to execute instructions, AX8052 requires between 1 and 11 clock cycles depending on instructions. 90% of the instructions are executed between 1 and 4 clock cycles.

Clocks to execute	1	2	2+	3	3+	4	4+	5	6	7	11
8052 instructions	21	23	8	21	17	10	2	4	3	1	2

Table 1: AX8052 performance

There is one clock cycle latency when reading data in the IRAM. It does not concern internal SFR read and write accesses nor IRAM write accesses. Those instructions are indicated with an plus (+) in the instruction set summary table, indicating that the latency depends on the memory space access. Instructions doing read or write accesses to the external SFR memory space are also indicated with an plus (+) as the latency depends on the peripheral.

There is as well one clock cycle latency when reading data in the XRAM. It is not the case for write accesses.

The 4 register banks are located in the IRAM. So R0 and R1 of the active register bank selected by PSW[4:3] are not easily accessible when doing indirect addressing. In order to speed up this addressing mode, the core has two internal shadow registers to store R0 and R1 images. Doing so, it is not necessary to read R0 or R1 each time the core makes an indirect access. Nevertheless, instructions that change PSW[4:3] flags require 4 clock cycles more in order to read the new active R0 and R1.

The bit addressable locations are also in the IRAM. Writing a bit to such memory locations implies a Read-Modify-Write operation, and so requires 2 clock cycles to read the byte and modify the appropriate bit, and one clock cycle to write the new byte in the IRAM.

2.2. Instruction set

All the AX8052 instructions are the binary and functional equivalent of the 8051 counterparts, including opcodes, addressing modes and effects on PSW.

The next table named "*Instruction set summary*" provides information about the arithmetic, logical, data transfer, boolean manipulation and program branching instructions. In order to simplify the table, different symbols are used. Their meanings are:

- **Rn**: Register R0-R7 of the currently selected register bank.
- **@Ri**: Data RAM location addressed indirectly through R0 or R1.
- **Rel**: 8-bit, signed (2's complement) offset relative to the first bytes of the following instruction.
- **Direct**: 8-bit internal data location's address. This could be direct-access Data RAM location (0x00-0x7F) or an SFR (0x80-0xFF).
- **#data**: 8 or 16-bit constants.
- **Bit**: Direct-accessed bit in Data Ram or SFR.
- **Addr11**: 11-bit destination address used by ACALL and AJMP. The destination must be within the same 2K-bytes page of program memory as the first byte of the following instruction.
- **Addr16**: 16-bit destination address used by LCALL and LJMP. The destination may be anywhere within the program memory.

Instruction	Description	Prgm bytes	Clock cycles
	ARITHMETIC OPERATIONS		
ADD A, Rn	Add register to Accumulator	1	2
ADD A, direct	Add direct byte to Accumulator	2	2+
ADD A, @Ri	Add indirect RAM to Accumulator	1	2

ADD A, #data	Add immediate data to Accumulator	2	2
ADDC A, Rn	Add register to Accumulator with carry	1	2
ADDC A, direct	Add direct byte to Accumulator with carry	2	2+
ADDC A, @Ri	Add indirect RAM to Accumulator with carry	1	2
ADDC A, #data	Add immediate data to ACC with carry	2	2
SUBB A, Rn	Subtract Register from ACC with borrow	1	2
SUBB A, direct	Subtract direct byte from ACC with borrow	2	2+
SUBB A, @Ri	Subtract indirect RAM from ACC with borrow	1	2
SUBB A, #data	Subtract immediate data from ACC with borrow	2	2
INC A	Increment Accumulator	1	1
INC Rn	Increment register	1	3
INC direct	Increment direct byte	2	3+
INC @Ri	Increment indirect RAM	1	3
INC DPTR	Increment Data Pointer	1	1
DEC A	Decrement Accumulator	1	1
DEC Rn	Decrement Register	1	3
DEC direct	Decrement direct byte	2	3+
DEC @Ri	Decrement indirect RAM	1	3
MUL AB	Multiply A and B	1	11
DIV AB	Divide A by B	1	11
DA A	Decimal Adjust Accumulator	1	1
LOGICAL OPERATIONS			
ANL A, Rn	AND Register to Accumulator	1	2
ANL A, direct	AND direct byte to Accumulator	2	2+
ANL A, @Ri	AND indirect RAM to Accumulator	1	2
ANL A, #data	AND immediate data to Accumulator	2	2
ANL direct, A	AND Accumulator to direct byte	2	3+
ANL direct, #data	AND immediate data to direct byte	3	3+
ORL A, Rn	OR register to Accumulator	1	2
ORL A, direct	OR direct byte to Accumulator	2	2+

ORL A, @Ri	OR indirect RAM to Accumulator	1	2
ORL A, #data	OR immediate data to Accumulator	2	2
ORL direct, A	OR Accumulator to direct byte	2	3+
ORL direct, #data	OR immediate data to direct byte	3	3+
XRL A, Rn	Exclusive-OR register to Accumulator	1	2
XRL A, direct	Exclusive-OR direct byte to Accumulator	2	2+
XRL A, @Ri	Exclusive-OR indirect RAM to Accumulator	1	2
XRL A, #data	Exclusive-OR immediate data to Accumulator	2	2
XRL direct, A	Exclusive-OR Accumulator to direct byte	2	3+
XRL direct, #data	Exclusive-OR immediate data to direct byte	3	3+
CLR A	Clear Accumulator	1	1
CPL A	Complement Accumulator	1	1
RL A	Rotate Accumulator left	1	1
RLC A	Rotate Accumulator left through the carry	1	1
RR A	Rotate Accumulator right	1	1
RRC A	Rotate Accumulator right through the carry	1	1
SWAP A	Swap nibbles within the Accumulator	1	1
DATA TRANSFER			
MOV A, Rn	Move register to Accumulator	1	1
MOV A, direct	Move direct byte to Accumulator	2	2+
MOV A, @Ri	Move indirect RAM to Accumulator	1	1
MOV A, #data	Move immediate data to Accumulator	2	2
MOV Rn, A	Move Accumulator to register	1	1
MOV Rn, direct	Move direct byte to register	2	3+
MOV Rn, #data	Move immediate data to register	2	2
MOV direct, A	Move Accumulator to direct byte	2	2+
MOV direct, Rn	Move register to direct byte	2	3+
MOV direct, direct	Move direct byte to direct	3	3+

MOV direct, @Ri	Move indirect RAM to direct byte	2	3+
MOV direct, #data	Move immediate data to direct byte	3	3+
MOV @Ri, A	Move Accumulator to indirect RAM	1	1
MOV @Ri, direct	Move direct byte to indirect RAM	2	3+
MOV @Ri, #data	Move immediate data to indirect RAM	2	2
MOV DPTR, #data	Load Data Pointer with a 16-bit constant	3	3
MOVC A, @A+DPTR	Move Code byte relative to DPTR to ACC	1	4
MOVC A, @A+PC	Move Code byte relative to PC to ACC	1	4
MOVC @DPTR, A	Move Accumulator To Program Memory	1	4
MOVX A, @Ri	Move external RAM (8-bit addr) to ACC	1	2
MOVX A, @DPTR	Move external RAM (16-bit addr) to ACC	1	2
MOVX @Ri, A	Move ACC to external RAM (8-bit addr)	1	1
MOVX @DPTR, A	Move ACC to external RAM (16-bit addr)	1	1
PUSH direct	Push direct byte onto stack	2	3+
POP direct	Pop direct byte from stack	2	3+
XCH A, Rn	Exchange register with Accumulator	1	3
XCH A, direct	Exchange direct byte with Accumulator	2	3+
XCH A, @Ri	Exchange indirect RAM with Accumulator	1	3
XCHD A, @Ri	Exchange low-order digit indirect RAM with ACC	1	3
BOOLEAN MANIPULATION			
CLR C	Clear carry	1	1
CLR bit	Clear direct bit	2	4
SETB C	Set carry	1	1
SETB bit	Set direct bit	2	4
CPL C	Complement carry	1	1
CPL bit	Complement direct bit	2	6
ANL C, bit	AND direct bit to carry	2	3
ANL C, /bit	AND complement of direct bit to carry	2	3
ORL C, bit	OR direct bit to carry	2	3

ORL C, /bit	OR complement of direct bit to carry	2	3
MOV C, bit	Move direct bit to carry	2	3
MOV bit, C	Move carry to direct bit	2	4
JC rel	Jump if carry is set	2	3
JNC rel	Jump if carry not set	2	3
JB rel	Jump if direct bit is set	3	5
JNB rel	Jump if direct bit is not set	3	5
JBC bit, rel	Jump if direct bit is set and clear bit	3	7
PROGRAM BRANCHING			
ACALL addr11	Absolute subroutine call	2	3
LCALL addr16	Long subroutine call	3	4
RET	Return from subroutine	1	6
RETI	Return from interrupt	1	6
AJMP addr11	Absolute jump	2	3
LJMP addr16	Long jump	3	4
SJMP rel	Short jump (relative addr)	2	3
JMP @A+DPTR	Jump indirect relative to the DPTR	1	3
JZ rel	Jump if Accumulator is zero	2	3
JNZ rel	Jump if Accumulator is not zero	2	3
CJNE A, direct, rel	Compare direct byte to ACC and jump if not equal	3	4+
CJNE A, #data, rel	Compare immediate to ACC and jump if not equal	3	4
CJNE Rn, #data, rel	Compare immediate to register and jump if not equal	3	5
CJNE @Ri, #data, rel	Compare immediate to indirect and jump if not equal	3	5
DJNZ Rn, rel	Decrement register and jump if not zero	2	4
DJNZ direct, rel	Decrement direct byte and jump if not zero	3	4+
NOP	No Operation	1	1

Table 2: Instruction set summary

2.3. Memory organization

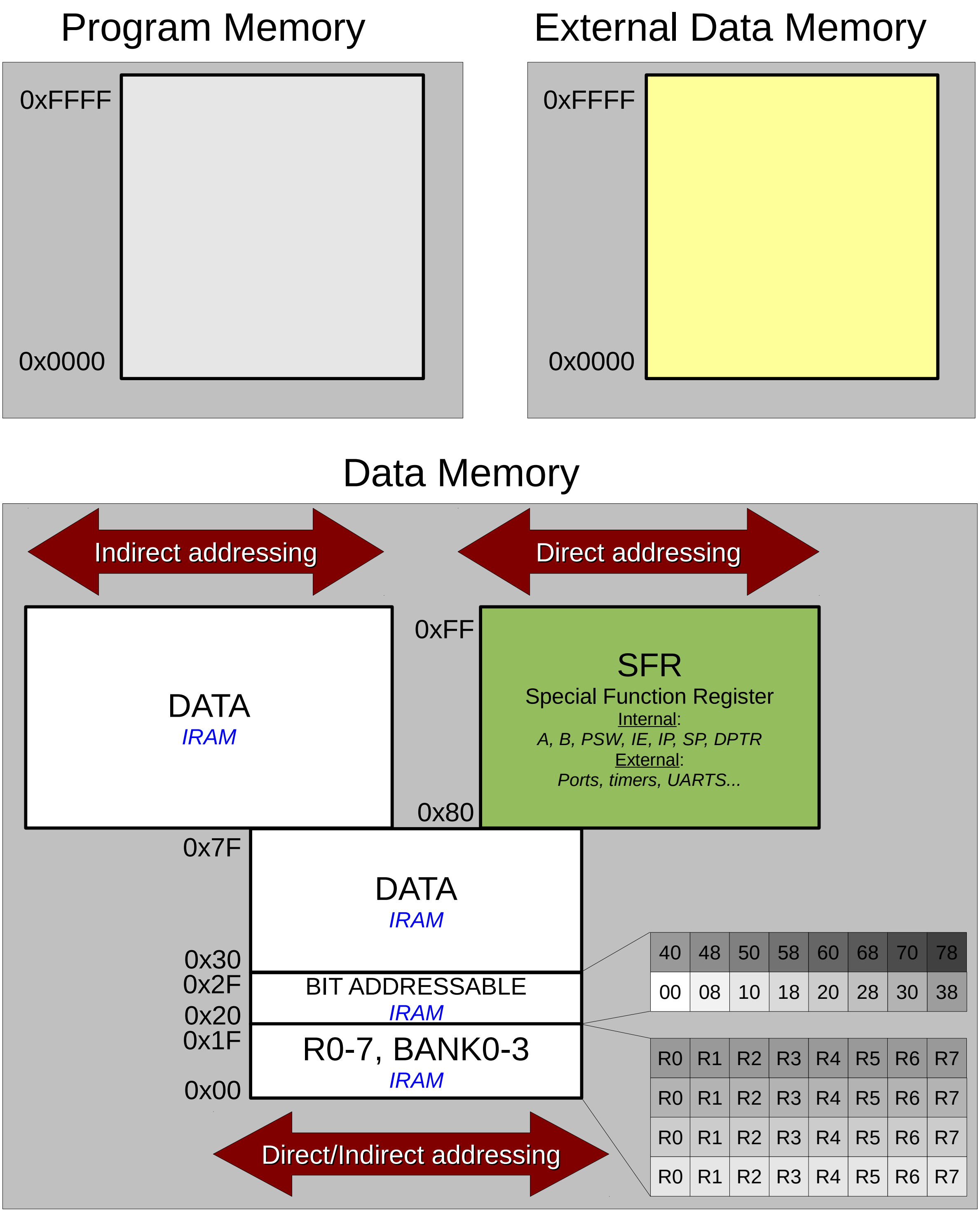


Figure 2: AX8052 memory organization

2.4. Data memory

The AX8052 has 256 bytes of data memory mapping called IRAM (*internal data*) or SFR (*Special Function Register*) depending on the addressing mode used and the address space access. The memory space goes from 0x00 to 0xFF.

The lower 128 bytes of data memory are used for general purpose registers, bits and scratch pad memory. Either direct or indirect addressing may be used to access the lower 128 bytes of data memory:

- Location 0x00 through 0x1F are addressable as four banks of general purpose registers, each bank consisting of eight byte-wide registers.

- The next 16 bytes locations 0x20 through 0x2F may either be addressed as bytes or as 128 bit locations accessible with the direct addressing mode.

The upper 128 bytes region represents the upper part of internal data memory and the SFR. They are physically separated and are accessible through different addressing modes:

- The upper 128 bytes of internal data memory is accessible only with indirect addressing.

- Special Function Registers are accessible on the same address space, using direct addressing.

2.4.1. Register banks

The AX8052 uses 8 “R” registers (*locations 0x00 through 0x1F*) which are used in many instructions. These “R” registers are numbered from 0 through 7 (*R0, R1, R2, R3, R4, R5, R6, R7*) and are generally used to assist in manipulating values and moving data from one memory location to another. The microcontroller has 4 distinct register banks and only one of these banks may be enabled at a time. When the CPU is first booted up, register bank 0 is used by default. However, your program may instruct the CPU to use one of the alternate register banks; i.e., register bank 1, 2 or 3. In this case, R4 (*for example*) will no longer be the same as internal RAM address 04h. Two bits in the Program Status Word (*PSW*), RS0 (*PSW.3*) and RS1 (*PSW.4*), select the active register bank.

Indirect addressing mode uses registers R0 and R1 as index registers. The AX8052 has a directly accessible image of the active R0 and R1, speeding up indirect accesses. Doing so, the core does not need to read R0 or R1 in IRAM before doing an indirect access. Each time the active R0 or R1 register is changed, or when RS0 and/or RS1 is modified, the core updates the R0 and R1 images.

Register bank 3	0x18	R0	R1	R2	R3	R4	R5	R6	R7	**RS0=1 RS1=1**
Register bank 2	0x10	R0	R1	R2	R3	R4	R5	R6	R7	**RS0=0 RS1=1**
Register bank 1	0x08	R0	R1	R2	R3	R4	R5	R6	R7	**RS0=1 RS1=0**
Register bank 0	0x00	R0	R1	R2	R3	R4	R5	R6	R7	**RS0=0 RS1=0**

IRAM 0x00 -> 0x1F

Figure 3: Register Banks

2.4.2. Bit addressable locations

The sixteen data memory location at 0x20 through 0x2F are also accessible as 128 individually addressable bits. Each bit has a bit address from 0x00 to 0x7F. Bit 0 of the byte at 0x20 has bit address 0x00 while bit 7 of the byte at 0x20 has bit address 0x07. Bit 7 of the byte at 0x2F has bit address 0x7F. A bit access is distinguished from a full byte access by the type of instruction used (*bit source or destination operands as opposed to a byte source or destination*).

0x2F	40	48	50	58	60	68	70	78
0x20	00	08	10	18	20	28	30	38

IRAM 0x20 -> 0x2F

Figure 4: Bit memory

2.4.3. Stack

A programmer's stack can be located anywhere in the 256-byte data memory. The stack area is designated using the Stack Pointer (*SP, 0x81*) SFR. The SP will point to the last location used. The next value pushed on the stack is place at SP+1, and then SP is incremented. A reset initializes the stack pointer to location 0x07. Therefore, the first value pushed on the stack is placed at location 0x08, which is also the first register (*R0*) of register bank 1. Thus, if more than one register bank is to be used, the SP should be initialized to a location in the data memory not being used for data storage. The stack depth can extend up to 256 bytes.

2.4.4. Special Function Registers (SFR)

The direct access data memory locations from 0x80 to 0xFF constitute the Special Function Registers (*SFRs*).

The internal SFR are the accumulator (*A or ACC*), the B register (*B*), the Stack Pointer (*SP*), the Program Status Word (*PSW*), the Interrupt Enable (*IE*) and Interrupt priority (*IP*) registers and the external Data Pointer register (*DPL and DPH, known as DPTR*). The word "*internal*" is used to describe those SFR because they are physically located inside the AX8052 core.

In opposition to "*internal*" SFR, it exists external SFRs that provide control and data exchange with the AX8052 and peripherals (*like ports, timers, UARTS...*). The word "*external*" is used because the peripherals implementing those SFR are located outside the core.

Direct addressing mode is used to access the SFR memory location, i.e. from 0x80 to 0xFF. SFRs with addresses ending in 0x0 or 0x8 (*e.g. ACC, B, PSW, IP, IE…*) are bit-addressable as well as byte-addressable. All other SFRs are byte-addressable only.

Unoccupied addresses in the SFR space are reserved for future use (*peripherals...*). Accessing these areas will have an indeterminate effect and should be avoided.

2.4.5. Internal SFR descriptions

2.4.5.1. Accumulator - A

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0

Reset value: 0x00
SFR address: 0xE0
Bit addressable: Yes

Bits 7-0: Accumulator (*A*).

2.4.5.2. B register – B

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0

Reset value: 0x00
SFR address: 0xF0
Bit addressable: Yes

Bits 7-0: B register (*B*).
This register serves as a second accumulator for some arithmetic operations.

2.4.5.3. Stack Pointer – SP

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SP.7	SP.6	SP.5	SP.4	SP.3	SP.2	SP.1	SP.0

Reset value: 0x07
SFR address: 0x81
Bit addressable: No

Bits 7-0: Stack Pointer (*SP*).
The stack pointer holds the top location of the stack.
It is incremented before every PUSH operations and decremented after every POP operations.

2.4.5.4. Program Status Word – PSW

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R
CY	AC	F0	RS1	RS0	OV	F1	P

Reset value: 0x00
SFR address: 0xD0
Bit addressable: Yes

Bit 7: Carry Flag (*CY*).
This bit is set when the last arithmetic operation resulted in a carry into (*addition*) or a borrow (*subtraction*). It is cleared to 0 by all other arithmetic operations.
Bit 6: Auxiliary Carry Flag (*AC*).
This bit is set when the last arithmetic operation resulted in a carry into (*addition*) or a borrow (*subtraction*) from the higher order nibble. It is cleared to 0 by all other arithmetic operations.
Bit 5: User Flag 0 (*F0*).
This is a bit-addressable, general purpose flag for use under software control.
Bit 4-3: Register Bank Select (*RS1-RS0*).
These bits select which register bank is used during register access.
0-0: Register Bank 0 is selected
0-1: Register Bank 1 is selected
1-0: Register Bank 2 is selected
1-1: Register Bank 3 is selected
Bit 2: Overflow Flag (*OV*).
This bit is set to 1 under the following circumstances:
An ADD, ADDC or SUBB instruction causes a sign-change overflow.
A MUL instruction results in an overflow (*result is greater than 255*).
A DIV instruction causes a divide-by-zero condition.
Bit 1: User Flag 1 (*F1*).
This is a bit-addressable, general purpose flag for use under software control.
Bit 0: Parity Flag (*P*).
This bit is set to logic 1 if the sum of the eight bits in the accumulator is odd and cleared if the sum is even.

2.4.5.5. Data Pointer – DPTR

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DPL.7	DPL.6	DPL.5	DPL.4	DPL.3	DPL.2	DPL.1	DPL.0

Reset value: 0x00
SFR address: 0x82
Bit addressable: No

Bits 7-0: Data Pointer Low (*DPL*).
The DPL register is the low byte of the 16-bit DPTR. DPTR is used to access indirectly addressed memory.

R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DPH.7	DPH.6	DPH.5	DPH.4	DPH.3	DPH.2	DPH.1	DPH.0

Reset value: 0x00
SFR address: 0x83
Bit addressable: No

Bits 7-0: Data Pointer High (*DPH*).
The DPL register is the high byte of the 16-bit DPTR. DPTR is used to access indirectly addressed memory.

2.5. Interrupts

AX8052 includes an interrupt system supporting a total of 22 interrupt sources with 2 priority levels.

If interrupts are enabled for the source, an interrupt request is generated when the interrupt input line is triggered. As soon as execution of the current instruction is complete, the CPU generates an LCALL to a predetermined address to begin execution of an interrupt service routine (ISR). Each ISR must end with an RETI instruction which returns program execution to the next instruction that would have been executed if the interrupt request had not occurred.

If interrupts are not enabled, the interrupt input line activity is ignored by the hardware and program execution continues as normal. Each interrupt source can be individually enabled or disabled through the use of an associated interrupt enable bit in an SFR (*IE*, *EIE*, *E2IE*). However, interrupts must first be globally enabled by setting the EA bit (IE.7) to logic 1 before the individual interrupt enables are recognized. Setting the EA bit to logic 0 disables all interrupt sources regardless of the individual interrupt-enable settings.

If an interrupt input line remains set after the CPU completes the return-from-interrupt (RETI) instruction, a new interrupt request will be generated immediately and the CPU will re-enter the ISR after completion of the next instruction.

2.5.1. Interrupt priority

Each interrupt source can be individually programmed to one of two priority levels: low or high. A low priority interrupt service routine can be preempted by a high priority interrupt. A high priority interrupt cannot be preempted. Each interrupt has an associated interrupt priority bit in an SFR (*IP*, *EIP*, *E2IP*) used to configure its priority level. Low priority is the default. If two interrupts are recognized simultaneously, the interrupt with the higher priority is serviced first. If both interrupts have the same priority level, a fixed priority order is used to arbitrate.

2.5.2. Interrupt latency

Interrupts response time depends on the state of the CPU when the interrupt occurs. Pending interrupts are sampled and priority decoded each system clock cycle. Therefore, the fastest possible response time is 5 clock cycles: 1 clock cycle to detect the interrupt,and 4 clock cycles to complete the LCALL to the ISR.

If the CPU is executing an ISR for an interrupt with equal or higher priority, the new interrupt will not be serviced until the current ISR completes, including the RETI and following instruction.

2.6. Interrupt Vectors

Address	**Vector**
0x0000	Reset
0x0003	External 0 Interrupt
0x000B	Wakeup Timer Interrupt
0x0013	External 1 Interrupt
0x001B	GPIO Interrupt
0x0023	Radio Interrupt
0x002B	Clock Management Interrupt (see OSCCALIB)
0x0033	Power Management Interrupt
0x003B	Timer 0 Interrupt
0x0043	Timer 1 Interrupt

0x004B	Timer 2 Interrupt
0x0053	SPI 0 Interrupt
0x005B	UART 0 Interrupt
0x0063	UART 1 Interrupt
0x006B	GPADC Interrupt
0x0073	DMA Interrupt
0x007B	Output Compare 0 Interrupt
0x0083	Output Compare 1 Interrupt
0x008B	Input Capture 0 Interrupt
0x0093	Input Capture 1 Interrupt
0x009B	Random Number Generator Interrupt
0x00A3	AES Interrupt
0x00AB	DebugLink Interrupt

3. AX8052 Registers

3.1. Register: SP

Name	Bits	R/W	Reset	Description
SP	7:0	RW	0x07	Stack Pointer

This is the stack pointer. See chapter 2.4.5.3 for additional documentation.

3.2. Register: DPH, DPL

Name	Bits	R/W	Reset	Description
DPTR0	15:0	RW	0x0000	Data Pointer

This is the data pointer register. It used in instructions that require a 16 Bit address. See chapter 2.4.5.5 for additional documentation.

3.3. Register: DPH1, DPL1

Name	Bits	R/W	Reset	Description
DPTR1	15:0	RW	0x0000	Second Data Pointer

This is the alternate data pointer register.

3.4. Register: DPS

The AX8052 features dual DPTR support to speed up X memory operations, such as memory copies. Bit 0 of the DPS register selects which DPTR is used during operations that reference the DPTR register.

Name	Bits	R/W	Reset	Description
DPS	0	RW	0	Data Pointer Select; 0=DPTR0, 1=DPTR1

3.5. Register: IE

Name	Bits	R/W	Reset	Description
IE0	0	RW	0	External 0 Interrupt Enable
IE1	1	RW	0	Wakeup Timer Interrupt Enable
IE2	2	RW	0	External 1 Interrupt Enable
IE3	3	RW	0	GPIO Interrupt Enable
IE4	4	RW	0	Radio Interrupt Enable
IE5	5	RW	0	Clock Management Interrupt Enable
IE6	6	RW	0	Power Management Interrupt Enable
EA	7	RW	0	Global Interrupt Enable

This register belongs to the interrupt controller. See chapter 2.5 for additional documentation about the interrupt subsystem. Interrupt sources enabled in this register may be used to wake up the microcontroller from sleep mode. EA does not need to be set to 1 for wake-up from sleep mode; it is sufficient for the interrupt source enable bit to be 1.

3.6. Register: IP

Name	Bits	R/W	Reset	Description
IP0	0	RW	0	External 0 Interrupt Priority
IP1	1	RW	0	Wakeup Timer Interrupt Priority
IP2	2	RW	0	External 1 Interrupt Priority
IP3	3	RW	0	GPIO Interrupt Priority
IP4	4	RW	0	Radio Interrupt Priority
IP5	5	RW	0	Clock Management Interrupt Priority
IP6	6	RW	0	Power Management Interrupt Priority

This register belongs to the interrupt controller. See chapter 2.5 for additional documentation about the interrupt subsystem.

3.7. Register: EIE

Name	Bits	R/W	Reset	Description
EIE0	0	RW	0	Timer 0 Interrupt Enable
EIE1	1	RW	0	Timer 1 Interrupt Enable
EIE2	2	RW	0	Timer 2 Interrupt Enable
EIE3	3	RW	0	SPI 0 Interrupt Enable
EIE4	4	RW	0	UART 0 Interrupt Enable
EIE5	5	RW	0	UART 1 Interrupt Enable
EIE6	6	RW	0	GPADC Interrupt Enable
EIE7	7	RW	0	DMA Interrupt Enable

This register belongs to the interrupt controller. See chapter 2.5 for additional documentation about the interrupt subsystem.

3.8. Register: EIP

Name	Bits	R/W	Reset	Description
EIP0	0	RW	0	Timer 0 Interrupt Priority
EIP1	1	RW	0	Timer 1 Interrupt Priority
EIP2	2	RW	0	Timer 2 Interrupt Priority
EIP3	3	RW	0	SPI 0 Interrupt Priority
EIP4	4	RW	0	UART 0 Interrupt Priority
EIP5	5	RW	0	UART 1 Interrupt Priority
EIP6	6	RW	0	GPADC Interrupt Priority
EIP7	7	RW	0	DMA Interrupt Priority

This register belongs to the interrupt controller. See chapter 2.5 for additional documentation about the interrupt subsystem.

3.9. Register: E2IE

Name	Bits	R/W	Reset	Description
E2IE0	0	RW	0	Output Compare 0 Interrupt Enable
E2IE1	1	RW	0	Output Compare 1 Interrupt Enable
E2IE2	2	RW	0	Input Capture 0 Interrupt Enable
E2IE3	3	RW	0	Input Capture 1 Interrupt Enable
E2IE4	4	RW	0	Random Number Generator Interrupt Enable
E2IE5	5	RW	0	AES Interrupt Enable
E2IE6	6	RW	0	DebugLink Interrupt Enable

This register belongs to the interrupt controller. See chapter 2.5 for additional documentation about the interrupt subsystem.

3.10. Register: E2IP

Name	Bits	R/W	Reset	Description
E2IP0	0	RW	0	Output Compare 0 Interrupt Priority
E2IP1	1	RW	0	Output Compare 1 Interrupt Priority
E2IP2	2	RW	0	Input Capture 0 Interrupt Priority
E2IP3	3	RW	0	Input Capture 1 Interrupt Priority
E2IP4	4	RW	0	Random Number Generator Interrupt Priority
E2IP5	5	RW	0	AES Interrupt Priority
E2IP6	6	RW	0	DebugLink Interrupt Priority

This register belongs to the interrupt controller. See chapter 2.5 for additional documentation about the interrupt subsystem.

3.11. Register: PSW

Name	Bits	R/W	Reset	Description
PARITY	0	RW	0	Accumulator Parity
F1	1	RW	0	User Flag 1
OV	2	RW	0	Overflow Flag
RS	4:3	RW	00	Register Bank Select
F0	5	RW	0	User Flag 0
AC	6	RW	0	Auxiliary Carry Flag
CY	7	RW	0	Carry Flag

This is the program status word. See chapter 2.4.5.4 for additional documentation.

3.12. Register: ACC

Name	Bits	R/W	Reset	Description
ACC	7:0	RW	0x00	Accumulator

This is the accumulator register. It is used as an operand in many instructions. See chapter 2.4.5.1 for additional documentation.

3.13. Register: B

Name	Bits	R/W	Reset	Description
B	7:0	RW	0x00	B Register

This is the B register. It is used to supply the second operand to the multiplication instruction. See chapter 2.4.5.2 for additional documentation.

3.14. Register: XPAGE

Name	Bits	R/W	Reset	Description
XPAGE	7:0	RW	0x00	XPAGE Register

The XPAGE register supplies the high byte of the X address for MOVX @Rx instructions. For compatibility with the SDCC runtime library, this register is also available under the name _XPAGE.

4. Address Space

The AX8052 uses a harvard architecture with 3.5 address spaces. P space is used by instruction fetch and can be accessed using `MOVC` instructions. I space can be accessed by direct `MOV` and indirect `MOV @Rx` instructions. The upper half of it may only be accessed by indirect moves `MOV @Rx`. SFR space can be accessed by direct `MOV` instructions. X space can be accessed by `MOVX` instructions.

<table>
<tr><th rowspan="2">Address</th><th rowspan="2">P (Code) Space</th><th rowspan="2">X Space</th><th colspan="2">I Space</th></tr>
<tr><th>Direct</th><th>Indirect</th></tr>
<tr><td>0000–007F</td><td rowspan="12">FLASH</td><td rowspan="3">XRAM</td><td>IRAM</td><td rowspan="2">IRAM</td></tr>
<tr><td>0080–00FF</td><td>SFR</td></tr>
<tr><td>0100–1FFF</td><td></td><td></td></tr>
<tr><td>2000–207F</td><td>IRAM</td><td></td><td></td></tr>
<tr><td>2080–3F7F</td><td></td><td></td><td></td></tr>
<tr><td>3F80–3FFF</td><td>SFR</td><td></td><td></td></tr>
<tr><td>4000–4FFF</td><td>RREG</td><td></td><td></td></tr>
<tr><td>5000–5FFF</td><td>RREG (nb)</td><td></td><td></td></tr>
<tr><td>6000–7FFF</td><td>XREG</td><td></td><td></td></tr>
<tr><td>8000–BFFF</td><td rowspan="3">FLASH</td><td></td><td></td></tr>
<tr><td>C000–DFFF</td><td></td><td></td></tr>
<tr><td>E000–FBFF</td><td></td><td></td></tr>
<tr><td>FC00–FFFF</td><td>Calibration</td><td>Calibration</td><td></td><td></td></tr>
</table>

XREG are the AX8052 registers that do not fit into SFR space. The XREG memory map can be found in chapter 4.3. The SFR memory map can be found in chapter 4.2. RREG are the registers of the radio chip. RREG (nb) is a mirror of the radio chip registers used for nonblocking access. The use of RREG and RREG (nb) is documented in chapter 11.2.

Some memories can appear in multiple address spaces, for example the FLASH and the IRAM. Accessing them through different address spaces accesses the same content.

The FLASH is organized as 64 1kByte pages. FLASH may be erased page-wise and written in 16Bit words. The last FLASH page is reserved for factory calibration data and should not be overwritten. It is highly recommended to call flash_apply_calibration() from libmf early in the startup sequence to ensure calibration data is applied to the AX8052.

The upper half of FLASH may also be accessed in X space. This reduces the need for generic pointers[2] by allowing to access constant and variable data through X space pointers only.

4.1. Memory Switch Architecture

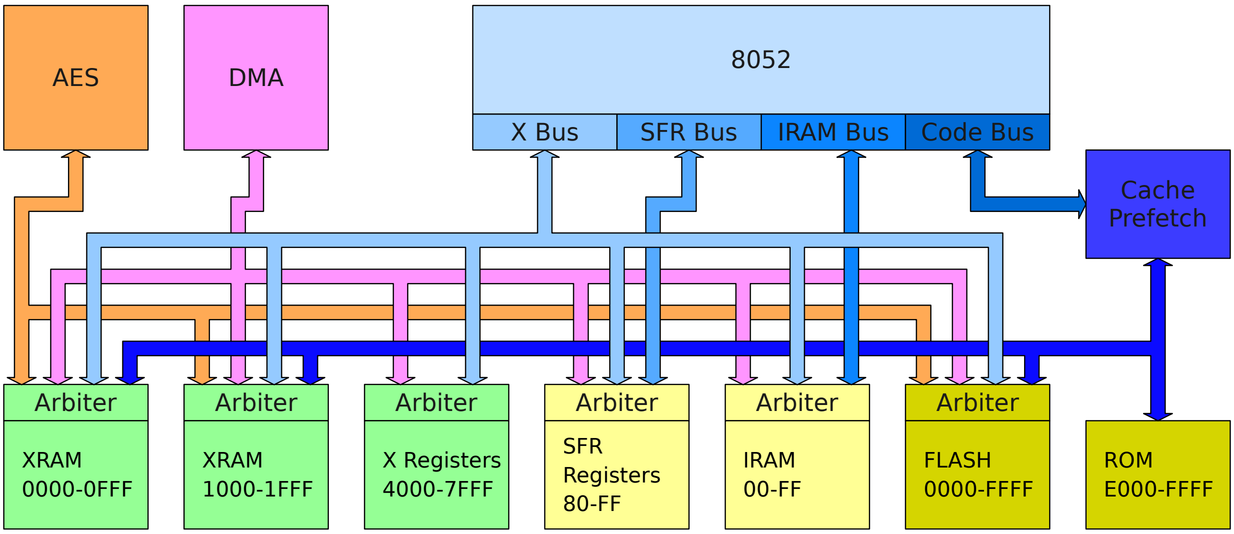


Figure 5: Memory Switch Architecture

Figure 5 shows the memory multiplexing and switching architecture. As can be seen, the chip contains three bus masters: the microcontroller (AX8052), the DMA controller, and the AES core. Care has been taken so that multiple bus masters can access independent memories concurrently.

2 Generic pointers are pointers that contain, besides the address, an address space tag

4.2. SFR Address Map

Address	Register							
Hex	0	1	2	3	4	5	6	7
	8	9	A	B	C	D	E	F
0x80–	PORTA	SP	DPL	DPH	DPL1	DPH1	DPS	PCON
0x8F	PORTB	DIRA	DIRB	DIRC	PORTR	PINR	DIRR	–
0x90–	PORTC	NVSTATUS	NVADDR0	NVADDR1	NVDATA0	NVDATA1	NVKEY	CODECONFIG
0x9F	EIE	T0MODE	T0CLKSRC	T0STATUS	T0CNT0	T0CNT1	T0PERIOD0	T0PERIOD1
0xA0–	E2IE	T1MODE	T1CLKSRC	T1STATUS	T1CNT0	T1CNT1	T1PERIOD0	T1PERIOD1
0xAF	IE	T2MODE	T2CLKSRC	T2STATUS	T2CNT0	T2CNT1	T2PERIOD0	T2PERIOD1
0xB0–	EIP	RADIOACC	RADIOADDR1	RADIOADDR0	RADIODATA3	RADIODATA2	RADIODATA1	RADIODATA0
0xBF	IP	OC0MODE	OC0PIN	OC0STATUS	OC0COMP0	OC0COMP1	RADIOSTAT0	RADIOSTAT1
0xC0–	E2IP	OC1MODE	OC1PIN	OC1STATUS	OC1COMP0	OC1COMP1	CLKCON	CLKSTAT
0xCF	PINA	ADCCONV	ADCCH0CONFIG	ADCCH1CONFIG	IC0MODE	IC0STATUS	IC0CAPT0	IC0CAPT1
0xD0–	PSW	ADCCLKSRC	ADCCH2CONFIG	ADCCH3CONFIG	IC1MODE	IC1STATUS	IC1CAPT0	IC1CAPT1
0xDF	–	XPAGE	WDTCFG	WDTRESET	SPMODE	SPSTATUS	SPSHREG	SPCLKSRC
0xE0–	ACC	ANALOGCOMP	DBGLNKSTAT	DBGLNKBUF	U0CTRL	U0STATUS	U0SHREG	U0MODE
0xEF	PINB	WTIRQEN	WTSTAT	WTCNTR1	U1CTRL	U1STATUS	U1SHREG	U1MODE

Address	Register							
Hex	0	1	2	3	4	5	6	7
	8	9	A	B	C	D	E	F
0xF0–	B	WTCFGA	WTCNTA0	WTCNTA1	WTEVTA0	WTEVTA1	WTEVTB0	WTEVTB1
0xFF	PINC	WTCFGB	WTCNTB0	WTCNTB1	WTEVTC0	WTEVTC1	WTEVTD0	WTEVTD1

4.3. X Register Address Map

Address	Register							
Hex	0	1	2	3	4	5	6	7
	8	9	A	B	C	D	E	F
0x3F80–	PORTA	–	–	–	–	–	–	PCON
0x3F8F	PORTB	DIRA	DIRB	DIRC	PORTR	PINR	DIRR	–
0x3F90–	PORTC	NVSTATUS	NVADDR0	NVADDR1	NVDATA0	NVDATA1	NVKEY	CODECONFIG
0x3F9F	–	T0MODE	T0CLKSRC	T0STATUS	T0CNT0	T0CNT1	T0PERIOD0	T0PERIOD1
0x3FA0–	–	T1MODE	T1CLKSRC	T1STATUS	T1CNT0	T1CNT1	T1PERIOD0	T1PERIOD1
0x3FAF	IE	T2MODE	T2CLKSRC	T2STATUS	T2CNT0	T2CNT1	T2PERIOD0	T2PERIOD1
0x3FB0–	–	RADIOACC	RADIOADDR1	RADIOADDR0	RADIODATA3	RADIODATA2	RADIODATA1	RADIODATA0
0x3FBF	IP	OC0MODE	OC0PIN	OC0STATUS	OC0COMP0	OC0COMP1	RADIOSTAT0	RADIOSTAT1

ON Semiconductor®

Address	Register							
Hex	0	1	2	3	4	5	6	7
	8	9	A	B	C	D	E	F
0x3FC0–	–	OC1MODE	OC1PIN	OC1STATUS	OC1COMP0	OC1COMP1	CLKCON	CLKSTAT
0x3FCF	PINA	ADCCONV	ADCCH0CONFIG	ADCCH1CONFIG	IC0MODE	IC0STATUS	IC0CAPT0	IC0CAPT1
0x3FD0–	–	ADCCLKSRC	ADCCH2CONFIG	ADCCH3CONFIG	IC1MODE	IC1STATUS	IC1CAPT0	IC1CAPT1
0x3FDF	–	–	WDTCFG	WDTRESET	SPMODE	SPSTATUS	SPSHREG	SPCLKSRC
0x3FE0–	–	ANALOGCOMP	DBGLNKSTAT	DBGLNKBUF	U0CTRL	U0STATUS	U0SHREG	U0MODE
0x3FEF	PINB	WTIRQEN	WTSTAT	WTCNTR1	U1CTRL	U1STATUS	U1SHREG	U1MODE
0x3FF0–	–	WTCFGA	WTCNTA0	WTCNTA1	WTEVTA0	WTEVTA1	WTEVTB0	WTEVTB1
0x3FFF	PINC	WTCFGB	WTCNTB0	WTCNTB1	WTEVTC0	WTEVTC1	WTEVTD0	WTEVTD1
0x7000–	INTCHGA	INTCHGB	INTCHGC	EXTIRQ	PINCHGA	PINCHGB	PINCHGC	ANALOGA
0x700F	PALTA	PALTB	PALTC	PINSEL	GPIOENABLE	–	–	–
0x7010–	DMA0ADDR0	DMA0ADDR1	DMA1ADDR0	DMA1ADDR1	DMA0CONFIG	DMA1CONFIG	–	–
0x701F	–	–	–	–	–	–	–	–
0x7020–	ADCCH0VAL0	ADCCH0VAL1	ADCCH1VAL0	ADCCH1VAL1	ADCCH2VAL0	ADCCH2VAL1	ADCCH3VAL0	ADCCH3VAL1
0x702F	ADCTUNE0	ADCTUNE1	ADCTUNE2	–	–	–	–	–
0x7040–	RADIOFDATAADDR0	RADIOFDATAADDR1	RADIOFSTATADDR0	RADIOFSTATADDR1	RADIOMUX	–	–	–
0x704F	–	–	–	–	–	–	–	–

Address	Register							
Hex	0	1	2	3	4	5	6	7
	8	9	A	B	C	D	E	F
0x7050–	OSCFORCERUN	OSCRUN	OSCREADY	OSCCALIB	LPXOSCGM	–	–	–
0x705F	–	–	–	–	–	–	–	–
0x7060–	LPOSCCONFIG	–	LPOSCKFILT0	LPOSCKFILT1	LPOSCREF0	LPOSCREF1	LPOSCFREQ0	LPOSCFREQ1
0x706F	LPOSCPER0	LPOSCPER1	–	–	–	–	–	–
0x7070–	FRCOSCCONFIG	–	FRCOSCKFILT0	FRCOSCKFILT1	FRCOSCREF0	FRCOSCREF1	FRCOSCFREQ0	FRCOSCFREQ1
0x707F	FRCOSCPER0	FRCOSCPER1	–	–	–	–	–	–
0x7080–	–	–	–	–	SCRATCH0	SCRATCH1	SCRATCH2	SCRATCH3
0x708F	–	–	–	–	–	–	–	–
0x7F00–	SILICONREV	MISCCTRL	–	–	–	–	–	–
0x7F0F	–	–	–	–	–	–	–	–
0x7F10–	–	–	–	–	–	–	–	–
0x7F1F	XTALOSC	XTALAMPL	XTALREADY	–	–	–	–	–

4.4. Register Overview

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
MCU Core													
81	SP	RW	R	00000111	SP(7:0)								Stack Pointer
82	DPL	RW	R	00000000	DPTR0(7:0)								Data Pointer
83	DPH	RW	R	00000000	DPTR0(15:8)								Data Pointer
84	DPL1	RW	R	00000000	DPTR1(7:0)								Second Data Pointer
85	DPH1	RW	R	00000000	DPTR1(15:8)								Second Data Pointer
86	DPS	RW	R	–––––––0	–	–	–	–	–	–	–	DPS	Data Pointer Select
87	PCON	RW	R	–––00000	BROWNOUT	WAKEUP	WDTRESET	SWRESET	XRAMKEEP		PWRMODE		Power Control
A8	IE	RW	R	00000000	EA	IE(6:0)							Interrupt Enable
B8	IP	RW	R	–0000000	–	IP(6:0)							Interrupt Priority
98	EIE	RW	R	00000000	EIE(7:0)								Extended Interrupt Enable
B0	EIP	RW	R	00000000	EIP(7:0)								Extended Interrupt Priority
A0	E2IE	RW	R	00000000	E2IE(7:0)								Extended 2 Interrupt Enable
C0	E2IP	RW	R	00000000	E2IP(7:0)								Extended 2 Interrupt Priority
D0	PSW	RW	R	00000000	CY	AC	F0	RS(1:0)		OV	F1	PARITY	Program Status Word
E0	ACC	RW	R	00000000	ACC(7:0)								Accumulator
F0	B	RW	R	00000000	B(7:0)								B Register
D9	XPAGE	RW	R	00000000	XPAGE(7:0)								XPAGE Register
E2	DBGLNKSTAT	RW	R	000–––––	DBG TXIE	DBG RXIE	DBGEN CHGIE	DBG EN	DBGTX UNDER	DBGTX EMPTY	DBGRX OVER	DBGRX FULL	DebugLink Status
E3	DBGLNKBUF	RW	R	––––––––	DBGBUF								DebugLink Buffer Register

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
GPIO													
80	PORTA	RW	R	00000000	PORTA(7:0)								PORTA Register
C8	PINA	R	R	————	PINA(7:0)								PINA Register
89	DIRA	RW	R	00000000	DIRA(7:0)								DIRA Register
88	PORTB	RW	R	00000000	PORTB(7:0)								PORTB Register
E8	PINB	R	R	————	PINB(7:0)								PINB Register
8A	DIRB	RW	R	00000000	DIRB(7:0)								DIRB Register
90	PORTC	RW	R	00000000	PORTC(7:0)								PORTC Register
F8	PINC	R	R	————	PINC(7:0)								PINC Register
8B	DIRC	RW	R	00000000	DIRC(7:0)								DIRC Register
8C	PORTR	RW	R	00000000	PORTR(7:0)								PORTR Register
8D	PINR	R	R	————	PINR(7:0)								PINR Register
8E	DIRR	RW	R	00000000	DIRR(7:0)								DIRR Register
7007	ANALOGA	RW	R	00000000	ANALOGA(7:0)								ANALOGA Register
7000	INTCHGA	RW	R	00000000	INTCHGA(7:0)								Port A Interrupt On Change Register
7001	INTCHGB	RW	R	00000000	INTCHGB(7:0)								Port B Interrupt On Change Register
7002	INTCHGC	RW	R	00000000	INTCHGC(7:0)								Port C Interrupt On Change Register
7003	EXTIRQ	RW	R	–000–000	–	EXTIRQ1 PIN	EXTIRQ1 MODE(1:0)		–	EXTIRQ0 PIN	EXTIRQ0 MODE(1:0)		External Interrupt Configuration
7004	PINCHGA	R	R	————	PINCHGA(7:0)								Port A Pin Change Register
7005	PINCHGB	R	R	————	PINCHGB(7:0)								Port B Pin Change Register
7006	PINCHGC	R	R	————	PINCHGC(7:0)								Port C Pin Change Register

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
7008	PALTA	RW	R	00000000	PALTA(7:0)								Port A Alternate Function Register
7009	PALTB	RW	R	––000000	–	–	PALTB(5:0)						Port B Alternate Function Register
700A	PALTC	RW	R	–––00000	–	–	–	PALTC(4:0)					Port C Alternate Function Register
700B	PINSEL	RW	R	00000000	PINSEL(7:0)								Alternate Function Input Pin Selection Register
700C	GPIOENABLE	RW	R	–––––––1	–	–	–	–	–	–	–	GPIO ENABLE	GPIO Port Enable
System Controller													
C6	CLKCON	RW	R	00001000	CLKMON(1:0)		CLKPRE(2:0)			CLKSRC(2:0)			Clock Control
C7	CLKSTAT	R	R	––––––––	CLK LOSS	–	CLKPREST(2:0)			CLKSRCST(2:0)			Clock Status
Wakeup Timer													
F1	WTCFGA	RW	R	––001111	–	–	WTDIVA(2:0)			WTSRCA(2:0)			Wakeup Timer A Configuration
F9	WTCFGB	RW	R	––001111	–	–	WTDIVB(2:0)			WTSRCB(2:0)			Wakeup Timer B Configuration
E9	WTIRQEN	RW	R	00000000	WTIBD	WTIBC	WTIBB	WTIBA	WTIAD	WTIAC	WTIAB	WTIAA	Wakeup Timer Interrupt Enable
EA	WTSTAT	RC	R	––––––––	WTSBD	WTSBC	WTSBB	WTSBA	WTSAD	WTSAC	WTSAB	WTSAA	Wakeup Timer Status
F2	WTCNTA0	R	R	––––––––	WTCNTA(7:0)								Wakeup Counter A
F3	WTCNTA1	R	R	––––––––	WTCNTA(15:8)								Wakeup Counter A
FA	WTCNTB0	R	R	––––––––	WTCNTB(7:0)								Wakeup Counter B
FB	WTCNTB1	R	R	––––––––	WTCNTB(15:8)								Wakeup Counter B
EB	WTCNTR1	R	R	––––––––	WTCNTR(15:8)								Wakeup Counter Register
F4	WTEVTA0	R	R	00000000	WTEVTA(7:0)								Wakeup Event A
F5	WTEVTA1	R	R	00000000	WTEVTA(15:8)								Wakeup Event A
F6	WTEVTB0	R	R	00000000	WTEVTB(7:0)								Wakeup Event B

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
F7	WTEVTB1	R	R	00000000	WTEVTB(15:8)								Wakeup Event B
FC	WTEVTC0	R	R	00000000	WTEVTC(7:0)								Wakeup Event C
FD	WTEVTC1	R	R	00000000	WTEVTC(15:8)								Wakeup Event C
FE	WTEVTD0	R	R	00000000	WTEVTD(7:0)								Wakeup Event D
FF	WTEVTD1	R	R	00000000	WTEVTD(15:8)								Wakeup Event D
Watchdog Timer													
DA	WDTCFG	RW	R	––000000	–	–	WDT LCK	WDT ENA	WDTDIV(3:0)				Watchdog Configuration
DB	WDTRESET	W	R	––––––––	WDTRESET(7:0)								Watchdog Reset
Low Power Oscillator Calibration													
7060	LPOSCCONFIG	RW	R	–0000001	–	LPOSC FAST	LPOSC PRESC(2:0)			LPOSC CALSRC(2:0)			Low Power Oscillator Configuration
7063	LPOSCKFILT1	RW	R	00100000	LPOSCKFILT(15:8)								Low Power Oscillator Calibration Filter Constant
7062	LPOSCKFILT0	RW	R	11000100	LPOSCKFILT(7:0)								Low Power Oscillator Calibration Filter Constant
7065	LPOSCREF1	RW	R	01100001	LPOSCREF(15:8)								Low Power Oscillator Calibration Reference
7064	LPOSCREF0	RW	R	10101000	LPOSCREF(7:0)								Low Power Oscillator Calibration Reference
7067	LPOSCFREQ1	RW	R	00000000	LPOSCFREQ(9:2)								Low Power Oscillator Calibration Frequency
7066	LPOSCFREQ0	RW	R	0000––––	LPOSCFREQ(1:-2)				–	–	–	–	Low Power Oscillator Calibration Frequency
7069	LPOSCPER1	RW	R	––––––––	LPOSCPER(15:8)								Low Power Oscillator Calibration Period
7068	LPOSCPER0	RW	R	––––––––	LPOSCPER(7:0)								Low Power Oscillator Calibration Period

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
Fast RC Oscillator Calibration													
7070	FRCOSCCONFIG	RW	R	00000000	FRCOSC PERGAIN	FRCOSC PRESC(3:0)				FRCOSC CALSRC(2:0)			Fast RC Oscillator Configuration
7073	FRCOSCKFILT1	RW	R	00100000	FRCOSCKFILT(15:8)								Fast RC Oscillator Calibration Filter Constant
7072	FRCOSCKFILT0	RW	R	11000100	FRCOSCKFILT(7:0)								Fast RC Oscillator Calibration Filter Constant
7075	FRCOSCREF1	RW	R	01100001	FRCOSCREF(15:8)								Fast RC Oscillator Calibration Reference
7074	FRCOSCREF0	RW	R	10101000	FRCOSCREF(7:0)								Fast RC Oscillator Calibration Reference
7077	FRCOSCFREQ1	RW	R	00000000	FRCOSCFREQ(9:2)								Fast RC Oscillator Calibration Frequency
7076	FRCOSCFREQ0	RW	R	0000——	FRCOSCFREQ(1:-2)				–	–	–	–	Fast RC Oscillator Calibration Frequency
7079	FRCOSCPER1	RW	R	————	FRCOSCPER(15:8)								Fast RC Oscillator Calibration Period
7078	FRCOSCPER0	RW	R	————	FRCOSCPER(7:0)								Fast RC Oscillator Calibration Period
Oscillator Control													
7050	OSCFORCERUN	RW	R	——0000	–	–	–	–	LPXOSC FRUN	XOSC FRUN	LPOSC FRUN	FRCOSC FRUN	Oscillator Force Run
7051	OSCRUN	R	R	————	–	–	–	–	LPXOSC RUN	XOSC RUN	LPOSC RUN	FRCOSC RUN	Oscillator Force Run
7052	OSCREADY	R	R	————	–	–	–	–	LPXOSC RDY	XOSC RDY	LPOSC RDY	FRCOSC RDY	Oscillator Force Run
7053	OSCCALIB	RW	R	———100	LPOSC CALIRQ	FRCOSC CALIRQ	LPOSC CALIRQM	FRCOSC CALIRQM	CLKLOSS	CLKLOSS IRQE	LPOSC CALIRQE	FRCOSC CALIRQE	Oscillator Interrupt Status
7054	LPXOSCGM	RW	R	1—01000	LPXOSC BOOST	–	–	LPXOSCGM(4:0)					Low Power Crystal Oscillator Transconductance
7F01	MISCCTRL	RW	R	———00	–	–	–	–	–	–	XOSC DIS	LPXOSC DIS	Miscellaneous Control

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
Scratch													
7084	SCRATCH0	RW	R	————	SCRATCH(7:0)								Scratch Register; State is maintained in Deep Sleep
7085	SCRATCH1	RW	R	————	SCRATCH(15:8)								Scratch Register; State is maintained in Deep Sleep
7086	SCRATCH2	RW	R	————	SCRATCH(23:16)								Scratch Register; State is maintained in Deep Sleep
7087	SCRATCH3	RW	R	————	SCRATCH(31:24)								Scratch Register; State is maintained in Deep Sleep
DMA Controller													
7010	DMA0ADDR0	RW	R	11111111	DMA0ADDR(7:0)								DMA Channel 0 Buffer Descriptor Address
7011	DMA0ADDR1	RW	R	11111111	DMA0ADDR(15:8)								DMA Channel 0 Buffer Descriptor Address
7012	DMA1ADDR0	RW	R	11111111	DMA1ADDR(7:0)								DMA Channel 1 Buffer Descriptor Address
7013	DMA1ADDR1	RW	R	11111111	DMA1ADDR(15:8)								DMA Channel 1 Buffer Descriptor Address
7014	DMA0CONFIG	RW	R	00–00000	D0RUN	D0IRQ	–	D0SOURCE(4:0)					DMA Channel 0 Configuration
7015	DMA1CONFIG	RW	R	00–00000	D1RUN	D1IRQ	–	D1SOURCE(4:0)					DMA Channel 1 Configuration
Radio Controller													
B4	RADIODATA3	RW	R	00000000	RADIODATA(31:24)								Radio Chip Register Access Data
B5	RADIODATA2	RW	R	00000000	RADIODATA(23:16)								Radio Chip Register Access Data
B6	RADIODATA1	RW	R	00000000	RADIODATA(15:8)								Radio Chip Register Access Data
B7	RADIODATA0	RW	R	00000000	RADIODATA(7:0)								Radio Chip Register Access Data
B2	RADIOADDR1	RW	R	–0000000	–	RADIOADDR(14:8)							Radio Chip Register Access Address
B3	RADIOADDR0	RW	R	00000000	RADIOADDR(7:0)								Radio Chip Register Access Address

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
BF	RADIOSTAT1	R	R	————	RADIOSTAT(15:8)								Radio Chip Status
BE	RADIOSTAT0	R	R	————	RADIOSTAT(7:0)								Radio Chip Status
B1	RADIOACC	RW	R	——0000	RC BUSY	RC WRITE	–	–	RADIOADDR-FMT(1:0)		RADIODATA-FMT(1:0)		Radio Chip Access Mode and Control
7040	RADIOFDATAADDR0	RW	R	00000000	RADIOFDATAADDR(7:0)								Radio Chip FIFO Data Register Address
7041	RADIOFDATAADDR1	RW	R	——0000	–	–	–	–	RADIOFDATAADDR(11:8)				Radio Chip FIFO Data Register Address
7042	RADIOFSTATADDR0	RW	R	00000000	RADIOFSTATADDR(7:0)								Radio Chip FIFO Status Register Address
7043	RADIOFSTATADDR1	RW	R	——0000	–	–	–	–	RADIOFSTATADDR(11:8)				Radio Chip FIFO Status Register Address
7044	RADIOMUX	RW	R	–0000111	–	RADIO SPI	RADIOCLK(1:0)		RADIO IRQ	RADIOSYSCLK(2:0)			Radio Controller Pin Multiplexing Control
Timer 0													
9C	T0CNT0	RW	R	00000000	T0CNT(7:0)								Timer 0 Counter
9D	T0CNT1	RW	R	00000000	T0CNT(15:8)								Timer 0 Counter
9E	T0PERIOD0	RW	R	00000000	T0PERIOD(7:0)								Timer 0 Period
9F	T0PERIOD1	RW	R	00000000	T0PERIOD(15:8)								Timer 0 Period
9A	T0CLKSRC	RW	R	00001111	T0 CLK SYNC	T0 CLK INV	T0CLKDIV(2:0)			T0CLKSRC(2:0)			Timer 0 Clock Source
99	T0MODE	RW	R	00000000	T0 IRQMU	T0 IRQMO	T0PRBUF(1:0)		T0 LBUF	T0MODE(2:0)			Timer 0 Mode
9B	T0STATUS	R	R	00———	T0 IRQMPU	T0 IRQMPE	T0 IRQPU	T0 IRQPE	T0 IRQEU	T0 IRQEO	T0 IRQRU	T0 IRQRO	Timer 0 Status
Timer 1													
A4	T1CNT1	RW	R	00000000	T1CNT(7:0)								Timer 1 Counter
A5	T1CNT1	RW	R	00000000	T1CNT(15:8)								Timer 1 Counter
A6	T1PERIOD0	RW	R	00000000	T1PERIOD(7:0)								Timer 1 Period

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
A7	T1PERIOD1	RW	R	00000000	T1PERIOD(15:8)								Timer 1 Period
A2	T1CLKSRC	RW	R	00001111	T1 CLK SYNC	T1 CLK INV	T1CLKDIV(2:0)			T1CLKSRC(2:0)			Timer 1 Clock Source
A1	T1MODE	RW	R	00000000	T1 IRQMU	T1 IRQMO	T1PRBUF(1:0)		T1 LBUF	T1MODE(2:0)			Timer 1 Mode
A3	T1STATUS	R	R	00———	T1 IRQMPU	T1 IRQMPE	T1 IRQPU	T1 IRQPE	T1 IRQEU	T1 IRQEO	T1 IRQRU	T1 IRQRO	Timer 1 Status
Timer 2													
AC	T2CNT2	RW	R	00000000	T2CNT(7:0)								Timer 2 Counter
AD	T2CNT1	RW	R	00000000	T2CNT(15:8)								Timer 2 Counter
AE	T2PERIOD0	RW	R	00000000	T2PERIOD(7:0)								Timer 2 Period
AF	T2PERIOD1	RW	R	00000000	T2PERIOD(15:8)								Timer 2 Period
AA	T2CLKSRC	RW	R	00001111	T2 CLK SYNC	T2 CLK INV	T2CLKDIV(2:0)			T2CLKSRC(2:0)			Timer 2 Clock Source
A9	T2MODE	RW	R	00000000	T2 IRQMU	T2 IRQMO	T2PRBUF(1:0)		T2 LBUF	T2MODE(2:0)			Timer 2 Mode
AB	T2STATUS	R	R	00———	T2 IRQMPU	T2 IRQMPE	T2 IRQPU	T2 IRQPE	T2 IRQEU	T2 IRQEO	T2 IRQRU	T2 IRQRO	Timer 2 Status
Output Compare 0													
BC	OC0COMP0	RW	R	00000000	OC0COMP(7:0)								Output Compare 0 Compare Value
BD	OC0COMP1	RW	R	00000000	OC0COMP(15:8)								Output Compare 0 Compare Value
B9	OC0MODE	RW	R	00000–00	OC0 IRQMU	OC0 IRQMO	OC0CMPBUF(1:0)		OC0 LBUF	–	OC0MODE(1:0)		Output Compare 0 Mode
BA	OC0PIN	RW	R	00000000	OC0PCU(1:0)		OC0PCO(1:0)		OC0PCF(1:0)		OC0PCR(1:0)		Output Compare 0 Output Pin Configuration

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
BB	OC0STATUS	RW	R	00————	OC0 IRQMCU	OC0 IRQMCE	OC0 CUNDER	OC0 CEMPTY	OC0 IRQEF	OC0 IRQER	OC0 IRQRF	OC0 IRQRR	Output Compare 0 Status
Output Compare 1													
C4	OC1COMP0	RW	R	00000000	OC1COMP(7:0)								Output Compare 1 Compare Value
C5	OC1COMP1	RW	R	00000000	OC1COMP(15:8)								Output Compare 1 Compare Value
C1	OC1MODE	RW	R	00000–00	OC1 IRQMU	OC1 IRQMO	OC1CMPBUF(1:0)		OC1 LBUF	–	OC1MODE(1:0)		Output Compare 1 Mode
C2	OC1PIN	RW	R	00000000	OC1PCU(1:0)		OC1PCO(1:0)		OC1PCF(1:0)		OC1PCR(1:0)		Output Compare 1 Output Pin Configuration
C3	OC1STATUS	RW	R	00————	OC1 IRQMCU	OC1 IRQMCE	OC1 CUNDER	OC1 CEMPTY	OC1 IRQEF	OC1 IRQER	OC1 IRQRF	OC1 IRQRR	Output Compare 1 Status
Input Capture 0													
F820	IC0CAPT0	R	R	————	IC0CAPT(7:0)								Input Capture 0 Capture Value
F821	IC0CAPT1	R	R	————	IC0CAPT(15:8)								Input Capture 0 Capture Value
CC	IC0MODE	RW	R	–0000000	–	IC0 IRQMCO	IC0 IRQMCF	IC0 LBUF	IC0EDGE(1:0)		IC0MODE(1:0)		Input Capture 0 Mode
CD	IC0STATUS	RW	R	0000——	IC0TRIG(3:0)				–	–	IC0 IRQE	IC0 IRQR	Input Capture 0 Status
Input Capture 1													
F820	IC1CAPT0	R	R	————	IC1CAPT(7:0)								Input Capture 1 Capture Value
F821	IC1CAPT1	R	R	————	IC1CAPT(15:8)								Input Capture 1 Capture Value
D4	IC1MODE	RW	R	–0000000	–	IC1 IRQMCO	IC1 IRQMCF	IC1 LBUF	IC1EDGE(1:0)		IC1MODE(1:0)		Input Capture 1 Mode
D5	IC1STATUS	RW	R	0000——	IC1TRIG(3:0)				–	–	IC1 IRQE	IC1 IRQR	Input Capture 1 Status
SPI													
DE	SPSHREG	RW	R	————	SPSHREG(7:0)								SPI Shift Register

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
DF	SPCLKSRC	RW	R	00000111	SP SCK PH	SP SCK INV	SPSCKDIV(2:0)			SPSCKSRC(2:0)			SPI Clock Source
DC	SPMODE	RW	R	—000000	–	–	SPSSIE	SPTXIE	SPRXIE	SPDIR	SPMODE(1:0)		SPI Mode
DD	T0STATUS	R	R	————	–	SP FIRST	SPSS STAT	SPSS CHG	SPTX UNDER	SPTX EMPTY	SPRX OVER	SPRX FULL	SPI Status
UART 0													
E6	U0SHREG	RW	R	————	U0SHREG(7:0)								UART 0 Shift Register
E7	U0MODE	RW	R	—000000	U0TXBRK	U0RXDGL	U0STOP	U0WL(2:0)			U0BRG(1:0)		UART 0 Mode
E4	U0CTRL	RW	R	00000000	U0 TX8	U0 MCE	U0 BRKIE	U0 FEIE	U0 TXIE	U0 RXIE	U0 TXEN	U0 RXEN	UART 0 Control
E5	U0STATUS	R	R	————	U0 RX8	–U0TX IDLE	U0 BRKDET	U0 FERR	U0TX UNDER	U0TX EMPTY	U0RX OVER	U0RX FULL	UART 0 Status
UART 1													
EE	U1SHREG	RW	R	————	U1SHREG(7:0)								UART 1 Shift Register
EF	U1MODE	RW	R	—000000	U1TXBRK	U1RXDGL	U1STOP	U1WL(2:0)			U1BRG(1:0)		UART 1 Mode
EC	U1CTRL	RW	R	00000000	U1 TX8	U1 MCE	U1 BRKIE	U1 FEIE	U1 TXIE	U1 RXIE	U1 TXEN	U1 RXEN	UART 1 Control
ED	U1STATUS	R	R	————	U1 RX8	–	U1 BRKDET	U1 FERR	U1TX UNDER	U1TX EMPTY	U1RX OVER	U1RX FULL	UART 1 Status
FLASH Controller													
92	NVADDR0	RW	R	00000000	NVADDR(7:0)								Flash Address Register
93	NVADDR1	RW	R	00000000	NVADDR(15:8)								Flash Address Register
94	NVDATA0	RW	R	00000000	NVDATA(7:0)								Flash Data Register
95	NVDATA1	RW	R	00000000	NVDATA(15:8)								Flash Data Register
91	NVSTATUS	R	R	————	–	–	–	–	–	–	NVUN LOCK	NV BUSY	Flash Status Register

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
		W	R	————	NVCMD(7:0)								Flash Command Register
96	NVKEY	W	R	————	NVKEY(7:0)								Flash Unlock Key Register
ROM Controller													
97	CODECONFIG	RW	R	110–0000	CACHE	PRE FETCH INSN	PRE FETCH MOVC	INVLD	XRAML8	0	0	0	Code Address Space Configuration
ADC													
D1	ADCCLKSRC	RW	R	00000111	ADC ACT	ADCCLKDIV(3:0)				ADCCLKSRC(2:0)			ADC Clock Source
C9	ADCCONV	RW	R	00000000	ADC IRQ	ADC BUSY	ADC PEND	–	–	CONVSRC(2:0)			ADC Conversion Control
7028	ADCTUNE0	RW	R	00000001	ADCTUNE0(7:0)								ADC Tuning 0
7029	ADCTUNE1	RW		00000010	ADCTUNE1(7:0)								ADC Tuning 1
CA	ADCCH0CONFIG	RW	R	11111111	CH0MODE(1:0)		CH0INN(2:0)			CH0INP(2:0)			ADC Channel 0 Configuration
CB	ADCCH1CONFIG	RW	R	11111111	CH1MODE(1:0)		CH1INN(2:0)			CH1INP(2:0)			ADC Channel 1 Configuration
D2	ADCCH2CONFIG	RW	R	11111111	CH2MODE(1:0)		CH2INN(2:0)			CH2INP(2:0)			ADC Channel 2 Configuration
D3	ADCCH3CONFIG	RW	R	11111111	CH3MODE(1:0)		CH3INN(2:0)			CH3INP(2:0)			ADC Channel 3 Configuration
7020	ADCCH0VAL0	R	R	————	CH0VAL(7:0)								ADC Channel 0 Value
7021	ADCCH0VAL1	R	R	————	CH0VAL(15:8)								ADC Channel 0 Value
7022	ADCCH1VAL0	R	R	————	CH1VAL(7:0)								ADC Channel 1 Value
7023	ADCCH1VAL1	R	R	————	CH1VAL(15:8)								ADC Channel 1 Value
7024	ADCCH2VAL0	R	R	————	CH2VAL(7:0)								ADC Channel 2 Value
7025	ADCCH2VAL1	R	R	————	CH2VAL(15:8)								ADC Channel 2 Value
7026	ADCCH3VAL0	R	R	————	CH3VAL(7:0)								ADC Channel 3 Value
7027	ADCCH3VAL1	R	R	————	CH3VAL(15:8)								ADC Channel 3 Value

Addr	Name	Dir	R	Reset	Bit								Description
Hex					7	6	5	4	3	2	1	0	
702A	ADCTUNE2	RW	R	11101000	WAKEP(2:0)			WAKEC(2:0)			PMODE(1:0)		ADC Power Saving Modes
E1	ANALOGCOMP	RW	R	—000000	ACOPM1 ST	ACOMP0 ST	ACOMP1 INV	ACOMP0 INV	ACOMP1 REF	ACOMP1 IN	ACOMP0 REF	ACOMP0 IN	Analog Comparators
Revision													
7F00	SILICONREV	R	R	1000111X	SILICONREV(7:0)								Silicon Revision
Crystal Oscillator													
7F18	XTALOSC	RW	R	----0100	–	–	–	–	XTALOSCGM(3:0)				GM of Crystal Oscillator
7F19	XTALAMPL	RW	R	1----000	XTAL REG ON	–	–	–	–	XTALREGVC(2:0)			Crystal Oscillator Critical Amplitude
7F1A	XTALREADY	RW	R	----0001	XTAL SIG DET	XTAL READY	–	–	XTAL AMP DIS	XTALREADYMODE(2:0)			Crystal Oscillator Ready Detection Mode

5. FLASH

The FLASH is the user rewritable non-volatile memory. It is organized as 64 1kByte pages. It may be erased page-wise, and written as 16Bit words. The small sector size enables the FLASH to be used for configuration data where traditionally E^2PROM would have been used, obviating the need for additional E^2PROM.

5.1. Features

- The FLASH size is 64kBytes
- Word size is 16 Bits
- Erase Sector size is 1kByte
- Secure erase ensures that the main flash data is fully erased before erasing the protect and key bits

5.2. Register: NVADDR0, NVADDR1

Name	Bits	R/W	Reset	Description
NVADDR	15:0	RW	0x0000	Flash Address Register

This register stores the address for FLASH write or page erase operations.

5.3. Register: NVDATA0, NVDATA1

Name	Bits	R/W	Reset	Description
NVDATA	15:0	RW	0x0000	Flash Data Register

This register stores the data for FLASH write operations.

5.4. Register: NVSTATUS

Name	Bits	R/W	Reset	Description
NVBUSY	0	R	–	Indicates the Flash Controller is Busy. This bit can only be observed when executing from memory other than Flash

NVUNLOCK	1	R	–	Indicates that the Flash Controller is unlocked, i.e. the unlock sequence has been successfully written to the NVKEY register
NVCMD	7:0	W	–	Flash Controller Commands Command – Meaning 0x00 – NOP 0x10 – Bulk Erase 0x20 – Flash Page Erase 0x30 – Flash Word Write 0x34 – Flash Protect Bits Write

The NVSTATUS register reports FLASH controller status when read and serves as a command register when written.

5.5. Register: NVKEY

Name	Bits	R/W	Reset	Description
NVKEY	7:0	W	–	Unlock Key Register

Normally, the Flash controller is locked, and commands (other than NOP) are not executed. This is indicated by NVUNLOCK in register NVSTATUS being zero.

To unlock the Flash controller, first 0x41, then 0x78, must be written to NVKEY within 16 System clock cycles.

To lock the Flash controller again, write any other value to the NVKEY register.

This mechanism ensures that the FLASH contents are not changed inadvertently.

5.6. FLASH Protect Bits

The FLASH controller features individual erase and write protect bits for each FLASH sector. To protect a sector, write the address from the table below into the NVADDR register, write all ones except for the bit corresponding to the sector to be protected, which should be cleared, to the NVDATA register, and then issue the Flash Protect Bits Write command.

Note that Protect bits can only be set by a Bulk Erase command, which clears the whole FLASH. So once protected, a sector cannot be unprotected except by clearing the complete FLASH.

Address	F E D C B A 9 8 7 6 5 4 3 2 1 0	
0x0000	$\overline{\text{ERASEPROTECT}}$(15:0)	Erase Protect Bits
0x0002	$\overline{\text{ERASEPROTECT}}$(31:16)	
0x0004	$\overline{\text{ERASEPROTECT}}$(47:32)	
0x0006	$\overline{\text{ERASEPROTECT}}$(63:48)	
0x0008	$\overline{\text{WRITEPROTECT}}$(15:0)	Write Protect Bits
0x000A	$\overline{\text{WRITEPROTECT}}$(31:16)	
0x000C	$\overline{\text{WRITEPROTECT}}$(47:32)	
0x000E	$\overline{\text{WRITEPROTECT}}$(63:48)	

6. Cache and Prefetch

AX8052 contains a small, 4 Words × 16 Bits fully associative write-through cache to hide the FLASH access latencies. The cache is consistent with respect to writes over the code bus (MOVC), i.e. a write that hits in the cache invalidates the respective cache line. The cache is not consistent with writes to the code memories over other interfaces, especially the FLASH using the NV controller, and to XRAM via the XDATA bus. In this case, the cache must be manually flushed. The replacement strategy is pseudo-LRU.

The cache also contains the prefetch controller. Prefetching hides memory latencies for streaming accesses (i.e. program fetching without jumps). Prefetching may be separately enabled/disabled for code fetches and MOVC reads.

6.1. Register: CODECONFIG

Name	Bits	R/W	Reset	Description
MBZ	0	RW	0	Must be zero
MBZ	1	RW	0	Must be zero
MBZ	2	RW	0	Must be zero
XRAML8	3	RW	0	Enable XRAM in code address space 0x0000–0x1FFF
INVLD	4	W	–	Writing this bit as 1 invalidates the cache
PREFETCHMOVC	5	RW	0	Enable Code Memory MOVC Prefetching
PREFETCHINSN	6	RW	1	Enable Code Memory Instruction Prefetching
CACHE	7	RW	1	Enable Code Memory Caching

The XRAML8 bit maps, if set, the 8kBytes XRAM into code memory space. This allows code to execute from XRAM.

7. RAM

AX8052 contains 256 Bytes of IRAM and 8 kBytes of XRAM. The 8 kBytes XRAM are split into two 4 kBytes halves. Both halves can be individually preserved or switched off during sleep mode (refer to the PCON register). The XRAM memory can be accessed in a cycle-steal fashion by the DMA controller and the AES crypto engine.

8. Debug Interface

The main purpose of the Debug Interface is to aid debugging by allowing the core to be stopped, to examine state, and to set breakpoints. It is also used for in-circuit system programming and for testing.

8.1. Features

- 3 Wire (1 dedicated, 2 shared) synchronous serial interface
- Code protection prevents unauthorized access to the Firmware, while still allowing full debug capability to authorized users
- if Code Protection is enabled, only Secure Bulk FLASH erase may be executed by unauthorized users
- Boundary Scan (proprietary, not 1149.1 compliant) can be emulated by injecting instructions to read/write GPIO port registers
- Unlimited number of break points
- Single stepping support
- DebugLink allows printf style debugging into a debugger window without the need for dedicated pins. DebugLink is accessed on the microcontroller side using the DBGLNKSTAT and DBGLNKBUF registers documented below. It is similar to using an UART.

8.2. Register: DBGLNKSTAT

Name	Bits	R/W	Reset	Description
DBGRXFULL	0	R	–	Rx Full interrupt request
DBGRXOVER	1	R	–	Rx Overrun interrupt request (clears on read)

DBGTXEMPTY	2	R	–	Tx Empty interrupt request
DBGTXUNDER	3	R	–	Tx Underrun interrupt request (clears on read)
DBGEN	4	R	–	Status of the DBG_EN input (i.e. 1 if the debug interface is enabled) (reading clears the DBG_EN change interrupt)
DBGENCHGIE	5	RW	0	DBG_EN change interrupt enable
DBGRXIE	6	RW	0	Receiver interrupt enable
DBGTXIE	7	RW	0	Transmitter interrupt enable

8.3. Register: DBGLNKBUF

Name	Bits	R/W	Reset	Description
DBGBUF	7:0	RW	–	DebugLink Buffer Register

9. System Controller

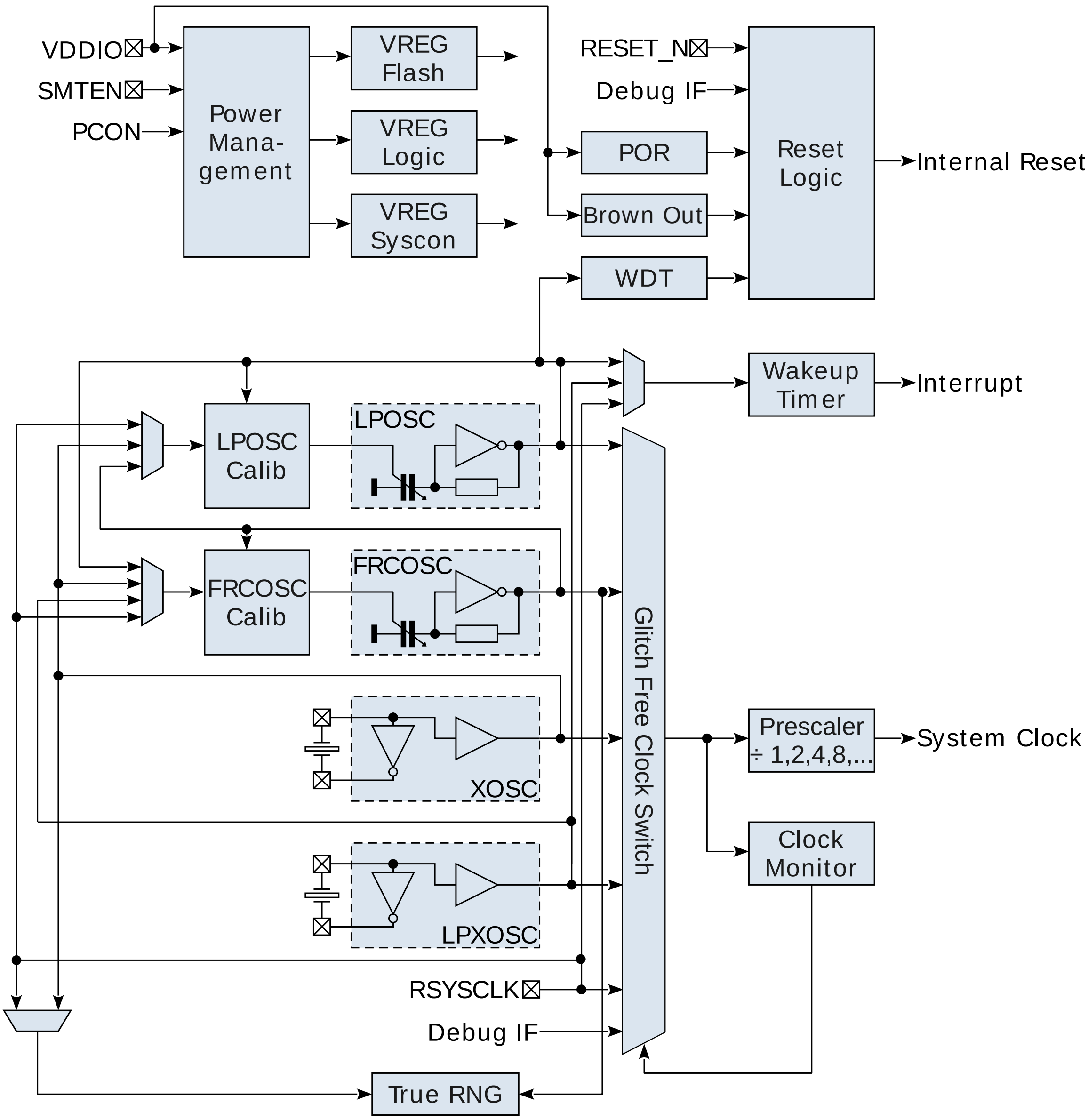


Figure 6: System Controller Block Diagram

The system controller generates the power supplies, reset and clock signals. Figure 6 shows the block diagram of the System Controller.

9.1. Features

The system controller contains several reset sources, namely:

- RESET_N Pin
- Power On Reset
- Brown Out Detector
- Watchdog Timer
- Software Reset (SWRESET Bit in PCON)

The PCON Register allows the software to query the reset cause. Since CPU registers are not preserved during sleep and deep sleep modes, after waking up from sleep or deep sleep the CPU starts executing at the reset vector as well. Bit 6 of PCON allows the software to distinguish wake up from reset events.

The system controller is also responsible for generating the system clock. A glitch free clock multiplexer, controlled by the CLKCON register, selects one clock source among the following oscillators:

- Fast RC Oscillator 20MHz
- Low Power Oscillator 10kHz/640Hz
- Radio Chip (SYSCLK)
- Fast Crystal Oscillator
- Low Power Tuning Fork Crystal Oscillator

Because clock switching must be glitch free, switching the clock source takes several cycles of the old as well as the new clock source. Clock switching status may be queried using the CLKSTAT register. An optional clock monitor may observe the output of the clock multiplexer and switch back to the fast RC oscillator, should the selected clock source cease to toggle. The clock multiplexer is followed by a prescaler, also controlled by the CLKCON register, which allows the CPU execution speed and thus current consumption to be closely tuned to the task at hand.

A wakeup timer with two counter (time) registers and four event (match) registers provides flexible timed wakeups.

Calibration logic for the Low Power oscillator and the Fast RC oscillator allows these oscillators to be automatically calibrated against a more precise (e.g. Crystal) oscillator. Calibration may be programmed to automatically run whenever the chosen reference oscillator is switched on.

A programmable watchdog may be used to detect erratic firmware behavior and to reset the chip on such an event.

9.2. Power Modes

AX8052 supports different power modes, that differ in current consumption, the amount of state (memory) retained, and the possible wakeup sources. Power modes are controlled via the PCON register.

Name	Wakeup	Current[3]	Retention	Description
Deep Sleep	one dedicated pin	50nA	32bit SCRATCH Register	lowest power, chip completely powered down except for SCRATCH Register
Sleep	Wakeup Timer (LPOSC/LPXOSC/ RSYSCLK), Interrupt on Port Change Pxx	500nA–1.5μA	32bit SCRATCH Register, IRAM, XRAM (programmable)	system clock stopped, FLASH powered down, all peripherals powered down
Standby	any enabled interrupt	85μA + running peripherals	RAM, all Registers	system clock running if needed, FLASH powered down, some peripherals may continue running depending on user choice
Running	n/a	150μA/MHz	n/a	different selectable clock sources

3 Current Consumption is approximate only, and may be different for different members of the AX8052 family. Please refer to the appropriate datasheet for accurate current consumption values

9.3. Register: PCON

<table>
<tr><th>Name</th><th>Bits</th><th>R/W</th><th>Reset</th><th>Description</th></tr>
<tr><td>PWRMODE</td><td>1:0</td><td>RW</td><td>00</td><td>Set Power Mode (see section 9.2 for a documentation of the Power Modes)
<table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>Running</td></tr><tr><td>01</td><td>Standby</td></tr><tr><td>10</td><td>Sleep</td></tr><tr><td>11</td><td>Deepsleep</td></tr></table></td></tr>
<tr><td>XRAMKEEP</td><td>3:2</td><td>RW</td><td>00</td><td>specify which of the XRAM halves should be kept on during sleep</td></tr>
<tr><td>SWRESET</td><td>4</td><td>RW</td><td>0</td><td>Software Reset; Writing 1 resets the Chip; Reading 1 indicates that the last reset was caused by a software reset</td></tr>
<tr><td>WDTRESET</td><td>5</td><td>R</td><td>–</td><td>Watchdog Reset; Reading 1 indicates that the last reset was caused by a watchdog reset</td></tr>
<tr><td>WAKEUP</td><td>6</td><td>R</td><td>–</td><td>Wakeup Reset; Reading 1 indicates that the last reset was caused by a wakeup from sleep or deep sleep</td></tr>
<tr><td>BROWNOUT</td><td>7</td><td>RC</td><td>–</td><td>Brownout (Interrupt)</td></tr>
</table>

When writing PWRMODE=01 to enter standby mode, the following instruction may be executed before the microcontroller is stopped. It is therefore recommended to use enter_standby() from libmf, or to add a NOP instruction after PWRMODE=01 writes.

When using the debugger, standby mode may be terminated by changing the PWRMODE bits of PCON to 00, using the register window.

When entering Sleep mode, the microcontroller and most peripherals, except the wakeup timer and the GPIO logic, are powered down. When entering Deep Sleep mode, the microcontroller and all peripherals are powered down, and the GPIO pins are frozen (GPIOENABLE is reset to zero). When waking up from Sleep or Deep Sleep mode, the microcontroller re-starts executing at address 0x0000, and the registers of powered-down peripherals are reset, similar to power-on reset or releasing RESET_N. Wakeup may be distinguished from reset by examining the WAKEUP bit of register PCON.

Since the debug interface logic is in the microcontroller power domain, a connected debugger prevents the microcontroller from being powered down. There are therefore some differences between running with debugger versus running stand-alone with respect to

Sleep and Deep Sleep modes. It is therefore recommended to use the enter_sleep() and enter_deepsleep() routines from libmf, which minimize the differences. Furthermore, power consumption in Sleep and Deep Sleep modes are considerably higher with the debugger connected.

9.4. Register: CLKCON

<table>
<tr><th>Name</th><th>Bits</th><th>R/W</th><th>Reset</th><th>Description</th></tr>
<tr><td>CLKSRC</td><td>2:0</td><td>RW</td><td>000</td><td>Requested System Clock Source
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>000</td><td>FRCOSC</td></tr>
<tr><td>001</td><td>LPOSC</td></tr>
<tr><td>010</td><td>XOSC</td></tr>
<tr><td>011</td><td>LPXOSC</td></tr>
<tr><td>100</td><td>RSYSCLK</td></tr>
<tr><td>101</td><td>invalid</td></tr>
<tr><td>110</td><td>invalid</td></tr>
<tr><td>111</td><td>invalid</td></tr>
</table></td></tr>
<tr><td>CLKPRE</td><td>5:3</td><td>RW</td><td>001</td><td>Requested System Clock Prescaler
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>000</td><td>÷1</td></tr>
<tr><td>001</td><td>÷2</td></tr>
<tr><td>010</td><td>÷4</td></tr>
<tr><td>011</td><td>÷8</td></tr>
<tr><td>100</td><td>÷16</td></tr>
<tr><td>101</td><td>÷32</td></tr>
<tr><td>110</td><td>÷64</td></tr>
<tr><td>111</td><td>÷128</td></tr>
</table></td></tr>
</table>

CLKMON	7:6	RW	00	Clock Monitor Period Bits / Meaning 00 Off 01 4 LPOSC Periods 10 16 LPOSC Periods 11 64 LPOSC Periods

Do not select XOSC or LPXOSC unless a crystal is connected – see OSCFORCERUN for details.

9.5. Register: CLKSTAT

Name	Bits	R/W	Reset	Description
CLKSRCST	2:0	R	–	Currently selected System Clock Source Bits / Meaning 000 FRCOSC 001 LPOSC 010 XOSC 011 LPXOSC 100 RSYSCLK 101 invalid 110 invalid 111 invalid

CLKPREST	5:3	R	–	Currently selected System Clock Prescaler Bits / Meaning 000 ÷1 001 ÷2 010 ÷4 011 ÷8 100 ÷16 101 ÷32 110 ÷64 111 ÷128
CLKLOSS	7	RC	–	Clock Loss Detected and Switched Back to Default

This register reflects the currently active clock settings. Normally, it reflects the values of the CLKCON register. Since clock switching takes time, however, it will not immediately take on new values.

9.6. Register: WTCFGA, WTCFGB

These two registers select the clock sources for both wakeup timer counter registers.

Name	Bits	R/W	Reset	Description
WTSRCA WTSRCB	2:0	RW	111 111	Bits – Meaning 000 – FRCOSC 001 – LPOSC 010 – XOSC 011 – LPXOSC 100 – RSYSCLK 101 – invalid 110 – invalid 111 – Off
WTDIVA WTDIVB	5:3	RW	001 001	Bits – Meaning 000 – ×2 001 – ×1 010 – ÷2 011 – ÷4 100 – ÷8 101 – ÷16 110 – ÷32 111 – ÷64

Do not select XOSC or LPXOSC unless a crystal is connected – see OSCFORCERUN for details.

9.7. Register: WTIRQEN

Name	Bits	R/W	Reset	Description
WTIAA	0	RW	0	Wakeup Timer Interrupt on Event A match with Counter A
WTIBA	1	RW	0	Wakeup Timer Interrupt on Event B match with Counter A
WTICA	2	RW	0	Wakeup Timer Interrupt on Event C match with Counter A
WTIDA	3	RW	0	Wakeup Timer Interrupt on Event D match with Counter A

WTIAB	4	RW	0	Wakeup Timer Interrupt on Event A match with Counter B
WTIBB	5	RW	0	Wakeup Timer Interrupt on Event B match with Counter B
WTICB	6	RW	0	Wakeup Timer Interrupt on Event C match with Counter B
WTIDB	7	RW	0	Wakeup Timer Interrupt on Event D match with Counter B

9.8. Register: WTSTAT

Name	Bits	R/W	Reset	Description
WTSAA	0	RC	–	Wakeup Timer Event A matched with Counter A; cleared on read
WTSBA	1	RC	–	Wakeup Timer Event B matched with Counter A; cleared on read
WTSCA	2	RC	–	Wakeup Timer Event C matched with Counter A; cleared on read
WTSDA	3	RC	–	Wakeup Timer Event D matched with Counter A; cleared on read
WTSAB	4	RC	–	Wakeup Timer Event A matched with Counter B; cleared on read
WTSBB	5	RC	–	Wakeup Timer Event B matched with Counter B; cleared on read
WTSCB	6	RC	–	Wakeup Timer Event C matched with Counter B; cleared on read
WTSDB	7	RC	–	Wakeup Timer Event D matched with Counter B; cleared on read

Match flags are cleared on read of this register. Note however that the match condition has level triggered semantics; if the match condition persists after reading this register, the match flag will immediately be set again. So the recommended sequence of events upon a wakeup timer match are:

1. Read the WTSTAT register to determine which events matched

2. Update the corresponding WTEVT registers

3. Reread WTSTAT to clear the match bits of the event registers modified in step 2 above; check if any new matches corresponding to other event registers happened in the meantime; if so, go back to step 2

WTSTAT bits are set on match regardless of the setting of the corresponding WTIRQEN bits. There is no mechanism available for atomic WTEVT updates, both halves need to be updated separately. This gives rise to the possibility of matches to intermediate values. Therefore, the WTSTAT bits corresponding to the WTEVT register being updated may be set. It is therefore recommended to read WTSTAT after updating a WTEVT register to clear the WTSTAT bits corresponding to the changed WTEVT register (but do not loose WTSTAT bits of other WTEVT registers). Due to the level triggered semantics of the WTSTAT bits, if WTEVT is still equal to a WTCNT register, the respective WTSTAT bit will be re-set to one immediately.

9.9. Register: WTCNTA0, WTCNTA1, WTCNTB0, WTCNTB1

Name	Bits	R/W	Reset	Description
WTCNTA WTCNTB	15:0	R	–	Wakeup Counter

9.10. Register: WTCNTR1

Name	Bits	R/W	Reset	Description
WTCNTR1	7:0	R	–	Wakeup Counter Register; when reading WTCNTA0, WTCNTA1 is latched into WTCNTR1; when reading WTCNTB0, WTCNTB1 is latched into WTCNTR1; this allows reading counter registers atomically

9.11. Register: WTEVTA0, WTEVTA1, WTEVTB0, WTEVTB1, WTEVTC0, WTEVTC1

The wakeup timer contains four event registers and two counter registers. All event registers are continuously compared to all counter registers, and match events are generated if any match is detected. These are the four match registers.

Name	Bits	R/W	Reset	Description
WTEVTA WTEVTB WTEVTC WTEVTD	15:0	RW	0x0000	Wakeup Timer Event Register

9.12. Register: WDTCFG

The watchdog timer runs off the low power oscillator. Once enabled, the watchdog timer must be periodically reset by writing to the WDTRESET (see below), otherwise the watchdog resets the CPU.

<table>
<tr><th>Name</th><th>Bits</th><th>R/W</th><th>Reset</th><th>Description</th></tr>
<tr><td>WDTDIV</td><td>3:0</td><td>RW</td><td>0x0000</td><td>Watchdog Timer Divider; This field can only be changed if both WDTENA and WDTLCK is zero.
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>0000</td><td>÷2</td></tr>
<tr><td>0001</td><td>÷4</td></tr>
<tr><td>0010</td><td>÷8</td></tr>
<tr><td>0011</td><td>÷16</td></tr>
<tr><td>0100</td><td>÷32</td></tr>
<tr><td>0101</td><td>÷64</td></tr>
<tr><td>0110</td><td>÷128</td></tr>
<tr><td>0111</td><td>÷256</td></tr>
<tr><td>1000</td><td>÷512</td></tr>
<tr><td>1001</td><td>÷1024</td></tr>
<tr><td>1010</td><td>÷2048</td></tr>
<tr><td>1011</td><td>÷4096</td></tr>
<tr><td>1100</td><td>÷8192</td></tr>
<tr><td>1101</td><td>÷16384</td></tr>
</table></td></tr>
</table>

				1110 ÷32768 1111 ÷65536
WDTENA	4	RW	0	Watchdog Timer Enable
WDTLCK	5	RW	0	Watchdog Timer Configuration Lock; when set to 1, this register may no longer be changed

9.13. Register: WDTRESET

Name	Bits	R/W	Reset	Description
WDTRESET	7:0	W	–	If 10101110 is written to this register, the watchdog is reset

9.14. Register: LPOSCCONFIG

Name	Bits	R/W	Reset	Description
LPOSC CALSRC	2:0	RW	001	Low Power Oscillator Calibration Source Bits Meaning 000 FRCOSC 001 Off (no Calibration) 010 XOSC 011 invalid 100 RSYSCLK 101 invalid 110 invalid 111 invalid

LPOSC PRESC	5:3	RW	000	Bits / Meaning: 000 ×2 001 ×1 010 ÷2 011 ÷4 100 ÷8 101 ÷16 110 ÷32 111 ÷64
LPOSC FAST	6	RW	0	Select the Frequency of the Low Power Oscillator. 0=640Hz, 1=10.24kHz

Note that the selected reference oscillator is not automatically turned on. The purpose of this feature is to allow "opportunistic" calibration, i.e. the calibration logic is always turned on, but is inactive until the reference oscillator is needed for another purpose. The reference oscillator can, however, be forced to run using the OSCFORCERUN register.

The Radio Clock does not feature a ready signal, therefore calibration starts immediately if the Radio Clock is selected as reference clock. It is the responsibility of the user to make sure the Radio Clock is enabled and runs before selecting it as reference clock. For Axsem radios, this usually means:

- setting the power mode register of the radio appropriately such that the oscillator circuitry is enabled
- configuring SYSCLK
- configuring PORTR, DIRR and RADIOMUX

libmf provides functions to enable and disable the Radio Clock for Axsem radios.

9.15. Register: LPOSCKFILT1, LPOSCKFILT0

Name	Bits	R/W	Reset	Description
LPOSCKFILT	15:0	RW	0x20C4	k_{FILT} (Low Power Oscillator Calibration Filter Constant)

The maximum value of k_{FILT}, that results in quickest calibration (single cycle), but no jitter suppression, is:

$$k_{FILT} = \left\lceil \frac{PRESCALER \cdot 2^{20}}{f_{REF} \cdot \tau_{base} \cdot k_{LPOSC}} \right\rceil = \left\lceil \frac{512000\text{Hz} \cdot PRESCALER \cdot 2^{20}}{f_{REF}} \right\rceil$$

Smaller values of k_{FILT} result in longer calibration, but increased jitter suppression.

9.16. Register: LPOSCREF1, LPOSCREF0

Name	Bits	R/W	Reset	Description
LPOSCREF	15:0	RW	0x61A8	LP Oscillator Reference Frequency Divider; set to $\frac{f_{REF}}{640\text{Hz} \cdot PRESCALER}$

9.17. Register: LPOSCFREQ1, LPOSCFREQ0

Name	Bits	R/W	Reset	Description
LPOSCFREQ	9:-2	RW	0x000	LP Oscillator Frequency Tune Value; in $\frac{1}{32}$ %.

9.18. Register: LPOSCPER1, LPOSCPER0

Name	Bits	R/W	Reset	Description
LPOSCPER	15:0	R	–	Last measured LP Oscillator Period

9.19. Register: FRCOSCCONFIG

Name	Bits	R/W	Reset	Description
FRCOSC CALSRC	2:0	RW	000	Fast RC Oscillator Calibration Source Bits – Meaning 000 – Off (no Calibration) 001 – LPOSC 010 – XOSC 011 – LPXOSC 100 – RSYSCLK 101 – invalid 110 – invalid 111 – invalid

FRCOSC PRESC	6:3	RW	0000	Bits / Meaning: 0000 ×2 0001 ÷1 0010 ÷2 0011 ÷4 0100 ÷8 0101 ÷16 0110 ÷32 0111 ÷64 1000 ÷128 1001 ÷256 1010 ÷512 1011 ÷1024 1100 ÷2048 1101 ÷4096 1110 ÷8192 1111 ÷16384
FRCOSC PERGAIN	7	RW	0	Period Gain; if 1, the measured period is multiplied by 16

Choose a reference clock source, and choose the reference clock prescaler such that the resulting reference frequency is above 500Hz. For best frequency measurement precision, the resulting reference frequency should be between 500Hz and 1kHz, but faster reference frequencies are permissible for faster acquisition.

Note that the selected reference oscillator is not automatically turned on. The purpose of this feature is to allow "opportunistic" calibration, i.e. the calibration logic is always turned on, but is inactive until the reference oscillator is needed for another purpose. The reference oscillator can, however, be forced to run using the OSCFORCERUN register.

The Radio Clock does not feature a ready signal, therefore calibration starts immediately if the Radio Clock is selected as reference clock. It is the responsibility of the user to make sure the Radio Clock is enabled and runs before selecting it as reference clock. For Axsem radios, this usually means:

- setting the power mode register of the radio appropriately such that the oscillator circuitry is enabled
- configuring SYSCLK
- configuring PORTR, DIRR and RADIOMUX

libmf provides functions to enable and disable the Radio Clock for Axsem radios.

9.20. Register: FRCOSCKFILT1, FRCOSCKFILT0

Name	Bits	R/W	Reset	Description
FRCOSCKFILT	15:0	RW	0x20C4	k_{FILT} (Fast RC Oscillator Calibration Filter Constant)

The maximum value of k_{FILT}, that results in quickest calibration (single cycle), but no jitter suppression, is:

For PERGAIN=0, it is:

$$k_{FILT} = \left\lceil \frac{2^{20}}{f_{base} \cdot \tau_{REF} k_{FRCOSC}} \right\rceil = \left\lceil \frac{f_{REF} \cdot 2^{20}}{15\text{kHz} \cdot PRESCALER} \right\rceil$$

For PERGAIN=1, it is:

$$k_{FILT} = \left\lceil \frac{2^{16}}{f_{base} \cdot \tau_{REF} \cdot k_{FRCOSC}} \right\rceil = \left\lceil \frac{f_{REF} \cdot 2^{16}}{15\text{kHz} \cdot PRESCALER} \right\rceil$$

Smaller values of k_{FILT} result in longer calibration, but increased jitter suppression.

9.21. Register: FRCOSCREF1, FRCOSCREF0

Name	Bits	R/W	Reset	Description
FRCOSCREF	15:0	RW	0x61A8	Fast RC Oscillator Reference Frequency Divider; set to $\frac{20\text{MHz} \cdot PRESCALER}{f_{REF}}$

9.22. Register: FRCOSCFREQ1, FRCOSCFREQ0

Name	Bits	R/W	Reset	Description
FRCOSCFREQ	9:-2	RW	0x000	Fast RC Oscillator Frequency Tune Value; in $\frac{1}{32}$ %.

9.23. Register: FRCOSCPER1, FRCOSCPER0

Name	Bits	R/W	Reset	Description
FRCOSCPER	15:0	R	–	Last measured Fast RC Oscillator Period

9.24. Register: OSCFORCERUN

Normally, oscillators are automatically enabled when needed, except for oscillator calibration. This register allows oscillators to be turned on even if no peripheral needs the clock. This can be used to force oscillator calibration, or to hide startup latencies of the crystal oscillators when their future need is anticipated.

Name	Bits	R/W	Reset	Description
FRCOSC FRUN	0	RW	0	If 1, the Fast RC Oscillator is always enabled
LPOSC FRUN	1	RW	0	If 1, the Low Power Oscillator is always enabled
XOSC FRUN	2	RW	0	If 1, the Crystal Oscillator is always enabled
LPXOSC FRUN	3	RW	0	If 1, the Low Power Crystal Oscillator is always enabled

Do not enable the Crystal Oscillator or the Low Power Crystal Oscillator unless an appropriate Crystal is connected to the Oscillator pins and the pins are configured as analog input. Once enabled, the Crystal Oscillators will not disable again until they see two falling edges on their Oscillator pins, or until reset is applied. If no crystal is connected, or if the pins are not configured as analog inputs, these falling edges will not be seen.

If accidentally enabled, the oscillator may be turned off again by calling libmf routines turn_off_xosc or turn_off_lpxosc. These routines, however, need to toggle pins PA0/PA1 or PA4/PA5, respectively, so they can only be used if toggling these pins does not upset the board periphery.

9.25. Register: OSCRUN

This register indicates which oscillator is turned on.

Name	Bits	R/W	Reset	Description
FRCOSC RUN	0	R	–	1 indicates the Fast RC oscillator is enabled
LPOSC RUN	1	R	–	1 indicates the Low Power oscillator is enabled
XOSC RUN	2	R	–	1 indicates the Crystal oscillator is enabled
LPXOSC RUN	3	R	–	1 indicates the the Low Power Crystal oscillator is enabled

9.26. Register: OSCREADY

This register indicates which oscillator is turned on and is operating. The crystal oscillators take some time between switching on and achieving stable oscillation. On these oscillators, RUN reads one and RDY reads zero during the startup period.

Name	Bits	R/W	Reset	Description
FRCOSC RDY	0	R	–	1 indicates the Fast RC oscillator is running and stable
LPOSC RDY	1	R	–	1 indicates the Low Power oscillator is running and stable

XOSC RDY	2	R	–	1 indicates the Crystal oscillator is running and stable
LPXOSC RDY	3	R	–	1 indicates the the Low Power Crystal oscillator is running and stable

9.27. Register: OSCCALIB

This register selects the events that will generate a clock management interrupt.

Name	Bits	R/W	Reset	Description
FRCOSCCALIRQE	0	RW	0	Fast RC Oscillator Calibration Interrupt Enable
LPOSCCALIRQE	1	RW	0	Low Power Oscillator Calibration Interrupt Enable
CLKLOSSIRQE	2	RW	1	Clock Loss Interrupt Enable
CLKLOSS	3	R	–	Clock Loss Status (same as CLKSTAT)
FRCOSCCALIRQM	4	R	–	Fast RC Oscillator Calibration Missed
LPOSCCALIRQM	5	R	–	Low Power Oscillator Calibration Missed
FRCOSCCALIRQ	6	R	–	Fast RC Oscillator Calibration Updated Interrupt
LPOSCCALIRQ	7	R	–	Low Power Oscillator Calibration Updated Interrupt

9.28. Register: LPXOSCGM

This register configures the low power (tuning fork) crystal oscillator.

Name	Bits	R/W	Reset	Description
LPXOSCGM	4:0	RW	01000	Load Capacitance Configuration Bits (4:0) / Meaning ⋮ ⋮ 00110 3.5µS ⋮ ⋮ 01000 4.6µS ⋮ ⋮ 01100 6.9µS ⋮ ⋮ 10000 9.1µS ⋮ ⋮
LPXOSCBOOST	7	RW	1	If set, the oscillator is boosted during startup (until the second falling edge is seen)

9.29. Register: MISCCTRL

This register contains miscellaneous control bits.

Name	Bits	R/W	Reset	Description
LPXOSCDIS	0	RW	0	If set, the LPXOSC is permanently disabled. Set this bit if no crystal is connected to PA3/PA4.
XOSCDIS	1	RW	0	If set, the XOSC is permanently disabled. Set this bit if no crystal is connected to PA0/PA1.

This register has no function in silicon revision V1.

9.30. Register: XTALOSC

This register allows the transconductance of the crystal oscillator to be configured. Normally, setting this register is not necessary, as a servo loop controls the transconductance to achieve low amplitude oscillation. This ensures the minimum possible current consumption.

Name	Bits	R/W	Reset	Description
XTALOSCGM	3:0	RW	0100	Gm (Gain) of the Crystal Oscillator Bits / Bias Current 0000: $0\mu S$ 0001: $25\mu S$ 0010: $50\mu S$ 0011: $75\mu S$ ⋮ : ⋮ 1110: $350\mu S$ 1111: $1000\mu S$

9.31. Register: XTALAMPL

This register controls the transconductance servo loop. This register should be left at the default value.

Name	Bits	R/W	Reset	Description
XTALREGVC	2:0	RW	000	Crystal Oscillator Amplitude Bits / Amplitude 000: 180mV 001: 195mV 010: 230mV ⋮ : ⋮ 111: 460mV
XTALREGON	7	RW	1	Crystal Oscillator Amplitude Regulator Enable

9.32. Register: XTALREADY

This register controls the generation of the crystal oscillator ready signal. It should not normally be changed from the default.

Name	Bits	R/W	Reset	Description
XTALREADYMODE	2:0	RW	001	Crystal Oscillator Ready Mode Bits / Meaning 000: Always Ready 001: Ready when XTAL READY = 1 010: Ready when XTAL SIG DET = 1 011: Ready when either XTAL READY = 1 or XTAL SIG DET = 1 (or both) 100: Never Ready 101: Invalid (never ready) 110: Invalid (never ready) 111: Ready when both XTAL READY = 1 and XTAL SIG DET = 1
XTAL AMP DIS	3	RW	0	If set, the crytal oscillator output amplifier is disabled
XTAL READY	6	R	–	Crystal Oscillator READY signal
XTAL SIG DET	7	R	–	Crystal Oscillator SIG DET signal

9.33. Register: SCRATCH0, SCRATCH1, SCRATCH2, SCRATCH3

The main purpose of the scratch registers is to provide 32bit of state storage during deep sleep mode.

Name	Bits	R/W	Reset	Description
SCRATCH	31:0	RW	–	Scratch Register; State is maintained in Deep Sleep

The scratch registers can only be read if GPIOENABLE (Register GPIOENABLE) is set to one.

9.34. Register: SILICONREV

This register returns a silicon revision identification number.

Name	Bits	R/W	Reset	Description
SILICONREV	7:0	RW	1000111X	Silicon Revision

This register may contain the following values:

SILICONREV	Revision Name
0x8E (10001110)	V1
0x8F (10001111)	V1C

10. DMA Controller

The DMA controller transfers data between selected peripherals and the XRAM memory autonomously in the background without microcontroller intervention. This is especially useful as transferring radio chip FIFO data otherwise uses significant microcontroller cycles.

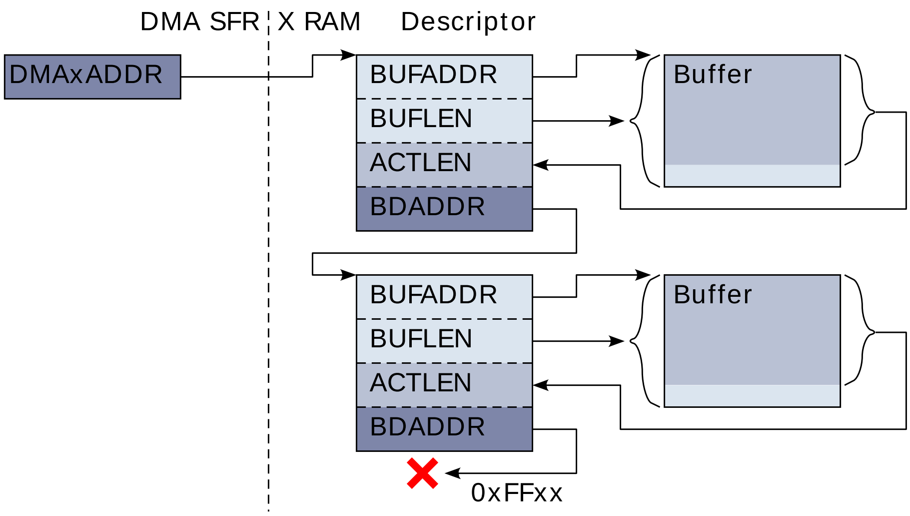


Figure 7: DMA Engine Data Structures

The advanced DMA engine uses buffer descriptors to describe one or multiple consecutive buffers in XRAM. The buffer descriptor is an 8 byte structure located in XRAM. Each buffer descriptor contains a pointer to a buffer fragment in XRAM, and its associated length. Once the DMA controller fills the buffer, the actual length is written back into the buffer descriptor, as well as flag bits, and the DMA engine follows another pointer to the next buffer descriptor. DMAxADDR points to the buffer descriptor the DMA engine is currently working on. It must be set by the Microcontroller before enabling the DMA engine. An address of 0xFFxx (the high byte set) as next buffer address stops the DMA engine. Each buffer descriptor may specify whether an interrupt should be signalled at completion of its buffer.

Figure 7 Shows an illustration of the DMA Engine data structures. This data structure is very flexible. If it is desired to indefinitely repeat the same data (such as in a waveform synthesis application), this can easily be achieved by pointing BDADDR of the last descriptor to the address of the first descriptor. With some care, it is even possible to manipulate the buffer descriptor structure while the DMA engine is running. When replacing a stop marker (0xFFxx) with a buffer descriptor address, always write the low byte of the BDADDR field first. When replacing a buffer address with a stop marker (0xFFxx) always write the high byte first (the low byte may be omitted). Never directly overwrite a buffer address with another one; always first write a stop marker and then overwrite the stop marker. This way, the DMA engine never sees an invalid (half old half new) address.

So to add a new buffer descriptor to the end of a running DMA buffer descriptor chain, first write the address of the new buffer descriptor into the BDADDR field of the last existing buffer descriptor, low byte first. Afterwards, check whether the DMAxADDR SFR has 0xFFxx. In this case, the DMA engine terminated the (old) chain before the microcontroller completed adding the new buffer to the chain. In this case, the DMA channel must be restarted at the newly inserted buffer descriptor.

10.1. Features

- Two DMA controller channels
- Operates on a linked list of buffer descriptors
- Accesses X-Bus and SFR-Bus peripherals in a cycle steal manner
- Most on-chip Peripherals are supported
- Both channels may be chained to support memory to memory copies

10.2. Register: DMA0ADDR0, DMA0ADDR1, DMA1ADDR0, DMA1ADDR1

Name	Bits	R/W	Reset	Description
DMA0ADDR DMA1ADDR	15:0	RW	0xFFFF 0xFFFF	Address of the DMA Buffer Descriptor; if DMAxADDR = 0xFFxx, the channel is switched off

10.3. Register: DMA0CONFIG, DMA1CONFIG

<table>
<tr><th>Name</th><th>Bits</th><th>R/W</th><th>Reset</th><th>Description</th></tr>
<tr><td>DxSOURCE</td><td>4:0</td><td>RW</td><td>00000</td><td>
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>00000</td><td>XRAM→Other DMA Channel (for memory to memory copies)</td></tr>
<tr><td>00001</td><td>SPI Transmit</td></tr>
<tr><td>00010</td><td>UART 0 Transmit</td></tr>
<tr><td>00011</td><td>UART 1 Transmit</td></tr>
<tr><td>00100</td><td>Timer 0 (ΣΔ)</td></tr>
<tr><td>00101</td><td>Timer 1 (ΣΔ)</td></tr>
</table>
</td></tr>
</table>

				00110 Timer 2 (ΣΔ)
				00111 Radio Transmit
				01000 Output Compare 0 (PWM)
				01001 Output Compare 1 (PWM)
				01XXX reserved
				10000 Other DMA Channel→XRAM (for memory to memory copies)
				10001 SPI Receive
				10010 UART 0 Receive
				10011 UART 1 Receive
				10100 ADC
				10101 reserved
				10110 reserved
				10111 Radio Receive
				11000 Input Capture 0
				11001 Input Capture 1
				11XXX reserved
DxIRQ	6	RW	0	1=interrupt has occurred; cleared by reading this register
DxRUN	7	RW	0	1=DMA channel running; automatically clears when reaching end of buffer chain

10.4. Buffer Descriptor Format

7	6	5	4	3	2	1	0
BUFADDR(7:0)							
BUFADDR(15:0)							
BUFLEN(7:0)							
IRQEN	ERREN	FMT		BUFLEN(11:8)			
ACTLEN(7:0)							
DONE	ERROR	0	0	ACTLEN(11:8)			
BDADDR(7:0)							
BDADDR(15:8)							

BUFADDR points to the actual buffer in XRAM. BUFLEN is the maximum (allocated) length of the buffer. IRQEN is a flag that enables an interrupt upon completion of this buffer. ACTLEN is the actual length of the buffer (may be less than BUFLEN). BDADDR is the pointer to the next Buffer Descriptor. If BDADDR is 0xFFxx, then this is the last buffer descriptor, the DMA engine will stop when finishing this buffer.

<table>
<tr><th>DxSOURCE</th><th>Source</th><th>FMT Meaning</th></tr>
<tr><td>00000
10000</td><td>XRAM→DMA
DMA→XRAM</td><td>FMT unused (set to 00)</td></tr>
<tr><td>00001</td><td>SPI Transmit</td><td>FMT unused (set to 00)</td></tr>
<tr><td>00010
00011</td><td>UART 0 Transmit
UART 1 Transmit</td><td><table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>All bytes sequentially written to UxSHREG</td></tr><tr><td>01</td><td>Bytes alternatively written to UxCTRL and UxSHREG</td></tr><tr><td>1X</td><td>Invalid</td></tr></table></td></tr>
<tr><td>10001</td><td>SPI Receive</td><td><table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>Bytes from UxSHREG / SPSHREG sequentially written to DMA Buffer</td></tr><tr><td>01</td><td>Bytes alternatively from UxSHREG / SPSHREG and UxSTATUS / SPSTATUS written to DMA Buffer</td></tr><tr><td>1X</td><td>Invalid</td></tr></table></td></tr>
<tr><td>10010</td><td>UART 0 Receive</td><td></td></tr>
<tr><td>10011</td><td>UART 1 Receive</td><td></td></tr>
<tr><td>00100
00101
00110</td><td>Timer 0 (ΣΔ)
Timer 1 (ΣΔ)
Timer 2 (ΣΔ)</td><td><table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>8 Bit data written to TxPERIOD1</td></tr><tr><td>01</td><td>16 Bit data (little endian) written to TxPERIOD0/TxPERIOD1</td></tr><tr><td>1X</td><td>Invalid</td></tr></table></td></tr>
</table>

DxSOURCE	Source	FMT Meaning
01000 01001	Output Compare 0 (PWM) Output Compare 1 (PWM)	**Bits** – **Meaning** 00 – 8 Bit data written to OCxCOMP1 01 – 16 Bit data (little endian) written to OCxCOMP0/OCxCOMP1 1X – Invalid
11000 11001	Input Capture 0 Input Capture 1	**Bits** – **Meaning** 00 – 8 Bit data (ICxCAPT1) written to DMA buffer 01 – 16 Bit data (little endian, ICxCAPT0/ICxCAPT1) written to DMA buffer 10 – 8 Bit data (ICxCAPT1) and status (ICxSTATUS) written to DMA buffer 11 – 16 Bit data (little endian, ICxCAPT0/ICxCAPT1) and status (ICxSTATUS) written to DMA buffer
10100	ADC	**Bits** – **Meaning** 00 – 8 Bit data (ADCCHxVAL1) written to DMA buffer 01 – 16 Bit data (little endian, ADCCHxVAL0/ADCCHxVAL1) written to DMA buffer 1X – Invalid
00111	Radio Transmit	**Bits** – **Meaning** 00 – All bytes sequentially written to FIFODATA 01 – Bytes alternatively written to FIFOCMD and FIFODATA 10 – All bytes sequentially written to FIFODATA, except the last, which is written to FIFOCMD 11 – Register Read/Write; buffer consists of 2 Bytes Address (topmost bit: 1=write, 0=read), followed by 1 byte data

DxSOURCE	Source	FMT Meaning	
10111	Radio Receive	Bits	Meaning
		00	Bytes from FIFODATA sequentially written to DMA Buffer
		01	Bytes alternatively from FIFODATA and “quickstatus” bytes written to DMA Buffer
		10	Invalid
		11	Register Read/Write; buffer consists of 2 Bytes Address (topmost bit: 1=write, 0=read), followed by 1 byte data

11. Radio Interface

The Radio Interface is a dedicated interface unit to Axsem Radio chips, as well as some other SPI slave devices. It features easy microcontroller access by mapping the Radio registers into X address space. Normally, 8 bit Register accesses are performed. The Radio Interface however may perform up to 32 bit Register accesses, needed with some Axsem Radio chips to atomically read some tracking registers; in this case, the top most byte is returned from the X address space access, the other data bytes may afterwards be read out from SFR registers.

11.1. Features

- Directly maps Radio Chip registers into the X Bus address space
- Non-blocking access mode may be used to hide the access latency

11.2. Direct Access via X-Space

When direct accesses are performed in the Radio Controller dedicated X address space, the controller performs the following actions:

RREG 4000–4FFF	R	1. Stall CPU core until the Radio Controller is not busy 2. Copy Address Bits 11:0 into RADIOADDR1,RADIOADDR0(11:0) 3. Perform Radio Register Read access 4. End CPU core access, return contents RADIODATA3
	W	1. Stall CPU core until the Radio Controller is not busy 2. Copy Address Bits 11:0 into RADIOADDR1,RADIOADDR0(11:0) 3. Write CPU data to RADIODATA3 4. Perform Radio Register Write access 5. End CPU core access
RREG (nb) 5000–5FFF	R	1. Stall CPU core until the Radio Controller is not busy 2. Copy Address Bits 11:0 into RADIOADDR1,RADIOADDR0(11:0) 3. Start Radio Register Read access 4. End CPU core access, return data undefined
	W	1. Stall CPU core until the Radio Controller is not busy 2. Copy Address Bits 11:0 into RADIOADDR1,RADIOADDR0(11:0)

		3. Write CPU data to RADIODATA3 4. Start Radio Register Write access 5. End CPU core access

The RREG area is blocking; the CPU is stalled until the register access is terminated. The RREG (nb) area is non-blocking. The register access is only initiated. The software is then responsible for polling the RCBUSY bit in RADIOACC and potentially retrieving the data from RADIODATA3.

11.3. Register: RADIODATA0, RADIODATA1, RADIODATA2, RADIODATA3

Name	Bits	R/W	Reset	Description
RADIODATA	31:0	RW	0x00000000	Radio Chip Register Read/Write Data

11.4. Register: RADIOADDR0, RADIOADDR1

Name	Bits	R/W	Reset	Description
RADIOADDR	14:0	RW	0x0000	Radio Chip Register Read/Write Address

11.5. Register: RADIOSTAT0, RADIOSTAT1

Name	Bits	R/W	Reset	Description
RADIOSTAT	15:0	R	–	Radio Chip "Quick Status" (shifted out during address phase)

11.6. Register: RADIOACC

Name	Bits	R/W	Reset	Description
RADIODATAFMT	1:0	RW	00	Radio Chip Data Access Width Bits: Meaning 00: 8 Bit 01: 16 Bit

				10 24 Bit 11 32 Bit
RADIOADDRFMT	3:2	RW	00	Radio Chip Address Access Width Bits Meaning 00 7 Bit 01 15 Bit 10 12 Bit 11 7/12 Bit automatic
RC WRITE	6	R	–	Current/Last Access: 0=Read, 1=Write
RC BUSY	7	R	–	Radio Controller is busy with a transfer if 1

11.7. Register: RADIOFDATAADDR0, RADIOFDATAADDR1

Name	Bits	R/W	Reset	Description
RADIOFDATAADDR	11:0	RW	0x000	Radio Chip FIFO Data Register Address

11.8. Register: RADIOFSTATADDR0, RADIOFSTATADDR1

Name	Bits	R/W	Reset	Description
RADIOFDATAADDR	11:0	RW	0x000	Radio Chip FIFO Status Register Address

11.9. Setup for Axsem Radio Chips

The following table lists how to set up Radio Interface and GPIO registers in order to successfully communicate with an Axsem Radio chip. Registers RADIOFDATAADDR and RADIOFSTATADDR are only required if DMA access to the Radio chip FIFO is desired.

It is recommended to use the appropriate routines from libmf, such as ax5031_reset(), ax5042_reset(), ax5043_reset(), or ax5051_reset(). These routines take care of setting up the AX8052 as well as the Radio chip registers correctly. Furthermore, they also work with the SOC AX8052F131, AX8052F142, AX8052F143, and AX8052F151.

Register	Value
PORTR	0xCB
DIRR	0x15

Chip	RADIOACC	RADIOMUX	RADIOFDATAADDR	RADIOFSTATADDR
AX5031	0x00	0x07	0x005	0x004
AX5042	0x00	0x07	0x005	0x004
AX5043	0x0C	0x07	0x029	0x028
AX5051	0x00	0x07	0x005	0x004

11.10. Access Waveforms

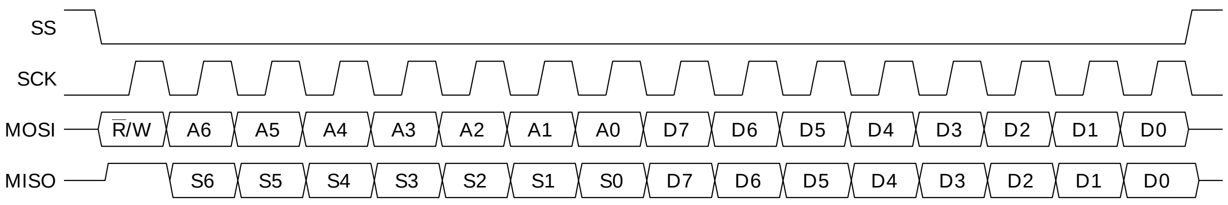


Figure 8: SPI 8bit Read/Write Access

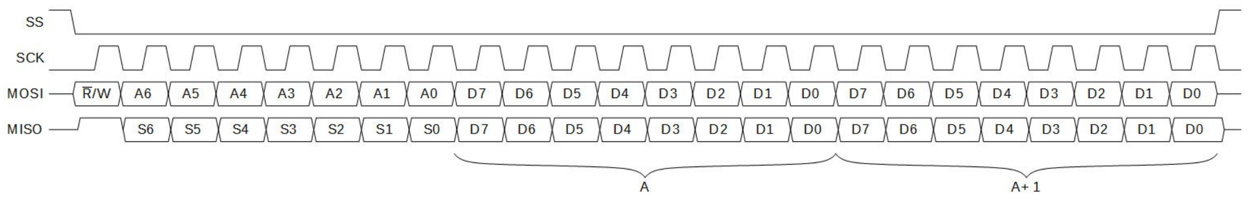


Figure 9: SPI 16bit Read/Write Access

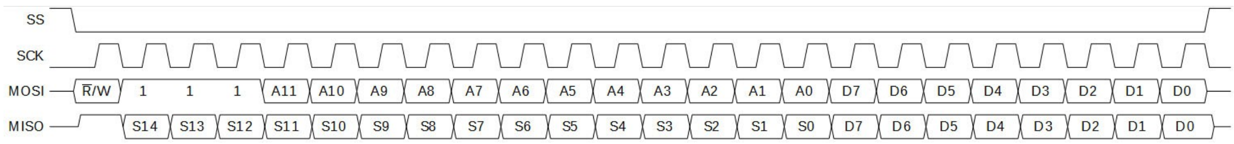


Figure 10: SPI 8bit Long Address Read/Write Access

Figures 8, 9 and 10 show typical access waveforms. The SPI access is divided into three portions: the read/write bit, the address, and the data. There are four different addressing modes: 7 bit, 15 bit, 12 bit, and 7/12 bit automatic. The data portion may be 8, 16, 24 or 32 bits wide. During the address phase, status bits are returned. Addressing modes are summarized in the table below:

Addressing Mode		**Condition**	**First Address Byte**	**Second Address Byte**	**Figure**
Code	**Description**				
00	7 bit	–	R/W A6:A0	omitted	8, 9
01	15 bit	–	R/W A14:A8	A7:A0	–
10	12 bit	–	R/W 111 A11:A8	A7:A0	10
11	7/12 bit automatic	ADDR<0x70	R/W A6:A0	omitted	8, 9
		ADDR≥0x70	R/W 111 A11:A8	A7:A0	10

12. GPIO

In order to maximize the number of user assignable pins, the AX8052 has only two dedicated pins (besides the supply pins), namely RESET_N and DBG_EN.

The AX8052 has three general purpose 8 bit GPIO Ports, Port A, B and C. An additional port, Port R, is dedicated to connecting a Radio chip. Most GPIO pins may also be connected to peripherals. The registers PALTA, PALTB and PALTC configure which GPIO pins serve as dedicated peripheral output. Peripheral inputs may be selected using the PINSEL register.

Each GPIO Pin features a programmable pull-up. Interrupt on pin change is provided for Ports A, B and C. Ports B, C and R are 5V tolerant. Port A pins should not exceed VDDIO. All digital inputs feature schmitt trigger buffers. Port A may be used as analog input. Digital input buffers should be switched off for Port A pins used as analog inputs by setting the appropriate bits of ANALOGA, to prevent additional current consumption.

In order to prevent GPIO pin glitches after wakeup from deep sleep, GPIO pin state is latched when entering deep sleep mode. After wakeup and setup of the GPIO registers, the latches must be switched to transparent mode by writing 1 to GPIOENABLE.

Depending on package, not all GPIO pins may be available, see the appropriate data sheet.

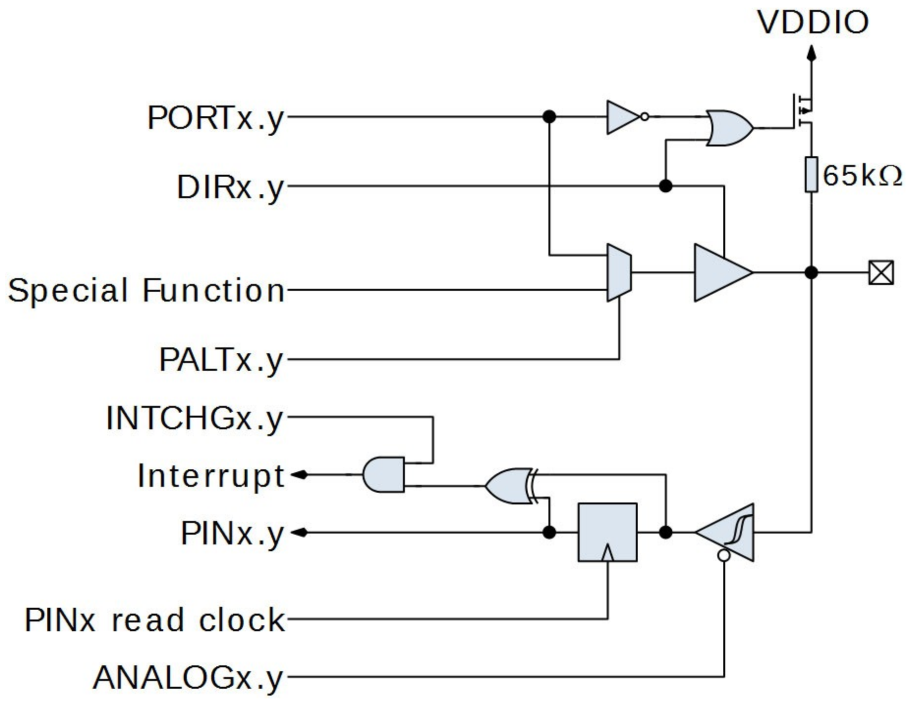


Figure 11: GPIO Pin Block Diagram

Figure 11 shows the block diagram of a GPIO pin.

12.1. Features

- Port, Pin and Direction registers
- Programmable Pull-up
- Port A: Pins programmable as Analog Pins (switches off the input schmitt trigger to save power)
- Interrupt on Port Change

12.2. Dedicated Pins

Name	Dir	Pull	Purpose
DBG_EN	I	PD	Debug Serial Interface Enable (active high)
RESET_N	I	PU	Reset Input (active low)

PD = Pull Down; PU = Pull Up

12.3. GPIO Pins

The following table lists alternate functions for all GPIO pins.

Name	Alternate Functions			
PA0	T0OUT	IC1	ANALOG0	XTALP
PA1	T0CLK	OC1	ANALOG1	XTALN
PA2	OC0	U1RX	ANALOG2	COMPI00
PA3	T1OUT		ANALOG3	LPXTALP
PA4	T1CLK	COMPO0	ANALOG4	LPXTALN
PA5	IC0	U1TX	ANALOG5	COMPI10
PA6	T2OUT	ADCTRIG	ANALOG6	COMPI01
PA7	T2CLK	COMPO1	ANALOG7	COMPI11
PB0	U1TX	IC1	EXTIRQ0	
PB1	U1RX	OC1		
PB2	IC0	T2OUT		
PB3	OC0	T2CLK	EXTIRQ1	
PB4	U0TX	T1CLK		
PB5	U0RX	T1OUT		
PB6	DBG_DATA			
PB7	DBG_CLK			
PC0	SSEL	T0OUT	EXTIRQ0	
PC1	SCK	T0CLK	COMPO1	

Name	Alternate Functions			
PC2	SMOSI	U0TX		
PC3	SMISO	U0RX	COMPO0	
PC4	COMPO1	ADCTRIG	EXTIRQ1	
PC5				
PC6				
PC7				
PR0	RSEL			
PR1	RSYSCLK			
PR2	RCLK			
PR3	RMISO			
PR4	RMOSI			
PR5	RIRQ			
PR6				
PR7				

12.4. Recommended Initialization Sequence

1. Set the registers DIRA, DIRB, DIRC, PORTA, PORTB, PORTC, ANALOGA according to the board circuitry. Unconnected pins should either be driven or should have their pull-up enabled; this prevents floating pins, which can cause elevated current consumption of their digital input buffer.

2. Call flash_apply_calibration(); This routine copies manufacturing calibration data from the FLASH into the corresponding chip registers.

3. If an Axsem Radio SoC is used or if an Axsem Radio Chip is connected to Port R, the following routines should be called:

   - On cold start (PCON bit 6 zero), call ax*_reset()

- On warm start (PCON bit 6 one), call ax*_comminit()

4. Set GPIOENABLE to one

5. Configure the digital pins of the Radio (such as SYSCLK)

The routines flash_apply_calibration(), ax*_reset() and ax*_comminit() can be found in libMF; please refer to the libMF documentation. Using these routines ensures that the interface between the microcontroller and the radio is set up correctly, and that the differences between the SoC and the corresponding discrete two-chip solution are handled transparently to the user software.

12.5. Register: PORTA, PORTB, PORTC, PORTR

Name	Bits	R/W	Reset	Description
PORTA PORTB PORTC PORTR	7:0	RW	0x00 0x00 0x00 0x00	Port Output Control

12.6. Register: PINA, PINB, PINC, PINR

Name	Bits	R/W	Reset	Description
PINA PINB PINC PINR	7:0	R	– – – –	Port Input

12.7. Register: DIRA, DIRB, DIRC, DIRR

Name	Bits	R/W	Reset	Description
DIRA DIRB DIRC DIRR	7:0	RW	0x00 0x00 0x00 0x00	Port Direction Control

DIRx.y	PORTx.y	Px.y Pin Drive
0	0	Z
0	1	Z, Pull-Up
1	0	0
1	1	1

12.8. Register: ANALOGA

Name	Bits	R/W	Reset	Description
ANALOGA	7:0	RW	0x00	Port A Analog Mode

Setting a bit in this register disables the digital input buffer of the corresponding port. This prevents additional current consumption when mid-scale analog voltages are applied to the corresponding port, which is used by an analog peripheral, such as the ADC, Comparators or Crystal Oscillators. Setting a bit causes the corresponding bit in the PINA and PINCHGA registers to become undefined.

12.9. Register: INTCHGA, INTCHGB, INTCHGC

Name	Bits	R/W	Reset	Description
INTCHGA INTCHGB INTCHGC	7:0	RW	0x00 0x00 0x00	Port Interrupt on Change

12.10. Register: EXTIRQ

Name	Bits	R/W	Reset	Description
EXTIRQ0MODE	1:0	RW	00	External Interrupt 0 Mode Bits – Meaning 00 – Level High 01 – Level Low 10 – Rising Edge 11 – Falling Edge
EXTIRQ0PIN	2	RW	0	0: EXTIRQ0 is Port B.0; 1: EXTIRQ0 is Port C.0
EXTIRQ1MODE	5:4	RW	00	External Interrupt 1 Mode Bits – Meaning 00 – Level High 01 – Level Low 10 – Rising Edge 11 – Falling Edge
EXTIRQ1PIN	6	RW	0	0: EXTIRQ1 is Port B.3; 1: EXTIRQ1 is Port C.4

12.11. Register: PINCHGA, PINCHGB, PINCHGC

Name	Bits	R/W	Reset	Description
PINCHGA PINCHGB PINCHGC	7:0	R	– – –	Port Change

12.12. Register: PALTA

Name	Bits	R/W	Reset	Description
PALTA0	0	RW	0	Port A.0 enable T0OUT
PALTA1	1	RW	0	Port A.1 enable OC1
PALTA2	2	RW	0	Port A.2 enable OC0

PALTA3	3	RW	0	Port A.3 enable T1OUT
PALTA4	4	RW	0	Port A.4 enable COMPO0
PALTA5	5	RW	0	Port A.5 enable U1TX
PALTA6	6	RW	0	Port A.6 enable T2OUT
PALTA7	7	RW	0	Port A.7 enable COMPO1

12.13. Register: PALTB

Name	Bits	R/W	Reset	Description
PALTB0	0	RW	0	Port B.0 enable U1TX
PALTB1	1	RW	0	Port B.1 enable OC1
PALTB2	2	RW	0	Port B.2 enable T2OUT
PALTB3	3	RW	0	Port B.3 enable OC0
PALTB4	4	RW	0	Port B.4 enable U0TX
PALTB5	5	RW	0	Port B.5 enable T1OUT

12.14. Register: PALTC

Name	Bits	R/W	Reset	Description
PALTC0	0	RW	0	Port C.0 enable T0OUT
PALTC1	1	RW	0	Port C.1 enable COMPO1
PALTC2	2	RW	0	Port C.2 enable U0TX
PALTC3	3	RW	0	Port C.3 enable COMPO0
PALTC4	4	RW	0	Port C.4 enable COMPO1

12.15. Register: PINSEL

Name	Bits	R/W	Reset	Description
PINSEL0	0	RW	0	0: U0RX is Port B.5; 1: U0RX is Port C.3
PINSEL1	1	RW	0	0: U1RX is Port B.1; 1: U1RX is Port A.2

PINSEL2	2	RW	0	0: IC0 is Port B.2; 1: IC0 is Port A.5
PINSEL3	3	RW	0	0: IC1 is Port B.0; 1: IC1 is Port A.0
PINSEL4	4	RW	0	0: T0CLK is Port C.1; 1: T0CLK is Port A.1
PINSEL5	5	RW	0	0: T1CLK is Port B.4; 1: T1CLK is Port A.4
PINSEL6	6	RW	0	0: T2CLK is Port B.3; 1: T2CLK is Port A.7
PINSEL7	7	RW	0	0: ADCTRIG is Port C.4; 1: ADCTRIG is Port A.6

12.16. Register: GPIOENABLE

Name	Bits	R/W	Reset	Description
GPIOENABLE	0	RW	1	0: All Port Pins keep their current state; 1: Port Pins normal operation

This bit is reset to 0 when waking up from deep sleep.

12.17. Register: RADIOMUX

<table>
<tr><th>Name</th><th>Bits</th><th>R/W</th><th>Reset</th><th>Description</th></tr>
<tr><td>RADIOSYSCLK</td><td>2:0</td><td>RW</td><td>111</td><td>Port R.1 Pin Function<table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>000</td><td>FRCOSC</td></tr><tr><td>001</td><td>LPOSC</td></tr><tr><td>010</td><td>XOSC</td></tr><tr><td>011</td><td>LPXOSC</td></tr><tr><td>100</td><td>invalid</td></tr><tr><td>101</td><td>invalid</td></tr><tr><td>110</td><td>System Clock</td></tr><tr><td>111</td><td>Off</td></tr></table></td></tr>
<tr><td>RADIOIRQ</td><td>3</td><td>RW</td><td>0</td><td>0: IRQ is Port R.5; 1: IRQ is Port R.6</td></tr>
</table>

RADIOCLK	5:4	RW	00	Radio Clock Source Bits / Meaning 00: Port R.1 01: Port B.0 10: Port B.7 11: Port C.4
RADIOSPI	6	RW	0	0: Port R.0, R.2, R.3 and R.4 is GPIO; 1: Port R.0, R.2, R.3 and R.4 is Radio SPI

13. ADC, Comparator, Temperature Sensor

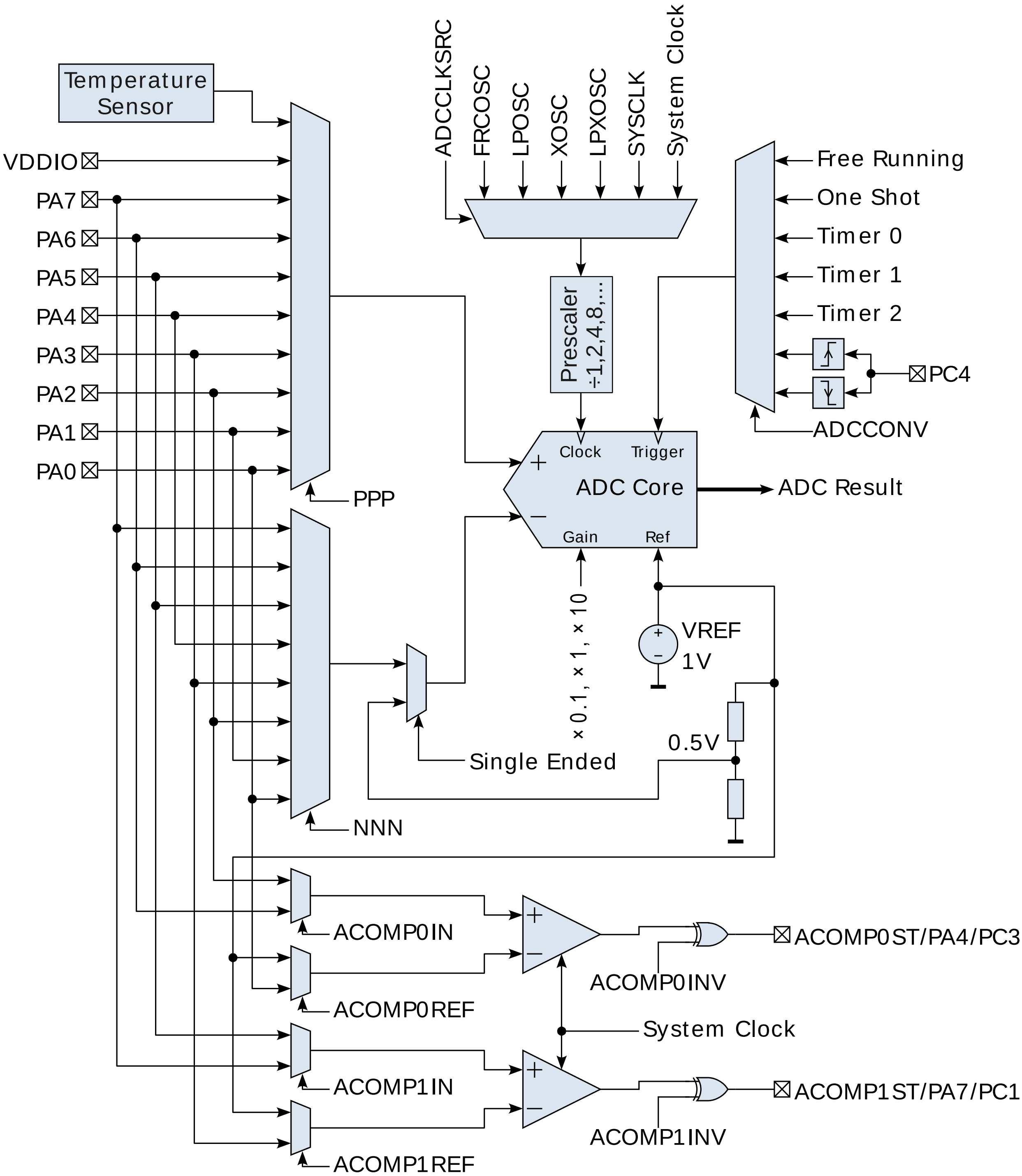


Figure 12: ADC Block Diagram

The ADC digitizes analog sensor signals. It also contains two general purpose comparators, as well as a temperature sensor.

The ADC controller supports up to 4 channels. A channel is a combination of multiplexer, single ended/differential, and gain settings. Each channel has its own result register. When triggered, the ADC converts all configured channels in sequence. If more than 4 channels are needed, software may first convert a group of 4 channels, read the result, reprogram the channel config registers, convert the next group of 4 signals, and so on.

The ADC needs a clock with a frequency of 16 times the sampling rate. This clock may be derived from any on-chip clock. A prescaler is also provided.

In order to achieve maximum precision, calibration data, stored during factory test in the uppermost 1kByte FLASH sector, needs to stored in the ADC core. Therefore, flash_apply_calibration() must be called and 0x06 must be written to ADCTUNE1 before the ADC is used.

13.1. Features

- 10 Bit ADC, 500kSamples/s
- Single Ended and Differential
- ×1, ×0.1 and ×10 gain settings
- 2 additional Comparators
- Temperature Sensor
- Built-in 1V Reference
- Conversion may be triggered by software, timer or external signal

13.2. Register: ADCCLKSRC

This register selects the clock source which provides the timing for the ADC circuitry. The sampling rate of the ADC is $\frac{1}{16}$ th of the clock frequency selected by this register.

Name	Bits	R/W	Reset	Description
ADCCLKSRC	2:0	RW	111	Clock Source Bits | Meaning 000 | FRCOSC 001 | LPOSC 010 | XOSC 011 | LPXOSC 100 | RSYSCLK 101 | invalid 110 | System Clock 111 | Off
ADCCLKDIV	6:3	RW	0000	Clock Prescaler Bits | Meaning 0000 | ÷1 0001 | ÷2 0010 | ÷4 0011 | ÷8 0100 | ÷16 0101 | ÷32 0110 | ÷64 0111 | ÷128 1000 | ÷256 1001 | ÷512 1010 | ÷1024 1011 | ÷2048

				1100 ÷4096 1101 ÷8192 1110 ÷16384 1111 ÷32768
ADCACT	7	R	–	ADC Active; the logical OR of ADCPEND and ADCBUSY in Register ADCCONV

Do not select XOSC or LPXOSC unless a crystal is connected – see OSCFORCERUN for details.

The System Clock may stop if the processor enters standby mode. Therefore, if System Clock is selected as ADC clock source, care has to be taken not to enter standby mode while a conversion is running.

13.3. Register: ADCCH0CONFIG, ADCCH1CONFIG, ADCCH2CONFIG, ADCCH3CONFIG

Name	Bits	R/W	Reset	Description
CH0CONFIG CH1CONFIG CH2CONFIG CH3CONFIG	7:0 7:0 7:0 7:0	RW RW RW RW	11111111 11111111 11111111 11111111	Bits / Meaning: 00NNNPPP: Differential Mode, Gain ×0.1, NNN=negative terminal, PPP=positive terminal 01NNNPPP: Differential Mode, Gain ×1, NNN=negative terminal, PPP=positive terminal 10NNNPPP: Differential Mode, Gain ×10, NNN=negative terminal, PPP=positive terminal 11001PPP: Single Ended Mode, Gain ×1, PPP=positive terminal 11011000: Temperature 11011001: VDDIO 11111111: Channel Off

PPP and NNN encode the input port to be used for the positive and negative input, respectively. For the single ended modes, the negative input is connected to the internal 0.5V reference voltage, and therefore implicit.

PPP, NNN	Port
000	A0
001	A1
010	A2
011	A3

PPP, NNN	Port
100	A4
101	A5
110	A6
111	A7

Ports used for analog voltages need their corresponding digital input buffer disabled in register ANALOGA. Failure to do so results in additional current consumption by the digital input buffer when a mid-scale voltage is applied to the port.

13.3.1. Measuring VDDIO

After conversion of the IO voltage using ADCCHxCONFIG=0xD9, the conversion result can be converted into a voltage using the following formula:

$$VDDIO = ADCCHxVAL \cdot \frac{10V}{2^{16}} - 4.5V$$

ADCCHxVAL must be treated as an unsigned 16 Bit value. In other words, VDDIO is sampled using a full scale range of 10V, with an offset voltage of 4.5V.

13.4. Register: ADCCH0VAL0, ADCCH0VAL1, ADCCH1VAL0, ADCCH1VAL1, ADCCH2VAL0, ADCCH2VAL1

Name	Bits	R/W	Reset	Description
CHxVAL	15:0	R	–	ADC Channel Conversion Result Registers

13.5. Register: ADCTUNE0

Name	Bits	R/W	Reset	Description
ADCTUNE0	7:0	RW	01	Must be set to 0x01

13.6. Register: ADCTUNE1

Name	Bits	R/W	Reset	Description
ADCTUNE1	7:0	RW	0	Must be set to 0x06

13.7. Register: ADCTUNE2

Name	Bits	R/W	Reset	Description
PMODE	1:0	RW	00	ADC Power Saving Mode Bits / Meaning: 00 No Powersave 01 Clock Stop between Bursts 10 Power Off between Bursts 11 Invalid
WAKEC	4:2	RW	010	Wakeup conversion cycles after Clock Stop Bits / Meaning: 000 0 001 1 010 2 011 3 100 4 101 6 110 8 111 12

WAKEP	7:5	RW	111	Wakeup conversion cycles after Power Up Bits / Meaning 000 0 001 1 010 2 011 3 100 4 101 6 110 8 111 12

13.8. Register: ADCCONV

Name	Bits	R/W	Reset	Description
CONVSRC	2:0	RW	000	Conversion Control Bits / Meaning 000 Idle 001 Start One-Shot Conversion (CONVSRC will read back as 0) 010 Pin Rising Edge 011 Pin Falling Edge 100 Timer 0 101 Timer 1 110 Timer 2 111 Continuous Free Running
ADCPEND	5	R	–	ADC Conversion is Pending
ADCBUSY	6	R	–	ADC is Busy
ADCIRQ	7	RC	–	ADC Conversion Done Interrupt active; this bit clears on read of this register

Whenever a conversion burst is triggered, for example by writing 001 to CONVSRC or if the selected event happens, the ADCPEND bit is set, and the ADC core starts to wake up from power down. When the ADC is ready to perform the conversion, ADCBUSY is set and

ADCPEND is cleared. After the conversion is done, ADCBUSY is cleared and the ADC core is powered down again.

In order to wait for results to be available, software should either use the ADC interrupt, poll until the ADCIRQ bit is set, or poll for both ADCPEND and ADCBUSY to go low. Alternatively, software can also poll until ADCACT in register ADCCLKSRC goes low, which preserves the state of ADCIRQ.

13.9. Register: ANALOGCOMP

Name	Bits	R/W	Reset	Description
ACOMP0IN	0	RW	0	Comparator 0 Input Select: 0=PA2; 1=PA6
ACOMP0REF	1	RW	0	Comparator 0 Reference Select: 0=PA0; 1=internal Reference
ACOMP1IN	2	RW	0	Comparator 1 Input Select: 0=PA5; 1=PA7
ACOMP1REF	3	RW	0	Comparator 1 Reference Select: 0=PA3; 1=internal Reference
ACOMP0INV	4	RW	0	Comparator 0 Inversion
ACOMP1INV	5	RW	0	Comparator 1 Inversion
ACOMP0ST	6	R	–	Comparator 0 Status
ACOMP1ST	7	R	–	Comparator 1 Status

14. SPI Master/Slave Interface

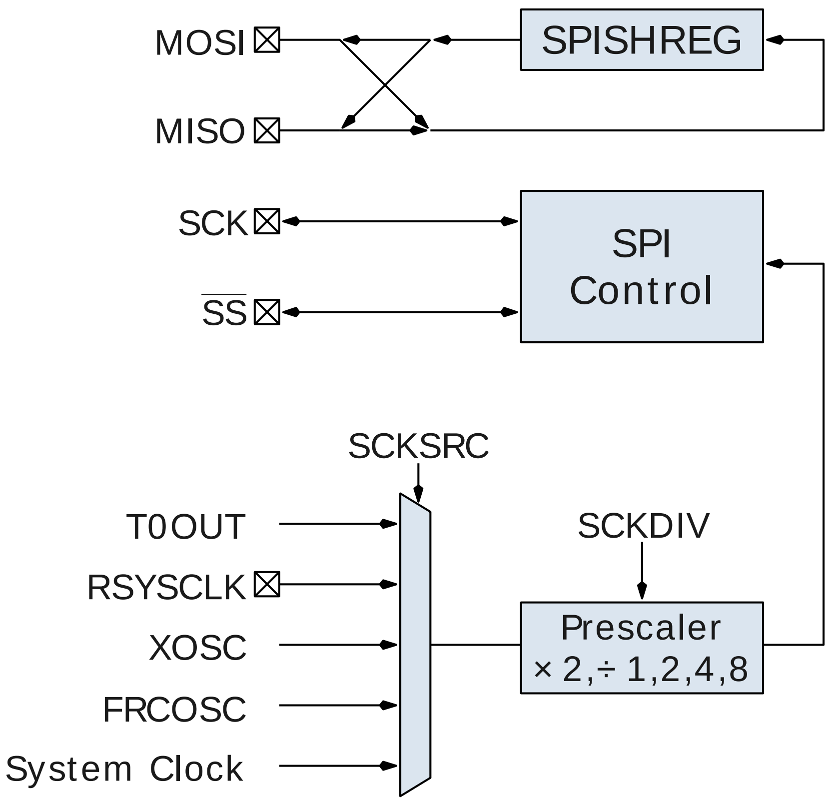


Figure 13: SPI Block Diagram

The SPI interface provides a general purpose SPI master or slave. Figure 13 shows the block diagram of the SPI interface. Its fully programmable, flexible clocking scheme allows it to connect to any SPI compatible peripheral or master.

14.1. Features

- 8-Bit
- Master
- 3 and 4 wire Slave

- in Master mode, SCLK can be the system clock, any of the oscillators or divided versions of it
- Programmable Clock Phase and Inversion

14.2. Register: SPSHREG

Name	Bits	R/W	Reset	Description
SPSHREG	7:0	RW	–	SPI Shift Register

14.3. Register: SPSCKSRC

Name	Bits	R/W	Reset	Description
SPSCKSRC	2:0	RW	111	Clock Source Bits – Meaning 000 – FRCOSC 001 – LPOSC 010 – XOSC 011 – LPXOSC 100 – RSYSCLK 101 – T0OUT 110 – System Clock 111 – Off

SPSCKDIV	5:3	RW	000	Clock Prescaler Bits / Meaning 000 ÷1 001 ÷2 010 ÷4 011 ÷8 100 ÷16 101 ÷32 110 ÷64 111 ÷128
SPSCKINV	6	RW	0	SPI Clock Invert
SPSCKPH	7	RW	0	SPI Clock Phase

Do not select XOSC or LPXOSC unless a crystal is connected – see OSCFORCERUN for details.

The System Clock may stop if the processor enters standby mode. Therefore, if System Clock is selected as SPI clock source, an ongoing SPI transaction may be halted while the processor is in standby mode.

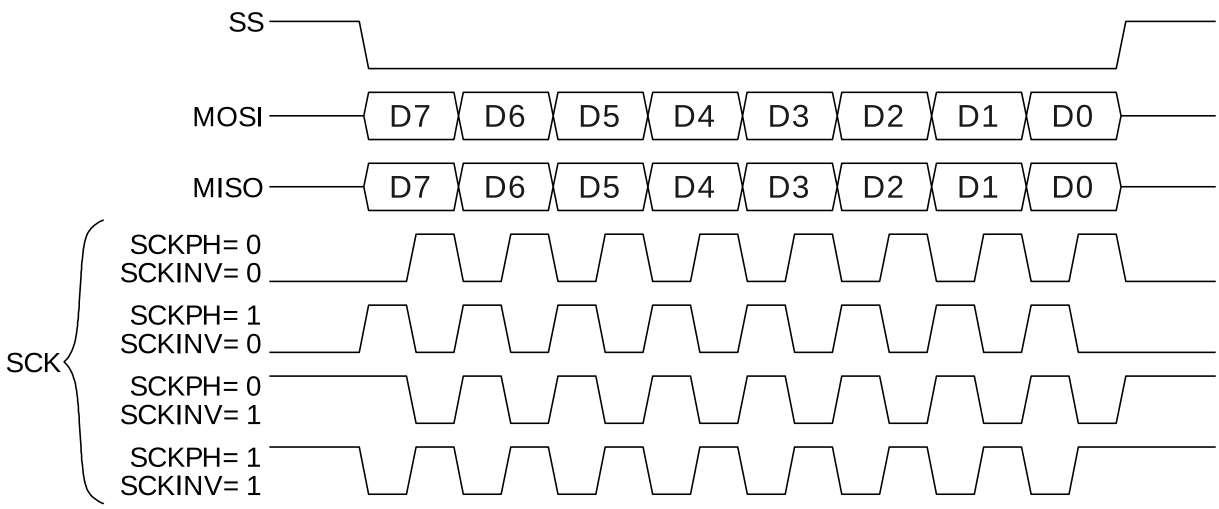


Figure 14: SPI Waveforms

Figure 14 shows the effect of the clock phase and inversion settings.

14.4. Register: SPMODE

Name	Bits	R/W	Reset	Description
SPMODE	1:0	RW	00	Operating Mode Bits – Meaning 00 – Off 01 – Master 10 – Slave, always selected 11 – Slave, selected when SS=0
SPDIR	2	RW	0	Shift Direction, 0 = MSB first, 1 = LSB first
SPRXIE	3	RW	0	SPI Receive interrupt enable
SPTXIE	4	RW	0	SPI Transmit interrupt enable
SPSSIE	5	RW	0	SPI SS change interrupt enable

When switching to slave mode (SPMODE=10 or SPMODE=11), the SPI bus must be idle and SCK must be at its idle state (SCK=0 when SCKINV=0, or SCK=1 when SCKINV=1). Otherwise, bit synchronization between master and slave may not be achieved. Alternatively, when switching to SPMODE 11, SS=1 will also ensure bit synchronization.

14.5. Register: SPSTATUS

Name	Bits	R/W	Reset	Description
SPRXFULL	0	R	–	Rx Full interrupt request
SPRXOVER	1	R	–	Rx Overrun interrupt request (clears on read)
SPTXEMPTY	2	R	–	Tx Empty interrupt request
SPTXUNDER	3	R	–	Tx Underrun interrupt request (clears on read)
SPSSCHG	4	R	–	SS change interrupt request (clears on read)
SPSSSTAT	5	R	–	Current SS status
SPFIRST	6	R	–	First Word

15. Timer Counter 0/1/2

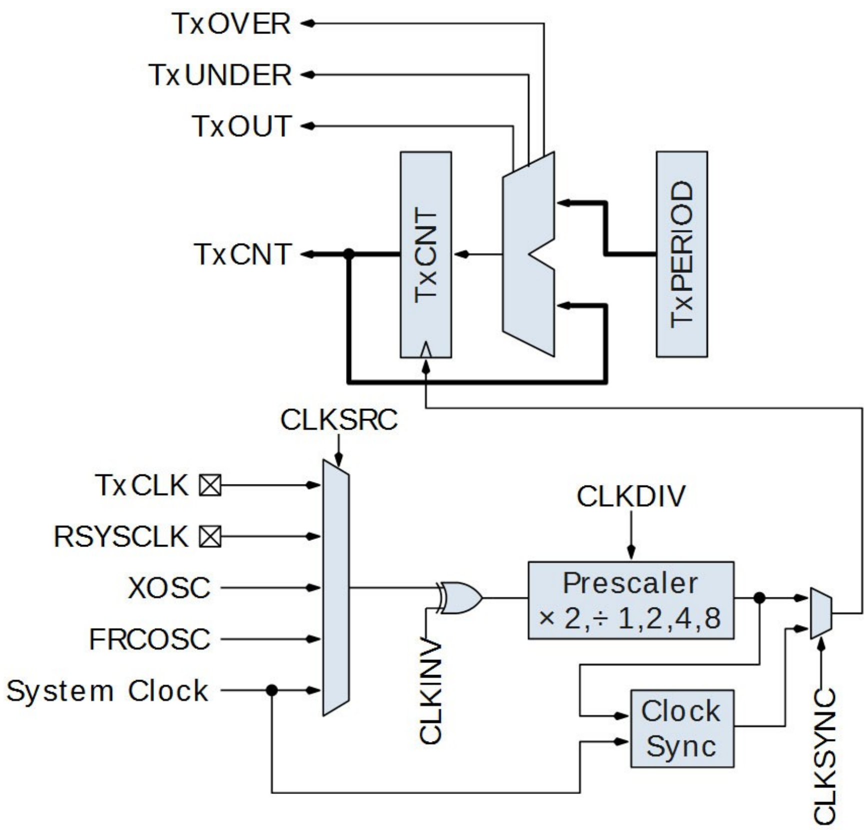


Figure 15: Timer Block Diagram

The 16bit Timer Counter peripheral provides the microcontroller with timing. Figure 15 shows the block diagram of the Timers. It can also interface to the analog world with its ΣΔ feature. PWM may be realized together with an output compare block. Input signal Timing may be captured together with an input capture block.

15.1. Features

- Supports the following modes
    - Timer
    - Prescaler

    - PWM
    - ΣΔ
    - Input Capture
- Can operate from the system clock or another asynchronous clock (note that some modes are not supported with asynchronous clock)
- Provides UART Baud Rate and ADC Sampling Rate clocks

15.2. Register: T0CNT0, T0CNT1

Name	Bits	R/W	Reset	Description
T0CNT	15:0	RW	0x0000	Timer 0 Counter

Since the timer may run at a different clock rate than the CPU, reading the timer register may not always be consistent. While every effort is made to make reads consistent, it is not always possible. Reads are consistent under the following conditions:

- All modes that change the counter register only by one at a time, regardless of clock frequencies
    - Divide Triangle
    - Divide Sawtooth with T0PERIOD=0xFFFF
    - Multiply Sawtooth/Triangle with T0PERIOD=1
- All modes if $PERIOD_{TIMERCLK} \geq 2 \cdot PERIOD_{SYSCLK}$
- All modes if T0CLKSYNC is enabled

What does consistency mean? A read is consistent if it returns either the current counter value, or any counter value up to $2 \cdot \text{PERIOD}_{SYSCLK}$ in the past.

When writing to this register, it takes up to two timer clock cycles until the value appears in the internal counter register.

15.3. Register: T0PERIOD0, T0PERIOD1

Name	Bits	R/W	Reset	Description
T0PERIOD	15:0	RW	0x0000	Timer 0 Period

15.4. Register: T0CLKSRC

Name	Bits	R/W	Reset	Description
T0CLKSRC	2:0	RW	111	Clock Source Bits / Meaning: 000 FRCOSC 001 LPOSC 010 XOSC 011 LPXOSC 100 RSYSCLK 101 T0CLK 110 System Clock 111 Off

T0CLKDIV	5:3	RW	001	Clock Prescaler Bits / Meaning 000: ×2 001: ×1 010: ÷2 011: ÷4 100: ÷8 101: ÷16 110: ÷32 111: ÷64
T0CLKINV	6	RW	0	Timer 0 Clock Invert
T0CLKSYNC	7	RW	0	Timer 0 Clock Synchronisation to System Clock

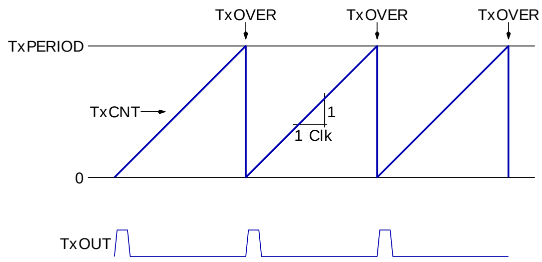
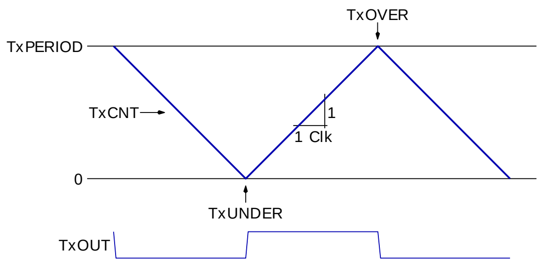
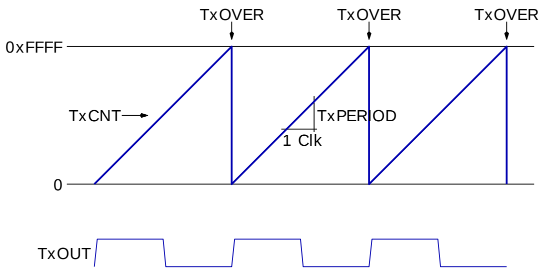
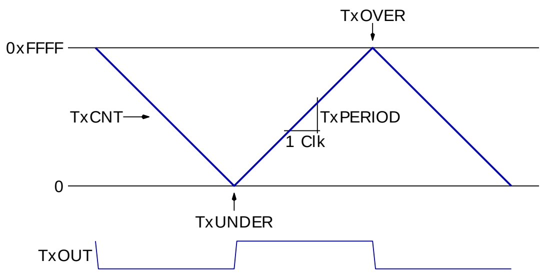
Do not select XOSC or LPXOSC unless a crystal is connected – see OSCFORCERUN for details.

The System Clock may stop if the processor enters standby mode. Therefore, if System Clock is selected as Timer clock source, the timer may pause counting if the processor enters standby mode.

15.5. Register: T0MODE

<table>
<tr><th>Name</th><th>Bits</th><th>R/W</th><th>Reset</th><th>Description</th></tr>
<tr><td>T0MODE</td><td>2:0</td><td>RW</td><td>000</td><td>Operating Mode (see table below)
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>000</td><td>Off</td></tr>
<tr><td>001</td><td>ΣΔ</td></tr>
<tr><td>010</td><td>Divide Sawtooth</td></tr>
<tr><td>011</td><td>Divide Triangle</td></tr>
<tr><td>100</td><td>Multiply Sawtooth</td></tr>
<tr><td>101</td><td>Multiply Triangle</td></tr>
<tr><td>110</td><td>invalid</td></tr>
<tr><td>111</td><td>invalid</td></tr>
</table></td></tr>
<tr><td>T0LBUF</td><td>3</td><td>RW</td><td>0</td><td>Counter Buffering; 0 = No Buffering, 1 = TxCNT0, TxPERIOD0 updated/written on access of TxCNT1, TxPERIOD1; reading TxCNT0 returns value latched when reading TxCNT1; TxPERIOD0 is never buffered on reads as TxPERIOD can change only under program control</td></tr>
<tr><td>T0PRBUF</td><td>5:4</td><td>RW</td><td>00</td><td>Period Buffering in Divide and Multiply Modes:
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>00</td><td>No Double Buffering; TxPERIOD0 buffering according to TxLBUF</td></tr>
<tr><td>01</td><td>Internal Buffer updated on TxOVER</td></tr>
<tr><td>10</td><td>Internal Buffer updated on TxUNDER</td></tr>
<tr><td>11</td><td>Internal Buffer updated on TxOVER and TxUNDER</td></tr>
</table>
Period Buffering in ΣΔ Mode:
<table>
<tr><th>Bits</th><th>Meaning</th></tr>
<tr><td>00</td><td>No Double Buffering; TxPERIOD0 buffering according to TxLBUF</td></tr>
<tr><td>01</td><td>Internal Buffer updated on T0OUT</td></tr>
<tr><td>10</td><td>Internal Buffer updated on T1OUT</td></tr>
<tr><td>11</td><td>Internal Buffer updated on T2OUT</td></tr>
</table></td></tr>
<tr><td>T0IRQMO</td><td>6</td><td>RW</td><td>0</td><td>Interrupt on TxOVER enable</td></tr>
</table>

T0IRQMU	7	RW	0	Interrupt on TxUNDER enable

	Sawtooth	Triangle
Div		
Multiply		

15.6. Register: T0STATUS

Name	Bits	R/W	Reset	Description
T0IRQRO	0	R	–	TxOVER interrupt request (clears on read)
T0IRQRU	1	R	–	TxUNDER interrupt request (clears on read)
T0IRQEO	2	R	–	TxOVER interrupt missed (clears on read)
T0IRQEU	3	R	–	TxUNDER interrupt missed (clears on read)
T0IRQPE	4	R	–	TxPERIOD empty interrupt request
T0IRQPU	5	R	–	TxPERIOD underrun interrupt request (clears on read)
T0IRQMPE	6	RW	0	TxPERIOD empty interrupt enable
T0IRQMPU	7	RW	0	TxPERIOD underrun interrupt enable

16. Output Compare 0/1

The output compare unit allows, together with a Timer, PWM waveforms to be generated.

16.1. Register: OC0COMP0, OC0COMP1

Name	Bits	R/W	Reset	Description
OC0COMP	15:0	RW	0x0000	Output Compare 0 Compare Value

16.2. Register: OC0MODE

<table>
<tr><th>Name</th><th>Bits</th><th>R/W</th><th>Reset</th><th>Description</th></tr>
<tr><td>OC0SRC</td><td>1:0</td><td>RW</td><td>00</td><td>Source Timer<table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>Off</td></tr><tr><td>01</td><td>Timer 0</td></tr><tr><td>10</td><td>Timer 1</td></tr><tr><td>11</td><td>Timer 2</td></tr></table></td></tr>
<tr><td>OC0LBUF</td><td>3</td><td>RW</td><td>0</td><td>Compare Buffering; 0 = No Buffering, 1 = OCxCOMP0 written on access of OCxCOMP1</td></tr>
<tr><td>OC0CMPBUF</td><td>5:4</td><td>RW</td><td>00</td><td>Period Buffering<table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>No Double Buffering; OCxCOMP0 buffering according to OC0LBUF</td></tr><tr><td>01</td><td>Internal Buffer updated on TxOVER</td></tr><tr><td>10</td><td>Internal Buffer updated on TxUNDER</td></tr><tr><td>11</td><td>Internal Buffer updated on TxOVER and TxUNDER</td></tr></table></td></tr>
<tr><td>OC0IRQMR</td><td>6</td><td>RW</td><td>0</td><td>Interrupt on compare rising edge</td></tr>
<tr><td>OC0IRQMF</td><td>7</td><td>RW</td><td>0</td><td>Interrupt on compare falling edge</td></tr>
</table>

16.3. Register: OC0PIN

<table>
<tr><th>Name</th><th>Bits</th><th>R/W</th><th>Reset</th><th>Description</th></tr>
<tr><td>OC0PCR</td><td>1:0</td><td>RW</td><td>00</td><td>Pin Change on Compare Rising Edge
<table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>No Change</td></tr><tr><td>01</td><td>Invalid</td></tr><tr><td>10</td><td>Clear</td></tr><tr><td>11</td><td>Set</td></tr></table></td></tr>
<tr><td>OC0PCF</td><td>3:2</td><td>RW</td><td>00</td><td>Pin Change on Compare Falling Edge
<table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>No Change</td></tr><tr><td>01</td><td>Invalid</td></tr><tr><td>10</td><td>Clear</td></tr><tr><td>11</td><td>Set</td></tr></table></td></tr>
<tr><td>OC0PCO</td><td>5:4</td><td>RW</td><td>00</td><td>Pin Change on TxOVER
<table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>No Change</td></tr><tr><td>01</td><td>Invalid</td></tr><tr><td>10</td><td>Clear</td></tr><tr><td>11</td><td>Set</td></tr></table></td></tr>
<tr><td>OC0PCU</td><td>7:6</td><td>RW</td><td>00</td><td>Pin Change on TxUNDER
<table><tr><th>Bits</th><th>Meaning</th></tr><tr><td>00</td><td>No Change</td></tr><tr><td>01</td><td>Invalid</td></tr><tr><td>10</td><td>Clear</td></tr><tr><td>11</td><td>Set</td></tr></table></td></tr>
</table>

16.4. Register: OC0STATUS

Name	Bits	R/W	Reset	Description
OC0IRQRR	0	R	–	Compare Rising interrupt request (clears on read)
OC0IRQRF	1	R	–	Compare Falling interrupt request (clears on read)
OC0IRQER	2	R	–	Compare Rising interrupt missed (clears on read)
OC0IRQEF	3	R	–	Compare Falling interrupt missed (clears on read)
OC0CEMPTY	4	R	–	Compare Register Empty (clears on writing OC0COMP)
OC0CUNDER	5	R	–	Compare Register Underrun (clears on read)
OC0IRQMCE	6	RW	0	Compare Register Empty Interrupt Enable
OC0IRQMCU	7	RW	0	Compare Register Underrun Interrupt Enable

17. Input Capture 0/1

The input capture units allow the timing of digital signals to be measured, together with a timer. On a programmable edge of the input signal (either rising or falling), the value of the selected timer is latched. Not only external signals may be measured; a wide variety of capture sources are programmable. Two timers plus an input capture unit may be used, for example, to measure the frequency of an unknown signal with respect to a known clock source.

17.1. Register: IC0CAPT0, IC0CAPT1

Name	Bits	R/W	Reset	Description
IC0CAPT	15:0	R	–	Input Capture 0 Capture Value

17.2. Register: IC0MODE

Name	Bits	R/W	Reset	Description
IC0SRC	1:0	RW	00	Source Timer Bits | Meaning 00 | Off 01 | Timer 0 10 | Timer 1 11 | Timer 2
IC0EDGE	3:2	RW	00	Trigger Signal Edge Bits | Meaning 00 | Invalid 01 | Rising Edge 10 | Falling Edge 11 | Invalid

IC0LBUF	4	RW	0	Capture Buffering; 0 = No Buffering, 1 = ICxCAPT0 updated on access of ICxCAPT1
IC0IRQMCF	5	RW	0	Capture Register Full Interrupt Enable
IC0IRQMCO	6	RW	0	Capture Register Overrun Interrupt Enable

17.3. Register: IC0STATUS

Name	Bits	R/W	Reset	Description
IC0IRQR	0	R	–	Capture interrupt request
IC0IRQE	1	R	–	Capture interrupt missed (clears on read)
IC0TRIG	7:4	RW	0000	Capture Trigger Signal Bits: Meaning 0000: IC0 Capture Pin (IC1: IC1 Capture Pin) 0001: IC1 Capture Pin (IC1: IC0 Capture Pin) 0010: OC0 Compare Pin (IC1: OC1 Compare Pin) 0011: OC1 Compare Pin (IC1: OC0 Compare Pin) 0100: Timer 0 Out 0101: Timer 0 Clock 0110: Timer 1 Out 0111: Timer 1 Clock 1000: Timer 2 Out 1001: Timer 2 Clock 1010: Analog Comparator 0 (IC1: Analog Comparator 1) 1011: Analog Comparator 1 (IC1: Analog Comparator 0) 1100: IRQ 1101: IC1 Capture Source (IC1: IC0 Capture Source) 1110: Capture Zero 1111: Capture One

18. UART 0/1

The UARTs provide two Universal Asynchronous Receiver Transmitters.

18.1. Features

- Standard Universal Asynchronous Receiver Transmitter
- Word Lengths: 5, 6, 7, 8, 9 bits
- Stop Bits: 1, 2 (note that this only affects the transmitter, the receiver will always accept 1 stop bit)
- Arbitrary baud rates can be generated using a timer as baud rate generator
- Parity generation: none, even, odd (Software)
- Break detection
- Optional Receiver Deglitching for improved noise immunity

18.2. Register: U0SHREG

Name	Bits	R/W	Reset	Description
U0SHREG	7:0	RW	–	UART 0 Shift Register

18.3. Register: U0MODE

Name	Bits	R/W	Reset	Description
U0BRG	1:0	RW	00	Baud Rate Generator (TxOUT = 16× Baudrate) Bits | Meaning 00 | Off 01 | Timer 0 10 | Timer 1 11 | Timer 2
U0WL	4:2	RW	000	Word Size Bits | Meaning 000 | 8 Bits 001 | 9 Bits 010 | invalid 011 | invalid 100 | invalid 101 | 5 Bits 110 | 6 Bits 111 | 7 Bits
U0STOP	5	RW	0	Stop Bits; 0 = 1 Stop Bit, 1 = 2 Stop Bits
U0RXDGL	6	RW	0	Receiver Deglitch (Figure 6)
U0TXBRK	7	RW	0	Transmit Break enable

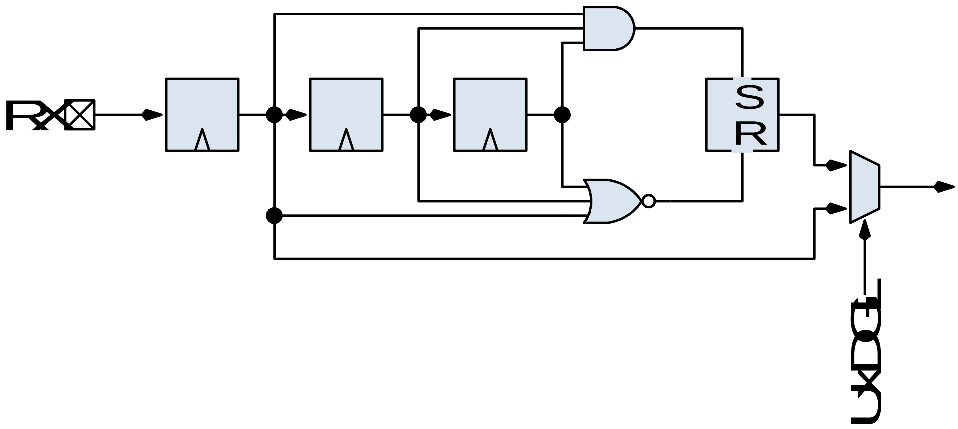


Figure 16: UART Deglitch

18.4. Register: U0CTRL

Name	Bits	R/W	Reset	Description
U0RXEN	0	RW	0	Receiver enable
U0TXEN	1	RW	0	Transmitter enable
U0RXIE	2	RW	0	Receiver interrupt enable
U0TXIE	3	RW	0	Transmitter interrupt enable
U0FEIE	4	RW	0	Receiver Framing error interrupt enable
U0BRKIE	5	RW	0	Receiver Break detect change interrupt enable (interrupt is activated if U0BRKDET changes, and cleared when U0BRKDET is read)
U0MCE	6	RW	0	Multiprocessor Communication Enable; If set, the received byte is only stored in the receive register if the topmost bit is set
U0TX8	7	RW	0	8th Tx Bit

18.5. Register: U0STATUS

Name	Bits	R/W	Reset	Description
U0RXFULL	0	R	–	Rx Full interrupt request
U0RXOVER	1	R	–	Rx Overrun interrupt request (clears on read)

U0TXEMPTY	2	R	–	Tx Empty interrupt request
U0TXUNDER	3	R	–	Tx Underrun interrupt request (clears on read) (does not make any sense)
U0FERR	4	R	–	Rx Framing Error interrupt request (clears on read)
U0BRKDET	5	R	–	Rx Break Detected (interrupt clears on read)
U0TXIDLE	6	R	–	Tx Idle
U0RX8	7	R	–	8th Rx Bit

19. Random Number Generator / AES

The Random Number Generator and Advanced Encryption Standard (AES) engine are documented in the AX8052 Cryptographic Functions Programming Manual.

PUBLICATION ORDERING INFORMATION
LITERATURE FULFILLMENT:
Literature Distribution Center for ON Semiconductor
19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA
Phone: 303-675-2175 or 800-344-3860 Toll Free USA/Canada
Fax: 303-675-2176 or 800-344-3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support: 800-282-9855 Toll Free USA/Canada.
Europe, Middle East and Africa Technical Support: Phone: 421 33 790 2910
Japan Customer Focus Center Phone: 81-3-5817-1050

Order Literature: http://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative

ON Semiconductor Website: **www.onsemi.com**