ON Semiconductor

Is Now

To learn more about onsemi™, please visit our website at
www.onsemi.com

AND9748/D

NCS36510
Programming Manual

ON Semiconductor®

www.onsemi.com

APPLICATION NOTE

DEVICE OVERVIEW

NCS36510 is an ultra−low power RF microcontroller System on Chip (SoC) that integrates the powerful and energy efficient ARM® Cortex®−M3 microprocessor, 640 kB of FLASH memory, 48 kB of RAM, 2.4 GHz 802.15.4 RF transceiver with hardware accelerated MAC, a DMA controller, and 18 GPIO. Peripherals include: UART (2), SPI (2), I^2C (2), timers (3), PWM, RTC, 10−bit SAR ADC, and internal voltage and temperature sensors. Security features include 128/256 hardware accelerated AES encryption engine and a true random number generator. NCS36510 implements advanced low−power modes for ultra−low power consumption.

Features

ARM® Cortex® −M3

- ARMv7−M Architecture
- Thumb® / Thumb−2 Subset Instruction Set
- Nested Vectored Interrupt Controller (NVIC) with 15 Built−in Exceptions and 20 External Interrupts with 4−bit Programmable Priority
- Non−Maskable Interrupt (NMI)
- Sleep, Deep Sleep, and Coma Mode Support
- Wake−up Interrupt Controller (WIC)
- SysTick Timer for Scheduler
- Bit Banding
- Little Endian

Debugger

- Serial−Wire Debug Access Port (SW−DAP)
- Breakpoint and Single Stepping Support
- Micro Trace Buffer (MTB)
- Standard Trace with ITM and DWT Triggers and Counters
- Full Debug with DWT Matching
- Debug Part Lockout

Memory

- 640 kB FLASH
  - Single Cycle 32−bit Fetch @ 32 MHz
  - Two Banks of 320 kB with Independent Power Controls
  - Both Banks Have Their Own 8 kB Onformation Block Section
  - 10,000 Program/Erase Cycles
  - 10 Year Data Retention at 85C
- 48 kB RAM
  - 16 kB or 32 kB Can be Retained in Coma Mode

Flexible Clocking

- On−Chip High Speed (32 MHz) RC Oscillator
- On−Chip Low Power (32.768 kHz) RC Oscillator
- High Speed Crystal Oscillator (32 MHz)
- Low Power Crystal Oscillator (32.768 kHz)
- Automatic Calibration of On−Chip RC Oscillators to External High Speed Crystal Oscillator

Timers

- General Purpose 16−bit Timers (3)
  - Pre−Load
  - Down Count
  - Interrupt on 0
  - Pre−scale Clock Divider
  - Free Running or Periodic Mode
- Integrated SysTick Timer
- Real Time Clock
  - On 32.768 kHz Clock Domain
  - Pre−Load
  - 15 bit Sub−Seconds Counter
  - 32 bit Seconds Counter
  - Both Counters Can Be Combined for ~ 136 Years and Can Implement a UNIX (POSIX or Epoch) Time Counter
- Watchdog Timer
  - On 32.768 kHz Clock Domain
    - Pre−Load
    - Max Timeout 30.5 us
    - Lockout Control

January, 2020 − Rev. 0

Publication Order Number:
AND9748/D

18 GPIO

- Programmable pull−up/pull−down
- Programmable drive strengths
- Open drain capable
- Programmable interrupts (edge/level, polarity)
- Allocation of GPIOn pins to peripherals through programmable crossbar

UART (2x)

- One full featured 16550 UART
- One reduced feature UART with transmit, receive, clear to send, and ready to send
- Programmable baud rate

Master/Slave SPI

- Programmable data width and direction
- Programmable phase and polarity

10−bit SAR ADC

- 10−bit 200 k−samples/second @ 4 Mhz
- Up to 4 external sample channels
- Single ended sampling
- Pseudo−differential sampling mode
- Ratio sampling mode
- Programmable input resistor divider
- Can sample internal power supply voltage and built−in temperature sensor

Security Features

- Hardware AES Acceleration
  - Supports 128−bit and 256−bit encryption/decryption
  - CCM, CTR, and CBC modes
- True random number generator
- Debug port lock

Table of Contents

BLOCK DIAGRAM

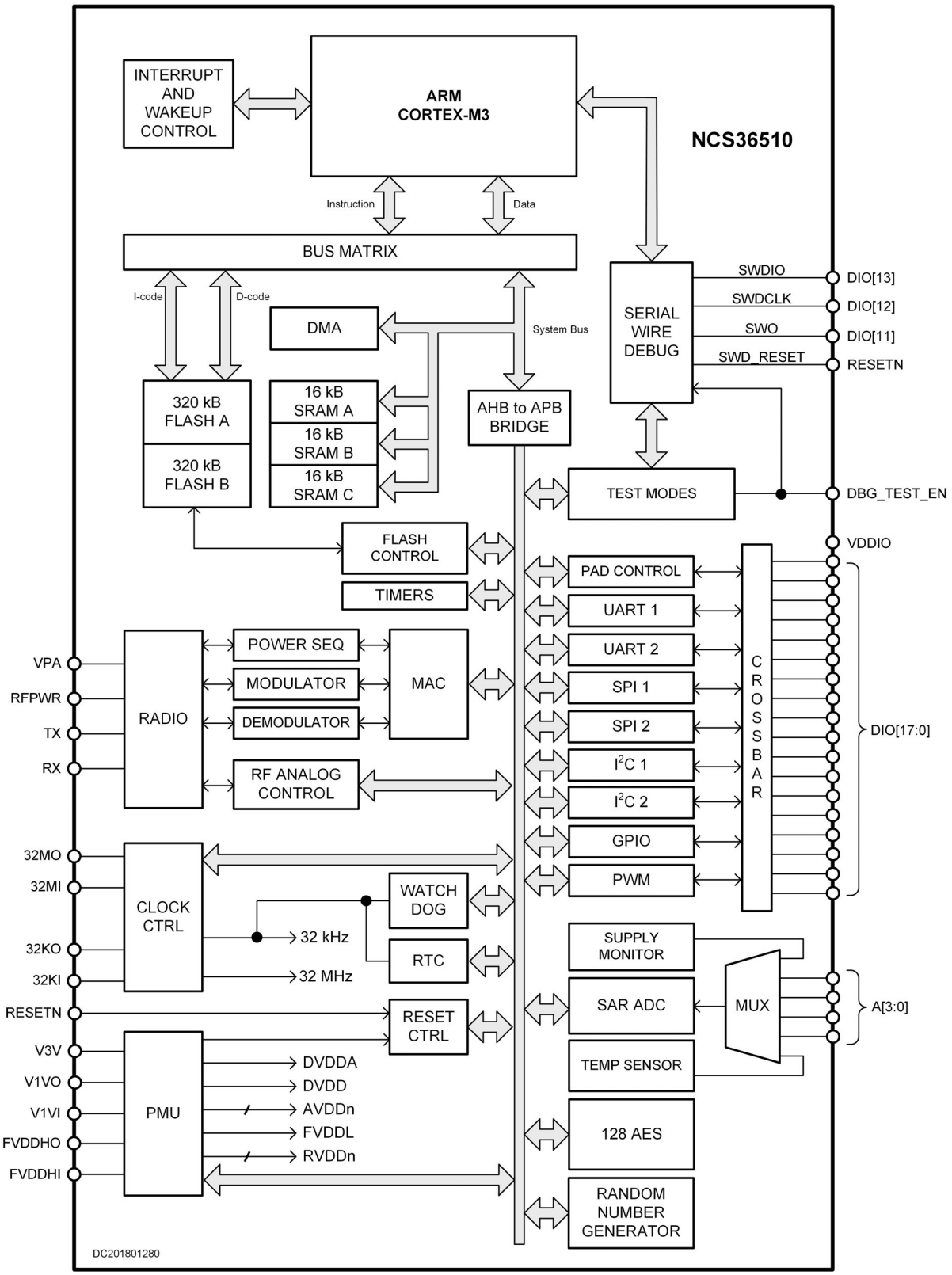


Figure 1. Block Diagram

CORTEX−M3 INSTRUCTION SET SUMMARY

The processor implements the ARMv7−M Thumb instruction set, including a number of 32−bit instructions that use Thumb−2 technology. The Cortex−M3 instruction set comprises:

- All of the 16−bit Thumb instructions from ARMv6 excluding SETEND and BLX.
- The 32−bit Thumb instructions excluding instructions related to Co−processor support (not supported on Cortex−M3) and the following HINT instructions (which behave as NOP if used): DBG, PLD, PLI, and YIELD.

The following instruction set summary is provided by ARM online at: https://developer.arm.com/docs/ddi0337/latest/programmers−model/instruction−set−summary/cortex−m3−instructions and is provided here for convenience.

The following table shows the Cortex−M3 instructions and their cycle counts. The cycle counts are based on a system with zero wait states.

Within the assembler syntax, depending on the operation, the `<op2>` field can be replaced with one of the following options:

- a simple register specifier, for example `Rm`
- an immediate shifted register, for example `Rm, LSL #4`
- a register shifted register, for example `Rm, LSL` R_S
- an immediate value, for example `#0xE000E000`.

For brevity, not all load and store addressing modes are shown.

The table below uses the following abbreviations in the Cycles column:

– P: The number of cycles required for a pipeline refill. This ranges from 1 to 3 depending on the alignment and width of the target instruction, and whether the processor manages to speculate the address early.
– B: The number of cycles required to perform the barrier operation. For and `DSB` and `DMB`, the minimum number of cycles is zero. For `ISB`, the minimum number of cycles is equivalent to the number required for a pipeline refill.
– N: The number of registers in the register list to be loaded or stored, including PC or LR.
– W: The number of cycles spent waiting for an appropriate event.

See the *ARMv7−M Architecture Reference Manual* for more information about the ARMv7−M Thumb instructions.

Table 1. Cortex−M3 Instruction Set Table

Operation	Description	Assembler	Cycles
Move	Register	MOV Rd, <op2>	1
	16−bit immediate	MOVW Rd, #<imm>	1
	Immediate into top	MOVT Rd, #<imm>	1
	To PC	MOV PC, Rm	1 + P
Add	Add	ADD Rd, Rn, <op2>	1
	Add to PC	ADD PC, PC, Rm	1 + P
	Add with carry	ADC Rd, Rn, <op2>	1
	Form address	ADR Rd, <label>	1
Subtract	Subtract	SUB Rd, Rn, <op2>	1
	Subtract with borrow	SBC Rd, Rn, <op2>	1
	Reverse	RSB Rd, Rn, <op2>	1
Multiply	Multiply	MUL Rd, Rn, Rm	1
	Multiply accumulate	MLA Rd, Rn, Rm	2
	Multiply subtract	MLS Rd, Rn, Rm	2
	Long signed	SMULL RdLo, RdHi, Rn, Rm	3 to 5 (Note 1)
	Long unsigned	UMULL RdLo, RdHi, Rn, Rm	3 to 5 (Note 1)
	Long signed accumulate	SMLAL RdLo, RdHi, Rn, Rm	4 to 7 (Note 1)
	Long unsigned accumulate	UMLAL RdLo, RdHi, Rn, Rm	4 to 7 (Note 1)
Divide	Signed	SDIV Rd, Rn, Rm	2 to 12 (Note 2)
	Unsigned	UDIV Rd, Rn, Rm	2 to 12 (Note 2)
Saturate	Signed	SSAT Rd, #<imm>, <op2>	1
	Unsigned	USAT Rd, #<imm>, <op2>	1

Table 1. Cortex−M3 Instruction Set Table

Operation	Description	Assembler	Cycles
Compare	Compare	CMP Rn, <op2>	1
	Negative	CMN Rn, <op2>	1
Logical	AND	AND Rd, Rn, <op2>	1
	Exclusive OR	EOR Rd, Rn, <op2>	1
	OR	ORR Rd, Rn, <op2>	1
	OR NOT	ORN Rd, Rn, <op2>	1
	Bit clear	BIC Rd, Rn, <op2>	1
	Move NOT	MVN Rd, <op2>	1
	AND test	TST Rn, <op2>	1
	Exclusive OR test	TEQ Rn, <op1>	
Shift	Logical shift left	LSL Rd, Rn, #<imm>	1
	Logical shift left	LSL Rd, Rn, Rs	1
	Logical shift right	LSR Rd, Rn, #<imm>	1
	Logical shift right	LSR Rd, Rn, Rs	1
	Arithmetic shift right	ASR Rd, Rn, #<imm>	1
	Arithmetic shift right	ASR Rd, Rn, Rs	1
Rotate	Rotate right	ROR Rd, Rn, #<imm>	1
	Rotate right	ROR Rd, Rn, Rs	1
	With extension	RRX Rd, Rn	1
Count	Leading zeroes	CLZ Rd, Rn	1
Load	Word	LDR Rd, [Rn, <op2>]	2 (Note 3)
	To PC	LDR PC, [Rn, <op2>]	2 (Note 3) + P
	Halfword	LDRH Rd, [Rn, <op2>]	2 (Note 3)
	Byte	LDRB Rd, [Rn, <op2>]	2 (Note 3)
	Signed halfword	LDRSH Rd, [Rn, <op2>]	2 (Note 3)
	Signed byte	LDRSB Rd, [Rn, <op2>]	2 (Note 3)
	User word	LDRT Rd, [Rn, #<imm>]	2 (Note 3)
	User halfword	LDRHT Rd, [Rn, #<imm>]	2 (Note 3)
	User byte	LDRBT Rd, [Rn, #<imm>]	2 (Note 3)
	User signed halfword	LDRSHT Rd, [Rn, #<imm>]	2 (Note 3)
	User signed byte	LDRSBT Rd, [Rn, #<imm>]	2 (Note 3)
	PC relative	LDR Rd,[PC, #<imm>]	2 (Note 3)
	Doubleword	LDRD Rd, Rd, [Rn, #<imm>]	1 + N
	Multiple	LDM Rn, {<reglist>}	1 + N
	Multiple including PC	LDM Rn, {<reglist>, PC}	1 + N + P
Store	Word	STR Rd, [Rn, <op2>]	2 (Note 3)
	Halfword	STRH Rd, [Rn, <op2>]	2 (Note 3)
	Byte	STRB Rd, [Rn, <op2>]	2 (Note 3)
	Signed halfword	STRSH Rd, [Rn, <op2>]	2 (Note 3)
	Signed byte	STRSB Rd, [Rn, <op2>]	2 (Note 3)
	User word	STRT Rd, [Rn, #<imm>]	2 (Note 3)
	User halfword	STRHT Rd, [Rn, #<imm>]	2 (Note 3)

Table 1. Cortex−M3 Instruction Set Table

Operation	Description	Assembler	Cycles
	User byte	STRBT Rd, [Rn, #<imm>]	2 (Note 3)
	User signed halfword	STRSHT Rd, [Rn, #<imm>]	2 (Note 3)
	User signed byte	STRSBT Rd, [Rn, #<imm>]	2 (Note 3)
	Doubleword	STRD Rd, Rd, [Rn, #<imm>]	1 + N
	Multiple	STM Rn, {<reglist>}	1 + N
Push	Push	PUSH {<reglist>}	1 + N
	Push with link register	PUSH {<reglist>, LR}	1 + N
Pop	Pop	POP {<reglist>}	1 + N
	Pop and return	POP {<reglist>, PC}	1 + N + P
Semaphore	Load exclusive	LDREX Rd, [Rn, #<imm>]	2
	Load exclusive half	LDREXH Rd, [Rn]	2
	Load exclusive byte	LDREXB Rd, [Rn]	2
	Store exclusive	STREX Rd, Rt, [Rn, #<imm>]	2
	Store exclusive half	STREXH Rd, Rt, [Rn]	2
	Store exclusive byte	STREXB Rd, Rt, [Rn]	2
	Clear exclusive monitor	CLREX	1
Branch	Conditional	B<cc> <label>	1 or 1 + P (Note 4)
	Unconditional	B <label>	1 + P
	With link	BL <label>	1 + P
	With exchange	BX Rm	1 + P
	With link and exchange	BLX Rm	1 + P
	Branch if zero	CBZ Rn, <label>	1 or 1 + P (Note 4)
	Branch if non−zero	CBNZ Rn, <label>	1 or 1 + P (Note 4)
	Byte table branch	TBB [Rn, Rm]	2 + P
	Halfword table branch	TBH [Rn, Rm, LSL#1]	2 + P
State change	Supervisor call	SVC #<imm>	−
	If−then−else	IT... <cond>	1 (Note 5)
	Disable interrupts	CPSID <flags>	1 or 2
	Enable interrupts	CPSIE <flags>	1 or 2
	Read special register	MRS Rd, <specreg>	1 or 2
	Write special register	MSR <specreg>, Rn	1 or 2
	Breakpoint	BKPT #<imm>	−
Extend	Signed halfword to word	SXTH Rd, <op2>	1
	Signed byte to word	SXTB Rd, <op2>	1
	Unsigned halfword	UXTH Rd, <op2>	1
	Unsigned byte	UXTB Rd, <op2>	1
Bit field	Extract unsigned	UBFX Rd, Rn, #<imm>, #<imm>	1
	Extract signed	SBFX Rd, Rn, #<imm>, #<imm>	1
	Clear	BFC Rd, Rn, #<imm>, #<imm>	1
	Insert	BFI Rd, Rn, #<imm>, #<imm>	1

Table 1. Cortex−M3 Instruction Set Table

Operation	Description	Assembler	Cycles
Reverse	Bytes in word	REV Rd, Rm	1
	Bytes in both halfwords	REV16 Rd, Rm	1
	Signed bottom halfword	REVSH Rd, Rm	1
	Bits in word	RBIT Rd, Rm	1
Hint	Send event	SEV	1
	Wait for event	WFE	1 + W
	Wait for interrupt	WFI	1 + W
	No operation	NOP	1
Barriers	Instruction synchronization	ISB	1 + B
	Data memory	DMB	1 + B
	Data synchronization	DSB <flags>	1 + B

1. UMULL, SMULL, UMLAL, and SMLAL instructions use early termination depending on the size of the source values. These are interruptible, that is abandoned and restarted, with worst case latency of one cycle.
2. Division operations use early termination to minimize the number of cycles required based on the number of leading ones and zeroes in the input operands.
3. Neighboring load and store single instructions can pipeline their address and data phases. This enables these instructions to complete in a single execution cycle.
4. Conditional branch completes in a single cycle if the branch is not taken.
5. An IT instruction can be folded onto a preceding 16−bit Thumb instruction, enabling execution in zero cycles.

CORTEX−M3 MICROCONTROLLER AND BUS FABRIC

NCS36510 integrates the powerful and energy efficient ARM Cortex−M3 processor that includes the integrated Nested Vectored Interrupt Controller (NVIC), Wake−up Interrupt Controller (WIC), and Debug Access Port (DAP). The processor uses the Thumb instruction set and is optimized for high performance with reduced code size and low power operation. The ARM Cortex−M3 efficiently handles multiple parallel peripherals and has integrated sleep modes. With industry standard tool chain and support, developing applications on the NCS36510 platform reduces time to market. Test and debug capability is enhanced with the ARM Serial Wire Debug Port with full debug capabilities. The microprocessor uses little−endian formatting.

The NCS36510 implementation of the ARM Cortex−M3 includes a 640 kB integrated FLASH memory (2 banks of 320 kB) with 48 kB of internal RAM memory (3 banks of 16 kB, 1 or 2 retainable in coma mode). The microprocessor, debug port, and memories are interconnected using the Advanced Microcontroller Bus Architecture (AMBA bus) AHB−Lite system interface bus. An AHB to APB Bridge is included to connect the peripherals.

Next to the regular ARM Cortex−M3 processor interrupts, the NCS36510 implements multiple external source interrupts for peripheral devices. A powerful nested, pre−emptive and priority based interrupt handling assure timely and flexible response to external events.

Low power features on NCS36510 include the WIC, adjustable clock rates, and different software controlled power modes to maximize opportunities to save power in final application.

Serial Wire Debug Access Port (SW−DAP)

The Debug Access Port (DAP) is included in this ARM Cortex−M3 implementation. Standard ARM Cortex−M3 Serial Wire Debugging (SWD) debugging is supported by NCS36510. JTAG is not supported

The NCS36510 implements full trace support for the Cortex−M3 which includes the Data Watchpoint and Trace Unit (DWT) with comparators and counters, Instrumentation Trace Macrocell (ITM), and Embedded Trace Macrocell (ETM).

The Trace Port Interface Unit (TPIU) supports Serial Wire Viewer (SWV) mode.

Refer to the ARM Cortex−M3 technical reference manual for a full definition of the debug system and its capabilities.

The Debug Port is disabled at power−up if the part is locked, and may be enabled by firmware. Driving the DBG_TEST_EN pin high will prevent the part from entering coma mode (see the PMU description).

The Debug Access Port interface implementation is the ARM Serial Wire Debug Port (SW−DP) connected to pins DIO[13] (SWCLK), DIO[12] (SWDIO), and DIO[11] (SWO). To enable the DAP drive the DBG_TEST_EN pin high and DIO[13] and DIO[12] are automatically reconfigured for DAP usage. The DIO[11] pin will only connect to SWO if the TRCENA bit is enabled.

Use any SWD compliant hardware debugger interface to interact with the internals of the NCS36510.

NCS36510 is optimized for battery powered applications and therefore uses reduced size digital drivers in the DIO pins. In 3V mode, the maximum practical DAP speed is 1 MHz. Adding external buffers could increase this speed but on the ON Semiconductor development board an upper reliable limit has been found at 1MHz.

Not all, if any, SWD debuggers support 1V mode operation.

Nested Vectored Interrupt Controller (NVIC)

The Cortex−M3 Nested Vectored Interrupt Controller (NVIC) supports priority based nested vectored interrupts. It includes 15 built−in or reserved exceptions and is configured with an additional 20 interrupts. Most interrupts have programmable priority. Priority levels available in the NVIC are 0, 64, 128, and 192. Lower numbers are higher priority. The priority of each group can be set separately by the firmware. While an interrupt is being serviced, only interrupts from a higher priority group will be serviced. If two interrupts of the same priority arrive at the same time, the earlier one (according to polling order) will be serviced first. The optional Wake−up Interrupt Controller (WIC) is included for low power mode support. Only a subset of the interrupts are included in the wake−up controller.

Table 2. Cortex−M3 Instruction Set Table

Exception Number	Exception Type	Priority	Description
1	Reset	−3 (Highest)	Reset
2	NMI	−2	Non−maskable interrupt. This is set to the watchdog interrupt.
3	Hard fault	−1	All fault conditions if the corresponding fault handler is not enabled.
4	MemManage fault	Programmable	Memory management fault
5	Bus fault	Programmable	Bus fault
6	Usage fault	Programmable	Exceptions resulting from program error.
7−10	Reserved		
11	SVC	Programmable	Supervisor Call
12	Debug Monitor	Programmable	Debug monitor (breakpoints, watchpoints, or external debug requests)
13	Reserved	Programmable	
14	PendSV	Programmable	Pendable Service Call
15	SYSTICK	Programmable	System Tick Timer
16	Timer 0	Programmable	Timer 0 interrupt
17	Timer 1	Programmable	Timer 1 interrupt
18	Timer 2	Programmable	Timer 2 interrupt
19	UART	Programmable	UART interrupt
20	SPI	Programmable	SPI interrupt
21	I2C	Programmable	I2C interrupt
22	GPIO	Programmable	GPIO interrupt
23	RTC	Programmable	Real−time−clock interrupt
24	Flash Controller	Programmable	Flash Controller.
25	MAC	Programmable	MAC interrupt
26	AES	Programmable	AES interrupt
27	ADC	Programmable	ADC interrupt
28	Clock Calibration	Programmable	Clock calibration interrupt
29	UART #2	Programmable	UART Interrupt
30	UVI	Programmable	Under Voltage Indicator Interrupt
31	DMA	Programmable	DMA interrupt
32	CDBGPWRUPREQ	Programmable	Debug request
33	SPI #2	Programmable	SPI #2 Interrupt
34	I2C #2	Programmable	I2C #2 Interrupt
35	FVDDH Comp	Programmable	FVDDH Supply Comparator Trip

MEMORY MAP

The 32 bit memory address space is broken up into regions for code, data, and multi−use. Memory elements consist of registers, RAM, and FLASH. A memory region is dedicated to the IOP and APB peripheral access. There are also regions for built in Cortex−M3 registers and NCS36510 peripherals. As in most Cortex−M3 designs there are many unused portions of the memory space. Individual memory regions and elements are described in subsequent sections.

Flash

NCS36510 contains a total of 640 kB of FLASH memory, organized as two banks of 320 kB each (81,920 words by 32 bits). Two independent FLASH banks are used to allow either OTA upgrades or dual stack applications. If a FLASH bank is unused, it can be powered down to save power.

Both main FLASH banks include an additional 8 kB information block (2048 words by 32 bits).

By default the FLASH A information block contains the bootloader and factory programmed trim values. There are a minimum of three application related trims that can be programmed by the customer: 32.768 kHz external oscillator, 32 MHz external oscillator, and the RSSI offset. These application trims can be determined during the design phase, and for a given PCB design they can be set to a constant value for all boards of the same design. At the factory these are set to a nominal value.

The bootloader is also stored in the FLASH A information block. To reprogram the bootloader it is required to drive the DBG_TEST_EN pin high and to write an unlock code to the FLASH. The factory trim contents must be read out, the entire FLASH information block erased, and then the bootloader and factory trims written back in. If the factory trims are lost on a device it will become inoperable as factory trims are not recoverable.

The FLASH B information block does not contain any factory trim information.

Flash Alias and Remap

NCS36510 has a FLASH remap feature that allows the FLASH A and FLASH B to change positions in the memory map when activated. This makes it easier to reboot the system from FLASH B if doing over the air firmware updates.

Another related feature is the FLASH alias. The FLASH alias allows the FLASH A and FLASH B contents to be visible in a fixed address space regardless of the remap setting.

Both FLASH remap and alias features are shown in the memory map diagrams.

During ON Semiconductor factory test, the factory trim values and bootloader are written to the FLASH A information block. No factory information is written to the FLASH B information block. Software must be careful when using the FLASH remap feature as the factory trims and bootloader could be missed since FLASH A and FLASH B change places in the memory map. An easy way to avoid issues is to use the FLASH A information block alias during boot up because the alias memory map does not depend on the remap setting. The ON Semiconductor software already takes care of this remap functionality, but this feature is important to understand for customers writing their own software.

RAM

The NCS36510 has three banks of 16kB each RAM. In coma mode either one or two of these banks can be retained.

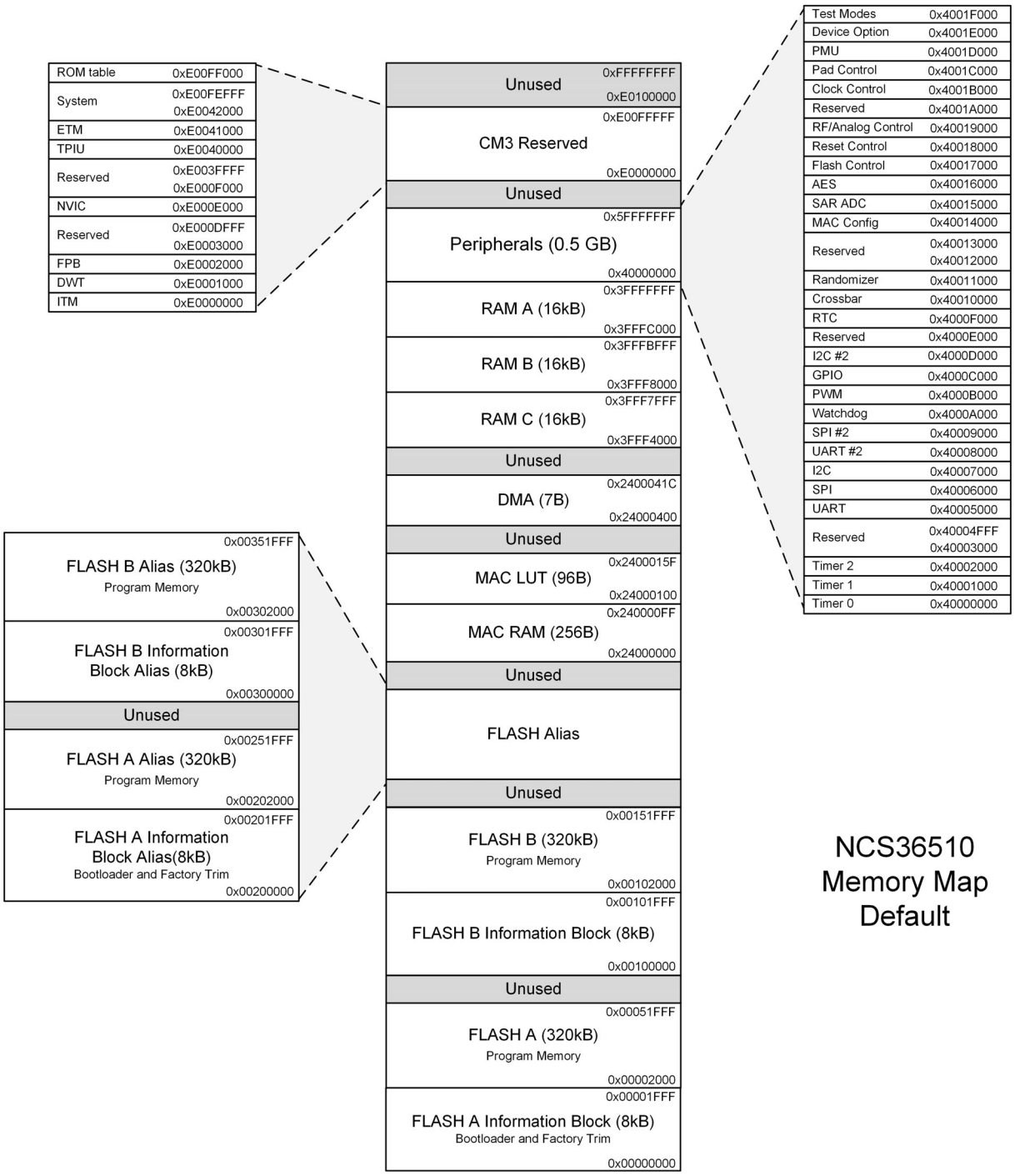


Figure 2. Default Memory Map

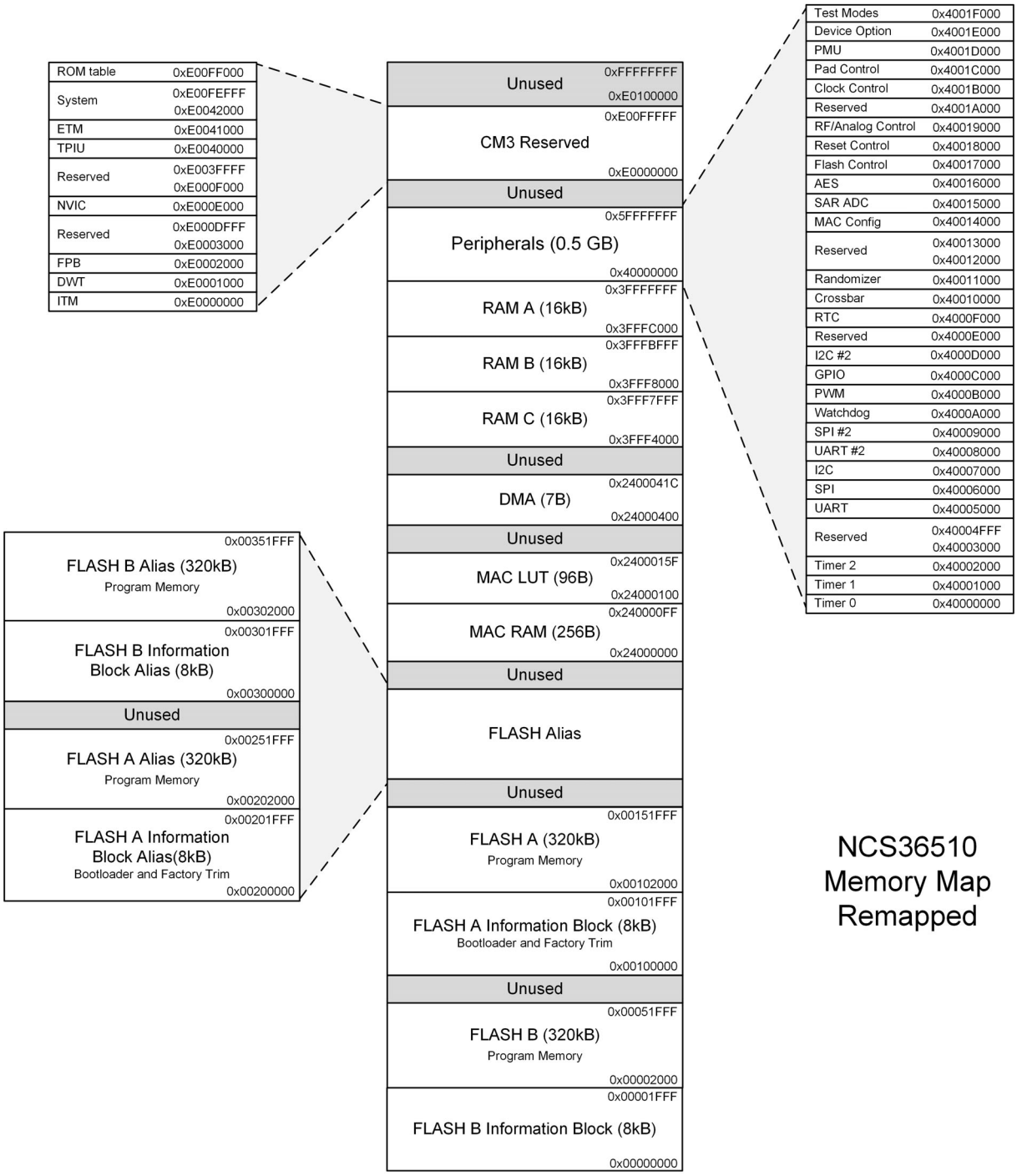


Figure 3. Remapped Memory Map

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
TIMER 0			
Load	0x4000_0000	R/W	Initial timer value
Value	0x4000_0004	RO	Current value of the timer
Control	0x4000_0008	R/W	Provides enable/disable, mode, and prescale configuration
Clear	0x4000_000C	WO	Clears the interrupt
TIMER 1			
Load	0x4000_1000	R/W	Initial timer value
Value	0x4000_1004	RO	Current value of the timer
Control	0x4000_1008	R/W	Provides enable/disable, mode, and prescale configuration
Clear	0x4000_100C	WO	Clears the interrupt
TIMER 2			
Load	0x4000_2000	R/W	Initial timer value
Value	0x4000_2004	RO	Current value of the timer
Control	0x4000_2008	R/W	Provides enable/disable, mode, and prescale configuration
Clear	0x4000_200C	WO	Clears the interrupt
UART 1			
Receive data	0x4000_5000	RO	Receive data
Transmit data	0x4000_5000	WO	Transmit data
Divisor latch LSB	0x4000_5000	RW	Least significant byte for input to baud generator
Divisor latch MSB	0x4000_5004	RW	Most significant byte for input to baud generator
Interrupt enable	0x4000_5004	RO	UART interrupt enables
Interrupt identification	0x4000_5008	RO	UART interrupt type/status register
FIFO control	0x4000_500C	WO	Enable FIFOs, clear FIFOs, etc.
Line control	0x4000_500C	RW	Specifies asynchronous data communications exchange format and sets the divisor latch access bit
Modem control	0x4000_5010	RW	Controls interface with the modem
Line status	0x4000_5014	RO	Data transfer status information
Modem status	0x4000_5018	RO	Current state of modem control lines
Scratch register	0x4000_501C	RW	Modem scratch register
SPI 1			
Transmit data	0x4000_6000	R/W	Transmit data
Receive data	0x4000_6004	R/W	Receive data
Serial clock divisor	0x4000_6008	R/W	SPI block clock divider setting (divide MCU clock)
Control	0x4000_600C	R/W	SPI control register
Status	0x4000_6010	RO	SPI status register
Slave select	0x4000_6014	R/W	SPI slave select control
Slave select polarity	0x4000_6018	R/W	SPI slave select polarity
Interrupt enable	0x4000_601C	R/W	SPI interrupt enables
Interrupt status	0x4000_6020	RO	SPI interrupt status register
Interrupt clear	0x4000_6024	WO	SPI interrupts clear register

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
SPI 1			
Transmit FIFO watermark	0x4000_6028	R/W	Set watermark for transmit FIFO half full flag
Receive FIFO watermark	0x4000_602C	R/W	Set watermark for receive FIFO half full flag
Transmit FIFO level	0x4000_6030	RO	Transmit FIFO level
Receive FIFO level	0x4000_6034	RO	Receive FIFO level
I2C 1			
Status	0x4000_7000	RO	I2C status register, clears upon read
Read data	0x4000_7004	RO	Read data FIFO access
Command	0x4000_7008	WO	I2C configuration
Interrupt Enable	0x4000_700C	R/W	Enable or disable I2C interrupts
Control	0x4000_7010	R/W	I2C control
Prescale	0x4000_7014	R/W	I2C block clock divider setting (divide MCU clock)
UART 2			
Receive data	0x4000_8000	RO	Receive data
Transmit data	0x4000_8000	WO	Receive data
Divisor latch LSB	0x4000_8000	RW	Least significant byte for input to baud generator
Divisor latch MSB	0x4000_8004	RW	Most significant byte for input to baud generator
Interrupt enable	0x4000_8004	RO	UART interrupt enables
Interrupt identification	0x4000_8008	RO	UART interrupt type/status register
FIFO control	0x4000_800C	WO	Enable FIFOs, clear FIFOs, etc.
Line control	0x4000_800C	RW	Specifies asynchronous data communications exchange format and sets the divisor latch access bit
Modem control	0x4000_8010	RW	Controls interface with the modem
Line status	0x4000_8014	RO	Data transfer status information
Modem status	0x4000_8018	RO	Current state of modem control lines
Scratch register	0x4000_801C	RW	Modem scratch register
SPI 2			
Transmit data	0x4000_9000	R/W	Transmit data
Receive data	0x4000_9004	R/W	Receive data
Serial clock divisor	0x4000_9008	R/W	SPI block clock divider setting (divide MCU clock)
Control	0x4000_900C	R/W	SPI control register
Status	0x4000_9010	RO	SPI status register
Slave select	0x4000_9014	R/W	SPI slave select control
Slave select polarity	0x4000_9018	R/W	SPI slave select polarity
Interrupt enable	0x4000_901C	R/W	SPI interrupt enables
Interrupt status	0x4000_9020	RO	SPI interrupt status register
Interrupt clear	0x4000_9024	WO	SPI interrupts clear register
Transmit FIFO watermark	0x4000_9028	R/W	Set watermark for transmit FIFO half full flag
Receive FIFO watermark	0x4000_902C	R/W	Set watermark for receive FIFO half full flag
Transmit FIFO level	0x4000_9030	RO	Transmit FIFO level
Receive FIFO level	0x4000_9034	RO	Receive FIFO level

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
WDT			
Load	0x4000_A000	R/W	Timer load value
Value	0x4000_A004	RO	Timer current value
Control	0x4000_A008	R/W	Enable
Kick	0x4000_A0C0	WO	Kick (reload timer from load value)
Lock	0x4000_A100	WO	Lock
Status	0x4000_A140	[3] R/W [2:0] RO	Status. Bit 3 is a R/W error bit (busy bit).
PWM			
Duty Cycle	0x4000_B000	WO	Duty cycle configuration
Enable	0x4000_B004	WO	PWM output enable
Disable	0x4000_B008	WO	PWM output disable
Prescale enable	0x4000_B00C	WO	Prescale select enable
Prescale disable	0x4000_B010	WO	Prescale select disable
Configuration and status	0x4000_B014	RO	Read PWM configuration and status
DIO			
State/Set	0x4000_C000	R/W	Read to see current state of synchronized input signals. Write ones to set corresponding outputs to 1, writes of 0 have no effect.
Interrupts/Clear	0x4000_C004	R/W	Read to see current state of interrupts. Write ones to set corresponding outputs to 0, writes of 0 have no effect.
Output Enable Set	0x4000_C008	WO	Write ones to set direction to output, writes of 0 have no effect on signal configuration
Output Enable Clear – Make input	0x4000_C00C	WO	Write ones to set direction to input (clears output enable), writes of 0 have no effect on signal configuration
Enable interrupts set	0x4000_C010	WO	Write ones to set enable interrupts, writes of 0 have no effect on interrupts configuration
Enable interrupts clear	0x4000_C014	WO	Write ones to set disable interrupts, writes of 0 have no effect on interrupts configuration
Edge interrupts select	0x4000_C018	WO	Write ones to set interrupt to edge−sensitive, writes of 0 have no effect on interrupts configuration
Level interrupts select	0x4000_C01C	WO	Write ones to set interrupt to level−sensitive, writes of 0 have no effect on interrupts configuration
Level interrupts set	0x4000_C020	WO	Write ones to set interrupts to active high or rising edge, writes of 0 have no effect on interrupts configuration
Level interrupts clear	0x4000_C024	WO	Write ones to clear interrupt for active low or falling edge, writes of 0 have no effect on interrupts configuration
Any edge interrupts set	0x4000_C028	WO	Write ones to override interrupt edge selection and interrupt on any edge, writes of 0 have no effect on interrupts configuration
Any edge interrupts clear	0x4000_C02C	WO	Write ones to clear edge selection override, writes of 0 have no effect on interrupts configuration
Interrupts clear	0x4000_C030	WO	Write ones to clear edge sensitive interrupts, writes of 0 have no effect on interrupts configuration

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
DIO			
Control	0x4000_C034	WO	Controls loopback/normal mode operation
I2C 2			
Status	0x4000_D000	RO	I2C status register, clears upon read
Read data	0x4000_D004	RO	Read data FIFO access
Command	0x4000_D008	WO	I2C configuration
Interrupt enable	0x4000_D00C	R/W	Enable or disable I2C interrupts
Control	0x4000_D010	R/W	I2C control
Prescale	0x4000_D014	R/W	I2C block clock divider setting (divide MCU clock)
RTC			
Sub−second counter	0x4000_F000	R/W	Sub−second counter. A write loads the written value and a read returns the current counter value.
Second counter	0x4000_F004	R/W	Second counter. A write loads the written value and a read returns the current counter value.
Sub−second alarm	0x4000_F008	R/W	Sub−second alarm value
Second alarm	0x4000_F00C	R/W	Second alarm value
RTC control	0x4000_F010	R/W	Control register for enables and interrupts
RTC status	0x4000_F014	WO/RO	Status register for errors, busy, and interrupts
RTC clear	0x4000_F018	WO	Clears the interrupt status
CROSSBAR (XBAR)			
DIO[0] control	0x4001_0000	R/W	Crossbar settings for given DIO pin
DIO[1] control	0x4001_0004	R/W	Crossbar settings for given DIO pin
DIO[2] control	0x4001_0008	R/W	Crossbar settings for given DIO pin
DIO[3] control	0x4001_000C	R/W	Crossbar settings for given DIO pin
DIO[4] control	0x4001_0010	R/W	Crossbar settings for given DIO pin
DIO[5] control	0x4001_0014	R/W	Crossbar settings for given DIO pin
DIO[6] control	0x4001_0018	R/W	Crossbar settings for given DIO pin
DIO[7] control	0x4001_001C	R/W	Crossbar settings for given DIO pin
DIO[8] control	0x4001_0020	R/W	Crossbar settings for given DIO pin
DIO[9] control	0x4001_0024	R/W	Crossbar settings for given DIO pin
DIO[10] control	0x4001_0028	R/W	Crossbar settings for given DIO pin
DIO[11] control	0x4001_002C	R/W	Crossbar settings for given DIO pin
DIO[12] control	0x4001_0030	R/W	Crossbar settings for given DIO pin
DIO[13] control	0x4001_0034	R/W	Crossbar settings for given DIO pin
DIO[14] control	0x4001_0038	R/W	Crossbar settings for given DIO pin
DIO[15] control	0x4001_003C	R/W	Crossbar settings for given DIO pin
DIO[16] control	0x4001_0040	R/W	Crossbar settings for given DIO pin
DIO[17] control	0x4001_0044	R/W	Crossbar settings for given DIO pin
TRNG			
Value	0x4001_1000	R/W	On a write sets the seed value, on a read returns random number
Control	0x4001_1004	R/W	Control register

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
TRNG			
Write buffer LSW	0x4001_1008	R/W	Byte swap write buffer – Least significant word
Write buffer MSW	0x4001_100C	R/W	Byte swap write buffer – Most significant word
Read buffer LSW	0x4001_1010	RO	Byte swap read buffer – Least significant word
Read buffer MSW	0x4001_1014	RO	Byte swap read buffer – Most significant word
Meta−stable latch TRNG value	0x4001_1018	RO	Meta−stable latch TRNG value
White noise TRNG value	0x4001_101C	RO	White noise TRNG value
IEEE 802.15.4 MEDIUM ACCESS CONTROL (MAC) – EXPERT REGISTERS			
Sequencer	0x4001_4000	R/W	Used to control MAC sequence operation
Sequence options	0x4001_4004	R/W	Set options that change behavior of basic events
Control	0x4001_4008	R/W	Control register
Status	0x4001_4010	RO	Status register
Options	0x4001_4014	R/W	Options register
PANID	0x4001_4018	R/W	Sets the MAC PAN ID
Short address	0x4001_401C	R/W	Device 16−bit short address
Long address (MSW)	0x4001_4020	R/W	The upper 32−bits of the device ID
Long address (LSW)	0x4001_4024	R/W	The lower 32−bits of the device ID
Divider	0x4001_4028	R/W	Prescaler divider for the protocol timer
RX/TX warmups	0x4001_402C	R/W	Set the warmup time for the transmitter & receiver
Clear interrupts	0x4001_4030	WO	Clears active interrupts
Enable interrupts	0x4001_4034	R/W	Enables/Disables certain interrupts
Interrupt status	0x4001_4038	RO	Interrupt status
Timer enable	0x4001_4040	R/W	Protocol timer control
Timer disable	0x4001_4044	R/W	Protocol timer control
Timer	0x4001_4048	R/W	Protocol timer
Start time	0x4001_404C	R/W	Event start time
Stop time	0x4001_4050	R/W	Event stop time
Timer status	0x4001_4054	RO	Start and stop timer status
Protocol timer state	0x4001_4058	RO	Protocol timer state
Finish time	0x4001_4060	RO	Records event finish time
Slot offset	0x4001_4064	R/W	Slot offset
Time stamp	0x4001_4068	RO	Records the protocol timer when the frame length field is received.
Coordinator short address	0x4001_406C	R/W	Coordinator short address
Coordinator long address	0x4001_4070	R/W	Coordinator long address (MSW)
Coordinator long address	0x4001_4074	R/W	Coordinator long address (LSW)
RX Length	0x4001_4088	RO	The length of the received frame
TX Length	0x4001_408C	R/W	The length of the transmit frame
TX sequence number	0x4001_4090	R/W	Sequence number to be used for the transmit frame
TX ACK delay	0x4001_4094	R/W	Delay from end of frame receive to start of ACK transmit

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
IEEE 802.15.4 MEDIUM ACCESS CONTROL (MAC) – EXPERT REGISTERS			
RX ACK delay	0x4001_4098	R/W	Delay from end of frame transmit to begin of automatic receive ACK
TX flush	0x4001_409C	R/W	Set extra time to hold transmitter on at the end of a transmitted frame
CCA	0x4001_40A0	R/W	Set CCA measurement length & delay between CCA measurements for a slotted mode transmission
ACK stop	0x4001_40A4	R/W	Length of time receive hardware will wait for incoming ACK
TX CCA delay	0x4001_40A8	R/W	Delay from last CCA measurement to start of frame transmit
Long address LUT	0x4001_40AC	R/W	Long address look up table (LUT)
Short Address LUT	0x4001_40B0	R/W	Short address look up table (LUT)
Frame match result	0x4001_40B4	RO	Result vector from frame matching
Frame match long address	0x4001_40B8	RO	Long address from LUT for frame matched addressed
Frame match short address	0x4001_40BC	RO	Short address from LUT for frame matched addressed
AGC control	0x4001_40C0	R/W	Receiver Automatic Gain Control (AGC) control register
AGC settings	0x4001_40C4	R/W	Receiver Automatic Gain Control (AGC) settings register
AGC status	0x4001_40C8	RO	Receiver Automatic Gain Control (AGC) status register
AGC gain table 0	0x4001_40CC	R/W	Receiver Automatic Gain Control (AGC) gain table 0
AGC gain table 1	0x4001_40D0	R/W	Receiver Automatic Gain Control (AGC) gain table 1
AGC gain table 2	0x4001_40D4	R/W	Receiver Automatic Gain Control (AGC) gain table 2
AGC gain table 3	0x4001_40D8	R/W	Receiver Automatic Gain Control (AGC) gain table 3
Demodulator control 0	0x4001_4100	R/W	Receiver demodulator control 0
Demodulator control 1	0x4001_4104	R/W	Receiver demodulator control 1
Demodulator control 2	0x4001_4108	R/W	Receiver demodulator control 2
Demodulator status	0x4001_410C	RO	Receiver demodulator status
SAR ADC			
Control	0x4001_5000	R/W	Control the ADC
Delay	0x4001_5004	R/W	Sets timing critical values of the ADC
Data	0x4001_5008	RO	Data from the ADC
Interrupt	0x4001_500C	R/W	Interrupt control
Prescaler	0x4001_5010	R/W	Prescaler for the ADC clock
Status	0x4001_5014	RO	Status
AES			
Key 0	0x4001_6000	WO	Least significant word of key
Key 1	0x4001_6004	WO	Next significant word of key

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
AES			
Key 2	0x4001_6008	WO	Next significant word of key
Key 3	0x4001_600C	WO	Most significant word of 128 bit key
Key 4	0x4001_6010	WO	Next significant word of key
Key 5	0x4001_6014	WO	Next significant word of key
Key 6	0x4001_6018	WO	Next significant word of key
Key 7	0x4001_601C	WO	Most significant word of 256 bit key
Counter 0	0x4001_6020	R/W	Least significant word of counter
Counter 1	0x4001_6024	R/W	Next significant word of counter
Counter 2	0x4001_6028	R/W	Next significant word of counter
Counter 3	0x4001_602C	R/W	Most significant word of counter
Counter result 0	0x4001_6030	RO	Least significant word of counter result
Counter result 1	0x4001_6034	RO	Next significant word of counter result
Counter result 2	0x4001_6038	RO	Next significant word of counter result
Counter result 3	0x4001_603C	RO	Most significant word of counter result
CBC result 0	0x4001_6040	RO	Least significant word of CBC result
CBC result 1	0x4001_6044	RO	Next significant word of CBC result
CBC result 2	0x4001_6048	RO	Next significant word of CBC result
CBC result 3	0x4001_604C	RO	Next significant word of CBC result
Control	0x4001_6050	R/W	Clear interrupt, clear CBC accumulator, start encryption command
Mode	0x4001_6054	R/W	Set encryption mode, length, and enable/disable interrupt mask
Status	0x4001_6058	RO	Status register
MAC Initial value 0	0x4001_605C	R/W	Least significant word of the 128−bit CBC initial data
MAC Initial value 1	0x4001_6060	R/W	Next significant word of the 128−bit CBC initial data
MAC Initial value 2	0x4001_6064	R/W	Next significant word of the 128−bit CBC initial data
MAC Initial value 3	0x4001_6068	R/W	Most significant word of the 128−bit CBC initial data
Data 0	0x4001_6070	R/W	Least significant word of data to encrypt
Data 1	0x4001_6074	R/W	Next significant word of data to encrypt
Data 2	0x4001_6078	R/W	Next significant word of data to encrypt
Data 3	0x4001_607C	R/W	Most significant word of data to encrypt
FLASH Control			
Status	0x4001_7000	RO	Flash Controller Status Registers
Control	0x4001_7004	R/W	Flash Control Register
Command	0x4001_7008	R/W	Flash Command Register
Address	0x4001_700C	R/W	Flash Address Register
Unlock command	0x4001_7010	R/W	Unlock command required before unlocking either FLASH A or B
Unlock FLASH A	0x4001_7014	R/W	Unlock FLASH A

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
FLASH Control			
Unlock FLASH B	0x4001_7018	R/W	Unlock FLASH B
Interrupt	0x4001_701C	RO	Interrupt status
Reset Control			
Reset sources	0x4001_8000	RO	Status of all reset sources
Clear reset sources	0x4001_8004	WO	Clears the reset sources register
Hardware revision number	0x4001_8008	RO	Specifies the hardware revision number
Control	0x4001_800C	R/W	External RESETN reset and WDT reset impact of debug logic
RF AND ANALOG CONTROL – EXPERT REGISTERS			
TX frequency control	0x4001_9000	R/W	Sets the transmit integer and fractional divider words for PLL
RX frequency control	0x4001_9004	R/W	Sets the receive integer and fractional divider words for PLL
Transmit power	0x4001_9010	R/W	Sets the output transmit power, not the same as settings required by IEEE 802.15.4 as handled by the MAC software. This is the raw power control to the power amplifier.
Receiver gain	0x4001_9014	RO	Current receiver gain value from AGC module
FVDDH supply comparator threshold	0x4001_9084	R/W	Sets the threshold to compare against
Transmitter trim	0x4001_9094	[3:0] R/W	Transmitter pre−driver tank circuit trim. Board design dependent.
CLOCK CONTROL			
Control	0x4001_B000	[4] R/W [3:2] WO [1:0] R/W	RTC clock enable Enable/disable 32MHz/32.768kHz internal oscillator calibration 32MHz oscillator source internal or external
Status	0x4001_B004	RO	Status register for state of DBG_TEST_EN pin and clock calibrations and clock source readiness
Interrupt enable	0x4001_B008	R/W	Clock calibration interrupt enable register
Clear	0x4001_B00C	WO	Clear the interrupt & status registers
Peripheral disable	0x4001_B010	R/W	Peripheral block disable register (gates clock to peripheral blocks)
FCLK prescaler	0x4001_B014	R/W	FCLK prescaler value
TRACECLK prescaler	0x4001_B018	R/W	Trace clock prescaler value
Internal 32MHz trim	0x4001_B020		The internal 32MHz RC oscillator trim value
Internal 32.768kHz trim	0x4001_B024		The internal 32kHz RC oscillator trim value
External 32MHz trim	0x4001_B028		The external 32MHz crystal oscillator trim and settings
External 32.768kHz trim	0x4001_B02C		The external 32.768kHz crystal oscillator trim and settings
DIO PAD CONTROL			
DIO[0] pad control	0x4001_C000	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[1] pad control	0x4001_C004	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
DIO PAD CONTROL			
DIO[2] pad control	0x4001_C008	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[3] pad control	0x4001_C00C	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[4] pad control	0x4001_C010	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[5] pad control	0x4001_C014	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[6] pad control	0x4001_C018	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[7] pad control	0x4001_C01C	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[8] pad control	0x4001_C020	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[9] pad control	0x4001_C024	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[10] pad control	0x4001_C028	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[11] pad control	0x4001_C02C	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[12] pad control	0x4001_C030	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[13] pad control	0x4001_C034	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[14] pad control	0x4001_C038	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[15] pad control	0x4001_C03C	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[16] pad control	0x4001_C040	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
DIO[17] pad control	0x4001_C044	R/W	Output driver configuration (open drain or push/pull), drive strength, and/or weak pull (up/down/both/none) setting for related DIO pad
POWER MANAGEMENT UNIT (PMU)			
Control	0x4001_D000		Control settings including UVI, Debugger power mode behavior, pre−regulator settings in active and coma modes, internal and external slow and fast clock controls, coma mode RAM controls, and power mode behavior after WFI instruction
Status	0x4001_D004	RO	Status

Table 3. PERIPHERAL REGISTER TABLE SUMMARY

Peripheral Register	Address	Access	Description
POWER MANAGEMENT UNIT (PMU)			
FVDD power up	0x4001_D00C		Flash power supply up timer
FVDD power down	0x4001_D010		Flash power supply down timer
UVI time base	0x4001_D018	[13:8] RO [5:0] R/W	Read only value for number of cycles comparator is high Threshold for comparator
RAM trim	0x4001_D01C		FMS?

Table 4. DMA REGISTER TABLE SUMMARY

Register	Address	Access	Description
DMA			
Control	0x2400_0400	R/W	Control register
Source address	0x2400_0404	R/W	Source address register
Destination address	0x2400_0408	R/W	Destination address register
Transfer size	0x2400_040C	R/W	Transfer size register
Status	0x2400_0410	RO	Status register
Interrupt enable	0x2400_0414	R/W	Interrupt enable register
Interrupt status	0x2400_0418	RO	Interrupt status register

DBG_TEST_EN Pin

The DBG_TEST_EN pin has several system level impacts that need to be understood by the software developer. This section documents all the interactions this pin's state has on the system. Further details are located throughout the document but summarized here.

The DBG_TEST_EN pin must be driven high to enable SW−DAP debugging. See the SW−DAP section for more details about debugging.

If programming the FLASH A information blocks, or in other words programming the bootloader or modifying factory trims, the DBG_TEST_EN pin must be high as well as the unlock codes written to the FLASH controller (the DBG_TEST_EN state influences some timeouts in the FLASH controller also). Further details are given in the FLASH controller section.

Warning: The NCS36510 has a POR test mode intended for use during ON Semiconductor factory testing only. If the DBG_TEST_EN pin is high, and the RESETN pin low while powering up the device, this POR test mode will be activated. The DIO[0] pin will mirror the internal POR signal. The only way to exit this mode is to power down the device, and restart with the DBG_TEST_EN pin low and or the RESETN pin high.

FLASH CONTROLLER

Description

The flash controller is used to interface with the two FLASH arrays A, and B.

During boot−up or when coming out of deep−sleep mode there is a delay before the FLASH is available for read, write, or erase operations. This delay is due to the start−up time of the internal FLASH power supplies. During this power−up time, the busy bit will be set to high in the status register. Since the processor may start to try to boot during this time, the FLASH controller will store the address and read command but not assert the ready signal until the FLASH becomes available and the correct address has been read. The read will complete at that point and the data will be returned on the bus. The error bits in the status register will be set to the code for attempted access while the FLASH array is busy powering up.

However if a write is attempted while the FLASH is busy powering up, that write will be ignored and the status register will show an error for attempted access while the array is busy powering up. If interrupts are enabled for access error, that interrupt will be generated.

Two types of erase operations are allowed: a page erase and a mass erase. A page erase will erase a single page in a memory region. A mass erase will erase an entire array. Once an erase operation is complete the FLASH arrays are written to via a write operation on the AHB bus.

It is only allowed to do two write cycles to a single location per erase cycle. Meaning for a 32 bit wide array, software can only write to a single location twice before the location must be erased again. Hardware does not perform any monitoring of this condition.

While a FLASH instance is being programmed, it cannot support any other access. In some circumstances, it may be advantageous to have the HREADYOUT signal simply prevent whatever is attempting access to the FLASH have all transactions freeze until the write operation completes. For other cases, having, for instance, the CPU hang until the write operation completes is not desired. The 'Block AHB bus during write' bit in the Control register allows for both methods. If this bit is cleared (0), then the processor, debugger, etc. can perform a write and continue on with other operations immediately. In this mode, the busy bits must be polled, or an interrupt used to determine when the flash becomes available. If this bit is set (1), then the process writing the FLASH will hang until the flash completes writing. This is mode is required to use DMA to write to the FLASH.

Table 5. REGISTERS

Function	Bits	Default	Type	Description
FLASH Controller Status Register: 0x4001_7000 Status register that indicates if any FLASH control errors are encountered, if FLASH A and/or FLASH B are unlocked, and if FLASH A and/or FLASH B are busy. Clears upon read.				
Error type	[7:5]	0x00	RO	000 – No error 111 – Attempt to access an array powering up 001 – Attempt to erase bootloader (DBG_TEST_EN = low) 010 – Attempt to access array during erase 100 – Attempt to access array during write
FLASH B unlocked indicator	[3]	0x0	RO	0 – FLASH B is locked 1 – FLASH B is unlocked
FLASH A unlocked indicator	[2]	0x0	RO	0 – FLASH A is locked 1 – FLASH A is unlocked
FLASH B busy indicator	[1]	0x0	RO	0 – FLASH B is not busy 1 – FLASH B is busy
FLASH A busy indicator	[0]	0x0	RO	0 – FLASH A is not busy 1 – FLASH A is busy
FLASH Controller Control Register: 0x4001_7004 FLASH controller control register for FLASH A and B to control AHB bus blocking, interrupt control, remap control, and power down controls.				
Block AHB bus during write	[6]	0x0	R/W	0 – AHB HREADYOUT high during write 1 – AHB HREADYOUT low during write
Error interrupts control	[5]	0x0	R/W	0 – Disable FLASH errors interrupts 1 – Enable FLASH errors interrupts
Erase interrupts control	[4]	0x0	R/W	0 – Disable FLASH erase complete interrupts 1 – Enable FLASH erase complete interrupts
Write interrupts control	[3]	0x0	R/W	0 – Disable FLASH write complete interrupts 1 – Enable FLASH write complete interrupts
Enable remap function	[2]	0x0	R/W	0 – Default memory map 1 – Remapped memory map
FLASH A power down control	[1]	0x0	R/W	0 – FLASH A powered on 1 – FLASH A powered off
FLASH B power down control	[0]	0x0	R/W	0 – FLASH B powered on 1 – FLASH B powered off
Function	**Bits**	**Default**	**Type**	**Description**
FLASH Controller Command Register: 0x4001_7008 FLASH controller command register to initiate read mode (no op), page erase, or mass erase.				
Command	[1:0]	0x00	R/W	00 – No operation, array is read–only 01 – Page erase 10 – Mass erase
FLASH Controller Address Register: 0x4001_700C FLASH controller address register for FLASH A and B. When doing a page erase the lower LSBs are ignored. During a mass erase the lower LSBs are ignored. For example if this register is set to 0x000102FF and a mass erase command is given, the entire main memory section will be erased. This would be the same as setting the address to 0x00100200.				

Table 5. REGISTERS

Function	Bits	Default	Type	Description
Address pointer	[31:0]	0x00000000	R/W	Address pointer for FLASH page and mass erase operations
Function	**Bits**	**Default**	**Type**	**Description**
FLASH Controller Unlock FLASH Register: 0x4001_7010 FLASH controller unlock register for FLASH A and B. This register must be written to 0xBB781AE9 to start the FLASH A or B unlock procedure. After this register is written with the proper unlock code, the corresponding unlock FLASH A and/or FLASH B registers must be written within 20 clock cycles (HCLK) or the FLASH controller automatically relocks. If the DBG_TEST_EN pin is high, there is no clock cycle limit. Both this unlock register and corresponding FLASH A and/or FLASH B unlock registers have to be unlocked before a valid erase or write operation can occur.				
FLASH unlock	[31:0]	0x4487E516	R/W	Write to 0xBB781AE9 to start unlock sequence. See note above for timing requirements and DBG_TEST_EN influence.
Function	**Bits**	**Default**	**Type**	**Description**
FLASH Controller Unlock FLASH A Register: 0x4001_7014 FLASH controller unlock register for FLASH A. This register must be written to 0xB56D9099 to complete the FLASH A unlock procedure. The unlock FLASH register 0x40017010 must be unlocked before this unlock can take effect. This register automatically relocks after the erase or write cycle completes on FLASH A.				
FLASH A unlock	[31:0]	0x4A926F66	R/W	Write to 0xB56D9099 to complete unlock sequence. Requires unlocking register 0x40017010 first.
Function	**Bits**	**Default**	**Type**	**Description**
FLASH Controller Unlock FLASH B Register: 0x4001_7018 FLASH controller unlock register for FLASH B. This register must be written to 0xB56D9099 to complete the FLASH B unlock procedure. The unlock FLASH register 0x40017010 must be unlocked before this unlock can take effect. This register automatically relocks after the erase or write cycle completes on FLASH B.				
FLASH B unlock	[31:0]	0x4A926F66	R/W	Write to 0xB56D9099 to complete unlock sequence. Requires unlocking register 0x40017010 first.
Function	**Bits**	**Default**	**Type**	**Description**
FLASH Controller Interrupt Status Register: 0x4001_701C FLASH related interrupts may be generated under three conditions: 1. Access errors. If the system attempts to access an array while it is busy powering up, being programmed, or being erased. If bit 5 of the FLASH Control Register is set to a 1, an attempt to access the array while it is busy will generate an interrupt. The interrupt may be cleared by reading the FLASH controller status register. 2. Erase complete. On completion of either a page erase, or a mass erase. If bit 4 of the FLASH control register is set to a 1, an interrupt will be generated when an erase operation completes, either page or mass erase. The interrupt may be cleared by reading the FLASH controller status register. 3. Program complete. On completion of writing data to an address. If bit 3 of the FLASH control register is set to a 1, an interrupt will be generated when a write to a single address completes. The interrupt may be cleared by reading the FLASH controller status register.				
Most recent interrupt type	[3:1]	0x0	RO	100 – Access error 010 – Erase complete 001 – Write complete 000 – No interrupt since reset
Interrupt pending	[0]	0x0	RO	0 – No interrupt pending 1 – Interrupt pending

RESET AND BROWNOUT CONTROL

Description

NCS36510 has various reset sources: Power−On Reset (POR) and Brownout (BO), external reset, software reset, and watchdog timer reset.

The Cortex−M3 lockup bit is also made available, but does not directly cause any reset to be applied.

Internal Power−On Reset and Brownout

The POR and BO functions are combined in the Power Management Unit (PMU). During startup, the POR is released when V3V is at a high enough voltage to support the internal digital logic voltage regulators. After power up the voltage at V1V is monitored and if it gets too low, a brownout reset is generated. A POR and a brownout have the same effect on the system which is a full reset including the processor debug logic. Upon POR or BO the processor jumps to the reset vector and the system reboots.

External Reset

When the external reset pin is driven low, the NCS36510 is held in reset. The processor debug logic is not reset.

Warning: The NCS36510 has a POR test mode intended for use during ON Semiconductor factory testing only. If the DBG_TEST_EN pin is high, and the RESETN pin low while powering up the device, this POR test mode will be activated. The DIO[0] pin will mirror the internal POR signal. The only way to exit this mode is to power down the device, and restart with the DBG_TEST_EN pin low and or the RESETN pin high.

Software Reset

Software reset can be called when switching from one application to the other, after remap of the FLASH banks, or on exit of a processor exception. The software requested reset will not reset all processor or peripheral device registers.

Watchdog Timer Reset

NCS36510 implements a programmable watchdog timer. The watchdog timer is disabled by default and the application software needs to instantiate the watchdog timer driver and enable it. The WDT has a register locking safety mechanism to prevent errant software from corrupting the WDT registers. While locked the only supported operation is a clear. The watchdog is on the 32.768 kHz clock domain so it has a minimum resolution of 30.5 μS. It is 18 bits wide giving it a maximum timeout time of 8 seconds. When the WDT overflows, the system is reset and the reset sources register is updated to indicate the system was reset by the watchdog timer. If a debugger is attached then the WDT is paused.

Table 6. REGISTERS

Function	Bits	Default	Type	Description
Reset Sources Register: 0x4001_8000 Register describing what source the last NCS36510 reset came from. Choices include POR/BO, software reset, external reset, and Watchdog.				
POR/BO Reset	[4]	0x0	RO	0 – POR/BO reset did NOT occur 1 – POR/BO reset DID occur
Software Reset – SYSRESE-TREQ	[3]	0x0	RO	0 – Software reset did NOT occur 1 – Software reset DID occur
External Reset	[2]	0x0	RO	0 – External reset did NOT occur 1 – External reset DID occur
Watchdog Reset	[1]	0x0	RO	0 – Watchdog reset did NOT occur 1 – Watchdog reset DID occur
Cortex−M3 lockup	[0]	0x0	RO	0 – MCU did NOT lockup 1 – MCU DID lockup Note: Lockup event does NOT reset NCS36510 and the lockup bit only indicates that a lockup occurred.
Function	**Bits**	**Default**	**Type**	**Description**
Clear Reset Sources Register: 0x4001_8004 Clear the reset sources register.				
Clear reset sources	[31:0]	n/a	WO	Write any value to this register to clear reset sources register.
Function	**Bits**	**Default**	**Type**	**Description**
Hardware Revision Register: 0x4001_8008 Hardware revision number is available in this register.				
Hardware revision number	[31:0]	0x80215405	RO	

Table 6. REGISTERS

Function	Bits	Default	Type	Description
Reset Sources Control Register: 0x4001_800C External reset and Watchdog reset impact on debug logic.				
External Reset and Watchdog impact on debug logic	[0]	0x0	R/W	0 – External reset and watchdog WILL reset debug logic 1 – External reset and watchdog will NOT reset debug logic

CLOCKS

Description

There are two major clock domains on NCS36510. There is a high speed 32 MHz oscillator domain and a low speed 32.768 kHz oscillator domain. During coma mode operation the only clock available is the 32.768 kHz oscillator.

The high speed 32MHz oscillator has two sources. Either the internal oscillator or an external crystal based oscillator. The crystal based clock is needed for 802.15.4 RF carrier frequency accuracy requirements. The internal oscillator is typically used for fast boot up before the external crystal oscillator is enabled and ready. The 32 MHz system clock (FCLK) can be divided by the following values: 1, 2, 3, 4, 5, 6, 7, and 8.

The low speed 32.768 kHz oscillator has two sources. Either the internal oscillator or an external crystal based oscillator. The 32.768 kHz internal oscillator is typically used during low power modes as it has the lowest power consumption. If accuracy is a higher priority over power consumption then the external 32.768 kHz oscillator can be used instead.

Both the internal 32MHz and internal 32.768 kHz oscillators can be calibrated against the required 32 MHz crystal oscillator. Both internal oscillators are sensitive to temperature changes of the NCS36510. Periodic calibration is recommended if frequency accuracy is important.

Both crystal oscillators have a boost mode that is automatically controlled during crystal oscillator startup. The boost mode injects extra energy into the resonant crystal circuit formed by the crystal and the internal amplifier and shunt caps. The gain of the internal amplifier can be set by software and the internal shunt capacitors can be programmed, allowing fine tune pulling of the external crystal for frequency fine−tuning. After startup, the boost mode automatically disables to lower power consumption.

By default, the system clock is gated to all the peripherals. The peripheral disable register bits for the required peripherals must be cleared to enable the system clock to clock the desired peripheral. When disabled, register writes to the peripherals will be silently ignored and will have no effect.

Debug Port Lockout

The debug access port (DAP) lock is used to disable the serial wire debug port to prevent access to the internal buses and memory for security sensitive applications. During power−up the word at address 0x00001FC8 in the FLASH memory will immediately be read by the ON Semiconductor bootloader firmware. If the value read is set to 0x00D1EDEB, the DAP lock enable bit will be set in the clock control register and the debug port will be disabled. The debug port is disabled by gating the clock, which is why the lock enable is in the clock control register.

Table 7. REGISTERS

Function	Bits	Default	Type	Description
Clock Control Register: 0x4001_B000 Control the NCS36510 clocking options. The Debug Access Port (DAP) lock enable is also in this register.				
Real Time Clock (RTC) control	[4]	0x0	R/W	0 – RTC disabled 1 – RTC enabled
Calibrate internal 32 MHz oscillator	[3]	0x0	WO	0 – Disable calibration 1 – Enable calibration
Calibrate internal 32.768 kHz oscillator	[2]	0x0	WO	0 – Disable calibration 1 – Enable calibration
DAP lock enable	[1]	0x0	R/W	0 – No effect, reset will disable dap lock 1 – Enable DAP lock
High speed oscillator select	[0]	0x0	R/W	0 – Use internal 32 MHz oscillator 1 – Use external 32 MHz crystal oscillator
Clock Status Register: 0x4001_B004 Various clock related status registers. Logic state of the DBG_TEST_EN pin also visible in this register.				

Table 7. REGISTERS

Function	Bits	Default	Type	Description
State of the DBG_TEST_EN pin	[31]	0x0	RO	0 – DBG_TEST_EN pin is at logic level 0 1 – DBG_TEST_EN pin is at logic level 1
Internal 32 MHz oscillator calibration complete	[5]	0x0	RO	0 – Calibration not started and/or not completed 1 – Calibration is finished
Internal 32 MHz oscillator calibration status	[4]	0x0	RO	0 – Not started or finished successfully 1 – Calibration failed
Internal 32.768 kHz oscillator calibration complete	[3]	0x0	RO	0 – Calibration not started and/or not completed 1 – Calibration is finished
Internal 32.768 kHz oscillator calibration status	[2]	0x0	RO	0 – Not started or finished successfully 1 – Calibration failed
External 32.768 kHz crystal oscillator status	[1]	0x0	RO	0 – Not running or not stable 1 – Running and stable
External 32 MHz crystal oscillator status	[0]	0x0	RO	0 – Not running or not stable 1 – Running and stable
Function	**Bits**	**Default**	**Type**	**Description**
Clock Interrupts Register: 0x4001_B008 Manage clock related interrupts.				
Calibrate internal 32 MHz oscillator interrupt enable	[1]	0x0	R/W	0 – Disabled 1 – Enabled
Calibrate internal 32.768 kHz oscillator interrupt enable	[0]	0x0	R/W	0 – Disabled 1 – Enabled
Function	**Bits**	**Default**	**Type**	**Description**
Clock Interrupts Register: 0x4001_B00C Clear clock related interrupts and status register.				
Clock interrupt and status register clear	[31:0]	0x0	n/a	Writing any value to this register will clear all clock related interrupts and the clock status register
Function	**Bits**	**Default**	**Type**	**Description**
Peripheral Clock Disable Register: 0x4001_B010 To save power, the clocks to the peripherals are gated by default. Clearing the disable will enable the peripheral to be clocked and therefore usable. If the peripheral clocks are disabled for a given peripheral, writes to their related registers will be ignored. The peripheral clock, PCLK, follows the FCLK if divided.				
NCS36510 test modes	[31]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
DMA	[30]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Power management unit	[29]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Pad control	[28]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Unused	[27]	0x1	R/W	
Digital I/O control	[26]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
RF and analog control	[25]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Unused	[24]	0x1	R/W	
FLASH control	[23]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
AES engine	[22]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated

Table 7. REGISTERS

Function	Bits	Default	Type	Description
SAR ADC	[21]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
802.15.4 hardware MAC	[20]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Unused	[19:18]	0x1	R/W	
TRNG	[17]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Crossbar	[16]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
RTC	[15]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Unused	[14]	0x1	R/W	
I2C 2	[13]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
GPIO	[12]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
PWM	[11]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Watchdog APB connected control register (Watchdog runs on 32.768 kHz domain, just the control register is on high speed clock domain)	[10]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
SPI 2	[9]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
UART 2	[8]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
I2C 1	[7]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
SPI 1	[6]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
UART 1	[5]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Unused	[4:3]	0x11	R/W	
Timer 2	[2]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Timer 1	[1]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Timer 0	[0]	0x1	R/W	1 – System clock to peripheral is gated 0 – System clock to peripheral is un−gated
Function	**Bits**	**Default**	**Type**	**Description**
System Clock Prescaler (FCLK) Register: 0x4001_B014 System clock divider, FCLK. Clock reference is either the internal 32 MHz oscillator or the external 32 MHz crystal oscillator.				
FCLK prescaler value	[2:0]	0x03	R/W	000 – Divide by 1 001 – Divide by 2 010 – Divide by 3 011 – Divide by 4 100 – Divide by 5 101 – Divide by 6 110 – Divide by 7 111 – Divide by 8 Note: It's recommended to sequence the divider values using grey encoding.

Table 7. REGISTERS

Function	Bits	Default	Type	Description
TRACECLK Prescaler Register: 0x4001_B018 Prescaler value for the debugger TRACECLK. Clock reference is either the internal 32 MHz oscillator or the external 32 MHz crystal oscillator (FCLK).				
TRACECLK prescaler value	[7:0]	0x00	R/W	The output frequency is given by:
Function	**Bits**	**Default**	**Type**	**Description**
Internal 32 MHz Oscillator Trim Value Register: 0x4001_B020 Trim value, typically used to center the internal 32 MHz oscillator as close as possible to the 32 MHz external crystal oscillator frequency. The calibration routine sets this value but it can also be written to by a debugger or software.				
Internal 32 MHz oscillator trim value	[5:0]	0x00	R/W	This value is set after calibration to the external 32 MHz crystal oscillator. Software and/or a debugger can overwrite it.
Function	**Bits**	**Default**	**Type**	**Description**
Internal 32.768 kHz Oscillator Trim Value Register: 0x4001_B024 Trim value, typically used to center the internal 32.768 kHz oscillator as close as possible to the reference based on the 32 MHz external crystal oscillator frequency. The calibration routine sets this value but it can also be written to by a debugger or software.				
Internal 32.768 kHz oscillator trim value	[10:0]	0x000	R/W	This value is set after calibration to the external 32 MHz crystal oscillator based reference. Software and/or a debugger can overwrite it.
Function	**Bits**	**Default**	**Type**	**Description**
External 32 MHz Oscillator Trim Value Register: 0x4001_B028 Trim values used to center the external crystal oscillator analog circuit for selected crystal. The amplifier drive strength, ready counter delay taps, and internal shunt capacitors are programmable.				
Internal analog amplifier gain value	[11:10]	0x11	R/W	00 – Off 01 – Low gain mode = about 300 μA 10 – Mid gain mode = about 600 μA 11 – High gain mode = about 900 μA
Ready signal delay control – fine tune the delay before oscillator signals the MCU it's ready	[9:8]	0x00	R/W	00 – Bit 12 = 4096 32 MHz clocks = 64 μs 01 – Bit 13 = 8192 32 MHz clocks = 128 μs 10 – Bit 14 = 16384 32 MHz clocks = 256 μs 11 – Bit 15 = 32768 32 MHz clocks = 512 μs
Boost done signal delay control – fine tune the duration of the boost	[7:6]	0x00	R/W	00 – Bit 12 = 4096 32 MHz clocks = 64 μs 01 – Bit 13 = 8192 32 MHz clocks = 128 μs 10 – Bit 14 = 16384 32 MHz clocks = 256 μs 11 – Bit 15 = 32768 32 MHz clocks = 512 μs
Trim cap value	[5:0]	0x20	R/W	000000 = about 24.35 pF 000001 = about 24.65 pF (approx. 0.3 per step) … 111111 = about 43.55 pF
Function	**Bits**	**Default**	**Type**	**Description**
External 32.768 kHz Oscillator Trim Value Register: 0x4001_B02C Trim values used to center the external crystal oscillator analog circuit for selected crystal. The amplifier drive strength, ready and boost counter delays, and internal shunt capacitors are programmable.				
Internal analog amplifier gain value	[11:10]	0x11	R/W	00 – Off 01 – Low gain mode = about 350 nA 10 – Mid gain mode = about 460 nA 11 – High gain mode = about 640 nA
Ready signal delay control – fine tune the delay before oscillator signals the MCU it's ready	[9:8]	0x00	R/W	00 – Bit 12 = 4096 32.768 kHz clocks = 125 ms 01 – Bit 13 = 8192 32.768 kHz clocks = 250 ms 10 – Bit 14 = 16384 32.768 kHz clocks = 500 ms 11 – Bit 15 = 32768 32.768 kHz clocks = 1000 sec

Table 7. REGISTERS

Function	Bits	Default	Type	Description
Boost done signal delay control – fine tune the duration of the boost	[7:6]	0x00	R/W	00 – Bit 12 = 4096 32.768 kHz clocks = 125 ms 01 – Bit 13 = 8192 32.768 kHz clocks = 250 ms 10 – Bit 14 = 16384 32.768 kHz clocks = 500 ms 11 – Bit 15 = 32768 32.768 kHz clocks = 1000 sec
Internal programmable shunt capacitors	[5:0]	0x20	R/W	000000 = about 20.9 pF 000001 = about 21.2 pF (approx. 0.3 per step) … 111111 = about 40.1 pF

Internal Oscillator Calibration

To calibrate the internal 32MHz oscillator to the external crystal 32 MHz oscillator an internal state machine is used that utilizes a binary search algorithm and a phase detector to align the internal oscillator as close as possible to the external crystal 32 MHz oscillator reference.

Typically this calibration is done after exiting sleep mode or anytime a large temperature variation is encountered or expected.

A typical 32 MHz calibration should take at most 450 μs and would entail the following steps:

1. Enable the external 32 MHz crystal oscillator and wait for it to stabilize
   a. Set bit 0 of the clock control register
   b. Wait for bit 0 of the clock control register to be a 1
2. Enable the internal 32 MHz oscillator and wait for it to stabilize
   a. Clear bit 5 of the PMU control register
   b. Wait at least 5μs
3. Set the interrupt enable
   a. Set bit 1 of the clock interrupt enable register
4. Enable the calibration
   a. Set bit 3 of the clock control register
5. Wait for the interrupt or poll the status register
   a. Polling can be accomplished by waiting for bit 5 of the clock status register to clear
6. Determine success of the calibration
   a. Read bit 4 of the clock status register

A typical 32.768 kHz calibration should take at most 760 μs and would entail the following steps:

1. Enable the external 32 MHz crystal oscillator and wait for it to stabilize
   a. Set bit 0 of the clock control register
   b. Wait for bit 0 of the clock control register to be a 1
2. Enable the internal 32.768 kHz oscillator and wait for it to stabilize
   a. Clear bit 4 of the PMU control register
   b. Wait at least 5 μs
3. Set the interrupt enable
   a. Set bit 0 of the clock interrupt enable register
4. Enable the calibration
   a. Set bit 2 of the clock control register
5. Wait for the interrupt or poll the status register
   a. Polling can be accomplished by waiting for bit 3 of the clock status register to clear
6. Determine success of the calibration
   a. Read bit 2 of the clock status register

TIMERS

SysTick Timer

Description

The SysTick timer is a standard Cortex−M3 timer that is on all instances of Cortex−M3 MCUs. This standard SysTick timer increases software portability between hardware platforms.

The SysTick timer is integrated with the NVIC and can be used to generate a SYSTICK exception. The SYSTICK timer is often used to generate interrupts for an operating system to manage system tasks.

Refer to the official ARM documentation for the usage model of the SysTick timer.

In the NCS36510 implementation the SysTick timer is always clocked by the MCU clock, FCLK. In coma mode this SysTick timer is not available and the Real Time Clock (RTC) must be used instead.

Table 8. REGISTERS

Function	Bits	Default	Type	Description
SysTick Control and Status Register: 0xE000_E010 Control and status register for SysTick timer.				
Count flag – COUNTFLAG	[16]	0x0	RO	Reads as 1 if counter reaches 0 since last time this register is read, clears on read
Clock source – CLKSOURCE	[2]	0x0	R/W	0 – External reference clock, STCLK 1 – FCLK, processor free running clock NOTE: This implementation does not use STCLK
Tick interrupt – TICKINT	[1]	0x0	R/W	0 – Do not generate tick interrupt 1 – Generate tick interrupt
SysTick timer enable – ENABLE	[0]	0x0	R/W	0 – Disable SysTick timer 1 – Enable SysTick timer
SysTick Reload Value Register: 0xE000_E014 SysTick reload value register.				
Reload value – RELOAD	[23:0]	0x0	R/W	Reload value for when timer reaches down count of 0
Function	**Bits**	**Default**	**Type**	**Description**
SysTick Current Value Register: 0xE000_E018 SysTick current value register.				
Current timer value – CURRENT	[23:0]	0x0	R/W	When read current timer value is returned When written the counter value is set to 0 and also clears COUNTFLAG in SysTick control/status register
Function	**Bits**	**Default**	**Type**	**Description**
SysTick Calibration Value Register: 0xE000_E01C SysTick calibration value register. Note for this implementation bit 31 is set and the remaining bits are set to 0. This implies this is a read only register.				
No external reference clock – NOREF	[31]	0x1	RO	Bit is hard−wired to 1 in hardware which means no external SysTick reference clock (STCLK) is available. Clock is always free running MCU clock (FCLK).
Calibration value skew – SKEW	[30]	0x0	RO	Calibration value is accurate In this implementation this bit is set to 0 in hardware
Calibration value for 10ms	[23:0]	0x0	RO	Calibration value for 10ms In this implementation these bits are set to 0 in hardware. This implies the calibration value is not available.

Real Time Clock (RTC) Control

Description

The RTC consists of two counters that are clocked by the 32.768 kHz clock. Both counters have their own alarm, interrupt, and clear functions.

The first counter is a 15 bit sub−second counter (2^15 − 1/32768 = 1s). It can be used for wait times less than one second.

The other counter is a 32 bit seconds counter. The seconds counter is incremented by the sub − second counter rollover.

The second counter can count up to ~136 years so it can be used as a UNIX (POSIX or Epoch) time counter if desired.

When both second and sub − second counters are enabled the RTC will generate an interrupt when both counters expire, allowing for non − integer second timing (for example 3.6 s).

All RTC registers are on the 32.768 kHz domain. A write to any of these registers requires four 32.768 kHz clocks to allow for PCLK (peripheral clock) event synchronization handshake. Attempting to write to a register a second time during this write cycle is not allowed. This illegal write condition will set an error bit in the status register and the 2nd write will be ignored since the results would be unpredictable. If a second write is required very soon after

a first write, software may check the status of the write by reading the write busy bits of the status register. If the write busy bit for the desired register reads a 1, then a consecutive write must wait.

Table 9. REGISTERS

Function	Bits	Default	Type	Description
RTC Sub−second Counter Value Register: 0x4000_F000 RTC sub−second counter value. A write loads the counter, a read returns current value.				
Sub−second counter value	[14:0]	0x0	R/W	Write loads the counter, a read returns the current value
Function	**Bits**	**Default**	**Type**	**Description**
RTC Second Counter Value Register: 0x4000_F004 RTC second counter value. A write loads the counter, a read returns current value.				
Second counter value	[31:0]	0x0	R/W	Write loads the counter, a read returns the current value
Function	**Bits**	**Default**	**Type**	**Description**
RTC Sub−second Alarm Register: 0x4000_F008 RTC sub−second alarm value register. A write loads the alarm value, a read returns current value.				
Sub−second alarm value	[14:0]	0x7FFF	R/W	Write loads the alarm value, a read returns the current value
Function	**Bits**	**Default**	**Type**	**Description**
RTC Second Alarm Register: 0x4000_F00C RTC second alarm value register. A write loads the alarm value, a read returns current value.				
Second alarm value	[31:0]	0xFFFFFFFF	R/W	Write loads the alarm value, a read returns the current value
Function	**Bits**	**Default**	**Type**	**Description**
RTC Control Register: 0x4000_F0010 RTC control register, enables/disables second and sub−second counters as well as their corresponding interrupts.				
Second interrupt enable	[3]	0x0	R/W	1 − Interrupt enabled 0 − Interrupt disabled
Sub−second interrupt enable	[2]	0x0	R/W	1 − Interrupt enabled 0 − Interrupt disabled
Second counter enable	[1]	0x0	R/W	1 − Counter enabled 0 − Counter disabled
Sub−second counter enable	[0]	0x0	R/W	1 − Counter enabled 0 − Counter disabled
Function	**Bits**	**Default**	**Type**	**Description**
RTC Status Register: 0x4000_F014 RTC status register, clears on any write.				
Busy with second counter interrupt clear write	[10]	0x0	R/W	1 − Busy 0 − Not busy
Busy with a sub−second interrupt clear write	[9]	0x0	R/W	1 − Busy 0 − Not busy
Busy with a control register write	[8]	0x0	R/W	1 − Busy 0 − Not busy
Busy with a second alarm register write	[7]	0x0	R/W	1 − Busy 0 − Not busy
Busy with a sub−second alarm register write	[6]	0x0	R/W	1 − Busy 0 − Not busy

Table 9. REGISTERS

Function	Bits	Default	Type	Description
Busy with a second counter register write	[5]	0x0	R/W	1 – Busy 0 – Not busy
Busy with a sub−second counter register write	[4]	0x0	R/W	1 – Busy 0 – Not busy
Busy with any write	[3]	0x0	R/W	1 – Busy 0 – Not busy
Reads error bit which is set when a write occurs before a previous write to the same register has completed	[2]	0x0	R/W	1 – Error 0 – No error
Second interrupt status	[1]	0x0	R/W	1 – Interrupt active 0 – No interrupt
Sub−second interrupt status	[0]	0x0	R/W	1 – Interrupt active 0 – No interrupt
Function	**Bits**	**Default**	**Type**	**Description**
RTC Clear Register: 0x4000_F018 Write a 1 to clear corresponding interrupt source.				
Second interrupt clear	[1]	n/a	WO	1 – Clear interrupt 0 – No effect
Sub−second interrupt clear	[0]	n/a	WO	1 – Clear interrupt 0 – No effect

Watchdog Timer (WDT)

Description

NCS36510 implements a programmable watchdog timer. The watchdog timer is disabled by default and the application software needs to instantiate the watchdog timer driver and enable it. The WDT has a register locking safety mechanism to prevent errant software from corrupting the registers. While locked the only supported operation is a clear. The watchdog is on the 32.768 kHz clock domain so it has a minimum resolution of 30.5 μs. It is 18 bits wide giving it a maximum timeout time of 8 seconds. When the WDT overflows, the system is reset and the reset sources register is updated to indicate the system was reset by the watchdog timer. If a debugger is attached then the WDT is paused.

All WDT registers are on the 32.768 kHz domain. A write to any of these registers requires four 32.768 kHz clocks to allow for PCLK (peripheral clock) event synchronization handshake. Attempting to write to a register a second time during this write cycle is not allowed. This illegal write condition will set an error bit in the status register and the 2nd write will be ignored since the results would be unpredictable. If a second write is required very soon after a first write, software may check the status of the write by reading the write busy bits of the status register. If the write busy bit for the desired register reads a 1, then a consecutive write must wait.

The WDT is not active in coma mode as the MCU is completely powered down. Use the RTC for coma mode.

Table 10. REGISTERS

Function	Bits	Default	Type	Description
WDT Load Value Register: 0x4000_A000 Watchdog timer load value register.				
Value that the WDT timer decrements from	[17:0]	0x3FFFF	R/W	After this register is written the WDT immediately restarts counting down from this value. The minimum value is 1.
Function	**Bits**	**Default**	**Type**	**Description**
WDT Current Value Register: 0x4000_A004 Watchdog timer current value register.				
Current down count value	[17:0]	0x3FFFF	R/W	Current WDT down count value

Table 10. REGISTERS

Function	Bits	Default	Type	Description
WDT Control Register: 0x4000_A008 Watchdog timer control register.				
Watchdog enable	[0]	0x0	R/W	0 – WDT disabled 1 – WDT enabled
WDT Kick Register: 0x4000_A0C0 Watchdog timer kick register. When written the watchdog timer starts over again at the load value.				
Watchdog kick	[31:0]	n/a	WO	A write of any value to this address will immediately restart the WDT down count from the load value
Function	Bits	Default	Type	**Description**
WDT Lock Register: 0x4000_A100 When the lock feature is activated, the other watchdog timer registers are read only. Write the unlock code to regain access to WDT registers. Reading this register only returns the LSB to show the lock status. To unlock, write the code 0x1ACCE551. This locking mechanism is meant to provide a safeguard against rogue software accessing the WDT.				
WDT register lock control	[0]	n/a	WO	0 – WDT registers are unlocked 1 – WDT registers are locked (see note above for unlock procedure)
Function	Bits	Default	Type	**Description**
WDT Status Register: 0x4000_A140 WDT status register.				
Register access error	[3]	0x0	RO	1 – Error, next register write occurred before previous write completed 0 – No error
Control register write busy	[2]	0x0	RO	1 – Busy 0 – Not busy
Load register write busy	[1]	0x0	RO	1 – Busy 0 – Not busy
Any register write busy	[0]	0x0	RO	1 – Busy 0 – Not busy

16−bit General Purpose Timers

Description

NCS36510 has three independent 16 bit down count timers (TIMER 0, 1, and 2). Each of the three independent timers can:

- Be clocked at either the system clock rate, or a choice of 8 prescale values; 0, 2, 8, 16, 32, 128, 256, and 1024
- Be loaded with a value from a preload register
- Generate an interrupt on 0 counts
- Be operated in free − running or periodic modes. In periodic mode the interrupt is generated one clock later than the pre − load value since 0 is included in the count.

Table 11. REGISTERS

Function	Bits	Default	Type	Description
Timer Load Value Register (TIMER 0, 1, 2): 0x4000_0000, 0x4000_1000, 0x4000_2000 Load value register for timer.				
Timer load value	[15:0]	0x0	R/W	Value used for pre−loading the timer's counter. Value gets reloaded at each cycle in periodic mode.
Function	Bits	Default	Type	**Description**
Timer Value Register (TIMER 0, 1, 2): 0x4000_0004, 0x4000_1004, 0x4000_2004 Current timer value register.				
Current timer value	[15:0]	0x0	R/W	Read to return current timer value

Table 11. REGISTERS

Function	Bits	Default	Type	Description
Timer Control Register (TIMER 0, 1, 2): 0x4000_0008, 0x4000_1008, 0x4000_2008 Timer control register, used to enable the timer, enable the timer interrupts, and to manage the pre−scale divider for the clock. Unused bits always read back 0.				
Interrupt status	[8]	0x0	RO	Returns interrupt active status 1 − Interrupt active 0 − No interrupt
Timer enable	[7]	0x0	R/W	1 − Enable timer 0 − Disable timer
Timer periodic mode enable	[6]	0x0	R/W	1 − Enable periodic mode 0 − Disable periodic mode
Timer clock prescale value	[4:2]	0x0	R/W	0 − No FCLK divider 1 − Divide by 16 2 − Divide by 256 3 − Divide by 2 4 − Divide by 8 5 − Divide by 32 6 − Divide by 128 7 − Divide by 1024
Function	**Bits**	**Default**	**Type**	**Description**
Timer Clear Register (TIMER 0, 1, 2): 0x4000_000C, 0x4000_100C, 0x4000_200C Timer interrupt clear registers, write−only.				
Clear interrupt	[15:0]	0x0	WO	Write any value to this register to clear the interrupt

DIGITAL INPUT/OUTPUT (DIO) CONTROL

Description

NCS36510 has 18 identical GPIO. The following list documents the programmable options available independently for each GPIO. Each GPIO input is compatible with CMOS levels and has hysteresis.

Options:

- Bi − directional capability
- Individually configurable interrupt lines
- Rising, falling, or both edge interrupt
- High, low, or both logic level interrupt
- Loopback mode
- Push pull or open drain
- Four programmable drive strengths
- Pullup, pulldown or neither

In the following registers, the DIO pins are aligned with their bit positions in the register. For example DIO11 corresponds to bit 11 in the register.

The DIO control and the crossbar must be configured to make sure that the DIO signals actually connect to the right DIO pads. See the crossbar section for more details.

Table 12. REGISTERS

Function	Bits	Default	Type	Description
DIO State/Set Register: 0x4000_C000 Read this register to determine the current state of the synchronized DIO input signals. Alternatively, write a 1 to desired DIO pins to drive the DIO signal high (set).				
Read synchronized DIO input or drive a DIO output high	[17:0]	n/a	WO	When read, the value returned is the current synchronized state for each DIO input. Each input must be enabled in the input direction register. When a 1 is written to any bit position, the corresponding DIO will output a 1 to the DIO pad (set). Each output must be enabled in the output direction register.
Function	**Bits**	**Default**	**Type**	**Description**
DIO Clear/Interrupts Register: 0x4000_C004 Read this register to determine the current state of the DIO interrupts. Alternatively, write a 1 to desired DIO pins to drive the DIO signal low (clear).				

Table 12. REGISTERS

Function	Bits	Default	Type	Description
Read DIO input interrupts status or drive a DIO output low	[17:0]	n/a	WO	When read, the value returned is the current interrupt state for each DIO input. Each input must be enabled in the input direction register. When a 1 is written to any bit position, the corresponding DIO will output a 0 to the DIO pad (clear). Each output must be enabled in the output direction register.
DIO Output Enable Set Register: 0x4000_C008 Each DIO that is selected will have the direction of output. Or in other words this is an output enable set function.				
DIO output enable set	[17:0]	0x7FFFF	R/W	1 – Enable selected DIO as an output 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Output Enable Clear Register: 0x4000_C00C Each DIO that is selected will have the direction of input. Or in other words this is an output enable clear function.				
DIO output enable clear – make input	[17:0]	0x7FFFF	R/W	1 – Disable selected DIO as an output (make an input) 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Interrupt Enable Set Register: 0x4000_C010 Each DIO that is selected will have its interrupt enabled. Or in other words this is an interrupt enable set function.				
DIO interrupt enable set	[17:0]	0x0	R/W	1 – Enable selected DIO interrupt 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Interrupt Enable Clear Register: 0x4000_C014 Each DIO that is selected will have its interrupt disabled. Or in other words this is an interrupt enable clear function.				
DIO interrupt enable clear	[17:0]	0x0	R/W	1 – Disable selected DIO Interrupt 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Set Edge Sensitive Interrupts Register: 0x4000_C018 Each DIO that is selected will have its interrupt be set to edge sensitive. Or in other words this is an edge sensitive interrupt enable set function.				
DIO edge sensitive interrupt set	[17:0]	0x0	R/W	1 – Enable edge sensitive interrupt 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Set Level Sensitive Interrupts Register: 0x4000_C01C Each DIO that is selected will have its interrupt be set to level sensitive. Or in other words this is a level sensitive interrupt enable set function.				
DIO level sensitive interrupt set	[17:0]	0x0	R/W	1 – Enable level sensitive interrupt 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Set Interrupt Level/Direction Register: 0x4000_C020 Each DIO that is selected will have its interrupt be set to be either rising edge triggered or logic level high triggered, depending on which was previously selected.				
DIO interrupt level/direction set	[17:0]	0x0	R/W	1 – Rising edge or logic high triggered interrupt 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Clear Interrupt Level/Direction Register: 0x4000_C024 Each DIO that is selected will have its interrupt be set to be either falling edge triggered or logic level low triggered, depending on which was previously selected.				

Table 12. REGISTERS

Function	Bits	Default	Type	Description
DIO interrupt level/direction clear	[17:0]	0x0	R/W	1 – Falling edge or logic low triggered interrupt 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Set Any Edge Interrupt Override Register: 0x4000_C028 Each DIO that is selected will have its interrupt be set to be either rising or falling edge triggered, regardless of what settings were previously selected.				
DIO interrupt any direction override set	[17:0]	0x0	R/W	1 – Rising or falling edge triggered interrupt 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Clear Any Edge Interrupt Override Register: 0x4000_C02C Each DIO that is selected will have its override interrupt cleared. This is the clear register for the DIO set any edge interrupt override register above.				
DIO interrupt any direction override clear	[17:0]	0x0	R/W	1 – Clear override interrupt, return to settings in non–override registers 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Clear Interrupt Register: 0x4000_C030 Each DIO that is selected will have its corresponding interrupt cleared.				
DIO interrupt clear	[17:0]	0x0	R/W	1 – Clear interrupt 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO Loopback Register: 0x4000_C034 Each DIO that is selected will have its output signal routed back to its input signal.				
DIO loopback enable	[17:0]	0x0	R/W	1 – Mirror output signal back in as an input 0 – No effect
Function	**Bits**	**Default**	**Type**	**Description**
DIO[0] Control Register: 0x4001_C000 **DIO[1] Control Register: 0x4001_C004** **DIO[2] Control Register: 0x4001_C008** **DIO[3] Control Register: 0x4001_C00C** **DIO[4] Control Register: 0x4001_C010** **DIO[5] Control Register: 0x4001_C014** **DIO[6] Control Register: 0x4001_C018** **DIO[7] Control Register: 0x4001_C01C** **DIO[8] Control Register: 0x4001_C020** **DIO[9] Control Register: 0x4001_C024** **DIO[10] Control Register: 0x4001_C028** **DIO[11] Control Register: 0x4001_C02C** **DIO[12] Control Register: 0x4001_C030** **DIO[13] Control Register: 0x4001_C034** **DIO[14] Control Register: 0x4001_C038** **DIO[15] Control Register: 0x4001_C03C** **DIO[16] Control Register: 0x4001_C040** **DIO[17] Control Register: 0x4001_C044** Each DIO has its own control register and can be used to manage the output type, drive strength, and pull. Please note when the DBG_TEST_EN pin is high, the DIO[11:13] pins get automatically reconfigured for SWD debugging. The output drivers on NCS36510 are smaller than general MCUs, so the programmer may desire putting the drive strength to maximum to better compatibility with external devices if the default drive strength is not sufficient. Drive strength capabilities are reduced at 1V VDDIO operation. Refer to NCS36510 datasheet.				

Table 12. REGISTERS

Function	Bits	Default	Type	Description
Output driver type	[5]	0x0	R/W	0 – Push/Pull 1 – Open drain pull down only (NMOS transistor)
Output drive strength, typical at 3V (refer to datasheet)	[4:2]	0x1	R/W	000 – 1.4 mA 001 – 2.7 mA 010 – 5.3 mA 011 – 10.4 mA
Pull enable and direction (approx. 10kΩ res, see product datasheet)	[1:0]	0x0	R/W	00 – Pull down active 01 – No pull 10 – No pull 11 – Pull up active

CROSSBAR CONTROL

Description

To connect the UARTs, SPIs, I2Cs, DIOs, and PWM to the external DIO pins the crossbar must be utilized. Not every peripheral can reach every DIO pad so the system designer must make sure the physical PCB layout and the desired software configuration of the peripherals through the crossbar is compatible with system requirements.

If the DBG_TEST_EN pin is high, the crossbar settings are superseded by the debugger logic for DIO[13], DIO[12] and DIO[11].

The following table summarizes what peripherals can be connected to what DIO pins.

		DIO																	
		17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UART 1	Transmit Data																		X
	Receive Data																	X	
	Clear To Send																X		
	Request To Send															X			
	Data Terminal Ready														X				
	Data Set Ready													X					
	Data Carrier Detect												X						
	Ring Indicator											X							
UART 2	Transmit Data										X								
	Receive Data									X									
	Clear To Send								X										
	Request To Send							X											
I2C 1	SCLK					X		X						X			X		
	SDATA						X		X						X	X			
I2C 2	SCLK	X			X														
	SDATA		X	X															

SPI 1	SCLK										X				X			
	SDATAO									X				X				
	SDATAI								X				X					
	SSNI							X				X						
	SSNO<0>						X				X							X
	SSNO<1>					X				X							X	
	SSNO<2>								X							X		
	SSNO<3>							X										
SPI 2	SCLK				X													
	SDATAO			X														
	SDATAI		X															
	SSNI	x																
	SSNO<0>	x																
PWM	Output					X	X			X		X	X					
D-I-O	17	x																
	16		X															
	15			X														
	14				X													
	13					X												
	12						X											
	11							X										
	10								X									
	9									X								
	8										X							
	7											X						
	6												X					
	5													X				
	4														X			
	3															X		
	2																X	
	1																	X
	0																	X

Table 13. REGISTERS

Function	Bits	Default	Type	Description
Crossbar Control Register for DIO[0]: 0x4001_0000 Crossbar control register.				
DIO[1]	[2:0]	0x0	R/W	111 – UART 1 Tx data 110 – SPI 1 SSNO[0] 000 – DIO[0]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[1]: 0x4001_0004 Crossbar control register.				
DIO[1]	[2:0]	0x0	R/W	111 – UART 1 Rx data 110 – SPI 1 SSNO[1] 000 – DIO[1]

Table 13. REGISTERS

Function	Bits	Default	Type	Description
Crossbar Control Register for DIO[2]: 0x4001_0008 Crossbar control register.				
DIO[2]	[2:0]	0x0	R/W	111 – UART 1 clear to send 110 – SPI 1 SSNO[2] 101 – I2C 1 I2C SCLK 000 – DIO[2]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[3]: 0x4001_000C Crossbar control register.				
DIO[3]	[2:0]	0x0	R/W	111 – UART 1 request to send 110 – SPI 1 SSNO[3] 101 – I2C 1 I2C SDATA 000 – DIO[3]
Function	Bits	Default	Type	Description
Crossbar Control Register for DIO[3]: 0x4001_0010 Crossbar control register.				
DIO[4]	[2:0]	0x0	R/W	111 – UART 1 Data terminal ready 110 – SPI 1 SCLK 101 – I2C 1 I2C SDATA 000 – DIO[4]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[5]: 0x4001_0014 Crossbar control register.				
DIO[5]	[2:0]	0x0	R/W	111 – UART 1 data set ready 110 – SPI 1 SDATA0 101 – I2C 1 I2C SCLK 000 – DIO[5]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[6]: 0x4001_0018 Crossbar control register.				
DIO[6]	[2:0]	0x0	R/W	111 – UART 1 data carrier detect 110 – SPI 1 SDATAI 100 – PWM output 000 – DIO[6]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[7]: 0x4001_001C Crossbar control register.				
DIO[7]	[2:0]	0x0	R/W	111 – UART 1 ring indicator 110 – SPI 1 SSNI 100 – PWM output 000 – DIO[7]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[8]: 0x4001_0020 Crossbar control register.				
DIO[8]	[2:0]	0x0	R/W	111 – UART 2 Tx data 110 – SPI 1 SCLK 011 – SPI 1 SSN[0] 000 – DIO[8]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[9]: 0x4001_0024 Crossbar control register.				

Table 13. REGISTERS

Function	Bits	Default	Type	Description
DIO[9]	[2:0]	0x0	R/W	111 – UART 2 Rx data 110 – SPI 1 SDATAO 100 – PWM output 011 – SPI 1 SSN[1] 000 – DIO[9]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[10]: 0x4001_0028 Crossbar control register.				
DIO[10]	[2:0]	0x0	R/W	111 – UART 2 clear to send 110 – SPI 1 SDATAI 101 – I2C 1 SDATA 011 – SPI 1 SSN[2] 000 – DIO[10]
Crossbar Control Register for DIO[11]: 0x4001_002C Crossbar control register.				
DIO[11]	[2:0]	0x0	R/W	111 – UART 2 request to send 110 – SPI 1 SSNI 101 – I2C 1 SCLK 011 – SPI 1 SSN[3] 000 – DIO[11]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[12]: 0x4001_0030 Crossbar control register.				
DIO[12]	[2:0]	0x0	R/W	110 – SPI 1 SSNO[0] 101 – I2C 1 SDATA 100 – PWM output 000 – DIO[12]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[13]: 0x4001_0034 Crossbar control register.				
DIO[13]	[2:0]	0x0	R/W	110 – SPI 1 SSNO[1] 101 – I2C 1 SCLK 100 – PWM output 000 – DIO[13]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[14]: 0x4001_0038 Crossbar control register.				
DIO[14]	[2:0]	0x0	R/W	110 – SPI 2 SCLK 101 – I2C 2 SCLK 000 – DIO[14]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[15]: 0x4001_003C Crossbar control register.				
DIO[15]	[2:0]	0x0	R/W	110 – SPI 2 SDATAO 101 – I2C 2 SDATA 000 – DIO[15]
Function	**Bits**	**Default**	**Type**	**Description**
Crossbar Control Register for DIO[16]: 0x4001_0040 Crossbar control register.				
DIO[16]	[2:0]	0x0	R/W	110 – SPI 2 SDATAI 101 – I2C 2 SDATA 000 – DIO[16]
Function	**Bits**	**Default**	**Type**	**Description**

Table 13. REGISTERS

Function	Bits	Default	Type	Description
Crossbar Control Register for DIO[17]: 0x4001_0040 Crossbar control register.				
DIO[17]	[2:0]	0x0	R/W	110 – SPI 2 SSNO[0] or SSNI 101 – I2C 2 SCLK 000 – DIO[17]

DIRECT MEMORY ACCESS (DMA) CONTROLLER

Description

The DMA controller is a single channel direct memory access controller. It can be used to directly transfer data from one memory to another.

After initial setup, the DMA controller initiates data transfer from the user specified source address to the internal FIFO. It then performs another transfer to move the data from the FIFO to the user specified destination address. The FIFO size is 32 words by 32 bits.

The DMA controller transfers data on word, 1/2 word, and byte aligned boundaries. The DMA controller supports beat modes of 4, 8, and 16 using incrementing bursts. Unspecified length bursts and single transfers are also supported. Wrapping bursts are not supported.

The bursts are automatically controlled and are based on the number of specified bytes to transfer. Interrupts are generated upon transfer completion and/or error conditions.

General programming sequence:

- Disable DMA in DMA control register
- Write source address, destination address, and transfer size to corresponding registers
- Enable the interrupt enable register
- Enable DMA in DMA control register
- Wait for interrupt or poll status register for done flag

Table 14. REGISTERS

Function	Bits	Default	Type	Description
DMA Control Register: 0x2400_0400 DMA control register.				
DMA mode	[2:1]	0x0	R/W	00 – Memory to memory
DMA enable	[0]	0x0	R/W	1 – Enabled 0 – Disabled
Function	**Bits**	**Default**	**Type**	**Description**
DMA Source Address Register: 0x2400_0404 DMA source address register.				
Source address to read data from	[31:0]	0x0	R/W	A write configures the source address where the DMA controller will read from. A read returns the written data. The source address can be on a word, 1/2 word or byte boundary, but is required to be specified as a 32 bit address.
Function	**Bits**	**Default**	**Type**	**Description**
DMA Destination Address Register: 0x2400_0408 DMA destination address register.				
Destination address to write data to	[31:0]	0x0	R/W	A write configures the destination address where the DMA controller will write to. A read returns the written data. The source address can be on a word, 1/2 word or byte boundary, but is required to be specified as a 32 bit address.
Function	**Bits**	**Default**	**Type**	**Description**
DMA Transfer Size Register: 0x2400_040C DMA transfer size register.				
Number of bytes to transfer	[31:0]	0x0	R/W	A write configures the size of the DMA data to be transferred in bytes. A read returns the written data. Transfer size does not have to be on a word aligned boundary. Maximum transfer size is 64 kB.
DMA Status Register: 0x2400_0410 DMA status register.				

Table 14. REGISTERS

Function	Bits	Default	Type	Description
Destination error	[2]	0x0	RO	0 – No error 1 – Error
Source error	[1]	0x0	RO	0 – No error 1 – Error
Transfer complete	[0]	0x0	RO	0 – Not complete 1 – Complete
Function	**Bits**	**Default**	**Type**	**Description**
DMA Interrupt Enable Register: 0x2400_0414 DMA interrupt enable register.				
Enable destination error interrupt	[2]	0x0	R/W	1 – Enable interrupt 0 – Disable interrupt
Enable source error interrupt	[1]	0x0	R/W	1 – Enable interrupt 0 – Disable interrupt
DMA transfer complete interrupt enable	[0]	0x0	R/W	1 – Enable interrupt 0 – Disable interrupt
Function	**Bits**	**Default**	**Type**	**Description**
DMA Interrupt Status Register: 0x2400_0418 DMA interrupt status register. Clear on read.				
Destination error interrupt	[2]	0x0	RO	1 – Interrupt active 0 – No interrupt
Source error interrupt	[1]	0x0	RO	1 – Interrupt active 0 – No interrupt
Transfer complete interrupt	[0]	0x0	RO	1 – Interrupt active 0 – No interrupt (or busy)

10 BIT SUCCESSIVE APPROXIMATION (SAR) ANALOG TO DIGITAL CONVERTER (ADC)

Description

The NCS36510 has a fully integrated 10 bit SAR ADC.

The ADC is single ended to reduce power consumption. A six input multiplexer allows up to four external signals to be measured. The other two inputs are for internal temperature and battery voltage (V3V) sensors.

WARNING: Warning: The user has the responsibility to respect the absolute maximum ratings. The voltage on the ADC pins cannot exceed V3V + 0.3 V, regardless of the input scaling. Several modes of ADC operation are available including absolute, ratio based, and pseudo − differential.

All ADC conversions are referenced to an internal fixed reference of nominally 950 mV. Absolute conversions represent the input signal compared to this fixed reference.

A programmable scale factor can be used to enable a resistive voltage divider at the A[x] pins to divide the incoming voltage. Please note that the only high impedance setting is the scale factor of 1.0. The following table documents the possible scale factor settings and their maximum input voltages.

The SAR ADC completes a conversion as fast as 5 μs with a 4 MHz sample clock in 20 clock cycles. To support a wide range of input voltages, there is a programmable resistive voltage divider on the external inputs. The table below shows the ideal settings.

Table 15. MINIMUM ROW TIME AND BLANKING NUMBERS

Scale Factor	Maximum Input Voltage	Input Resistance
1.00	0.950 V	High Impedance
0.69	1.3 V	80 kΩ
0.53	1.7 V	52 kΩ
0.43	2.1 V	43 kΩ
0.36	2.5 V	38 kΩ
0.31	2.9 V	36 kΩ
0.28	3.3 V	34 kΩ
0.24	3.6 V	32 kΩ

For applications that require measuring an external signal versus an external reference, a ratio based conversion mode is supported. Two ADC conversions are done back to back and automatically divided to get a ratio. Pseudo − differential mode is similar to ratio based mode, except the final computation is the difference of the two signals instead of the ratio.

Ratio based pseudo − differential mode consists of three conversions. The first two are taken as a differential signal, the third is considered as a reference. The resulting differential voltage is automatically divided by the third conversion resulting in a pseudo − differential ratio.

This SAR ADC, like all ADCs, has a finite input time constant. The mux has a finite resistance, and the input of the ADC has a finite capacitance. The input voltage must be fully settled to get an accurate conversion. The input time constant also depends on the scale factors programmed. The following table contains typical values for time constants as a function of the input scale factor. For maximum accuracy, at least 7 time constants of settling time are recommended. The following table has recommended settling times for various scale factors.

Table 16. MINIMUM ROW TIME AND BLANKING NUMBERS

Scale Factor	Time constant, τ	7 τ
1.00	160 ns	1.12 μs
0.69	110 ns	770 ns
0.53	85 ns	595 ns
0.43	68 ns	476 ns
0.36	58 ns	406 ns
0.31	50 ns	350 ns
0.28	44 ns	308 ns
0.24	39 ns	273 ns

The ADC can be used in either single or continuous conversion modes.

There is a status bit that can be polled to determine if the ADC is busy. An interrupt can also be configured to avoid consuming processor cycles polling the status bit.

Averaging can improve the accuracy of the SAR ADC conversions as compared to single shot measurements.

Table 17. RECOMMENDED GAIN SETTINGS

Function	Bits	Default	Type	Description
SAR ADC Control Register: 0x4001_5000 SAR ADC control register.				
Reference channel select	[14:12]	0x0	R/W	000 – A[0] 001 – A[1] 010 – A[2] 011 – A[3] 110 – Internal temperature sensor 111 – Internal supply voltage sensor

Table 17. RECOMMENDED GAIN SETTINGS

Function	Bits	Default	Type	Description
Measurement channel select	[10:8]	0x0	R/W	000 – A[0] 001 – A[1] 010 – A[2] 011 – A[3] 110 – Internal temperature sensor 111 – Internal supply voltage sensor
Measurement type select	[3]	0x0	R/W	0 – Relative 1 – Absolute
Measurement abort	[2]	0x0	R/W	0 – No effect 1 – Aborts the continuous conversion
Measurement start	[1]	0x0	R/W	0 – No effect 1 – Start a conversion
Measurement mode	[0]	0x0	R/W	0 – Single sample 1 – Continuous sampling
Function	**Bits**	**Default**	**Type**	**Description**
SAR ADC Delay Settings Register: 0x4001_5004 SAR ADC delay settings register. Allows fine tuning of warm up periods. Typically this is left at the default settings.				
Sample delay	[31:24]	0x1A	R/W	Number of ADC clock cycles that the controller dwells in the measurement state. The minimum value of this delay is decimal 20.
Warm up delay	[23:16]	0x05	R/W	Number of ADC clock cycles the controller dwells in the warm up state (before conversion)
Sample period	[15:0]	0x0034	R/W	The number of ADC clock cycles between measurements when in continuous mode. Set this value to 0 for maximum throughput. Values greater than 0 but less than 2 plus the sum of the sample delay plus the warm−up delay are not allowed in absolute mode. Values greater than 0 but less than 4 plus the sum of 2 times the sample delay plus 2 times the warm−up delay are not allowed in relative mode.
Function	**Bits**	**Default**	**Type**	**Description**
SAR ADC Result Register: 0x4001_5008 SAR ADC measurement result register.				
SAR ADC measurement result	[9:0]	0x0	RO	Measurement result from ADC conversion. This value is updated 42 FCLK cycles after the end of the programmed sample delay.
Function	**Bits**	**Default**	**Type**	**Description**
SAR ADC Interrupt Register: 0x4001_500C SAR ADC interrupt configuration register.				
SAR ADC interrupt control	[0]	0x0	R/W	0 – Interrupt disabled 1 – Interrupt enabled
Function	**Bits**	**Default**	**Type**	**Description**
SAR ADC Clock Prescaler Register: 0x4001_5010 SAR ADC clock divider prescaler register.				
SAR ADC clock control	[8]	0x0	R/W	0 – ADC clock disabled 1 – ADC clock enabled
SAR ADC clock divider prescaler	[7:0]	0x07	R/W	ADC clock = FCLK / (ADC clock prescaler + 1) Minimum recommended value is 0x07

Table 17. RECOMMENDED GAIN SETTINGS

Function	Bits	Default	Type	Description
SAR ADC Clock Prescaler Register: 0x4001_5014 SAR ADC clock divider prescaler register.				
SAR ADC conversion status	[0]	0x0	RO	When the Interrupt Enable bit in the Interrupt Register is asserted, this output is asserted on the same clock cycle as valid data is presented to the Data Register. It remains high until it is cleared by clearing the Interrupt Enable bit or by a reset.

PWM

Description

The PWM peripheral generates a programmable duty cycle digital output signal. The duty cycle can be programmed from 0% to 100% with 8 bit resolution.

The PWM frequency can be either the system clock divided by 256 or the system clock divided by 4096.

The system clock is typically 32 MHz giving the lower PWM frequency of 7.8125 kHz or the upper frequency of 125 kHz.

The crossbar can be used to connect the PWM to a DIO.

Table 18. REGISTERS

Function	Bits	Default	Type	Description
PWM Duty Cycle Register: 0x4001_B000 Value in this register corresponds to the desired duty cycle to be output by the PWM hardware.				
PWM duty cycle value	[7:0]	0x0	WO	Writes to this location configure the duty cycle for the PWM output. 0x00 – Always off … 0x80 – 50% duty cycle … 0xFF – Always on Reads from this location return the duty cycle value.
Function	**Bits**	**Default**	**Type**	**Description**
PWM Enable Register: 0x4001_B004 Write any non−zero value to enable the PWM.				
PWM enable	[31:0]	0x0	WO	Write anything non−zero to enable
Function	**Bits**	**Default**	**Type**	**Description**
PWM Disable Register: 0x4001_B008 Write any non−zero value to disable the PWM.				
PWM enable	[31:0]	0x0	WO	Write anything non−zero to enable
Function	**Bits**	**Default**	**Type**	**Description**
PWM Prescale Enable Register: 0x4001_B00C Write any non−zero value to enable the PWM prescaler.				
PWM prescale enable	[31:0]	0x0	WO	Prescaler is PCLK/4096
Function	**Bits**	**Default**	**Type**	**Description**
PWM Prescale Disable Register: 0x4001_B010 Write any non−zero value to disable the PWM prescaler.				
PWM prescale disable	[31:0]	0x0	WO	Prescaler is PCLK/256
Function	**Bits**	**Default**	**Type**	**Description**

Table 18. REGISTERS

Function	Bits	Default	Type	Description
PWM Read Status and Control Register: 0x4001_B010 Read PWM current status and control settings.				
Current state of PWM output	[10]	0x0	RO	
Current state of PWM enable	[9]	0x0	RO	1 – PWM enabled 0 – PWM disabled
Current state of PWM prescaler	[8]	0x0	RO	1 – PWM prescaler enabled 0 – PWM prescaler disabled
Current duty cycle value	[7:0]	0x0	RO	0x00 – Always off … 0x80 – 50% duty cycle … 0xFF – Always on

IEEE 802.15.4 MAC

Description

The NCS36510 MAC hardware peripheral is designed to automate the low−level transmit and receive functions needed to support the IEEE 802.15.4−2006 Low−Rate Wireless Personal Area Network (LR−WPAN) standard. The system is designed so that time−critical sequences, such as frame timing, are performed by the MAC hardware peripheral, leaving only event−driven actions, such as packet origination, to be performed in software.

In transmit mode the peripheral constructs the packet to be transmitted, including the PHY synchronization and frame headers, the MAC frame header, and the CRC footer, and performs portions of the IEEE 802.15.4−2006 CSMA−CA algorithm. In addition, it can require the reception of a correct ACK before reporting a successful transmission.

In receive mode, the peripheral can be configured to test the correctness of the CRC value in a received frame, correlate the received address against an address stored in memory, and construct and transmit an acknowledgement, all without the services of the host MCU. More specifically, an outgoing ACK will be generated if the frame just received requested an ACK, and an incoming ACK must be received if the frame just transmitted requested an ACK. In the latter case, the received ACK will be validated (meaning that the received sequence number must match the transmitted sequence number). A transmitted ACK will contain the sequence number of the just−received frame.

In normal operation, only frames whose addresses match the device address are received. The entire frame, from the frame length field to the last byte of MAC payload, is written to the RX packet buffer register, for use by the MCU. (Only the CRC is not written.) During a frame transmission, the MAC hardware uses data from a TX packet buffer register to build a packet. The frame's CRC will be generated as the data is transmitted and the completed CRC will be sent at the end of frame.

Operation of the MAC peripheral is designed around a "command – interrupt" model, in which the MCU submits a sequence command to the peripheral by writing to the MAC sequencer register and, upon completion of the command, an IRQ is generated and sent to the interrupt controller. At that time, the sequence status can be read from the 0x4001_4010 MAC Status Register. There are maskable interrupts for the start and completion of events, the start of frame reception, valid data received, frame reception failure, PAN ID conflicts, and the matching of frame source addresses against a lookup table. While not necessary, it is possible for the MCU to be halted with its clocks gated off during operation of the MAC peripheral.

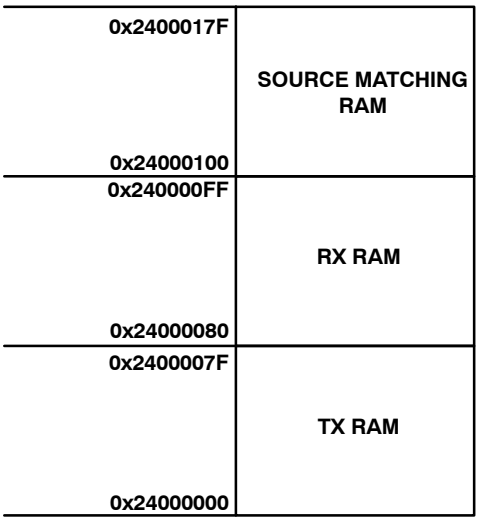


Figure 4.

Memory Map

The following figure depicts the memory map of the MAC peripheral.

The MAC hardware has a single memory buffer, divided into partitions for transmit, receive and source address matching. The RAM consists of three 128−byte regions: one for a TX buffer containing data to be transmitted, one for an RX buffer for storing incoming frame data and payload, and

one to contain a lookup table (LUT) of source addresses against which the MAC will compare the source address received from the incoming frame.

The memory from address 0 (0 hex) to 127 (7f hex) is used to store the data to be transmitted. The MAC hardware will start transmitting with the byte at address 0 and increment the address for subsequent bytes. Each byte is sent MSB first. The MAC RAM appears on the AHB bus and can be accessed by byte, half word, or word. While any data can be programmed in the MAC RAM to be transmitted, it is expected that a valid IEEE 802.15.4−2006 frame is written to this area. The value programmed in the *txlength* field of the 0x4001_408C MAC Tx Length register determines how much of the data will be transmitted and also determines when the hardware stops updating the CRC calculation (and then starts transmitting the resulting calculated CRC value).

During a receive operation, each byte of the received frame is written to the MAC memory buffer. It does this with an offset of 128 bytes (32 words) from the transmit partition. In addition, the byte stream is written offset by one byte relative to the transmitted data. This is to make room for the received frame length field, which is the first byte written. (The frame length is received as part of the PHY header, and otherwise would not be present at the MAC layer.) While the received data is always written to the MAC memory buffer, the MAC hardware will not generate the event−complete IRQ with status “Success” unless the incoming address field and CRC validate (address matches, and good CRC) – unless these requirements are waived by setting the Promiscuous Mode Enable and/or Frame Check Override bits in the 0x4001_4004 MAC Sequencer Options register.

When the MAC is performing source address matching, it uses the lookup table (LUT) located in the memory from location 256 (100 hex) to 383 (17F hex). These locations have dedicated connections to the MAC hardware to allow simultaneous parallel matching on all locations. Each 32−bit location in the LUT can have one of two possible configurations:

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	PAN ID																Short Address															

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	Long Address (HI or LOW)																															

Registers

Mac Sequencer

Table 19. RECOMMENDED GAIN SETTINGS

Function	Bits	Default	Type	Description
MAC Sequencer Register: 0x4001_4000 MAC sequencer register to initiate MAC operations. Sequencer options registers and other radio registers should be updated before initiating the selected sequence. Event completion is flagged by interrupts.				
MAC sequence	[2:0]	0x00	RW	Initiate selected sequence according to pre−programmed options 0x00 – No Op 0x03 – Receive 0x04 – Transmit 0x05 – Energy Detect 0x06 – Clear Channel Assessment

MAC Sequencer Options

The MAC sequencer options register sets options that change the behavior of the MAC sequencer operations.

If the Auto Receiver bit is set, when the first valid frame is successfully received, a data−arrived IRQ will be generated. (The IRQ status will remain “Not Complete” until the sequence is terminated, most conveniently via use of the stop timer). When a valid frame is successfully received, the receiver will turn off and remain off for *rxautodelay* (0x4000_4098: *mac_rx_ack_delay*) bit durations, and then restart, so there will be a period during which frames cannot be received (*rxautodelay* includes both the off time and the warm−up time). If the Auto Receiver bit is cleared, when the first valid frame is successfully received the event−complete IRQ status will change from “Not Complete” to “Success” (0x4000_4010: *mac_status*) and the receiver will turn off, completing the sequence.

If the Auto Receiver bit is set, the ACK Disable bit must also be set. Operation with the Auto Receiver bit set and the ACK Disable bit cleared, is undefined. (This is to prevent attempts at simultaneous transmission and reception.) If the incoming frame is requesting an ACK, but the ACK Disable bit is set, the MAC receiver will ignore the request and just restart the receiver operation.

Regardless of the state of the Auto Receiver bit setting, if an invalid frame is received (wrong address, CRC fails, etc.), the receiver will restart and continue without issuing an IRQ of any type. The IRQ status will remain “Not Complete”.

Upon reception of a frame, if the Promiscuous Mode bit is set (promiscuous mode enabled) an ACK will not be sent, regardless of the value of the Acknowledgement Request subfield in the Frame Control field of the received frame, and regardless of the value of the ACK Disable bit.

Upon transmission of a frame, if the Promiscuous Mode bit is set (promiscuous mode enabled) the event−complete IRQ status will change from “Not Complete” to “Success” (0x4000_4010: *mac_status*) whether or not a requested ACK was received, or the value of the ACK Disable bit. However, if the Promiscuous Mode bit is cleared (promiscuous mode disabled), a status of “No ACK” will be returned if the requested ACK is not received. For this reason, in normal operation it is recommended that the Promiscuous Mode bit be cleared for transmission.

Table 20. RECOMMENDED GAIN SETTINGS

Function	Bits	Default	Type	Description
MAC Sequencer Options Register: 0x4001_4004 MAC sequencer options register to initiate MAC operations. Select desired options before triggering the MAC sequencer register.				
Frame filter – ACK Frames	[20]	0x1	RW	Allows reception of ACK frames
Frame filter – Beacon Frames	[19]	0x1	RW	Allows reception of Beacon frames
Frame filter – Command Frames	[18]	0x1	RW	Allows reception of command frames
Frame filter – Data Frames	[17]	0x1	RW	Allows reception of data frames
Frame filter – Reserved Frame Types	[16]	0x1	RW	Allows reception of reserved frame types
Load Slot Timer with Rx Slot Offset (rstr)	[15]	0	RW	1 – Slot timer is loaded with the Rx slot offset stored in register 0x4000_0064 when the first bit of the frame length field is received 0 – No slot timer loaded
Load Slot Timer with Tx Slot Offset (rstt)	[14]	0	RW	1 – Slot timer is loaded with the TX slot offset stored in register 0x4000_0064 when the first bit of the frame length field is transmitted 0 – No slot timer loaded
PAN Coordinator Set	[13]	0	RW	1 – Device is the PAN coordinator 0 – Device is not the PAN coordinator
Frame Check Sequence Override	[12]	0	RW	1 – Return a valid received frame regardless of frame check sequence (FCS) 0 – Require a valid FCS to return a received frame
Promiscuous Mode	[11]	0	RW	1 – Return valid received frame if CRC test passes (no packet filtering), no ACKs generated 0 – Only return a valid received frame if CRC and packet filtering tests pass
Unused	[10]	0	RW	Reserved.
Start Sequence Control	[9]	0	RW	1 – Start sequence/event immediately 0 – Use MAC timer to start event (setup the MAC timer first)
Unused	[8]	0	RW	Reserved
Auto Receiver Enable	[7]	0	RW	1 – Automatically restart the receiver at the end of a valid received frame 0 – Do not automatically restart the receiver at the end of a valid received frame
Unused	[6:4]	0	RW	Reserved.
Force Transmission	[3]	0	RW	1 – Force transmission even if the channel is busy 0 – Require completion of CSMA−CA algorithm before transmission (slotted and non−slotted modes only, ignored in normal mode)

Table 20. RECOMMENDED GAIN SETTINGS

Function	Bits	Default	Type	Description
ACK Disable	[2]	0	RW	If set, disable automatic transmission and reception of ACKs. If cleared, perform automatic transmission and reception of ACKs. (But see text below concerning the effects of the *prm* bit on ACKs.) 1 – Automatic transmit and receive ACKs are disabled 0 – Automatic transmit and receive ACKs are enabled (see note about promiscuous mode)
Transmit Mode	[1:0]	00	RW	0x0 – Normal mode transmit. No CSMA−CA CCA performed and transmit starts immediately 0x1 – Non−slotted mode transmit. A single CSMA−CA CCA measurement is done before transmit. No time slot alignment. 0x2 – Slotted mode transmit. A double CSMA−CA CCA measurement is done before transmit and is slot aligned.

Table 21. MAC CONTROL

Function	Bits	Default	Type	Description
MAC Control Register: 0x4001_4008 A MAC Synchronous Reset request resets the values of the internal MAC state machine variables, but does not affect the memory−mapped MAC register values. The MAC Synchronous Reset bit is not self−clearing and a “0” must be written to MAC Synchronous Reset to release the synchronous reset.				
MAC Clock Divider	[2]	0x0	RW	1 – Override the current clock divider values with 0 (use for synchronous reset) 0 – Do not override the current clock divider values with 0
MAC Clock Enable	[1]	0x0	RW	1 – MAC clock is on 0 – MAC clock is off
MAC Synchronous Reset	[0]	0x0	RW	1 – Force an asynchronous reset of MAC peripheral 0 – No reset

Table 22. MAC STATUS

Function	Bits	Default	Type	Description
MAC Status Register: 0x4001_4010 After a MAC sequence completes, an event complete IRQ is generated (0x4000_4038), and the event status can be read in the MAC Status register as shown below. If the status register is read during a sequence, the incomplete sequence status will be returned.				
MAC Status Timeout	[15]		RO	1 – MAC Status timeout 0 – No timeout
Channel Busy	[13]		RO	1 – Channel busy as determined by RSSI > RSSI Threshold 0 – Channel not busy

Table 22. MAC STATUS

Function	Bits	Default	Type	Description
MAC Status Over Range	[12]		RO	1 – MAC sequence started without DMA Rx pointer setup 0 – MAC sequence started with DMA Rx pointer set-up
MAC Event Code	[3:0]		RO	0x0 – MAC Sequence completed successfully 0x1 – MAC timer expired 0x2 – Channel busy 0x3 – CRC Failure (only returned if MAC Sequence Options Frame Check Sequence Enable is set) 0x5 – No ACK 0x6 – PLL unlocked 0x7 – Bad start (timed event requested but start time is in the past) 0x8 – ACK frame pending Rx (ACK frame received with pending bit set) 0x9 – ACK frame pending Tx (ACK frame was sent with pending bit set) 0xA – Failed packet filtering 0xB – PAN ID conflict 0xF – Incomplete sequence

Table 23. MAC OPTIONS

Function	Bits	Default	Type	Description
MAC Options Register: 0x4001_4014 The MAC Options register controls how the transmitted data packet is built. It contains general purpose options that in normal operation will never be used.				
Tx Frame Pending Bit Override (tfpo)	[5]	0	RW	TX Frame Pending Bit override (signals an override of the Frame Pending Bit value in the Frame Control field of ACKs sent in response to a data request command.)
Set Sampling Phase Relative to Bit Clock (ssp)	[4]	0	RW	Set sampling phase of received bit stream relative to the bit clock. 1 – Sampling is done near the negative−going edge of the bit clock 0 – Sampling is done near the positive−going edge of the bit clock.
Shift Direction of Bits (sdb)	[3]	0	RW	Shift direction (bits) 1 – Reverses the bit order in each 4−bit symbol 0 – Does nothing.
Invert Chips (Ic)	[2]	0	RW	Invert chips 1 – Inverts chips (1→0, 01). 0 – Does nothing
Shift Direction of Chips (sdc)	[1]	0	RW	Shift direction (chips) 1 – Reverses the transmission order of the 32−bit chip sequence. 0 – Does nothing
Tx Frame Pending Bit (tfp)	[0]	0	RW	TX Frame Pending Bit (Frame Pending Bit value set by SW in the Frame Control field of ACKs sent in response to a data request command). Use with Tx Frame Pending Bit Override (tfpo).

Table 24. MAC PAN ID

Function	Bits	Default	Type	Description
MAC PAN ID Register: 0x4001_4018 The MAC PAN ID of the device.				
Device PAN ID	[15:0]	0xFFFF	RW	The 16 bit PAN ID of the device

Table 25. MAC SHORT ADDRESS

Function	Bits	Default	Type	Description
MAC Short Address Register: 0x4001_401C The MAC short address of the device.				
Device Short Address	[15:0]	0x0000	RW	The 16 bit short address of the device

Table 26. MAC LONG ADDRESS HIGH

Function	Bits	Default	Type	Description
MAC Long Address MSW Register: 0x4001_4020 The upper word of the device's long MAC address.				
MSW Device Long Address	[31:0]	0x00000000	RW	The most significant word, or upper 32 bits, of the device's MAC long address

Table 27. MAC LONG ADDRESS LOW

Function	Bits	Default	Type	Description
MAC Long Address LSW Register: 0x4001_4024 The lower word of the device's long MAC address.				
LSW Device Long Address	[31:0]	0x00000000	RW	The least significant word, or lower 32 bits, of the device's MAC long address

Table 28. MAC DIVIDER

Function	Bits	Default	Type	Description
MAC Divider Register: 0x4001_4028 MAC clock divider control register provides the time base for the protocol timer and sets the transmit data rate, all based on the microprocessor clock frequency, FCLK, in MHz.				
Chip Clock Divider	[23:16]	0x0F	RW	The chip clock divider value is (FCLK/ 2) − 1
System Clock Divider	[15:8]	0x1F	RW	The system clock divider value is FCLK− 1
Bit Clock Divider	[7:0]	0x7F	RW	The bit clock divider value is (4 * FCLK) − 1. The bit clock sets not only the transmitted bit rate, but also serves as the time base for the protocol timer. The bit clock divider value should produce a bit clock with a period of 4 μs.

Table 29. MAC RX/TX WARMUPS

Function	Bits	Default	Type	Description
MAC Rx/Tx Warmups Register: 0x4001_402C The MAC Rx/Tx Warmups register sets the warmup time duration for the transmitter and receiver, in units of bits (4 μs).				
Transmit Warmup	[27:16]	0x017	RW	
Receive Warmup	[11:0]	0x017	RW	

Table 30. MAC CLEAR INTERRUPTS

Function	Bits	Default	Type	Description
MAC Clear Interrupts Register: 0x4001_4030 Clear selected MAC interrupts (all at once or individually). These bits are self−clearing.				
PAN ID conflict IRQ	[6]	0x0	RO	1 − Clear the PAN ID conflict IRQ
Frame−match−done IRQ	[5]	0x0	RO	1 − Clear the frame−match−done IRQ

Table 30. MAC CLEAR INTERRUPTS

Function	Bits	Default	Type	Description
Failed−packet IRQ	[4]	0x0	RO	1 − Clear the failed−packet IRQ
Frame−started IRQ	[3]	0x0	RO	1 − Clear the frame−started IRQ
Data−arrived IRQ	[2]	0x0	RO	1 − Clear the data−arrived IRQ
Event−started IRQ	[1]	0x0	RO	1 − Clear the event−started IRQ
Event−complete IRQ	[0]	0x0	RO	1 − Clear the event−complete IRQ

Table 31. MAC INTERRUPT MASK

Function	Bits	Default	Type	Description
MAC Interrupts Mask Register: 0x4001_4034 Setup MAC interrupt mask (all at once or individually).				
PAN ID conflict IRQ	[6]	0x0	RW	1 − Enable the PAN ID conflict IRQ 0 − Mask the PAN ID conflict IRQ
Frame−match−done IRQ	[5]	0x0	RW	1 − Enable the frame−match−done IRQ 0 − Mask the frame−match−done IRQ
Failed−packet IRQ	[4]	0x0	RW	1 − Enable the failed−packet IRQ 0 − Mask the failed−packet IRQ
Frame−started IRQ	[3]	0x0	RW	1 − Enable the frame−started IRQ 0 − Mask the frame−started IRQ
Data−arrived IRQ	[2]	0x0	RW	1 − Enable the data−arrived IRQ 0 − Mask the data−arrived IRQ
Event−started IRQ	[1]	0x0	RW	1 − Enable the event−started IRQ 0 − Mask the event−started IRQ
Event−complete IRQ	[0]	0x0	RW	1 − Enable the event−complete IRQ 0 − Mask the event−complete IRQ

Table 32. MAC INTERRUPT STATUS

Function	Bits	Default	Type	Description
MAC Interrupts Status Register: 0x4001_4038 MAC interrupt status.				
PAN ID conflict IRQ	[6]	0x0	RW	1 − PAN ID conflict IRQ occurred 0 − PAN ID conflict IRQ did not occur
Frame−match−done IRQ	[5]	0x0	RW	1 − Frame−match−done IRQ occurred 0 − Frame−match−done IRQ did not occur
Failed−packet IRQ	[4]	0x0	RW	1 − Failed−packet IRQ occurred 0 − Failed−packet IRQ did not occur
Frame−started IRQ	[3]	0x0	RW	1 − Frame−started IRQ occurred 0 − Frame−started IRQ did not occur
Data−arrived IRQ	[2]	0x0	RW	1 − Data−arrived IRQ occurred 0 − Data−arrived IRQ did not occur
Event−started IRQ	[1]	0x0	RW	1 − Event−started IRQ occurred 0 − Event−started IRQ did not occur
Event−complete IRQ	[0]	0x0	RW	1 − Event−complete IRQ occurred 0 − Event−complete IRQ did not occur

Table 33. MAC TIMER ENABLE

Function	Bits	Default	Type	Description
MAC Timer Enable Register: 0x4001_4040 Enable MAC Timer.				
Stop Timer Enable	[1]	0x0	RW	1 – Enable the stop timer
Start Timer Enable	[0]	0x0	RW	1 – Enable the start timer

Table 34. MAC TIMER ENABLE

Function	Bits	Default	Type	Description
MAC Timer Enable Register: 0x4001_4044 Disable MAC Timer.				
Stop Timer Enable	[1]	0x0	RW	1 – Disable the stop timer
Start Timer Enable	[0]	0x0	RW	1 – Disable the start timer

Table 35. MAC TIMER LOAD

Function	Bits	Default	Type	Description
MAC Timer Load Register: 0x4001_4048 Write a MAC timer value.				
MAC Timer Value	[31:0]	0x00000000	RW	Load value for the MAC protocol timer. Time base is the bit clock (0x4000_4028 MAC divider)

Table 36. MAC START TIME

Function	Bits	Default	Type	Description
MAC Start Time Register: 0x4001_404C Start time when using timed MAC events.				
MAC Start Time	[31:0]	0x00000000	RW	The time an event is started when timed start is used. This value is compared to the value in the protocol timer and offset by the higher of the Transmit and Receive Warmup times (0x4000_402C MAC Rx/Tx Warmups)

Table 37. MAC TIMER STATUS

Function	Bits	Default	Type	Description
MAC Timer Status Register: 0x4001_4054 Show status of the stop timer.				
Stop Timer Status	[1]		RO	1 – On 0 – Off
Start Timer Status	[0]		RO	1 – On 0 – Off

Table 38. MAC PROTOCOL TIMER

Function	Bits	Default	Type	Description
MAC Protocol Timer Register: 0x4001_4058 Show the current status of the protocol timer.				
Protocol Timer Status	[31:0]		RO	Current state of the protocol timer. The time base of the timer is the bit clock (0x4000_4028 MAC divider)

Table 39. MAC FINISH TIME

Function	Bits	Default	Type	Description
MAC Finish Time Register: 0x4001_4060 Returns the value of the protocol timer when the last event complete IRQ was generated.				
MAC Finish Time	[31:0]		RO	Value of the protocol timer when the last event complete IRQ was generated

Table 40. MAC SLOT OFFSET

Function	Bits	Default	Type	Description
MAC Slot Offset Register: 0x4001_4064 Returns the value of the protocol timer when the last event complete IRQ was generated.				
Rx Slot Offset	[27:16]	0x007	RW	If the reset slot offset function is enabled (0x4000_4004 MAC Sequencer Options, rstr), this value is loaded into the slot timer as the first bit of the frame length field of the packet being received. The time unit is 1 bit (4 μs)
Tx Slot Offset	[11:0]	0x008	RW	If the reset slot offset function is enabled (0x4000_4004 MAC Sequencer Options, rstt), this value is loaded into the slot timer as the first bit of the frame length field of the packet being transmitted. The time unit is 1 bit (4 μs)

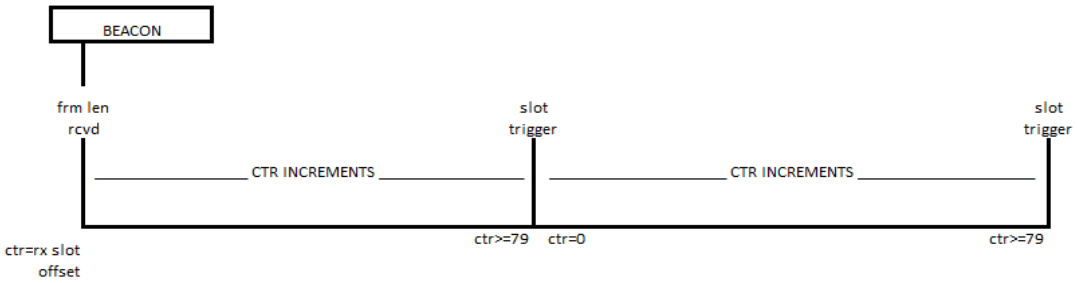


Figure 5.

Table 41. MAC TIME STAMP

Function	Bits	Default	Type	Description
MAC Time Stamp Register: 0x4001_4068 Returns the value of the protocol timer when the frame length field of the received packet is received.				
MAC Time Stamp	[31:0]		RO	Value of the protocol timer when the frame length field of the received packet is received

Table 42. MAC COORDINATOR SHORT ADDRESS

Function	Bits	Default	Type	Description
MAC Coordinator Short Address Register: 0x4001_4070 Control and values related to the MAC coordinator short address. If bit [30] is set, the contents of this register contain a valid short address for the PAN coordinator. If the bit is equal to 0, the address is not valid. This allows the PAN Coordinator address to have a value of 16'h0000 without it appearing that the register was not written.				
PAN Coordinator Short Address Indicator	[31]	0x0	RW	Indicates whether the address in bits [15:0] is for the PAN coordinator. 1 – The short address is for the PAN coordinator 0 – There is no valid short address in bits [15:0]

Table 42. MAC COORDINATOR SHORT ADDRESS

Function	Bits	Default	Type	Description
PAN Coordinator Long Address Indicator	[30]	0x0	RW	Indicates whether the long address in registers 0x4000_4074 and 0x4000_4078 is for the PAN coordinator. 1 – The long address is for the PAN coordinator 0 – There is no valid long address in registers 0x4000_4074 and 0x4000_4078
Device Associated Indicator	[29]	0x0	RW	Indicates whether this device is an associated device. 1 – The device is associated 0 – The device is not associated
PAN Coordinator 16 bit short address	[15:0]	0x0000	RW	Stores the 16–bit short address of the PAN Coordinator

Table 43. MAC COORDINATOR LONG ADDRESS HIGH

Function	Bits	Default	Type	Description
MAC Coordinator Long Address High Register: 0x4001_4074 MSW of the unique 64 bit extended address of the PAN coordinator. If bit [30] of register 0x4000_4070 is set, the contents of this register concatenated with the contents of register 0x4000_4078 contain a valid address for the PAN coordinator. If the bit is equal to 0, the concatenated contents of the two registers are not valid. This allows the PAN Coordinator address to have a value of 64'h00000000 without it appearing that the registers were not written.				
MSW PAN Coordinator Long Address	[31:0]	0x00000000	RW	Most significant word of the PAN coordinator unique 64 bit long address

Table 44. MAC COORDINATOR LONG ADDRESS LOW

Function	Bits	Default	Type	Description
MAC Coordinator Long Address Low Register: 0x4001_4078 LSW of the unique 64 bit extended address of the PAN coordinator.				
LSW PAN Coordinator Long Address	[31:0]	0x00000000	RW	Least significant word of the PAN coordinator unique 64 bit long address

Table 45. MAC RX LENGTH

Function	Bits	Default	Type	Description
MAC Rx Length Register: 0x4001_4088 Value is the number of bytes received by the PSDU of the MAC. This includes the PHY header.				
Number of Received Bytes	[7:0]	0x00	RO	Number of bytes received by the PSDU of the MAC including the PHR

Table 46. MAC TX LENGTH

Function	Bits	Default	Type	Description
MAC Tx Length Register: 0x4001_408C Number of bytes transmitted by the PSDU of the MAC including the PHY header. Number of warmup chips specified here also.				
Number of Warmup Chips to Transmit	[11:8]	0x00	RW	Number of chips to transmit before the start of the transmitted frame
Number of Transmitted Bytes	[6:0]	0x00	RW	Number of bytes transmitted by the PSDU of the MAC including the PHR

Table 47. MAC TX SEQUENCE NUMBER

Function	Bits	Default	Type	Description
MAC Tx Sequence Number Register: 0x4001_4090 Transmit sequence number.				
MAC Tx Sequence Number	[7:0]	0x00	RW	MAC Tx sequence number is a unique number that is a property of the transmitted packet. When an ACK of the transmitted packet is expected, the sequence number should be written to this register for comparison to the sequence number received in the ACK frame. This register therefore contains a copy of the sequence number of the transmitted frame.

Table 48. MAC TX ACK DELAY

Function	Bits	Default	Type	Description
MAC Tx ACK Delay Register: 0x4001_4094 When Tx auto ACK is enabled (0x4000_4004 MAC Sequencer Options), this delay defines the time from the end of a received frame to the start of an automatically transmitted ACK (including the Tx warmup time). Time is measured in bit time which is 4 μs per bit.				
MAC Tx Sequence Number	[11:0]	0x22	RW	MAC Tx sequence number is a unique number that is a property of the transmitted packet. When an ACK of the transmitted packet is expected, the sequence number should be written to this register for comparison to the sequence number received in the ACK frame. This register therefore contains a copy of the sequence number of the transmitted frame.

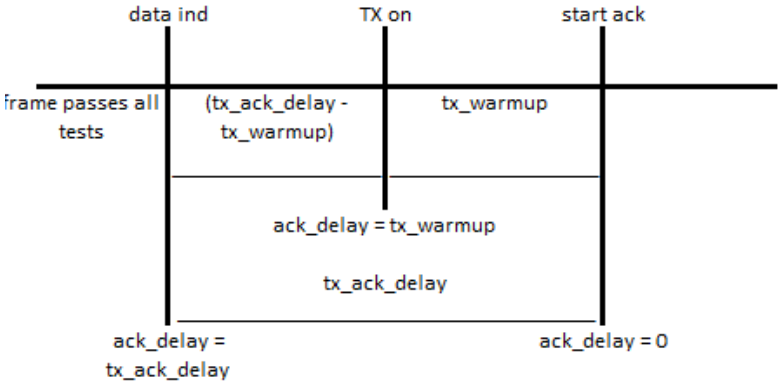
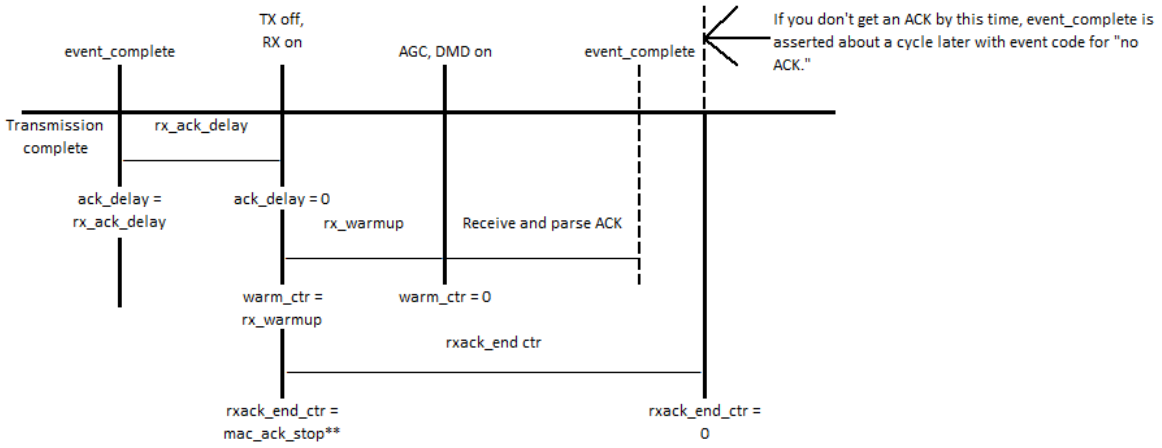


Figure 6.

Table 49. MAC RX ACK DELAY

Function	Bits	Default	Type	Description
MAC Rx ACK Delay Register: 0x4001_4098 When Rx auto restart is enabled (0x4001_0004 MAC Sequencer Options) this delay is the delay from the end of the previous receive event to the start of a new receive event. The time is measured in bit time, or 4 μs per bit.				
Rx Auto Delay	[27:16]	0x30	RW	When Rx auto restart is enabled this delay is the delay from the end of the previous receive event to the start of a new event in bit time units of 4μs per bit
Rx ACK Delay	[11:0]	0x4	RW	The delay from the end of the transmitted frame to the start of the receiver for an ACK in bit time units of 4 μs per bit



**NOTE: mac_ack_stop (Reg. 0a4) is different depending on whether you are in slot mode or not.

MODE	mac_ack_stop
Slot	rxslotted_end
Non-slot	rxack_end

Figure 7.

Table 50. MAC TX FLUSH

Function	Bits	Default	Type	Description
MAC Tx Flush Register: 0x4001_409C How long the transmitter stays on after the frame is transmitted.				
Tx Flush Time	[11:0]	0x4	RW	The time the transmitter stays on after transmitting a frame in units of 4 μs per bit

Table 51. MAC CCA DELAY

Function	Bits	Default	Type	Description
MAC Tx Flush Register: 0x4001_40A0 Clear channel assessment (CCA) parameter control.				
CCA Duration (cca_length)	[27:16]	0x03	RW	Time allocated for CCA or ED measurement as determined by the operation mode (0x4000_4004 MAC Sequencer Options). Time is measured in units of 4 μs. Slotted mode – The value of CCA duration is just the measurement period and does not include the warmup time Non–Slotted mode – The value of CCA duration is just the measurement period and does not include the warmup time Normal mode – The value of CCA duration includes the warmup time
CCA Delay (cca_delay)	[11:0]	0x1F	RW	The time between CCA measurements in a slotted system measured in units of 4 μs

Table 52. MAC ACK STOP

Function	Bits	Default	Type	Description
MAC Tx Flush Register: 0x4001_40A4 After a transmit sequence the MAC can be programmed to switch to receive mode to wait for an ACK. The timeout time is set using this register. The value depends on if slotted or non−slotted operation is being used.				
Rx Slotted Stop Time	[27:16]	0x170	RW	Time period after a transmit sequence for which the receiver remains on waiting for an ACK in a slotted system. This value must be greater than the Receive Warmup time in register 0x4000_402C.
Rx Non−Slotted Stop Time	[11:0]	0x170	RW	Time period after a transmit event for which the receiver remains on waiting for an ACK in a non−slotted system. This value must be greater than the Receive Warmup time in register 0x4000_402C.

Table 53. MAC TX CCA

Function	Bits	Default	Type	Description
MAC Tx CCA Register: 0x4001_40A8 Time delay between completion of CCA and the start of the frame transmission.				
Transmit CCA Delay	[11:0]	0x30	RW	Time from the end of a successful CCA measurement to the frame transmission start. Time measured in units of 4 μs per bit

Table 54. MAC LONG ADDRESS POINTER

Function	Bits	Default	Type	Description
MAC Long Address Pointer Register: 0x4001_40A8 Shows which addresses in the matching lookup table (LUT) contain long (64 bit) addresses. Each bit points to the corresponding location pair in the LUT. Example: 0x001 points to LUT locations 0 and 1.				
Long Address Pointer	[11:0]	0x000	RW	Shows which addresses in the matching lookup table (LUT) contain long (64 bit) addresses. Each bit points to the corresponding location pair in the LUT. Example: 0x001 points to LUT locations 0 and 1.

Table 55. MAC SHORT ADDRESS POINTER

Function	Bits	Default	Type	Description
MAC Short Address Pointer Register: 0x4001_40B0 Shows which addresses in the matching lookup table (LUT) contain short (32−bit) address/PAN ID pairs. Each bit points to the corresponding location in the LUT.				
Short Address Pointer	[23:0]	0x000000	RW	Shows which addresses in the matching lookup table (LUT) contain short (32−bit) address/PAN ID pairs. Each bit points to the corresponding location in the LUT.

Table 56. MAC FRAME MATCH RESULT

Function	Bits	Default	Type	Description
MAC Frame Match Result Register: 0x4001_40B4 Result vector for frame matching operation.				
Frame Match Result − Short or Long Address	[5]		RO	1 − Extended long address 0 − Short address
Frame Match Result	[4:0]		RO	The bit index of the least significant 1 in either registers 0x4000_40B8 or 0x4000_40BC. 0x3F − No match

Table 57. MAC FRAME MATCH LONG ADDRESS POINTER

Function	Bits	Default	Type	Description
MAC Frame Match Long Address Pointer Register: 0x4001_40B8 Returns which long addresses in the matching look up table (LUT) matched the incoming long (64 bit) source address. Each bit corresponds to the corresponding location pair in the LUT.				
Frame Match Long Address Pointer	[11:0]		RO	Returns which long addresses in the matching look up table (LUT) matched the incoming long (64 bit) source address. Each bit corresponds to the corresponding location pair in the LUT.

Table 58. MAC FRAME MATCH SHORT ADDRESS POINTER

Function	Bits	Default	Type	Description
MAC Frame Match Short Address Pointer Register: 0x4001_40BC Returns which short PAN ID/addresses in the matching look up table (LUT) matched the incoming short (16 bit) source address and 16 bit PAN ID. Each bit corresponds to the corresponding location pair in the LUT.				
Frame Match Short Address Pointer	[11:0]		RO	Returns which short PAN ID/addresses in the matching look up table (LUT) matched the incoming short (16 bit) source address and 16 bit PAN ID. Each bit corresponds to the corresponding location pair in the LUT.

Table 59. MAC AGC CONTROL

Function	Bits	Default	Type	Description
MAC AGC Control Register: 0x4001_40C0 Control settings for the receiver automatic gain control (AGC) function.				
Gain Start Value	[31:28]	0x0	RW	1 – The AGC algorithm starts with the gain value in this register 0 – The AGC gain setting is held at the value in this register
Freeze Gain After Preamble	[2]	0x0	RW	1 – Freeze AGC gain value when preamble is detected 0 – AGC gain can be frozen if the start of frame delimiter is found Both options are disabled if the register Allow Freeze AGC is low
Allow Freeze AGC	[1]	0x1	RW	1 – Allows AGC setting to be frozen in response to a preamble detection or start of frame delimiter detection 0 – AGC operates continuously
Allow AGC	[0]	0x1	RW	1 – AGC active 0 – AGC gain held at Gain Start Value above

Table 60. MAC AGC SETTINGS

Function	Bits	Default	Type	Description
MAC AGC Settings Register: 0x4001_40C4 Receiver automatic gain control (AGC) settings.				
AGC Low Threshold	[31:28]	0x1	RW	Value of the low AGC threshold. Gain values are relative to the AGC high threshold below. 0x0000: –9 dB 0x0001: –7 dB

Table 60. MAC AGC SETTINGS

Function	Bits	Default	Type	Description
AGC High Threshold	[27:24]	0x2	RW	Value of the high AGC threshold. Gain values are relative to the full scale value of the receiver path ADC. 0x0000: 0dB 0x0001: −1dB 0x0010: −3 dB 0x0011: −5 dB
AGC Clock Divider	[23:16]	0x01	RW	Clock divider for AGC. AGC Clock = FCLK/(divider + 1). FCLK is the MCU clock.
AGC Measurement Delay	[15:8]	0x017	RW	The number of clock cycles over which the input signal is sampled for each AGC step adjustment. This value is offset by 1 (a setting of 0 is actually 1)
AGC Settle Delay	[7:0]	0x07	RW	The number of clock cycles of delay after an AGC step adjustment and before the beginning of the next AGC measure cycle. This value is offset by 1 (a setting of 0 is actually 1)

Table 61. MAC AGC STATUS

Function	Bits	Default	Type	Description
MAC AGC Settings Register: 0x4001_40C8 Receiver automatic gain control (AGC) status register. The value of each register is captured at the end of packet reception or at the end of ED/CCA measurements.				
AGC State	[11:8]		RO	AGC state machine state. Allowed values are 0 – 11 (decimal). A value of 0 indicates the min gain, and 11 is the max gain. Each state value maps to the corresponding gain registers in 0x4001_40CC and 0x4001_40D4 (MAC AGC gain tables). 0x0 – Value specified in gain state 0 0x1 – Value specified in gain state 1 0x2 – Value specified in gain state 2 0x3 – Value specified in gain state 3 0x4 – Value specified in gain state 4 0x5 – Value specified in gain state 5 0x6 – Value specified in gain state 6 0x7 – Value specified in gain state 7 0x8 – Value specified in gain state 8 0x9 – Value specified in gain state 9 0xA – Value specified in gain state 10 0xB – Value specified in gain state 11
Gain of 3rd IF Amplifier	[6]		RO	0 – 0 dB 1 – 18 dB
Gain of 2nd IF Amplifier	[5:3]		RO	000 – 0 dB 001 – 6 dB 010 – 12 dB 011 – 18 dB 1xx – 24 dB
Gain of 1st IF Amplifier	[2:0]		RO	000 – 0 dB 001 – 6 dB 010 – 12 dB 011 – 18 dB 1xx – 24 dB

Table 62. MAC AGC GAIN TABLE 0

Function	Bits	Default	Type	Description
MAC AGC Gain Table 0 Register: 0x4001_40CC Receiver automatic gain control (AGC) gain table 0 register. The MAC AGC control manipulates 3 variable gain stages in the IF amplifier chain in the receiver. Settings cover 12 distinct gain states. Because there are multiple gain settings of the IF amplifiers that can result in the same gain, it is necessary to define a unique set of gain settings for each state. The settings are stored in registers 0x4001_40CC, 0x4001_40D0, 0x4001_40D4, and 0x4001_40D8. Because the AGC control mechanism defines only 12 states (0 to 11), 0x4001_40D8 (states 12−15) is effectively unused.				
IF Amplifier Gain Settings for AGC state 0	[30:24]	0x00	RW	IF amplifier gain settings for AGC state 0
IF Amplifier Gain Settings for AGC state 1	[22:16]	0x08	RW	IF amplifier gain settings for AGC state 1
IF Amplifier Gain Settings for AGC state 2	[14:8]	0x10	RW	IF amplifier gain settings for AGC state 2
IF Amplifier Gain Settings for AGC state 3	[6:0]	0x40	RW	IF amplifier gain settings for AGC state 3

Table 63. MAC AGC GAIN TABLE 1

Function	Bits	Default	Type	Description
MAC AGC Gain Table 1 Register: 0x4001_40D0 Receiver automatic gain control (AGC) gain table 1 register. The MAC AGC control manipulates 3 variable gain stages in the IF amplifier chain in the receiver. Settings cover 12 distinct gain states. Because there are multiple gain settings of the IF amplifiers that can result in the same gain, it is necessary to define a unique set of gain settings for each state. The settings are stored in registers 0x4001_40CC, 0x4001_40D0, 0x4001_40D4, and 0x4001_40D8. Because the AGC control mechanism defines only 12 states (0 to 11), 0x4001_40D8 (states 12−15) is effectively unused.				
IF Amplifier Gain Settings for AGC state 4	[30:24]	0x48	RW	IF amplifier gain settings for AGC state 4
IF Amplifier Gain Settings for AGC state 5	[22:16]	0x50	RW	IF amplifier gain settings for AGC state 5
IF Amplifier Gain Settings for AGC state 6	[14:8]	0x58	RW	IF amplifier gain settings for AGC state 6
IF Amplifier Gain Settings for AGC state 7	[6:0]	0x60	RW	IF amplifier gain settings for AGC state 7

Table 64. MAC AGC GAIN TABLE 2

Function	Bits	Default	Type	Description
MAC AGC Gain Table 2 Register: 0x4001_40D4 Receiver automatic gain control (AGC) gain table 2 register. The MAC AGC control manipulates 3 variable gain stages in the IF amplifier chain in the receiver. Settings cover 12 distinct gain states. Because there are multiple gain settings of the IF amplifiers that can result in the same gain, it is necessary to define a unique set of gain settings for each state. The settings are stored in registers 0x4001_40CC, 0x4001_40D0, 0x4001_40D4, and 0x4001_40D8. Because the AGC control mechanism defines only 12 states (0 to 11), 0x4001_40D8 (states 12−15) is effectively unused.				
IF Amplifier Gain Settings for AGC state 8	[30:24]	0x61	RW	IF amplifier gain settings for AGC state 8
IF Amplifier Gain Settings for AGC state 9	[22:16]	0x62	RW	IF amplifier gain settings for AGC state 9
IF Amplifier Gain Settings for AGC state 10	[14:8]	0x63	RW	IF amplifier gain settings for AGC state 10
IF Amplifier Gain Settings for AGC state 11	[6:0]	0x64	RW	IF amplifier gain settings for AGC state 11

Table 65. MAC AGC GAIN TABLE 3

Function	Bits	Default	Type	Description
MAC AGC Gain Table 3 Register: 0x4001_40D8 Receiver automatic gain control (AGC) gain table 2 register. The MAC AGC control manipulates 3 variable gain stages in the IF amplifier chain in the receiver. Settings cover 12 distinct gain states. Because there are multiple gain settings of the IF amplifiers that can result in the same gain, it is necessary to define a unique set of gain settings for each state. The settings are stored in registers 0x4001_40CC, 0x4001_40D0, 0x4001_40D4, and 0x4001_40D8. Because the AGC control mechanism defines only 12 states (0 to 11), 0x4001_40D8 (states 12−15) is effectively unused.				
IF Amplifier Gain Settings for AGC state 12	[30:24]	0x00	RW	IF amplifier gain settings for AGC state 12
IF Amplifier Gain Settings for AGC state 13	[22:16]	0x00	RW	IF amplifier gain settings for AGC state 13
IF Amplifier Gain Settings for AGC state 14	[14:8]	0x00	RW	IF amplifier gain settings for AGC state 13
IF Amplifier Gain Settings for AGC state 15	[6:0]	0x00	RW	IF amplifier gain settings for AGC state 14

Table 66. MAC DEMODULATOR CONTROL 0

Function	Bits	Default	Type	Description
MAC Demodulator Control 0 Register: 0x4001_4100 Various demodulator settings.				
Frequency Offset Range (dfr)	[30:16]	0x7FFF	RW	Selectively enables individual frequency offsets used during preamble search. Each of the 15 bits in this field correspond to 15 different frequency offsets. Bits set include the particular frequency offset from the search. Bits cleared exclude the particular frequency offset form the search. [16] – Frequency offset index 0 (−7 * 31.25 kHz) … [30] – Frequency offset index 14 (+7 * 31.25 kHz) The default value is for all bits in the field to have value 1, resulting in a search range of +/− 7.5*31.25 kHz (+/− 94.5 ppm at 2.48 GHz). If the TX or RX is known to have high stability, or if the frequency offset is known from previous packet reception results, a smaller range of offsets can be used. The term “frequency offset” refers to the adjustment applied by the demodulator. For example, if frequency offset index 0 is enabled, the demodulator will frequency−shift the signal down by 7*31.25 kHz prior to correlation.
RSSI and Digital Gain Control time decay constant (dtc)	[6]	0x1	RW	1 – Time constant set to 1/4th of a symbol period. This produces a faster response with slightly more variance in the RSSI calculation. Recommended for most cases. 0 – Time constant set to one symbol period. This produces a slower response time but more stable RSSI values. Not recommended for use with antenna diversity.
Antenna Selection (as)	[5]	0x0	RW	1 – Antenna 1 0 – Antenna 0
Antenna Selection Mode (asm)	[4]	0x0	RW	1 – Automatic antenna selection 0 – Manual antenna selection

Table 66. MAC DEMODULATOR CONTROL 0

Function	Bits	Default	Type	Description
Preamble Mode (pm)	[3]	0x0	RW	1 – Mode 1 (low false detection) – Preamble detection is based on a span of 4 consecutive symbol periods. Each symbol period produces a time index and frequency index corresponding to the largest correlation peak. If all four symbol periods produce time/frequency index values that meet a set of similarity criteria, then preamble detection is declared. This mode enforces a more strict detection rule and therefore offers lower rate of false preamble detection at the expense of higher missed detection. 0 – Mode 0 (high sensitivity) – Preamble detection is based on observation of a regular pattern of correlation peaks over a span of 5 consecutive symbol periods. Each symbol period produces a time index and frequency index corresponding to the largest correlation peak. If 4 out of 5 symbol periods produce time/frequency index values that meet a set of similarity criteria, then preamble detection is declared. This mode improves preamble detection rate by tolerating one corrupt correlation result in the span of 5 symbols. However, the relaxed detection rule allows a higher rate of false preamble detection when no signal is present.
Low IF Frequency Selection (lif)	[2]	0x1	RW	1 – Demodulator performs up−conversion from −1.23 MHz to baseband 0 – Demodulator performs down−conversion from +1.23 MHz to baseband
Swap I/Q Data	[1]	0x0	RW	1 – Swap In−Phase (I) and Quadrature (Q) components 0 – Normal, no swapping

Table 67. MAC DEMODULATOR CONTROL 1

Function	Bits	Default	Type	Description
MAC Demodulator Control 1 Register: 0x4001_4104 Various demodulator settings.				
Demodulator Correlator Threshold (dct)	[27:24]	0x8	RW	In order for preamble detection to be declared, the correlation peaks must exceed a threshold. The threshold is computed dynamically and includes a programmable scale factor: 1 + bit[27]/2 + bit[26]/4 + bit[25]/8 + bit[24]/16 The default value of 1.5 is recommended for manual−antenna selection, while a value of 1.75 is recommended for automatic antenna selection.
Preamble Frequency Span (dpf)	[19:16]	0x1	RW	The similarity criteria used for preamble detection includes a rule that all frequency index values must occupy a span equal to or less than this value. The default span of 0001 means that the difference between largest frequency index and smallest frequency index must be less than or equal to 1.

Table 67. MAC DEMODULATOR CONTROL 1

Function	Bits	Default	Type	Description
Preamble Time Span (dpt)	[13:8]	0x03	RW	The similarity criteria used for preamble detection includes a rule that all time index values must occupy a span equal to or less than this value. The default span of 0011 means that the correlation peaks must span a range of 3Ts, where Ts is the sample period = 0.25 μs. This value is recommended for typical multipath conditions. Very long−range applications may benefit from a higher value.
SFD Timeout (dst)	[3:0]	0x8	RW	This value specifies the SFD search period in symbols. After preamble detection, the demodulator begins symbol recovery and searches for the start−of−frame delimiter (SFD). If the SFD is not found within the number of symbols specified, the preamble detection flag is cleared and a new preamble search is initiated. The default value of 8 symbols should be sufficient for 802.15.4 compliant applications.

Table 68. MAC DEMODULATOR CONTROL 2

Function	Bits	Default	Type	Description
MAC Demodulator Control 2 Register: 0x4001_4108 Various demodulator settings.				
RSSI Calibration Value	[13:8]	0x00	RW	Calibration constant added to the RSSI calculation. The 6−bit field is treated as a signed value in two's complement format with values from −32 to +31 dB.
RSSI Threshold	[7:0]	0xFF	RW	Threshold value used to determine clear channel assessment (CCA) result. The channel is declared busy if RSSI > threshold.

Table 69. MAC DEMODULATOR STATUS

Function	Bits	Default	Type	Description
MAC Demodulator Status Register: 0x4001_410C Demodulator status register.				
Baseband RSSI Component	[19:16]		RO	Magnitude of the baseband digital signal (units are relative to ADC saturation). The value is updated until the AGC is frozen. The value is captured at the end of packet reception or at the end of ED/CCA measurements. 0000: below −42 dB 0001: −39 to −42 dB … 1110: 0 to −3 dB 1111: above 0 dB
Receive Antenna	[12]		RO	1 − Antenna 1 0 − Antenna 0

Table 69. MAC DEMODULATOR STATUS

Function	Bits	Default	Type	Description
Frequency Offset	[11:8]		RO	Frequency correction applied to the received packet. The value is captured at the end of packet reception or at the end of the ED/CCA measurements. 0000: −7 * 31.25 kHz 0001: −6 * 31.25 kHz … 0111: 0 * 31.25 kHz … 1110: +7 * 31.25 kHz
RSSI Value	[7:0]		RO	The value is captured at the end of packet reception or at the end of ED/CCA measurements and is interpreted in dBm as follows. 00000000: −127 dBm (or below) 00000001: −126 dBm … 01111111: 0 dBm (or above) 1xxxxxxxx: Not used

RADIO FREQUENCY (RF) CONTROL

Description

Most of the RF control settings are determined by ON Semiconductor and shall not normally be changed by the user. However, there are some values that the programmer needs access to and are documented here.

For determining the RF transmit and receive fractional−N PLL integer and fractional words use the following formulas:

$$\text{Divider Value} = \frac{RF_{\text{carrier frequency}}}{F_{\text{crystal frequency}}} \quad \text{(eq. 1)}$$

$$\text{Integer Byte} = \text{round (Divider Value)} \quad \text{(eq. 2)}$$

$$\text{Fractional Word} = (\text{Divider Value} - \text{Integer Byte}) \cdot 262147 \quad \text{(eq. 3)}$$

$$F_{\text{crystal frequency}} = 32\text{MHz} \quad \text{(eq. 4)}$$

An example follows…

$$RF_{\text{carrier frequency}} = 2.401\text{GHz} \quad \text{(eq. 5)}$$

$$F_{\text{crystal frequnecy}} = 32\ \text{MHz} \quad \text{(eq. 6)}$$

$$\text{Divider Value} = \frac{RF_{\text{carrier frequency}}}{F_{\text{crystal frequency}}} = \frac{2.401\ \text{GHz}}{32\ \text{MHz}} = 75.03125 \quad \text{(eq. 7)}$$

$$\text{Integer Byte} = \text{round (Divider Value)} = \text{round}(75.03125) = 75 = 0\text{x}4\text{B} \quad \text{(eq. 8)}$$

$$\text{Fractional Word} = (75.03125 - 75) \cdot 262147 = 8192 = 0\text{x}2000 \quad \text{(eq. 9)}$$

Typically the ON Semiconductor software will handle calculating these values, but the programmer may want to implement a CW test mode and as such will need to be able to set these values. See the test mode control section for details on how to enable CW mode.

Table 70. REGISTERS

Function	Bits	Default	Type	Description
RF Tx Frequency Control Register: 0x4001_9000 RF transmit frequency control register setting the fractional and integer words for the fractional−N frequency synthesizer used to generate the RF carrier frequency. This is normally only useful for CW tone generation. See notes in description for how to calculate.				
Integer divide portion of transmit frequency for fractional−N PLL used to generate carrier frequency	[31:24]	0x0	R/W	See notes in description section for how to calculate
Fractional divide portion of transmit frequency for fractional−N PLL used to generate carrier frequency	[23:0]	0x0	R/W	See notes in description section for how to calculate
Function	**Bits**	**Default**	**Type**	**Description**
RF Rx Frequency Control Register: 0x4001_9004 RF receive frequency control register setting the fractional and integer words for the fractional−N frequency synthesizer used to generate the RF carrier frequency. See notes in description for how to calculate.				
Integer divide portion of receive frequency for fractional−N PLL used to generate carrier frequency	[31:24]	0x0	R/W	See notes in description section for how to calculate
Fractional divide portion of receive frequency for fractional−N PLL used to generate carrier frequency	[23:0]	0x0	R/W	See notes in description section for how to calculate
Function	**Bits**	**Default**	**Type**	**Description**
RF Rx Control Register: 0x4001_900C RF receive frequency control register setting the fractional and integer words for the fractional−N frequency synthesizer used to generate the RF carrier frequency. See notes in description for how to calculate.				
LNA gain mode	[0]	0x0	R/W	0 − Normal gain mode 1 − High gain mode (enabled to reach datasheet sensitivity limits with combined Tx/Rx match circuit, cost is about 600uA)
Function	**Bits**	**Default**	**Type**	**Description**
RF Tx Output Power Control Register: 0x4001_9010 Transmitter output power control.				
Tx output power control	[7:0]	0x0	R/W	0x00 − Minimum transmitter power … 0x14 − Maximum transmitter power
Function	**Bits**	**Default**	**Type**	**Description**
RF Tx Trim Register: 0x4001_9094 Transmitter trim parameter. This parameter is board sensitive and will vary from PCB design to PCB design. Ideally one code will be used for all boards of a given design. This trim adjusts the capacitance of an internal LC tank circuit that resonates the gate node of the internal NMOS PA transistor. Due to parasitic capacitance from gate to drain of this PA transistor, this value is influenced by the parasitics on the PCB. The system designer may want to adjust this trim during RF matching to fully optimize the design. To use this trim setting, the RF designer will adjust the RF match external components and this register value can be adjusted until the best possible RF match is obtained. For details contact the factory.				
Tx tune trim	[3:0]	0x0	R/W	See note above

TEST MODE CONTROL

Description

The NCS36510 contains a number of test modes that are only used by ON Semiconductor. However there are a few test modes that must be exposed to enable the engineer to design and characterize their system.

As mentioned in the DBG_TEST_EN section the DBG_TEST_EN pin must be driven high as a first prerequisite to get into test mode. The next step involves unlocking a test register and then setting the proper bits to enable and turn on the RF transmitter in CW mode.

After the test registers are unlocked there are two registers that must be set to enable a particular function. This locking and double redundant enable scheme is provided to help prevent rogue software from crashing the system.

There are two main functions that the designer may want to use in test mode: 1) The transmitter CW mode, and 2) Oscillator test mode (for clock trimming/measurement).

Table 71. REGISTERS

Function	Bits	Default	Type	Description
Test Mode Unlock Register: 0x4001_F000 If the DBG_TEST_EN in is high, and the unlock code is written to this register, then the remaining test mode registers can be written.				
Unlock register	[31:0]	0xA1313F21	R/W	Write a value of 0x5ECECODE to unlock, any other value is ignored and the test registers remain locked. The DBG_TEST_EN pin must be high before the unlock code is written or else the unlock code is ignored.
Function	**Bits**	**Default**	**Type**	**Description**
Test Mode Override Enable Register: 0x4001_F004 If the DBG_TEST_EN in is high, and the unlock code is already written (the test mode is unlocked), then setting the bits in this register pre−enable a particular test mode function as listed.				
Receiver chain pre−enable	[10]	0x0	R/W	1 − Pre−enabled 0 − Disabled
Frequency synthesizer power management pre−enable	[8]	0x0	R/W	1 − Pre−enabled 0 − Disabled
Transmitter power amplifier pre−enable	[7]	0x0	R/W	1 − Pre−enabled 0 − Disabled
Transmitter frequency synthesizer pre−enable	[6]	0x0	R/W	1 − Pre−enabled 0 − Disabled
Receiver frequency synthesizer pre−enable	[5]	0x0	R/W	1 − Pre−enabled 0 − Disabled
Pre−enable external 32.768 kHz crystal oscillator	[3]	0x0	R/W	1 − Pre−enabled 0 − Disabled
Pre−enable internal 32.768 kHz oscillator	[2]	0x0	R/W	1 − Pre−enabled 0 − Disabled
Pre−enable external 32 MHz crystal oscillator	[1]	0x0	R/W	1 − Pre−enabled 0 − Disabled
Pre−enable internal 32 MHz oscillator	[0]	0x0	R/W	1 − Pre−enabled 0 − Disabled
Function	**Bits**	**Default**	**Type**	**Description**
Test Mode Override Values Register: 0x4001_F008 If the DBG_TEST_EN in is high, the unlock code is already written (the test mode is unlocked), the corresponding override enable bits are already set, then setting the bits in this register enable a particular test mode function as listed.				
Receiver chain enable	[10]	0x0	R/W	1 − Enabled 0 − Disabled
Frequency synthesizer power management enable	[8]	0x0	R/W	1 − Enabled 0 − Disabled

Table 71. REGISTERS

Function	Bits	Default	Type	Description
Transmitter power amplifier enable	[7]	0x0	R/W	1 – Enabled 0 – Disabled
Transmitter frequency synthesizer enable	[6]	0x0	R/W	1 – Enabled 0 – Disabled
Receiver frequency synthesizer enable	[5]	0x0	R/W	1 – Enabled 0 – Disabled
Enable external 32.768 kHz crystal oscillator	[3]	0x0	R/W	1 – Enabled 0 – Disabled
Enable internal 32.768 kHz oscillator	[2]	0x0	R/W	1 – Enabled 0 – Disabled
Enable external 32 MHz crystal oscillator	[1]	0x0	R/W	1 – Enabled 0 – Disabled
Enable internal 32 MHz oscillator	[0]	0x0	R/W	1 – Enabled 0 – Disabled
Function	**Bits**	**Default**	**Type**	**Description**
Test Mode Digital Test Mux Register: 0x4001_F100 If the DBG_TEST_EN in is high and the unlock code is already written (the test mode is unlocked) then it's possible to route some internal signals through to DIO pads for testing.				
DIO[9] test signal mux	[27:24]	0x0	R/W	0011 – Internal 32.768 kHz oscillator clock
DIO[8] test signal mux	[23:20]	0x0	R/W	0010 – External 32 MHz crystal oscillator clock
DIO[7] test signal mux	[19:16]	0x0	R/W	0010 – Internal 32 MHz oscillator clock
DIO[6] test signal mux	[15:12]	0x0	R/W	0010 – External 32.768 kHz crystal oscillator clock
Function	**Bits**	**Default**	**Type**	**Description**
Test Mode DIO Re−router Register: 0x4001_F128 If the DBG_TEST_EN in is high and the unlock code is already written (the test mode is unlocked) then it's possible to re−route some internal signals through to other DIO pads which may be more convenient for testing. Please note that the signal continues to come out the original DIO despite being also re−routed to the new DIO.				
DIO[9] re−route	[17:15]	0x0	R/W	000 – No re−route 001 – DIO[4] 010 – DIO[5] 011 – DIO[6] 100 – DIO[7] 101 – DIO[8] 110 – DIO[10]
DIO[8] re−route	[14:12]	0x0	R/W	000 – No re−route 001 – DIO[4] 010 – DIO[5] 011 – DIO[6] 100 – DIO[7] 101 – DIO[9] 110 – DIO[10]

Table 71. REGISTERS

Function	Bits	Default	Type	Description
DIO[7] re−route	[11:9]	0x0	R/W	000 − No re−route 001 − DIO[4] 010 − DIO[5] 011 − DIO[6] 100 − DIO[8] 101 − DIO[9] 110 − DIO[10]
DIO[6] re−route	[8:6]	0x0	R/W	000 − No re−route 001 − DIO[4] 010 − DIO[5] 011 − DIO[7] 100 − DIO[8] 101 − DIO[9] 110 − DIO[10]

Example of how to enable CW mode

1. Drive DBG_TEST_EN pin high
2. Write unlock code in test mode unlock register
3. Write the pre−enable bits in the test mode override pre−enable register for bits 8, 7, 6, and 1. This will pre−enable the synthesizer power management unit (bit 8), pre−enable the transmitter power amplifier (bit 7), pre−enable the transmitter frequency synthesizer (bit 6), and pre−enable the external 32 MHz crystal oscillator (bit 1).
4. Write the enable bits in the test mode override values register for bits 8, 7, 6, and 1. This will enable the synthesizer power management unit (bit 8), enable the transmitter power amplifier (bit 7), enable the transmitter frequency synthesizer (bit 6), and enable the external 32 MHz crystal oscillator (bit 1).

The CW frequency will be 500 kHz lower than expected because the transmitter is effectively transmitting all zeros. So to compensate for this simply add 500 kHz to the frequency before calculating the integer and fractional words for the synthesizer.

To change CW mode frequencies, clear the bits in the test mode override enable register, update the integer and fractional words for the transmit synthesizer, and then set the bits in the test mode override enable register again to reactivate the transmitter in CW mode. Alternatively you can clear and set the override values register instead.

To set or change the CW frequency or the output power level, refer to the RF control register section.

Example of how to bring out the 32 MHz crystal oscillator clock

To allow measuring the 32 MHz crystal oscillator frequency test mode access is provided. While in the test mode the normal clock trim registers mentioned elsewhere in this document are used to make adjustments. The following example is how the programmer would make the 32 MHz crystal oscillator clock signal visible outside of the NCS36510 chip.

1. Drive DBG_TEST_EN pin high
2. Write unlock code in test mode unlock register
3. Write the pre−enable bits in the test mode override pre−enable register for bit 1. This will pre−enable the external 32 MHz crystal oscillator (bit 1).
4. Write the enable bits in the test mode override values register for bit 1. This will enable the external 32 MHz crystal oscillator (bit 1).
5. Write 0x0010 to the digital test mux register to drive the clock signal to DIO[8]
6. If desired, the clock signal can also be made available on other pins by using the DIO re−router documented above

Note that the same procedure with appropriate modifications can be used to pin out the other three oscillator signals including the internal 32.768 kHz, internal 32 MHz, and the external 32.768 kHz clocks.

POWER MANAGEMENT

Power Modes

There are four operational modes with varying power consumption. These modes can be used by application software to minimize power consumption of both dynamic and static power. Reduced functionality and retention occurs with each consecutive level of lower power mode selected.

NCS36510 will default to the highest power mode (run mode) upon startup. The lower power modes are all entered through software (WFI/WFE) commands. The software needs to configure which mode will be entered upon these commands. See Power Management Unit section.

Run Mode

In Run mode all digital systems are powered and running including external and/or internal oscillators. The processor is executing code. Various options exist to reduce power within this mode. Individual peripheral clocks can be gated based on configuration. You can also reduce the clock frequency of the system clock for power reduction. See the clock control section.

Sleep Mode

In Sleep Mode the processor clock (HCLK) is gated and no code is executing. The FLASH and RAMs are still powered. The processor waits for an interrupt from either the WIC or the NVIC. Once the interrupt is received, the processor goes back to Run Mode and code execution starts from the last known location.

Deep Sleep Mode

Deep Sleep Mode is the same as Sleep Mode except the FLASH is also powered down.

Coma Mode

In Coma Mode almost the entire digital system is powered down. Both the internal and external 32 MHz oscillators are powered down and the system is clocked by the 32.768 kHz derivative (either internal or external). All the Cortex−M3, peripheral, and trim registers will retain their values in Coma Mode. To get out of Coma Mode, the system waits for an interrupt from the WIC which can come from MAC symbol clock timer, the RTC, or a DIO. After an interrupt occurs the system will go to Run Mode and code execution will restart at the last known location before the Coma Mode was entered.

Table 72. POWER MODE TABLE

Power Mode	Digital	FLASH	Retention RAM A	Retention RAM B	32 MHz Clock
Run	ON	ON	ON	ON	ON
Sleep	ON	ON	ON	ON	ON
Deep Sleep	ON	OFF	ON	ON	ON
Coma	OFF	OFF	Programmable	Programmable	OFF

Power Management Unit (PMU)

Description

NCS36510 has an advanced PMU supporting two voltage supply modes: 3 V and 1 V, and four operating modes: Run, Sleep, Deep Sleep, and Coma.

Pre−Regulator

In 3 V mode the V3V voltage is pre − regulated to a voltage of about 1.1 V. The default voltage regulator is the linear regulator. The application software is responsible for controlling the switching regulator, including monitoring the V3V voltage with the internal voltage sensor. The switching regulator is only allowed if V3V > 2.6 V.

In 1 V mode, the pre − regulator is disabled. Connecting the V3V and V1VO/I pins to 1 V will automatically generate an internal logic signal and the system will be configured in 1 V mode.

Application diagrams of 1 V and 3 V modes are provided in the NCS36510 datasheet.

The internal voltage sensor can be used to monitor the power supply voltage. Another resource that can be used is the Under Voltage Indicator (UVI). In 3 V mode, this information can be used to decide to use the switching or the linear regulator as this decision is not automatic in hardware.

Under Voltage Indicator (UVI)

The under−voltage indicator tests for the condition where either the V1V switching regulator (3V mode) or the FVDDH switching regulator (1V mode) cannot supply sufficient current to support its load. It operates by counting over a 1 us interval the number of cycles of the 32 MHz system clock for which the comparator output in the selected regulator circuit is high (indicating a low instantaneous voltage). If the count exceeds the programmed threshold value, a low−voltage indicator is generated.

The low−voltage indicator is updated at 1 us intervals. The recommended threshold setting is 25 (decimal).

System of linear regulators

The pre − regulator voltage (V1V) is the input voltage (V1VI) for an array of internal linear regulators that are automatically enabled and disabled in the appropriate modes to minimize power consumption. These internal regulators supply the internal analog and digital blocks.

Embedded flash power supplies

To support the embedded FLASH two internal power supplies are implemented. One is the FVDDH, which is 1.8 V, and the other is FVDDL, which is 1.2 V.

In 3 V mode, the FVDDH is generated by a linear regulator powered by V3V.

In 1 V mode, a voltage multiplier is used to boost the V3V input voltage to the required level. The FVDDH voltage is split into two pins, FVDDHO (FVDDH Output), and FVDDHI (FVDDHI Input). Between these pins a power supply filter must be put on the application board to suppress the voltage multiplier noise. Throughout this document this voltage may be referred to as simply FVDDH. The user has the responsibility to filter this voltage as specified to obtain the published performance specifications.

Table 73. REGISTERS

Function	Bits	Default	Type	Description
PMU Control Register: 0x4001_D000 Control register for the Power Management Unit (PMU).				
UVI reset	[11]	0x1	R/W	0 – Not reset 1 – Reset Synchronous reset that should be applied after the UVI circuit is enabled and before the UVI results are used in the system
UVI input	[10]	0x0	R/W	0 – V1V supply 1 – FVDDH supply
UVI control	[9]	0x0	R/W	0 – Disabled 1 – Enabled
PMU behavior while debugging	[8]	0x0	R/W	0 – Normal power behavior when debugging 1 – When debugger connected, NCS36510 can only enter Deep Sleep mode and FLASH always remains powered up. And the 32 MHz oscillator derivative cannot be powered down.
V1V regulator selection – run, sleep, or deep sleep modes	[7]	0x0	R/W	0 – Linear regulator (LDO) 1 – Switching regulator
V1V regulator in coma mode	[6]	0x0	R/W	0 – Linear regulator (LDO) 1 – Switching regulator
Internal 32 MHz oscillator disable	[5]	0x0	R/W	0 – Enabled 1 – Disabled This bit will automatically get cleared when exiting Coma, or Deep Sleep modes of operation. This bit should be set by software after switching over to the external 32 MHz oscillator using the Oscillator Select bit in the Clock Control register.
Internal 32.768 kHz oscillator disable	[4]	0x0	R/W	0 – Enabled 1 – Disabled Hardware guarantees that this oscillator cannot be powered down if the external 32.768 kHz external crystal oscillator is already powered down.
External 32.768 kHz crystal oscillator disable	[3]	0x1	R/W	0 – Enabled 1 – Disabled Hardware guarantees that this oscillator cannot be powered down if the internal 32.768 kHz oscillator is already powered down.
RAM B coma mode retention enable	[2]	0x1	R/W	0 – Retention in coma mode enabled 1 – No retention in coma mode
RAM A coma mode retention enable	[1]	0x1	R/W	0 – Retention in coma mode enabled 1 – No retention in coma mode
Power mode to go to after WFI instruction	[0]	0x0	R/W	0 – Sleep or Deep Sleep Mode depending on system control register 1 – Coma Mode
Function	**Bits**	**Default**	**Type**	**Description**
PMU Status Register: 0x4001_D004 PMU status register, contains the UVI result as well as the battery detect bit.				
UVI result	[1]	0x0	RO	0 – Battery can support current power consumption 1 – Battery cannot support current power consumption
Voltage mode detect (battery detect)	[0]	0x0	RO	0 – 1V mode/battery detected 1 – 3V mode/battery detected

Table 73. REGISTERS

Function	Bits	Default	Type	Description
Function	**Bits**	**Default**	**Type**	**Description**
PMU UVI Time Base Register: 0x4001_D018 Time base and threshold for UVI circuit. A threshold of decimal 25 is recommended.				
UVI value	[13:8]	0x0	RO	Number of cycles over the previous 1 us interval where the UVI comparator was high, indicating a low voltage. This value is updated every 1 us.
UVI threshold	[5:0]	0x0	R/W	Threshold value is compared to the UVI value to determine if an overvoltage condition exists. If the UVI value is greater than or equal to the threshold, then an under voltage condition exists. The recommended setting is decimal 25.

EXTERNAL COMMUNICATION INTERFACES

Universal Asynchronous Receiver Transmitter (UART1 and UART2)

Description

NCS36510 implements two UART devices, UART 1 and UART 2.

UART 1 is a complete implementation of a 16550 UART. By configuring the crossbar UART1 can be set up as a complete 16550 UART, with all control wires.

UART 2 is a reduced functionality version of UART1. Specifically the crossbar can be setup to support transmit and receive along with clear to send and request to send. UART 1 can also be configured the same way if desired.

The UART baud rate generator produces timing strobes at the baud rate (for the transmitter) and at 16 times the selected baud rate (for the receiver). The receiver examines the incoming data and uses the first edge of the start bit to determine the bit timing. Bits can be received with up to half a bit time error and still be captured properly. Transmit and receive paths can be configured to use a single register for data or to use FIFOs.

The UART 1 and UART 2 FIFO buffers are 16 by 8 bits.

Interrupts are identified by an interrupt pending flag with more detailed interrupt status registers that can be read. The statuses include (in descending priority order): receiver line status, received data available, character timeout, transmitter holding register empty, and modem status.

Baud rate generation

The baud rate generator can be configured to generate a wide range of baud rates, depending on the system clock frequency and the divisor latch. The divisor latch and the system clock frequency are related to the baud rate by the following expression:

For example, to configure the UART for a baud rate of 9600 with a system clock frequency of 32 MHz, the Divisor Latch would need to be decimal 208 (0x00D0), so the most significant byte of the divisor latch would be programmed 0x00 and the least significant byte of the divisor latch would be programmed 0xD0. To access the divisor latch, the Divisor Latch Access Bit (DLAB) must be set in the LCR.

The divisor latch can be programmed values ranging 0x0001 to 0xFFFF.

The receiver samples data at 16 times the baud rate.

Table 74. REGISTERS

Function	Bits	Default	Type	Description
UART 1 Data Buffer Register: 0x4001_5000 **UART 2 Data Buffer Register: 0x4001_8000** Data buffer for UART.				
Receive/Transmit buffer	[7:0]	0x0	R/W	Read for received data Write for transmitting data
Function	**Bits**	**Default**	**Type**	**Description**
UART 1 Interrupt Enable Register: 0x4001_5004 **UART 2 Interrupt Enable Register: 0x4001_8004** UART interrupt configuration.				
Power down interrupt	[5]	0x0	R/W	1 – Interrupt enabled 0 – Interrupt disabled
Modem status interrupt	[3]	0x0	R/W	1 – Interrupt enabled 0 – Interrupt disabled

Table 74. REGISTERS

Function	Bits	Default	Type	Description
Receiver line status interrupt	[2]	0x0	R/W	1 – Interrupt enabled 0 – Interrupt disabled
Transmitter holding interrupt	[1]	0x0	R/W	1 – Interrupt enabled 0 – Interrupt disabled
Received data interrupt	[0]	0x0	R/W	1 – Interrupt enabled 0 – Interrupt disabled
Function	**Bits**	**Default**	**Type**	**Description**
UART 1 Interrupt Identification and FIFO Control Register: 0x4001_5008 **UART 2 Interrupt Identification and FIFO Control Register: 0x4001_8008** Dual purpose register. When reading the register returns interrupt identification and FIFO enabled status. When written allows FIFO control.				
FIFOs	[7:6]	0x0	RO	00 – FIFOs disabled 11 – FIFOs enabled
Interrupt identification	[3:1]	0x0	RO	011 – Receive line status (highest priority) 010 – Receive data available (2nd priority) 110 – Character timeout (2nd priority) 001 – Transmitter holding register empty (3rd priority) 000 – Modem status (4th priority)
Interrupt pending	[0]	0x0	RO	1 – Interrupt pending 0 – No interrupt pending
FIFO trigger level	[7:6]	n/a	WO	00 – 1 byte 01 – 4 bytes 10 – 8 bytes 11 – 14 bytes
Reset transmit FIFO	[2]	n/a	WO	Writing a 1 triggers a one cycle reset pulse
Reset receive FIFO	[1]	n/a	WO	Writing a 1 triggers a one cycle reset pulse
FIFO enable	[0]	n/a	WO	1 – FIFO enabled 0 – FIFO disabled
Function	**Bits**	**Default**	**Type**	**Description**
UART 1 Line Control Register: 0x4001_500C **UART 2 Line Control Register: 0x4001_800C** Control register to manage divisor latch, parity, stop bits, character length, etc.				
Address pointer type select	[7]	0x0	R/W	1 – Enable access to the divisor latch register at address 0x4400_0000 0 – Return register access to transmit and receive buffers at address 0x4400_0000
SOUT force low	[6]	0x0	R/W	1 – Force SOUT line to 0 0 – Normal operation
Parity control	[5:3]	0x0	R/W	xx0 – Parity disabled 001 – Odd parity 011 – Even parity 101 – Stick parity, checked as 1 111 – Stick parity, checked as 0
Number of stop bits	[2]	0x0	R/W	0 – 1 stop bit 1 – 2 stop bits
Character length	[1:0]	0x0	R/W	00 – 5 bits 01 – 6 bits 10 – 7 bits 11 – 8 bits
Function	**Bits**	**Default**	**Type**	**Description**
UART 1 Modem Control Register: 0x4001_5010 **UART 2 Modem Control Register: 0x4001_8010** Modem control register.				

Table 74. REGISTERS

Function	Bits	Default	Type	Description
Power down	[7]	0x0	R/W	1 – Enable power down 0 – Disable power down
Loopback mode	[4]	0x0	R/W	1 – Loopback mode selected 0 – Normal mode
Request to send – RTS	[1]	0x0	R/W	1 – RTSn = 0 0 – RTSn = 1
Data terminal ready – DTR	[0]	0x0	R/W	1 – DTRn = 0 0 – DTRn = 1
Function	**Bits**	**Default**	**Type**	**Description**
UART 1 Line Status Register: 0x4001_5014 **UART 2 Line Status Register: 0x4001_8014** Line status register.				
Receive FIFO error	[7]	0x0	RO	1 – Receive FIFO error 0 – No error
Transmitter empty	[6]	0x0	RO	1 – Transmitter empty 0 – Transmitter not empty
Transmitter holding register empty	[5]	0x0	RO	1 – Transmitter holding register empty 0 – Transmitter holding register not empty
Break interrupt	[4]	0x0	RO	1 – Break interrupt 0 – No break interrupt
Framing error	[3]	0x0	RO	1 – Framing error 0 – No framing error
Parity error	[2]	0x0	RO	1 – Parity error 0 – No parity error
Overrun error	[1]	0x0	RO	1 – Overrun error 0 – No overrun error
Received data ready	[0]	0x0	RO	1 – Received data ready 0 – No received data
Function	**Bits**	**Default**	**Type**	**Description**
UART 1 Modem Status Register: 0x4001_5018 **UART 2 Modem Status Register: 0x4001_8018** Modem status register.				
DCDn input state	[7]	0x0	RO	1 – High 0 – Low
RIn input state	[6]	0x0	RO	1 – High 0 – Low
DSRn input state	[5]	0x0	RO	1 – High 0 – Low
CTSn input state	[4]	0x0	RO	1 – High 0 – Low
DCDn has changed since last read of the status register	[3]	0x0	RO	1 – Yes 0 – No
RIn has changed since last read of the status register	[2]	0x0	RO	1 – Yes 0 – No
DSRn has changed since last read of status register	[1]	0x0	RO	1 – Yes 0 – No
CTSn has changed since last read of status register	[0]	0x0	RO	1 – Yes 0 – No
Function	**Bits**	**Default**	**Type**	**Description**

Table 74. REGISTERS

Function	Bits	Default	Type	Description
UART 1 Scratch Register: 0x4001_501C **UART 2 Scratch Register: 0x4001_801C** Modem scratch register.				
Scratch register for temporary data	[7:0]	0x0	R/W	
Function	**Bits**	**Default**	**Type**	**Description**
UART 1 Divisor Latch LSB Register: 0x4001_5020 **UART 2 Divisor Latch LSB Register: 0x4001_8020** LSB for baud rate generator.				
Divisor latch value LSB				8 bit value used to generate baud rate, 0x0001 to 0xFFFF are valid
Function	**Bits**	**Default**	**Type**	**Description**
UART 1 Divisor Latch MSB Register: 0x4001_5024 **UART 2 Divisor Latch MSB Register: 0x4001_8024** MSB for baud rate generator.				
Divisor latch value MSB				8 bit value used to generate baud rate, 0x0001 to 0xFFFF are valid

Master/Slave SPI Controller (SPI)

Description

NCS36510 implements two SPI Bus controllers, SPI1 and SPI2. SPI1 supports up to 4 slave selects. SPI2 supports 1.

The SPI bus controller can be configured under software control to be a master or slave device. The data is transmitted synchronously with the MOSI relative to the SCLK generated by the master device. The master also receives data on the MISO signal in a full duplex fashion. When the core is configured as a slave, the MISO signal is tri−stated to allow for multiple slaves to transmit data to the master when the slave's slave select control is enabled.

The SPI can operate in 8, 16, or 32 bit mode. The SPI FIFO is 16 x 32 bits.

SCLK is a divided version of the APB clock based on the MCU HCLK. The divider is programmable from 0x00 to 0xFF. The resulting SPI master clock is given by:

$$SCLK_{frequency} = \frac{PCLK_{frequency}}{2 \cdot (divider\ value + 1)} \quad \text{(eq. 10)}$$

In slave mode, the SPI master clock divider can only be as low as a factor of 8 due to several clock cycles of delay in internal clock domain crossing synchronizers.

Interrupts are generated for the following conditions: receive (RX) FIFO full, RX FIFO half full, RX FIFO not empty, transmit (TX) FIFO not full, TX FIFO half empty, TX FIFO empty, transfer error, and slave select synchronized to

APB clock.

SPI2 implements one slave select and is selected by the 0 bit in the slave select register.

Table 75. REGISTERS

Function	Bits	Default	Type	Description
SPI 1 Transmit Data Register: 0x4001_6000 **SPI 2 Transmit Data Register: 0x4001_9000** Transmit data buffer.				
SPI transmit data buffer, pad with leading zeros if using 8 or 16 bit transfers	[31:0]	0x0	WO	
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Receive Data Register: 0x4001_6004 **SPI 2 Receive Data Register: 0x4001_9004** Receive data buffer.				

Table 75. REGISTERS

Function	Bits	Default	Type	Description
SPI receive data buffer, padded with leading zeros if using 8 or 16 bit transfers	[31:0]	0x0	RO	
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 SCLK Divider Value Register: 0x4001_6008 **SPI 2 SCLK Divider Value Register: 0x4001_9008** Divider value for SCLK.				
SCLK divider value	[7:0]	0x0	R/W	0x00 to 0xFF
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Control Register: 0x4001_600C **SPI 2 Control Register: 0x4001_900C** SPI control register.				
SPI transaction word length	[7:6]	0x00	R/W	00 – 8 bits 01 – 16 bits 10 – 32 bits
SPI master mode enable	[5]	0x0	R/W	0 – Slave mode 1 – Master mode
CPOL – SCLK polarity	[4]	0x0	R/W	0 – SCLK low when idle (active high) 1 – SCLK high when idle (active low)
CPHA – SCLK phase	[3]	0x0	R/W	0 – First transmit occurs before first edge of SCLK 1 – First transmit is on the first edge of SCLK
Word endianness	[2]	0x0	R/W	0 – Little endian (LSB first) 1 – Big endian (MSB first)
Direction of SCLK for data sampling	[1]	0x0	R/W	0 – Sample incoming data on opposite edge of SCLK from when outgoing data is driven 1 – Sample incoming data on the same edge of SCLK when outgoing data is driven
SPI enable	[0]	0x0	R/W	0 – Disabled 1 – Enabled
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Status Register: 0x4001_6010 **SPI 2 Status Register: 0x4001_9010** SPI status register.				
Receive FIFO full	[7]	0x0	RO	0 – Not full 1 – Full
Receive FIFO watermark full	[6]	0x0	RO	0 – Watermark not yet reached 1 – Watermark reached
Receive FIFO empty	[5]	0x0	RO	0 – Not empty 1 – Empty
Transmit FIFO full	[4]	0x0	RO	0 – Not full 1 – Full
Transmit FIFO watermark full	[3]	0x0	RO	0 – Watermark not yet reached 1 – Watermark reached
Transmit FIFO empty	[2]	0x0	RO	0 – Not empty 1 – Empty
Transfer error	[1]	0x0	RO	0 – No error 1 – Error
Transfer in progress	[0]	0x0	RO	0 – No transfer in progress 1 – Transfer in progress

Table 75. REGISTERS

Function	Bits	Default	Type	Description
SPI 1 Slave Select Register: 0x4001_6014 **SPI 2 Slave Select Register: 0x4001_9014** SPI slave select control register.				
Burst mode	[4]	0x0	R/W	0 – Normal behavior 1 – Maintain SSN active between transfers as long as transmit FIFO is not empty
Slave select signals	[3:0]	0x00	R/W	1000 – SSN[3] 0100 – SSN[2] 0010 – SSN[1] 0001 – SSN[0]
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Slave Select Polarity Register: 0x4001_6018 **SPI 2 Slave Select Polarity Register: 0x4001_9018** SPI slave select polarity control register.				
Polarity control of SSN, bit aligned with slave select in slave select register [3:0]	[3:0]	0x00	R/W	In **master** mode 0 – Slave select active low 1 – Slave select active high In **slave** mode 0 – SSN is interpreted as an active low signal 1 – SSN is interpreted as an active high signal
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Interrupt Enable Register: 0x4001_601C **SPI 2 Interrupt Enable Register: 0x4001_901C** SPI interrupt configuration register.				
Enable receive FIFO full interrupt	[7]	0x0	R/W	0 – Disabled 1 – Enabled
Enable receive FIFO watermark hit interrupt	[6]	0x0	R/W	0 – Disabled 1 – Enabled
Enable receive FIFO empty interrupt	[5]	0x0	R/W	0 – Disabled 1 – Enabled
Enable transmit FIFO full interrupt	[4]	0x0	R/W	0 – Disabled 1 – Enabled
Enable transmit FIFO watermark hit interrupt	[3]	0x0	R/W	0 – Disabled 1 – Enabled
Enable transmit FIFO empty interrupt	[2]	0x0	R/W	0 – Disabled 1 – Enabled
Enable transfer error interrupt	[1]	0x0	R/W	0 – Disabled 1 – Enabled
Enable slave select conditionally inverted and synchronized to PCLK interrupt	[0]	0x0	R/W	0 – Disabled 1 – Enabled
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Interrupt Status Register: 0x4001_6020 **SPI 2 Interrupt Status Register: 0x4001_9020** SPI interrupt status register.				
Receive FIFO full interrupt	[7]	0x0	RO	0 – No interrupt 1 – Interrupt active
Receive FIFO watermark hit interrupt	[6]	0x0	RO	0 – No interrupt 1 – Interrupt active

Table 75. REGISTERS

Function	Bits	Default	Type	Description
Receive FIFO empty interrupt	[5]	0x0	RO	0 – No interrupt 1 – Interrupt active
Transmit FIFO full interrupt	[4]	0x0	RO	0 – No interrupt 1 – Interrupt active
Transmit FIFO watermark hit interrupt	[3]	0x0	RO	0 – No interrupt 1 – Interrupt active
Transmit FIFO empty interrupt	[2]	0x0	RO	0 – No interrupt 1 – Interrupt active
Transfer error interrupt	[1]	0x0	RO	0 – No interrupt 1 – Interrupt active
Slave select conditionally inverted and synchronized to PCLK interrupt	[0]	0x0	RO	0 – No interrupt 1 – Interrupt active
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Interrupt Clear Register: 0x4001_6024 **SPI 2 Interrupt Clear Register: 0x4001_9024** SPI interrupt clear register.				
Clear receive FIFO full interrupt	[7]	0x0	WO	
Clear receive FIFO watermark hit interrupt	[6]	0x0	WO	
Clear receive FIFO empty interrupt	[5]	0x0	WO	
Clear transmit FIFO full interrupt	[4]	0x0	WO	
Clear transmit FIFO watermark hit interrupt	[3]	0x0	WO	
Clear transmit FIFO empty interrupt	[2]	0x0	WO	
Clear transfer error interrupt	[1]	0x0	WO	
Clear slave select conditionally inverted and synchronized to PCLK interrupt	[0]	0x0	WO	
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Transmit FIFO Watermark Register: 0x4001_6028 **SPI 2 Transmit FIFO Watermark Register: 0x4001_9028** SPI transmit FIFO watermark value register.				
Transmit FIFO watermark (half full) value	[4:0]	0x0	R/W	
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Receive FIFO Watermark Register: 0x4001_602C **SPI 2 Receive FIFO Watermark Register: 0x4001_902C** SPI receive FIFO watermark value register.				
Receive FIFO watermark (half full) value	[4:0]	0x0	R/W	
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Transmit FIFO Fill Level Register: 0x4001_6030 **SPI 2 Transmit FIFO Fill Level Register: 0x4001_9030** SPI transmit FIFO fill level value register.				

Table 75. REGISTERS

Function	Bits	Default	Type	Description
Transmit FIFO fill value	[4:0]	0x0	R/W	
Function	**Bits**	**Default**	**Type**	**Description**
SPI 1 Receive FIFO Fill Level Register: 0x4001_6034 **SPI 2 Receive FIFO Fill Level Register: 0x4001_9034** SPI receive FIFO fill level value register.				
Receive FIFO fill value	[4:0]	0x0	R/W	

I2C Controller

Description

NCS36510 implements two I2C bus master interfaces, I2C1 and I2C2. Both are identical.

The I2C bus is an industry − standard two − wire (clock and data) serial communication bus. The I2C bus is a single master, multiple slave bus that uses a two−wire interface including a bidirectional clock line (SCL) and bidirectional data line (SDA). I2C specifies that the I2C master will initiate all read and write transactions, and that the I2C slave will respond to these requests.

The I2C command FIFO is 32 x 8 bits and the read FIFO is 16 x 8 bits.

SCL is derived from the internal APB PCLK as follows:

The I2C internal clock is always a factor of 4 faster than PCLK to allow for proper internal clock phasing and synchronization. This is where the factor of 4 in the denominator comes from.

Interrupts are generated when the read and/or command FIFOs are not empty and upon errors.

Table 76. REGISTERS

Function	Bits	Default	Type	Description
I2C1 Status Register: 0x4000_7000 **I2C2 Status Register: 0x4000_D000** I2C status register.				
Command FIFO full	[5]	0x0	RO	0 – Command FIFO not full 1 – Command FIFO full
Command FIFO overflow	[4]	0x0	RO	0 – Command FIFO not overflowed 1 – Command FIFO overflowed
Read data FIFO underflow	[3]	0x0	RO	0 – Read data FIFO not underflowed 1 – Read data FIFO overflowed
I2C bus error	[2]	0x0	RO	0 – No bus error 1 – I2C bus error occurred An I2C bus error is when the I2C master expects an ACK (NACK) but receives a NACK (ACK) condition on the I2C bus from the addressed I2C Slave instead. Clear on read
Read data ready	[1]	0x0	RO	0 – Read data not ready 1 – Read data ready
Command FIFO empty	[0]	0x0	RO	0 – Command FIFO not empty 1 – Command FIFO empty
Function	**Bits**	**Default**	**Type**	**Description**
I2C1 Read Data Register: 0x4000_7004 I2C2 Read Data Register: 0x4000_D004 I2C read data register.				
Read data register	[7:0]	0x0	RO	Data read from I2C bus
Function	**Bits**	**Default**	**Type**	**Description**
I2C1 Command Register: 0x4000_7008 I2C2 Command Register: 0x4000_D008 I2C command register.				

Table 76. REGISTERS

Function	Bits	Default	Type	Description
Command sequence	[7:0]	0x0	WO	0x00 – No−op 0x10 – Transmit one bit of logic ’0’ 0x11 – Transmit one bit of logic ’1’ 0x12 – Transmit one byte of data 0x13 – Receive one byte of data 0x14 – Transmit Stop command 0x15 – Transmit Start command 0x16 – Verify received ACK 0x17 – Verify received NACK
Function	**Bits**	**Default**	**Type**	**Description**
I2C1 Interrupt Enable Register: 0x4000_700C I2C2 Interrupt Enable Register: 0x4000_D00C I2C interrupt enable configuration register.				
I2C interrupt enable	[2]	0x0	R/W	0 – Disabled 1 – Enabled
Read data FIFO not empty interrupt enable	[1]	0x0	R/W	0 – Disabled 1 – Enabled
Command FIFO empty interrupt enable	[0]	0x0	R/W	0 – Disabled 1 – Enabled
Function	**Bits**	**Default**	**Type**	**Description**
I2C1 Control Register: 0x4000_7010 I2C2 Control Register: 0x4000_D010 I2C control register.				
I2C enable	[7]	0x0	R/W	0 – Disabled 1 – Enabled
I2C clock divider enable	[5]	0x0	R/W	0 – Disabled 1 – Enabled
I2C APB PCLK clock divider value, bottom 5 bits	[4:0]	0x0	R/W	SCL frequency is 1/4 the divided I2C system clock, see description section.
Function	**Bits**	**Default**	**Type**	**Description**
I2C1 Prescaler Value Register: 0x4000_7014 I2C2 Prescaler Value Register: 0x4000_D014 I2C prescaler value register.				
I2C APB PCLK clock divider value, upper 8 bits	[7:0]	0x0	R/W	SCL frequency is 1/4 the divided I2C system clock, see description section.

SECURITY FUNCTIONS

True Random Number Generator (TRNG)

Description

The True Random Number Generator (TRNG) is able to produce true 32−bit random numbers. The TRNG uses on−chip sources to generate a string of random bits. This is in contrast to pseudo−random number generators often used, which only look random but are in fact generated by a deterministic algorithm.

There are two sources of random numbers in the TRNG block. There is a MSL source and a White noise source.

A seed value can be set in the TRNG value register.

Table 77. TRUE RANDOM NUMBER GENERATOR (TRNG) REGISTERS

Function	Bits	Default	Type	Description
TRNG Value Register: 0x4001_1000 TRNG value register. Returns a 32 bit random number when read. Software can program a seed value by writing this register.				
TRNG value	[31:0]	0x0	R/W	Write 32 bit seed or read a 32 bit random number which is updated according to mode bits in the TRNG control register

Table 77. TRUE RANDOM NUMBER GENERATOR (TRNG) REGISTERS

Function	Bits	Default	Type	Description
TRNG Control Register: 0x4001_1004 TRNG control register.				
Metastable latch (MSL) TRNG enable	[4]	0x0	R/W	0 – Disabled 1 – Enabled
White noise TRNG enable	[3]	0x0	R/W	0 – Disabled 1 – Enabled
MSL speed control	[2]	0x0	R/W	0 – Fast mode 1 – Slow mode
Byte swap control	[1]	0x0	R/W	0 – 32−bit byte swap 1 – 64−bit byte swap
Mode register	[0]	0x0	R/W	0 – Updated on every rising edge of PCLK 1 – Updated on a read event
Function	**Bits**	**Default**	**Type**	**Description**
TRNG Write Buffer LSW Register: 0x4001_1008 TRNG least significant word byte swap write buffer register.				
Byte 3	[31:24]	0x0	R/W	
Byte 2	[23:16]	0x0	R/W	
Byte 1	[15:8]	0x0	R/W	
Byte 0	[7:0]	0x0	R/W	
Function	**Bits**	**Default**	**Type**	**Description**
TRNG Write Buffer MSW Register: 0x4001_100C TRNG most significant word byte swap write buffer register.				
Byte 7	[31:24]	0x0	R/W	
Byte 6	[23:16]	0x0	R/W	
Byte 5	[15:8]	0x0	R/W	
Byte 4	[7:0]	0x0	R/W	
Function	**Bits**	**Default**	**Type**	**Description**
TRNG Read Buffer LSW Register: 0x4001_1010 TRNG least significant word byte swap read buffer register.				
Byte 4 or 0	[31:24]	0x0	RO	If Byte Swap Control, returns byte 4. Else returns byte 0.
Byte 5 or 1	[23:16]	0x0	RO	If Byte Swap Control, returns byte 5. Else returns byte 1.
Byte 6 or 2	[15:8]	0x0	RO	If Byte Swap Control, returns byte 6. Else returns byte 2.
Byte 7 or 3	[7:0]	0x0	RO	If Byte Swap Control, returns byte 7. Else returns byte 3.
Function	**Bits**	**Default**	**Type**	**Description**
TRNG Read Buffer MSW Register: 0x4001_1014 TRNG most significant word byte swap read buffer register.				
Byte 0	[31:24]	0x0	RO	
Byte 1	[23:16]	0x0	RO	
Byte 2	[15:8]	0x0	RO	

Table 77. TRUE RANDOM NUMBER GENERATOR (TRNG) REGISTERS

Byte 3	[7:0]	0x0	RO	
Function	**Bits**	**Default**	**Type**	**Description**
TRNG MSL Value Register: 0x4001_1018 TRNG metastable latch (MSL) value register.				
MSL TRNG value	[31:0]	0x0	RO	
Function	**Bits**	**Default**	**Type**	**Description**
TRNG White Noise Value Register: 0x4001_101C TRNG white noise value register.				
White noise value	[31:0]	0x0	RO	

AES Accelerator

Descritpion

The AES accelerator provides hardware support for the encryption and decryption operations used in 802.15.4. To support security in 802.15.4, the use of Counter with CBC−MAC (CCM) is required. CCM is a combination of counter mode (CTR) and cipher block chaining (CBC). It can perform a ”Counter” (CTR) or a Cipher Block Chaining (CBC) encryption in 12 clocks for 128 bit encryption, or 16 clocks for 256 bit encryption. The definition of CCM mode encryption is documented in the NIST publication SP800−38C. Details of the implementation of the AES module can be found in federal information processing standard fips197.

In CTR mode, a software counter function creates a nonce and counter input to the encryption engine which is encrypted using the key programmed into the key register. The encrypted counter is then exored with the plaintext payload from the data registers. While AES is a block cipher, the use of an encrypted counter converts it to a stream cipher. The initial value of the counter and the nonce are sent un−encrypted as plaintext to initialize the counter and nonce on the receiving end. If the receiving end has the same key as the transmitting end, the plaintext payload can be recovered. This mode is used to provide privacy, but will not prove that the message has arrived unmodified.

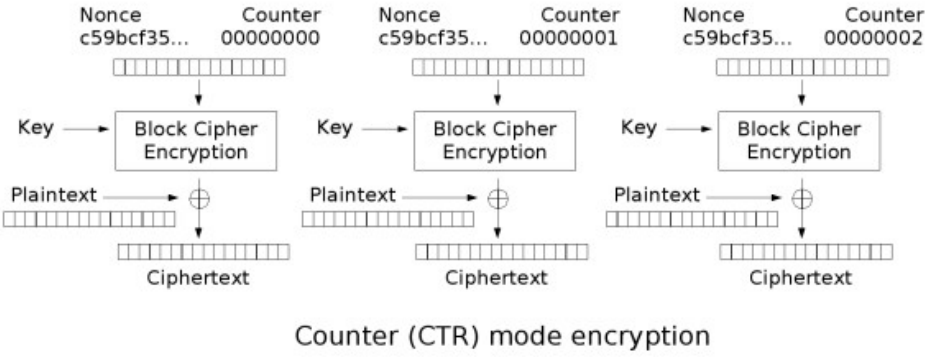


Figure 8.

In CBC mode (cipher block chaining), feedback is established in the encryption engine. A 128 bit payload block is exored with the previous AES engine output. This value is the plaintext input to the AES engine. The MAC initial value register allows a non−zero value to be supplied for the initial XOR with the payload, rather than all 0s, as would occur after clearing the CBC result register. The MAC Initial Value is used only once after it is written. The final encrypted result is an accumulation of all the previous results and is the MAC (message authentication code). Only a device that has knowledge of the key can reproduce the MAC value and authenticate that the message has not been modified. This mode will provide proof that the message has arrived unmodified, but does not provide privacy.

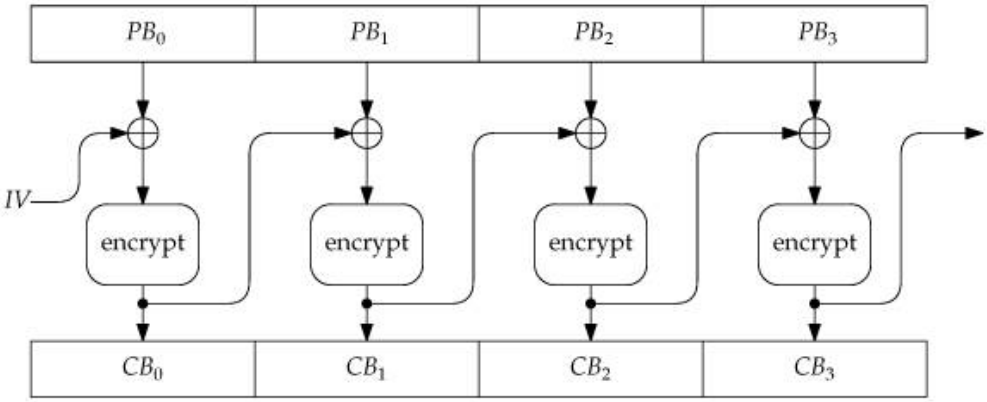


Figure 9.

The security of the system only depends on the secrecy of the key. To further this requirement, the key register is a write only register. A read of this register will return all "0's". It is not possible for a secondary process running on the processor to read the key value.

Table 78. REGISTERS

Function	Bits	Default	Type	Description
AES Key Register(s): **LSW = Least Significant Word, NSW = Next Significant Word, MSW = Most Significant Word** **AES Key 0 – LSW: 0x4001_6000** **AES Key 1 – NSW: 0x4001_6004** **AES Key 2 – NSW: 0x4001_6008** **AES Key 3 – NSW: 0x4001_600C** **AES Key 4 – NSW: 0x4001_6010** **AES Key 5 – NSW: 0x4001_6014** **AES Key 6 – NSW: 0x4001_6018** **AES Key 7 – MSW: 0x4001_601C** Write only AES key registers.				
Key 0 value	[31:0]	0x0	RO	Key bits [31:0]
Key 1 value	[31:0]	0x0	RO	Key bits [63:32]
Key 2 value	[31:0]	0x0	RO	Key bits [95:64]
Key 3 value	[31:0]	0x0	RO	Key bits [127:96]
Key 4 value	[31:0]	0x0	RO	Key bits [159:128]
Key 5 value	[31:0]	0x0	RO	Key bits [191:160]
Key 6 value	[31:0]	0x0	RO	Key bits [223:192]
Key 7 value	[31:0]	0x0	RO	Key bits [255:224]
Function	**Bits**	**Default**	**Type**	**Description**
AES Counter Mode Counter Value Register(s): **LSW = Least Significant Word, NSW = Next Significant Word, MSW = Most Significant Word** **AES Counter 0 – LSW: 0x4001_6020** **AES Counter 1 – NSW: 0x4001_6024** **AES Counter 2 – NSW: 0x4001_6028** **AES Counter 3 – NSW: 0x4001_602C** Counter mode counter values.				
Counter 0 counter mode value	[31:0]	0x0	R/W	Counter mode counter bits [31:0]
Counter 1 counter mode value	[31:0]	0x0	R/W	Counter mode counter bits [63:32]
Counter 2 counter mode value	[31:0]	0x0	R/W	Counter mode counter bits [95:64]

Table 78. REGISTERS

Function	Bits	Default	Type	Description
Counter 3 counter mode value	[31:0]	0x0	R/W	Counter mode counter bits [127:96]
AES Counter Result Register(s): **LSW = Least Significant Word, NSW = Next Significant Word, MSW = Most Significant Word** **AES Counter 0 – LSW: 0x4001_6030** **AES Counter 1 – NSW: 0x4001_6034** **AES Counter 2 – NSW: 0x4001_6038** **AES Counter 3 – NSW: 0x4001_603C** Counter values.				
Counter 0 value	[31:0]	0x0	R/W	Counter bits [31:0]
Counter 1 value	[31:0]	0x0	R/W	Counter bits [63:32]
Counter 2 value	[31:0]	0x0	R/W	Counter bits [95:64]
Counter 3 value	[31:0]	0x0	R/W	Counter bits [127:96]
Function	**Bits**	**Default**	**Type**	**Description**
AES Counter Result Register(s): **LSW = Least Significant Word, NSW = Next Significant Word, MSW = Most Significant Word** **AES Counter 0 – LSW: 0x4001_6030** **AES Counter 1 – NSW: 0x4001_6034** **AES Counter 2 – NSW: 0x4001_6038** **AES Counter 3 – NSW: 0x4001_603C** Counter CBC values.				
Counter 0 CBC value	[31:0]	0x0	R/W	Counter CBC bits [31:0]
Counter 1 CBC value	[31:0]	0x0	R/W	Counter CBC bits [63:32]
Counter 2 CBC value	[31:0]	0x0	R/W	Counter CBC bits [95:64]
Counter 3 CBC value	[31:0]	0x0	R/W	Counter CBC bits [127:96]
AES Control Register: 0x4001_6050 Control register for AES hardware accelerator.				
Interrupt clear	[2]	0x0	R/W	0 – No effect 1 – Clear interrupt
Clear CBC accumulator	[1]	0x0	R/W	0 – No effect 1 – Clear CBC accumulator
Start encryption	[0]	0x0	R/W	0 – No effect 1 – Start encryption process
Function	**Bits**	**Default**	**Type**	**Description**
AES Mode Register: 0x4001_6054 Mode register for AES hardware accelerator.				
Encryption key length	[3]	0x0	R/W	0 – 128 bits 1 – 256 bits
Interrupt mask	[2]	0x0	R/W	0 – Interrupt disabled 1 – Interrupt enabled
Mode	[0]	0x0	R/W	0 – Counter mode 1 – CBC mode
Function	**Bits**	**Default**	**Type**	**Description**
AES Status Register: 0x4001_6058 Status register for AES hardware accelerator.				
Status	[0]	0x0	RO	0 – Busy 1 – Complete
Function	**Bits**	**Default**	**Type**	**Description**

Table 78. REGISTERS

Function	Bits	Default	Type	Description
AES Initial Value Register(s): **LSW = Least Significant Word, NSW = Next Significant Word, MSW = Most Significant Word** **AES Initial Value 0 – LSW: 0x4001_605C** **AES Initial Value 1 – NSW: 0x4001_6060** **AES Initial Value 2 – NSW: 0x4001_6064** **AES Initial Value 3 – NSW: 0x4001_6068** Initial values for CBC counters.				
Counter 0 CBC initial value	[31:0]	0x0	R/W	Counter CBC bits [31:0]
Counter 1 CBC initial value	[31:0]	0x0	R/W	Counter CBC bits [63:32]
Counter 2 CBC initial value	[31:0]	0x0	R/W	Counter CBC bits [95:64]
Counter 3 CBC initial value	[31:0]	0x0	R/W	Counter CBC bits [127:96]
Function	**Bits**	**Default**	**Type**	**Description**
AES Data Register(s): **LSW = Least Significant Word, NSW = Next Significant Word, MSW = Most Significant Word** **AES Data 0 – LSW: 0x4001_6070** **AES Data 1 – NSW: 0x4001_6074** **AES Data 2 – NSW: 0x4001_6078** **AES Data 3 – NSW: 0x4001_607C** Data to be encrypted, 128 bits.				
Data 0 value	[31:0]	0x0	R/W	Data to encrypt, bits [31:0]
Data 1 value	[31:0]	0x0	R/W	Data to encrypt, bits [63:32]
Data 2 value	[31:0]	0x0	R/W	Data to encrypt, bits [95:64]
Data 3 value	[31:0]	0x0	R/W	Data to encrypt, bits [127:96]

Arm and Cortex are registered trademark of ECOS Consulting.

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

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

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

For additional information, please contact your local Sales Representative