SE1100

Specification

(Revision: 1.0)

File Name	**SE1100 Specification**		
S/N			
Revision	1.0		
Classification			
Category			
Submit		**Date**	
Review		**Date**	
Approved		**Date**	

Revision History

Date	Revision	Description	Submit
2013-06-25	1.0	Create	Robin

Content

Features:

*** High Performance, Low Power 8-bit Microcontroller**

*** Advanced RISC Architecture**

– 77 Powerful Instructions - Most Single Clock Cycle Execution

– Up to 1 MIPS Throughput at 4 MHz

*** Nonvolatile Program and Data Memories**

– 60 K Bytes of In-System Self-Programmable Flash, Endurance: 10,000 Write/Erase Cycles

– Optional Boot Code Section with Independent Lock Bits

In-System Programming by On-chip Boot Program

True Read-While-Write Operation

– 512 Bytes EEPROM, Endurance: 100,000 Write/Erase Cycles

– 2K Bytes Internal SRAM

– Programming Lock for Software Security

*** On-chip Debugging**

– Available through JTAG interface

*** Battery Management Features**

– Two, Three, or Four Cells in Series

– Deep Under-voltage Protection

– Over-current Protection (Charge and Discharge)

– Short-circuit Protection (Discharge)

– Integrated Cell Balancing FETs

– High Voltage Outputs to Drive Charge/Precharge/Discharge FETs

*** Peripheral Features**

– Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler

– Timer1 module: 16-bit timer/counter

– Timer2 module: 8-bit timer/counter with 8-bit period register (time base for PWM)

– Secondary oscillator clock option with Timer1

– Capture/Compare/PWM (CCP) modules;

Capture input: 16-bit, Compare: 16-bit, PWM output: PWM resolution is 1 to 10-bit

– Master Synchronous Serial Port (MSSP): I2C Master and Slave mode

– UART module

– Programmable Watchdog Timer

– 12-bit Voltage ADC, Eight External and Two Internal ADC Inputs

– High Resolution Coulomb Counter ADC for Current Measurements

*** Special Microcontroller Features**

– Power-on Reset

– On-chip Voltage Regulator

– External and Internal Interrupt Sources

– Four Sleep Modes: Idle, Power-save, Power-down, and Power-off

*** Packages**

– 48 pin LQFP

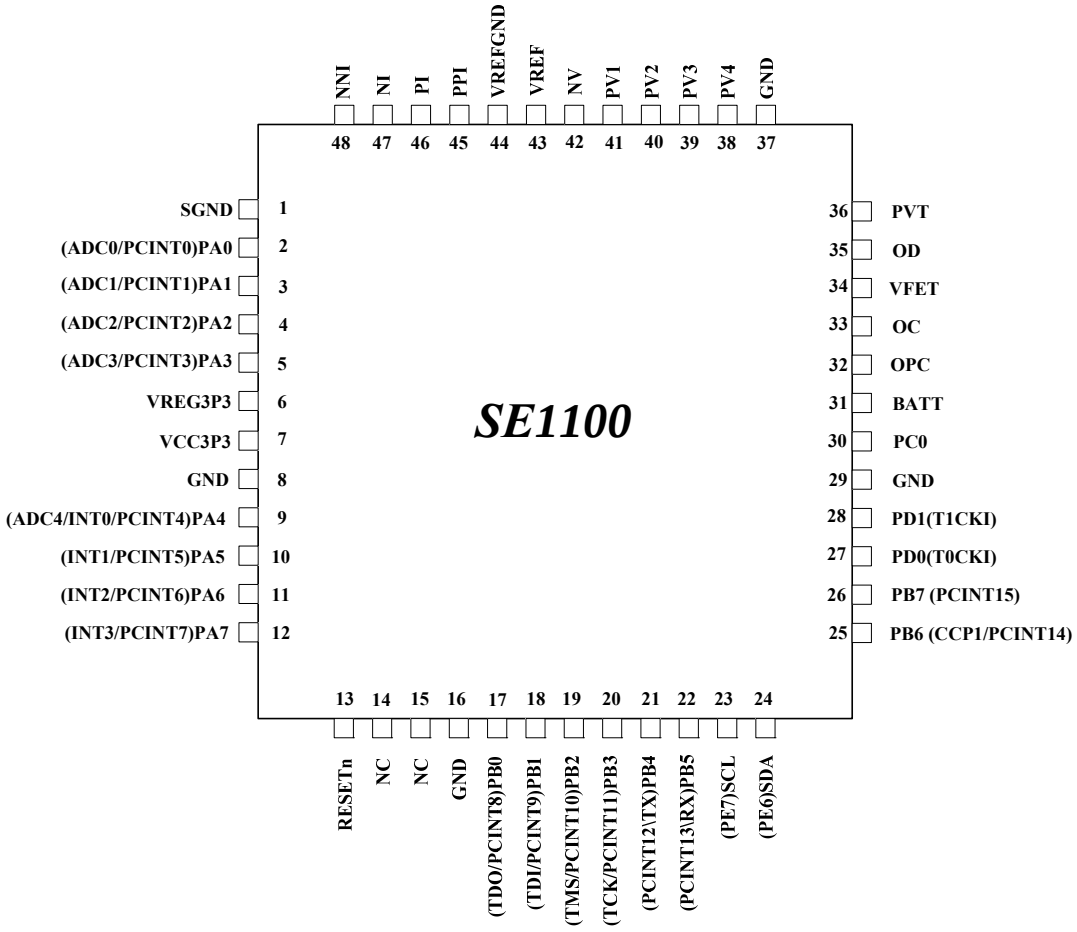
*** Operating Voltage: 4 - 25V**

*** Maximum Withstand Voltage (High-voltage pins): 28V**

*** Temperature Range: -40°C to 85°C**

1. Pin Configurations

Figure 1-1. Pinout SE1100.



1.1 Disclaimer

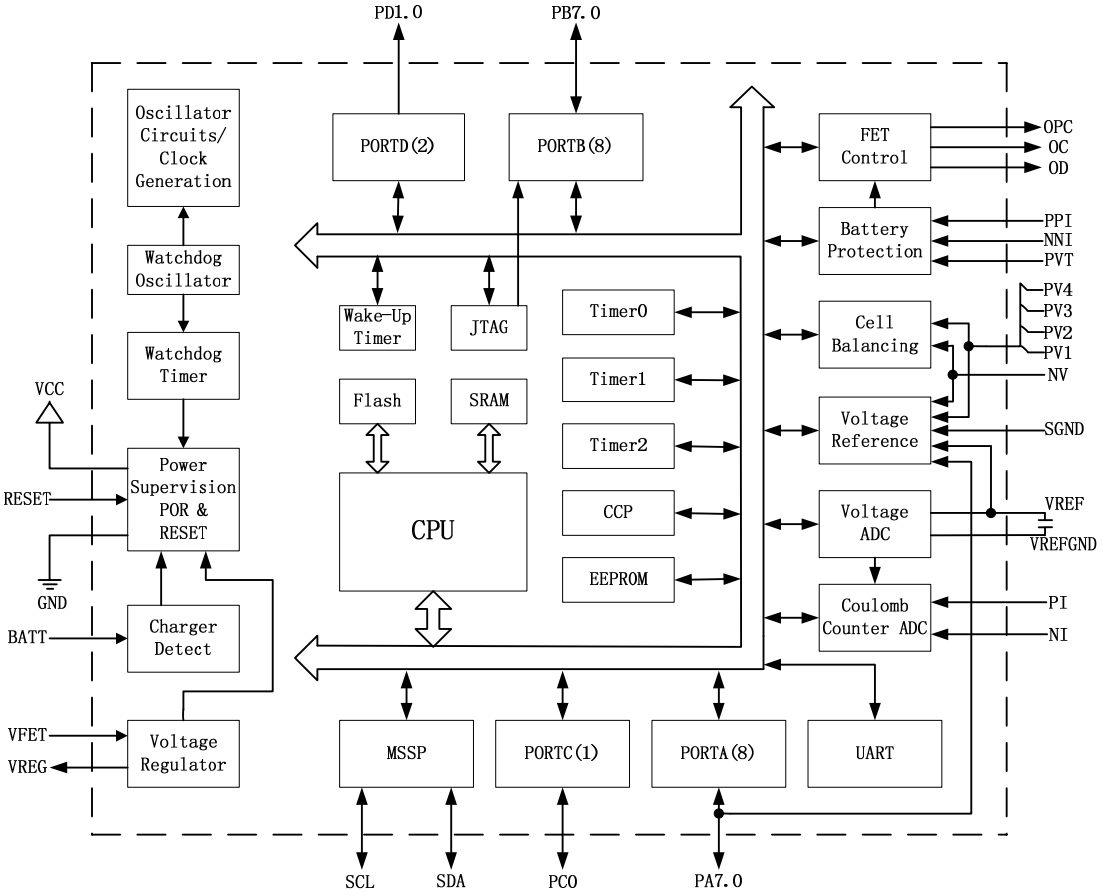
Typical values contained in this datasheet are based on simulations. Min and Max values will be available after the device is characterized.

2. Overview

The SE1100 is a low-power CMOS 8bit microcontroller based on the enhanced RISC architecture. By executing powerful instructions in four clock cycle, the SE1100 achieves throughputs approaching 1 MIPS at 4 MHz.

2.1 Block Diagram

Figure 2-1. Block Diagram



The SE1100 provides the following features: a Voltage Regulator, dedicated Battery Protection Circuitry, integrated cell balancing FETs, high-voltage analog front-end, and an MCU with two ADCs with On-chip voltage reference for battery fuel gauging.

The voltage regulator operates at a wide range of voltages, 4.0 - 25 volts. This voltage is regulated to a constant supply voltage of nominally 3.3 volts , for the integrated logic and analog functions.

The battery protection monitors the battery voltage and charge/discharge current to detect illegal conditions and protect the battery from these when required. The illegal conditions are deep

under-voltage during discharging, short-circuit during discharging and over-current during charging and discharging.

The integrated cell balancing FETs allow cell balancing algorithms to be implemented in software. The MCU provides the following features: 60K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512 bytes EEPROM, 2K byte SRAM, 18 general purpose I/O lines, 11 high-voltage I/O lines, a JTAG Interface for On-chip Debugging support and programming, three flexible Timer/Counters with PWM and compare modes, one Wake-up Timer, an SM-Bus compliant I2C module, internal and external interrupts, a 12-bit Sigma Delta ADC for voltage and temperature measurements, a high resolution Sigma Delta ADC for Coulomb Counting and instantaneous current measurements, a programmable Watchdog Timer with internal Oscillator, and four software selectable power saving modes. The SE1100 core combines a rich instruction set .The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The Idle mode stops the CPU while allowing the other chip function to continue functioning. The Power-save mode allows the voltage regulator, battery protection, regulator current detection, Watchdog Timer, and Wake-up Timer to operate, while disabling all other chip functions until the next Interrupt or Hardware Reset. In Power-off mode, only charge detector module run .

The device is manufactured using high voltage high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System, by a conventional non-volatile memory programmer or by an On-chip Boot program running on the SE1100 core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8bit RISC CPU with In-System Self-Programmable Flash, fuel gauging ADCs, dedicated battery protection circuitry, Cell Balancing FETs, and a voltage regulator on a monolithic chip, the SE1100 is a powerful microcontroller that provides a highly flexible and cost effective solution for Li-ion Smart Battery applications.

The SE1100 is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, and On-chip Debugger.

2.2 Pin Descriptions

2.2.1 VFET

High voltage supply pin. This pin is used as supply for the internal voltage regulator,. In addition the voltage level on this pin is monitored by the battery protection circuit, for deep-under-voltage protection.

2.2.2 VCC

Digital supply voltage. Normally connected to VREG.

2.2.3 VREG3P3

Output from the internal Voltage Regulator. Used for external decoupling to ensure stable regulator operation.

2.2.4 VREF

Internal Voltage Reference for external decoupling.

2.2.5 VREFGND

Ground for decoupling of Internal Voltage Reference.

2.2.6 GND

Ground

2.2.7 SGND

Signal ground pin, used as reference for Voltage-ADC conversions.

2.2.8 Port A (PA7:PA0)

PA3:PA0 serves as the analog inputs to the Voltage A/D Converter.

Port A also serves as a low-voltage 8bit bi-directional I/O port with internal pull-up resistors (selected for each bit). As inputs, Port A pins that are externally pulled low will source current if

the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port A also serves the functions of various special features of the SE1100 .

2.2.9 Port B (PB7:PB0)

Port B is a low-voltage 8bit bi-directional I/O port with internal pull-up resistors (selected for each bit). As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port B also serves the functions of various special features of the SE1100 .

2.2.10 Port C (PC0)

Port C is a high voltage Open Drain output port.

2.2.11 Port D (PD1:PD0)

Port D is a low-voltage 2-bit bi-directional I/O port with internal pull-up resistors (selected foreach bit). As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port D also serves the functions of various special features of the SE1100

2.2.12 SCL(PE7)

SMBUS clock, Open Drain bidirectional pin.

2.2.13 SDA(PE6)

SMBUS data, Open Drain bidirectional pin.

2.2.14 OC/OD/OPC

High voltage output to drive external Charge/Discharge/Pre-charge FETs.

2.2.15 PPI/NNI

Unfiltered positive/negative input from external current sense resistor, used by the battery protection circuit, for over-current and short-circuit detection.

2.2.16 PI/NI

Filtered positive/negative input from external current sense resistor, used to by the Coulomb Counter ADC to measure charge/discharge currents flowing in the battery pack.

2.2.17 NV/PV1/PV2/PV3/PV4

NV, PV1, PV2, PV3, and PV4 are the inputs for battery cells 1, 2, 3 and 4, used by the Voltage ADC to measure each cell voltage.

2.2.18 PVT

PVT defines the pull-up level for the OD output.

2.2.19 BATT

Input for detecting when a charger is connected. This pin also defines the pull-up level for OC and OPC outputs.

2.2.20 RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is 1Us.

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on https://www.nblsemi.com.

4. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler

documentation for more details.

5. MEMORY ORGANIZATION

There are three memory blocks in Enhanced MCU devices. These memory blocks are:

• Enhanced Flash Program Memory

• Data Memory

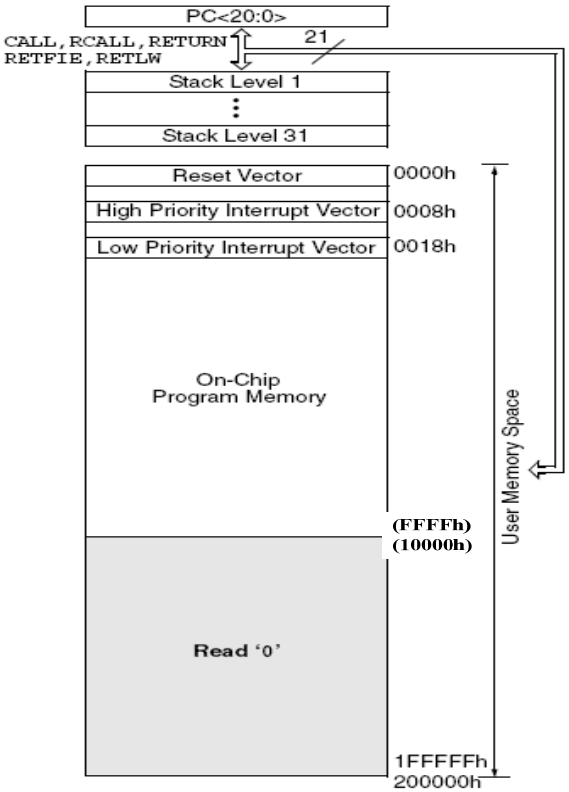
• EEPROM Data Memory

Data and program memory use separate busses, which allows concurrent access of these blocks. Additional detailed information on data EEPROM and Flash program memory is provided **"Data EEPROM Memory"** and **"Flash Program Memory"**, respectively.

5.1 Program Memory Organization

The SE1100 devices have a 21-bit program counter that is capable of addressing a 2-Mbyte program memory space. The Reset vector address is at 0000h and the interrupt vector addresses are at 0008h and 0018h.

FIGURE 5-1: PROGRAM MEMORY MAP AND STACK FOR SE1100



5.2 Return Address Stack

The return address stack allows any combination of up to 31 program calls and interrupts to occur. The PC (Program Counter) is pushed onto the stack when a PUSH, CALL or RCALL instruction is executed, or an interrupt is Acknowledged. The PC value is pulled off the stack on a RETURN, RETLW or a RETFIE instruction. PCLATU and PCLATH are not affected by any of the RETURN instructions. The stack operates as a 31-word by 21-bit stack memory and a 5-bit Stack Pointer register, with the Stack Pointer initialized to 00000b after all Resets. There is no RAM associated with Stack Pointer 00000b. This is only a Reset value. During a CALL type instruction, causing a push onto the stack, the Stack Pointer is first incremented and the RAM location pointed to by the Stack Pointer is written with the contents of the PC. During a RETURN type instruction, causing a pop from the stack, the contents of the RAM location indicated by the STKPTR are transferred to the PC and then the Stack Pointer is decremented. The stack space is not part of either program or data space. The Stack Pointer is readable and writable and the data on the top of the stack is readable and writable through SFR registers. Status bits indicate if the stack pointer is at or beyond the 31 levels provided.

5.2.1 TOP-OF-STACK ACCESS

The top of the stack is readable and writable. Three register locations, TOSU, TOSH and TOSL allow access to the contents of the stack location indicated by the STKPTR register. This allows users to implement a software stack, if necessary. After a CALL, RCALL or interrupt, the software can read the pushed value by reading the TOSU, TOSH and TOSL registers. These values can be placed on a user defined software stack. At return time, the software can replace the TOSU, TOSH and TOSL and do a return. The user should disable the global interrupt enable bits during this time to prevent inadvertent stack operations.

5.2.2 RETURN STACK POINTER (STKPTR)

The STKPTR register contains the Stack Pointer value, the STKFUL (Stack Full) status bit and the STKUNF (Stack Underflow) status bits. Register 4-1 shows the STKPTR register. The value of the Stack Pointer can be 0 through 31. The Stack Pointer increments when values are pushed onto the stack and decrements when values are popped off the stack. At Reset, the Stack Pointer value will be ‘0’. The user may read and write the Stack Pointer value. This feature can be used by a Real-Time Operating System for return stack maintenance. After the PC is pushed onto the stack 31 times (without popping any values off the stack), the STKFUL bit is set. The STKFUL bit can only be cleared in software or by a POR. The action that takes place when the stack becomes full depends on the state of the STVREN (Stack Overflow Reset Enable) configuration bit. If STVREN is set (default), the 31st push will push the (PC + 2) value onto the stack, set the STKFUL bit and reset the device. The STKFUL bit will remain set and the Stack Pointer will be set to ‘0’. If STVREN is cleared, the STKFUL bit will be set on the 31st push and the Stack Pointer will increment to 31. The 32nd push will overwrite the 31st push (and so on), while

STKPTR remains at 31. When the stack has been popped enough times to unload the stack, the next pop will return a value of zero to the PC and sets the STKUNF bit, while the stack pointer remains at '0'. The STKUNF bit will remain set until cleared in software or a POR occurs.

REGISTER 5-1: STKPTR: STACK POINTER REGISTER

Read/Wite (0XFFC) Bit	R/C-0	R/C-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	STKFUL	STKUNF	—	SP4	SP3	SP2	SP1	SP0	STKPTR
	bit 7							bit 0	

bit 7 **STKFUL**: Stack Full Flag bit

1 = Stack became full or overflowed

0 = Stack has not become full or overflowed

bit 6 **STKUNF**: Stack Underflow Flag bit

1 = Stack underflow occurred

0 = Stack underflow did not occur

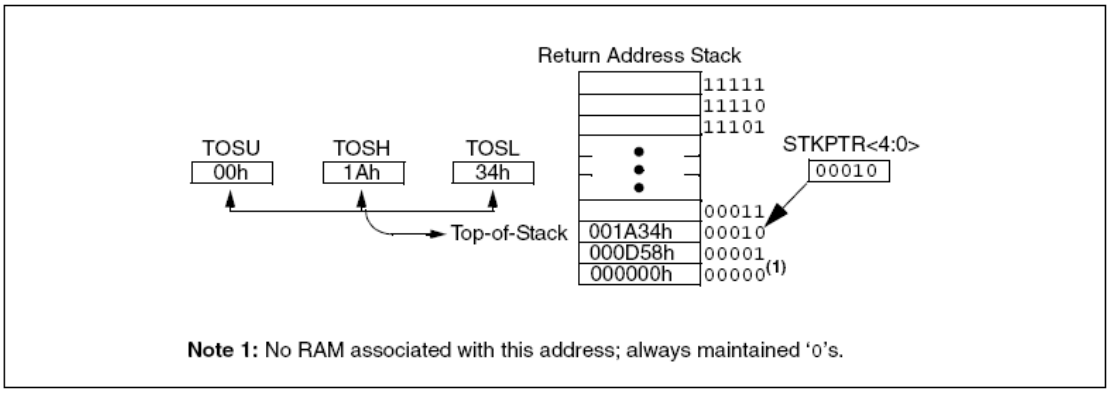
bit 5 **Unimplemented**: Read as '0'

bit 4-0 **SP4:SP0**: Stack Pointer Location bits

Note: Bit 7 and bit 6 need to be cleared following a stack underflow or a stack overflow.

Legend:
R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
-n = Value at POR '1' = Bit is set '0' = Bit is cleared C = Clearable bit

FIGURE 5-3: RETURN ADDRESS STACK AND ASSOCIATED REGISTERS



5.2.3 PUSH AND POP INSTRUCTIONS

Since the Top-of-Stack (TOS) is readable and writable, the ability to push values onto the stack and pull values off the stack, without disturbing normal program execution, is a desirable option. To push the current PC value onto the stack, a PUSH instruction can be executed. This will increment the Stack Pointer and load the current PC value onto the stack. TOSU, TOSH and TOSL can then be modified to place a return address on the stack.

The POP instruction discards the current TOS by decrementing the Stack Pointer. The previous value pushed onto the stack then becomes the TOS value.

5.2.4 STACK FULL/UNDERFLOW RESETS

These Resets are enabled by programming the STVREN configuration bit. When the STVREN bit is disabled, a full or underflow condition will set the appropriate STKFUL or STKUNF bit, but not cause a device Reset. When the STVREN bit is enabled, a full or underflow condition will set the appropriate STKFUL or STKUNF bit and then cause a device Reset. The STKFUL or STKUNF bits are only cleared by the user software or a POR.

5.3 Fast Register Stack

A “fast return” option is available for interrupts and calls. A fast register stack is provided for the Status, WREG and BSR registers and is only one layer in depth. The stack is not readable or writable and is loaded with the current value of the corresponding register when the processor vectors for an interrupt. The values in the fast register stack are then loaded back into the working registers if the FAST RETURN instruction is used to return from the interrupt.

A low or high priority interrupt source will push values into the stack registers. If both low and high priority interrupts are enabled, the stack registers cannot be used reliably for low priority interrupts. If a high priority interrupt occurs while servicing a low priority interrupt, the stack register values stored by the low priority interrupt will be overwritten.

If high priority interrupts are not disabled during low priority interrupts, users must save the key registers in software during a low priority interrupt.

If no interrupts are used, the fast register stack can be used to restore the Status, WREG and BSR registers at the end of a subroutine call. To use the fast register stack for a subroutine call, a FAST CALL instruction must be executed.
Example 5-1 shows a source code example that uses the fast register stack.

```
CALL SUB1, FAST     ;STATUS, WREG, BSR
                    ;SAVED IN FAST REGISTER
                    ;STACK
        •
        •
SUB1    •
        •
        •
    RETURN FAST     ;RESTORE VALUES SAVED
                    ;IN FAST REGISTER STACK
```

5.4 PCL, PCLATH and PCLATU

The Program Counter (PC) specifies the address of the instruction to fetch for execution. The PC is 21 bits wide. The low byte is called the PCL register. This register is readable and writable. The high byte is called the PCH register. This register contains the PC<15:8> bits and is not directly readable or writable. Updates to the PCH register may be performed through the PCLATH register. The upper byte is called PCU. This register contains the PC<20:16> bits and is not directly

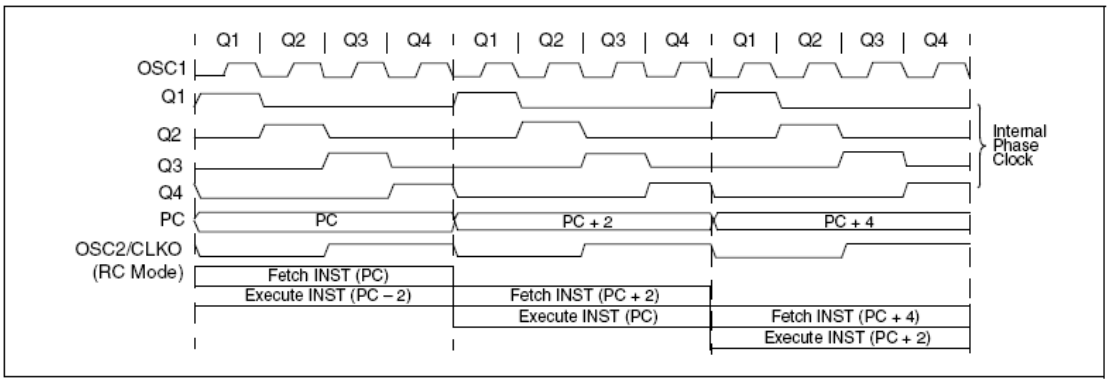
readable or writable. Updates to the PCU register may be performed through the PCLATU register.

The PC addresses bytes in the program memory. To prevent the PC from becoming misaligned with word instructions, the LSb of PCL is fixed to a value of '0'. The PC increments by 2 to address sequential instructions in the program memory.
The CALL, RCALL, GOTO and program branch instructions write to the program counter directly. For these instructions, the contents of PCLATH and PCLATU are not transferred to the program counter. The contents of PCLATH and PCLATU will be transferred to the program counter by an operation that writes PCL. Similarly, the upper two bytes of the program counter will be transferred to PCLATH and PCLATU by an operation that reads PCL.

5.5 Clocking Scheme/Instruction Cycle

The clock is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the Program Counter (PC) is incremented every Q1, the instruction is fetched from the program memory and latched into the instruction register in Q4. The instruction is decoded and executed during the following Q1 through Q4. The clocks and instruction execution flow are shown in Figure 5-4.

FIGURE 5-4: CLOCK/INSTRUCTION CYCLE



5.6 Instruction Flow/Pipelining

An "Instruction Cycle" consists of four Q cycles (Q1,Q2, Q3 and Q4). The instruction fetch and execute are pipelined such that fetch takes one instruction cycle, while decode and execute take another instruction cycle. However, due to the pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g., GOTO), two cycles are required to complete the instruction (Example 5-2). A fetch cycle begins with the Program Counter (PC) incrementing in Q1. In the execution cycle, the fetched instruction is latched into the "Instruction Register" (IR) in cycle Q1. This instruction is then decoded and executed during the Q2, Q3 and Q4 cycles. Data memory is read during Q2 (operand read) and written during Q4 (destination write).

5.7 Instructions in Program Memory

The program memory is addressed in bytes. Instructions are stored as two bytes or four bytes in program memory. The Least Significant Byte of an instruction word is always stored in a program memory location with an even address (LSB = 0). Figure 5-3 shows an example of how instruction words are stored in the program memory. To maintain alignment with instruction boundaries, the PC increments in steps of 2 and the LSB will always read '0' (see **Section 5.4 "PCL, PCLATH and PCLATU"**). The CALL and GOTO instructions have an absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1>, which accesses the desired byte address in program memory. Instruction #2 in Example 5-3 shows how the instruction "GOTO 000006h" is encoded in the program memory. Program branch instructions that encode a relative address offset operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions by which the PC will be offset.

EXAMPLE 5-2: INSTRUCTION PIPELINE FLOW

	TCY0	TCY1	TCY2	TCY3	TCY4	TCY5
1. MOVLW 55h	Fetch 1	Execute 1				
2. MOVWF PORTB		Fetch 2	Execute 2			
3. BRA SUB_1			Fetch 3	Execute 3		
4. BSF PORTA, BIT3 (Forced NOP)				Fetch 4	Flush	
5. Instruction @ address SUB_1					Fetch SUB_1	Execute SUB_1

Note: All instructions are single cycle, except for any program branches. These take two cycles, since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.

EXAMPLE 5-3: INSTRUCTIONS IN PROGRAM MEMORY

Instruction	Opcode	Memory	Address
—			000007h
MOVLW 055h	0E55h	55h	000008h
		0Eh	000009h
GOTO 000006h	0EF03h, 0F000h	03h	00000Ah
		0EFh	00000Bh
		00h	00000Ch
		0F0h	00000Dh
MOVFF 123h, 456h	0C123h, 0F456h	23h	00000Eh
		0C1h	00000Fh
		56h	000010h
		0F4h	000011h
—			000012h

5.8 Look-up Tables

Look-up tables are implemented two ways. These are:

- Computed GOTO
- Table Reads

5.8.1 COMPUTED GOTO

A computed GOTO is accomplished by adding an offset to the program counter (ADDWF PCL). A look-up table can be formed with an ADDWF PCL instruction and a group of RETLW 0xnn instructions. WREG is loaded with an offset into the table before executing a call to that table. The first instruction of the called routine is the ADDWF PCL instruction. The next instruction executed will be one of the RETLW 0xnn instructions that returns the value 0xnn to the calling function. The offset value (value in WREG) specifies the number of bytes that the program counter should advance. In this method, only one data byte may be stored in each instruction location and room on the return address stack is required.

5.8.2 TABLE READS/TABLE WRITES

A better method of storing data in program memory allows 2 bytes of data to be stored in each instruction location. Look-up table data may be stored as 2 bytes per program word by using table reads and writes. The Table Pointer (TBLPTR) specifies the byte address and the Table Latch (TABLAT) contains the data that is read from, or written to, program memory. Data is transferred to/from program memory, one byte at a time.

5.9 Data Memory Organization

The data memory is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4096 bytes of data memory. Figure 5-6 shows the data memory organization for the SE1100 devices. The data memory map is divided into as many as 16 banks that contain 256 bytes each. The lower 4 bits of the Bank Select Register (BSR<3:0>) select which bank will be accessed. The upper 4 bits for the BSR are not implemented. The data memory contains Special Function Registers(SFRs) and General Purpose Registers (GPRs). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratchpad operations in the user's application. The SFRs start at the last location of Bank 15 (FFFh) and grow downwards. GPRs start at the first location of Bank 0 and grow upwards. Any read of an unimplemented location will read as '0's. The entire data memory may be accessed directly or indirectly. Direct addressing may require the use of the BSR register. Indirect addressing requires the use of the File Select Register (FSR). Each FSR holds a 12-bit address value that can be used to access any location in the data memory map without banking. The instruction set and architecture allow operations across all banks. This may be accomplished by indirect addressing or by the use of the MOVFF instruction. The MOVFF instruction is a two-word/two-cycle instruction, that moves a value from one register to another. To ensure that commonly used registers (SFRs and select GPRs) can be accessed in a single cycle, regardless of the current BSR values, an Access Bank is implemented. A segment of Bank 0 and a segment of Bank 15 comprise the Access RAM.

5.9.1 GENERAL PURPOSE REGISTER FILE

The register file can be accessed either directly or indirectly. Indirect addressing operates through the File Select Registers (FSR). The operation of indirect addressing is shown in **Section 5.12 "Indirect Addressing, INDF and FSR Registers"**. Enhanced MCU devices may have banked memory in the GPR area. GPRs are not initialized by a Power-on Reset and are unchanged on all other Resets. Data RAM is available for use as GPR registers by all instructions. Bank 14 (E00h to E7Fh)and bank 15 (F80h to FFFh) contains SFRs. All other banks of data memory contain GPR registers, starting with Bank 0.

5.9.2 SPECIAL FUNCTION REGISTERS

The Special Function Registers (SFRs) are registers used by the CPU and peripheral modules for controlling the desired operation of the device. These registers are implemented as static RAM. A list of these registers is given in Table 5-1. The SFRs can be classified into two sets: those associated with the "core" function and those related to the peripheral functions. Those registers related to the "core" are described in this section, while those related to the operation of the peripheral features are described in the section of that peripheral feature. The SFRs are typically distributed among the peripherals whose functions they control. The unused SFR locations will be unimplemented and read as '0's. See Table 5-1 for addresses for the SFRs.

FIGURE 5-6: DATA MEMORY MAP FOR SE1100

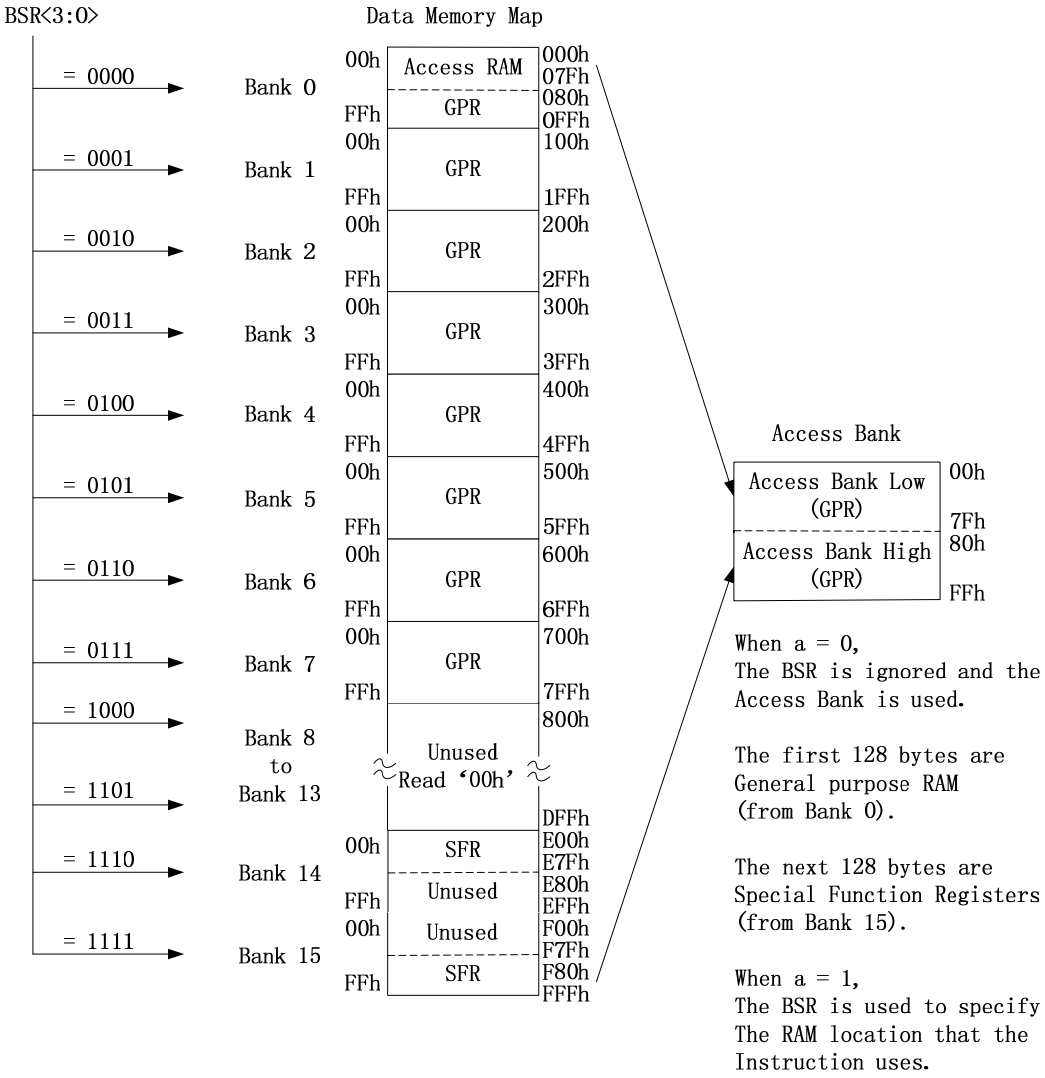


TABLE 5-1: SPECIAL FUNCTION REGISTER MAP

ADDR	NAME	ADDR	NAME	ADDR	NAME	ADDR	NAME
FFFh	TOSU	**FDFh**	INDF2	**FBFh**	CCPR1H	**F9Fh**	IPR1
FFEh	TOSH	**FDEh**	POSTINC2	**FBEh**	CCPR1L	**F9Eh**	PIR1
FFDh	TOSL	**FDDh**	POSTDEC2	**FBDh**	CCP1CON	**F9Dh**	PIE1
FFCh	STKPTR	**FDCh**	PREINC2	**FBCh**	-----	**F9Ch**	PCICR
FFBh	PCLATU	**FDBh**	PLUSW1	**FBBh**	-----	**F9Bh**	PCIFR
FFAh	PCLATH	**FDAh**	FSR2H	**FBAh**	-----	**F9Ah**	PCMSK1
FF9h	PCL	**FD9h**	FSR2L	**FB9h**	CLKPR	**F99h**	PCMSK0
FF8h	TBLPTRU	**FD8h**	STATUS	**FB8h**	OSICSR	**F98h**	-----
FF7h	TBLPTRH	**FD7h**	TMR0H	**FB7h**	SMCR	**F97h**	-----
FF6h	TBLPTRL	**FD6h**	TMR0L	**FB6h**	-----	**F96h**	DDRE
FF5h	TABLAT	**FD5h**	T0CON	**FB5h**	-----	**F95h**	DDRD
FF4h	PRODH	**FD4h**	CACHE_CFG	**FB4h**	-----	**F94h**	-----
FF3h	PRODL	**FD3h**	FBUSY	**FB3h**	-----	**F93h**	DDRB
FF2h	INTCON	**FD2h**	LVDCON	**FB2h**	-----	**F92h**	DDRA
FF1h	INTCON2	**FD1h**	WDTCON	**FB1h**	-----	**F91h**	-----

FF0h	INTCON3	**FD0h**	RCON	**FB0h**	-----	**F90h**	-----
FEFh	INDF0	**FCFh**	TMR1H	**FAFh**	SPBRG	**F8Fh**	-----
FEEh	POSTINC0	**FCEh**	TMR1L	**FAEh**	RCREG	**F8Eh**	-----
FEDh	POSTDEC0	**FCDh**	T1CON	**FADh**	TXREG	**F8Dh**	PINE
FECh	PREINC0	**FCCh**	TMR2	**FACh**	TXSTA	**F8Ch**	PIND
FEBh	PLUSW0	**FCBh**	PR2	**FABh**	RCSTA	**F8Bh**	------
FEAh	FSR0H	**FCAh**	T2CON	**FAAh**	EEADRH	**F8Ah**	PINB
FE9h	FSR0L	**FC9h**	SSPBUF	**FA9h**	EEADR	**F89h**	PINA
FE8h	WREG	**FC8h**	SSPADD	**FA8h**	EEDATA	**F88h**	-----
FE7h	INDF1	**FC7h**	SSPSTAT	**FA7h**	EECON2	**F87h**	-----
FE6h	POSTINC1	**FC6h**	SSPCON1	**FA6h**	EECON1	**F86h**	-----
FE5h	POSTDEC1	**FC5h**	SSPCON2	**FA5h**	-----	**F85h**	-----
FE4h	PREINC1	**FC4h**	FWREN	**FA4h**	-----	**F84h**	PORTE
FE3h	PLUSW1	**FC3h**	FWRDI	**FA3h**	-----	**F83h**	PORTD
FE2h	FSR1H	**FC2h**	FRDSR	**FA2h**	IPR2	**F82h**	PORTC
FE1h	FSR1L	**FC1h**	FSE	**FA1h**	PIR2	**F81h**	PORTB
FE0h	BSR	**FC0h**	FPP	**FA0h**	PIE2	**F80h**	PORTA

TABLE 5-2: SPECIAL FUNCTION REGISTER MAP

ADDR	NAME	ADDR	NAME	ADDR	NAME	ADDR	NAME
E7Fh	BPPLR	**E6Fh**	CADCSRC	**E5Fh**	VADCH	**E4F**	-----
E7Eh	BPCR	**E6Eh**	CADCSRB	**E5Eh**	VADCL	**E4E**	-----
E7Dh	BPHCTR	**E6Dh**	CADCSRA	**E5Dh**	DIDR0	**E4D**	-----
E7Ch	BPOCTR	**E6Ch**	CADICH	**E5Ch**	FOSCCAL	**E4C**	-----
E7Bh	BPSCTR	**E6Bh**	CADICL	**E5Bh**	-----	**E4B**	-----
E7Ah	BPCHCD	**E6Ah**	CADAC3	**E5Ah**	-----	**E4A**	-----
E79h	BPDHCD	**E69h**	CADAC2	**E59h**	-----	**E49**	-----
E78h	BPCOCD	**E68h**	CADAC1	**E58h**	-----	**E48**	-----
E77h	BPDOCD	**E67hh**	CADAC0	**E57h**	-----	**E47**	-----
E76h	BPSCD	**E66h**	CHGDCSR	**E56h**	-----	**E46**	-----
E75h	BPIFR	**E65h**	BGCSR	**E55h**	-----	**E45**	-----
E74h	BPIMSK	**E64h**	BGCRR	**E54h**	-----	**E44**	-----
E73h	CBCR	**E63h**	BGCCR	**E53h**	-----	**E43**	-----
E72h	FCSR	**E62h**	ROCR	**E52h**	-----	**E42**	-----
E71h	CADRDC	**E61h**	VADMUX	**E51h**	-----	**E41**	-----

E70h	CADRCC	**E60h**	VADCSR	**E50h**	-----	**E40**	-----

5.10 Access Bank

The Access Bank is an architectural enhancement that is very useful for C compiler code optimization. The techniques used by the C compiler are also useful for programs written in assembly. This data memory region can be used for:

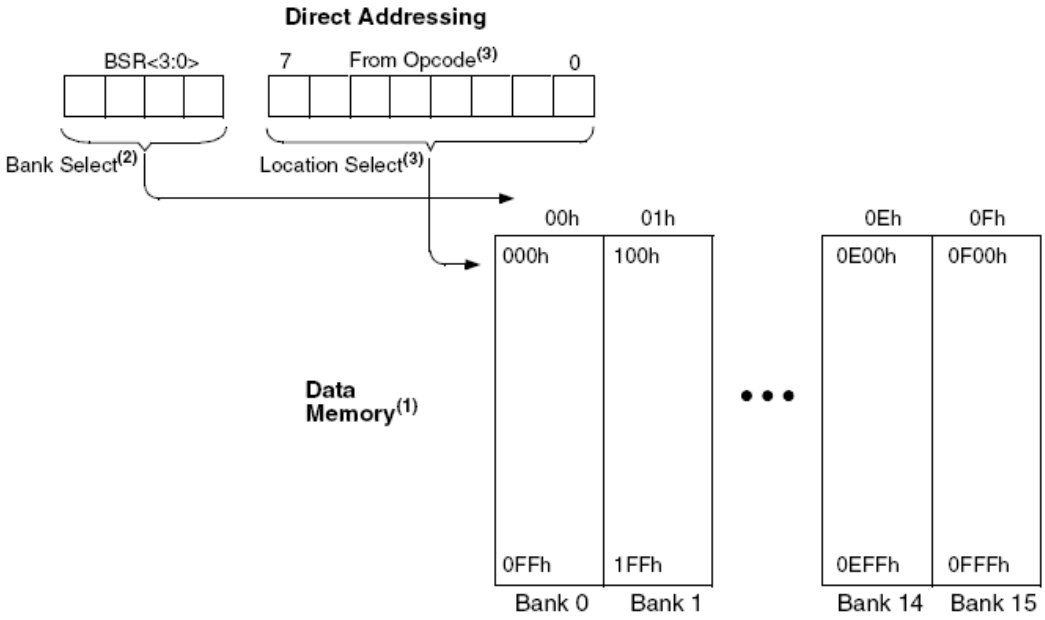
• Intermediate computational values
• Local variables of subroutines
• Faster context saving/switching of variables
• Common variables
• Faster evaluation/control of SFRs (no banking)

The Access Bank is comprised of the upper 128 bytes in Bank 15 (SFRs) and the lower 128 bytes in Bank 0. These two sections will be referred to as Access Bank High and Access Bank Low, respectively. Figure 5-6 indicates the Access Bank areas. A bit in the instruction word specifies if the operation is to occur in the bank specified by the BSR register or in the Access Bank. When forced in the Access Bank (a = 0), the last address in Access Bank Low is followed by the first address in Access Bank High. Access Bank High maps most of the Special Function Registers so that these registers can be accessed without any software overhead.

5.11 Bank Select Register (BSR)

The need for a large general purpose memory space dictates a RAM banking scheme. The data memory is partitioned into sixteen banks. When using direct addressing, the BSR should be configured for the desired bank. BSR<3:0> holds the upper 4 bits of the 12-bit RAM address. The BSR<7:4> bits will always read ‘0’s and writes will have no effect. A MOVLB instruction has been provided in the instruction set to assist in selecting banks. If the currently selected bank is not implemented, any read will return all ‘0’s and all writes are ignored. The Status register bits will be set/cleared as appropriate for the instruction performed. Each Bank extends up to FFh (256 bytes). All data memory is implemented as static RAM. A MOVFF instruction ignores the BSR since the 12-bit addresses are embedded into the instruction word.

FIGURE 5-7: DIRECT ADDRESSING



Note 1: For register file map detail, see Table 5-1,5-2.

2: The access bit of the instruction can be used to force an override of the selected bank (BSR<3:0>) to the registers of the Access Bank.

3: The MOVFF instruction embeds the entire 12-bit address in the instruction.

5.12 Indirect Addressing, INDF and FSR Registers

Indirect addressing is a mode of addressing data memory where the data memory address in the instruction is not fixed. A SFR register is used as a pointer to the data memory location that is to be read or written. Since this pointer is in RAM, the contents can be modified by the program. This can be useful for data tables in the data memory and for software stacks. Figure 5-8 shows the operation of indirect addressing. This shows the moving of the value to the data memory address specified by the value of the FSR register. Indirect addressing is possible by using one of the INDF registers. Any instruction using the INDF register actually accesses the register indicated by the File Select Register, FSR. Reading the INDF register itself, indirectly (FSR = 0), will read 00h. Writing to the INDF register indirectly, results in a no operation. The FSR register contains a 12-bit address which is shown in Figure 5-8. The INDFn (0 ≤n ≤2) register is not a physical register. Addressing INDFn actually addresses the register whose address is contained in the FSRn register (FSRn is a pointer). This is indirect addressing. Example 5-5 shows a simple use of indirect addressing to clear the RAM in Bank 1 (locations 100h-1FFh) in a minimum number of instructions.

EXAMPLE 5-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
          LFSR      FSR0,     100h    ;
NEXT      CLRF      POSTINC0          ; Clear INDF register & inc pointer
          BTFSS     FSR0H,    1       ; All done w/ Bank1?
          BRA       NEXT              ; No, clear next
CONTINUE                              ;
    :                                 ; Yes, continue
```

There are three indirect addressing registers. To address the entire data memory space (4096

bytes), these registers are 12 bits wide. To store the 12 bits of addressing information, two 8bit registers are required. These indirect addressing registers are:

1. FSR0: composed of FSR0H:FSR0L
2. FSR1: composed of FSR1H:FSR1L
3. FSR2: composed of FSR2H:FSR2L

In addition, there are registers INDF0, INDF1 and INDF2, which are not physically implemented. Reading or writing to these registers activates indirect addressing, with the value in the corresponding FSR register being the address of the data. If an instruction writes a value to INDF0, the value will be written to the address indicated by FSR0H:FSR0L. A read from INDF1 reads the data from the address indicated by FSR1H:FSR1L. INDFn can be used in code anywhere an operand can be used.

If INDF0, INDF1 or INDF2 are read indirectly via an FSR, all '0's are read (zero bit is set). Similarly, if INDF0, INDF1 or INDF2 are written to indirectly, the operation will be equivalent to a NOP instruction and the Status bits are not affected.

5.12.1 INDIRECT ADDRESSING OPERATION

Each FSR register has an INDF register associated with it, plus four additional register addresses. Performing an operation on one of these five registers determines how the FSR will be modified during indirect addressing.

• When data access is done to one of the five INDFn locations, the address selected will configure the FSRn register to:

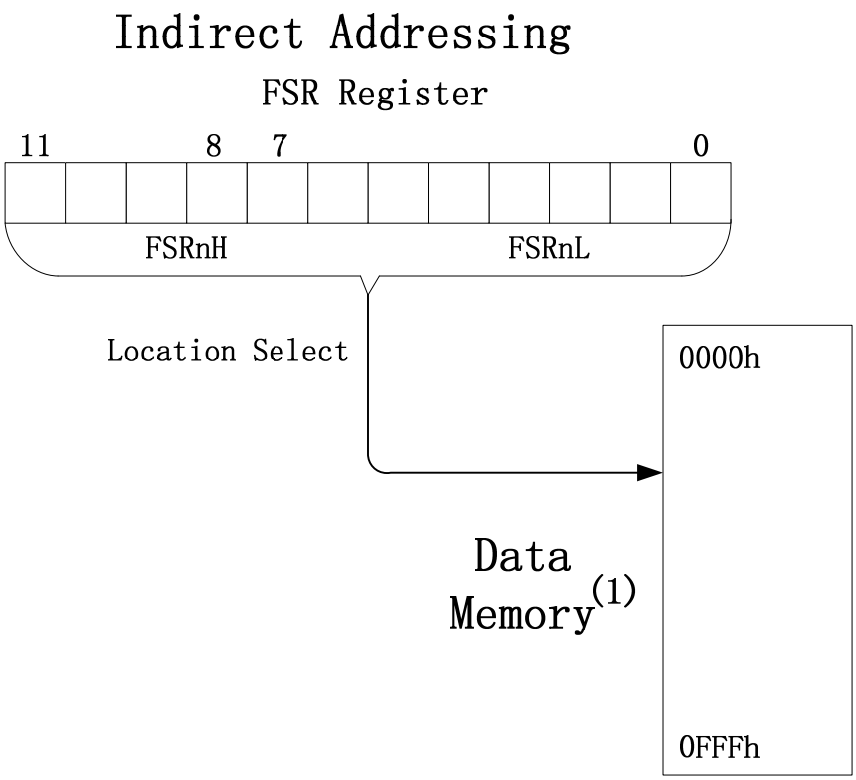
- Do nothing to FSRn after an indirect access (no change) – INDFn
- Auto-decrement FSRn after an indirect access (post-decrement) – POSTDECn
- Auto-increment FSRn after an indirect access (post-increment) – POSTINCn
- Auto-increment FSRn before an indirect access (pre-increment) – PREINCn
- Use the value in the WREG register as an offset to FSRn. Do not modify the value of the WREG or the FSRn register after an indirect access (no change) – PLUSWn

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the Status register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set. Incrementing or decrementing an FSR affects all 12 bits. That is, when FSRnL overflows from an increment, FSRnH will be incremented automatically. Adding these features allows the FSRn to be used as a software stack pointer in addition to its uses for table operations in data memory.

Each FSR has an address associated with it that performs an indexed indirect access. When a data access to this INDFn location (PLUSWn) occurs, the FSRn is configured to add the 2's complement value in the WREG register and the value in FSR to form the address before an indirect access. The FSR value is not changed. If an FSR register contains a value that indicates one of the INDFn, an indirect read will read 00h (zero bit is set), while an indirect write will be equivalent to a NOP (Status bits are not affected). If an indirect addressing operation is done where the target address is an FSRnH or FSRnL register, the write operation will dominate over

the pre- or post-increment/decrement functions.

FIGURE 5-8: INDIRECT ADDRESSING



Note 1: For register file map detail, see Table 5-1.

5.13 Status Register

The Status register, shown in Register 5-2, contains the arithmetic status of the ALU. The Status register can be the destination for any instruction, as with any other register. If the Status register is the destination for an instruction that affects the Z, DC, C, OV or N bits, then the write to these five bits is disabled. These bits are set or cleared according to the device logic. Therefore, the result of an instruction with the Status register as destination may be different than intended.

For example, CLRF STATUS will clear the upper three bits and set the Z bit. This leaves the Status register as 000u u1uu (where u = unchanged). It is recommended, therefore, that only BCF, BSF, SWAPF, MOVFF and MOVWF instructions are used to alter the Status register, because these instructions do not affect the Z, C, DC, OV or N bits from the Status register. For other instructions which do not affect the status bits.

REGISTER 5-2: STATUS REGISTER

Read/Wite (0XFD8) Bit	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	
	—	—	—	N	OV	Z	DC	C	STATUS
	bit 7							bit 0	

bit 10-5 **Unimplemented:** Read as '0'

bit 4 **N:** Negative bit

This bit is used for signed arithmetic (2's complement). It indicates whether the result of the ALU operation was negative (ALU MSb = 1).

1 = Result was negative

0 = Result was positive

bit 3 **OV:** Overflow bit

This bit is used for signed arithmetic (2's complement). It indicates an overflow of the 10-bit magnitude which causes the sign bit (bit 7) to change state.

1 = Overflow occurred for signed arithmetic (in this arithmetic operation)

0 = No overflow occurred

bit 2 **Z:** Zero bit

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC:** Digit Carry/Borrow bit

For ADDWF, ADDLW, SUBLW and SUBWF instructions:

1 = A carry-out from the 4th low-order bit of the result occurred

0 = No carry-out from the 4th low-order bit of the result

Note: For Borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRCF, RRNCF, RLCF and RLNCF) instructions, this bit is loaded with either bit 4 or bit 3 of the source register.

bit 0 **C:** Carry/Borrow bit

For ADDWF, ADDLW, SUBLW and SUBWF instructions:

1 = A carry-out from the Most Significant bit of the result occurred

0 = No carry-out from the Most Significant bit of the result occurred

Note: For Borrow, the polarity is reversed. A subtraction is executed by adding the 2's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high or low-order bit of the source register.

5.14 RCON Register

The Reset Control (RCON) register contains flag bits that allow differentiation between the sources of a device Reset. These flags include the TO, PD, POR, BOR and RI bits. This register is readable and writable.

Note 1: If the BOREN configuration bit is set, BOR is '1' on Power-on Reset. If the BOREN configuration bit is clear, BOR is unknown on Power-on Reset. The BOR status bit is a "don't care" and is not necessarily predictable if the brown out circuit is disabled (the BOREN configuration bit is clear). BOR must then be set by the user and checked on subsequent Resets to see if it is clear, indicating a brown-out has occurred.

2: It is recommended that the POR bit be set after a Power-on Reset has been detected, so that subsequent Power-on Resets may be detected.

REGISTER 5-3: RCON: RESET CONTROL REGISTER

BIT	7	6	5	4	3	2	1	0	
(0XFD0)	IPEN	CKOE	-	RI	TO	PD	POR	BOR	RCON
Read/Write	R/W	R/W	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	1	1	1	0	0	

bit 7 **IPEN:** Interrupt Priority Enable bit

1 = Enable priority levels on interrupts

0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6 **CKOE:** Clock Output Enable bit

When this bit is written to one, the CPU clock divided by 4 is output on the PB1 pin.

bit 5 **Unimplemented:** Read as '0'

bit 4 **RI:** RESET Instruction Flag bit

1 = The RESET instruction was not executed

0 = The RESET instruction was executed causing a device Reset

(must be set in software after a Brown-out Reset occurs)

bit 3 **TO:** Watchdog Time-out Flag bit

1 = After power-up, CLRWDT instruction or SLEEP instruction

0 = A WDT time-out occurred

bit 2 **PD:** Power-down Detection Flag bit

1 = After power-up or by the CLRWDT instruction

0 = By execution of the SLEEP instruction

bit 1 **POR:** Power-on Reset Status bit

1 = A Power-on Reset has not occurred

0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **BOR:** Brown-out Reset Status bit

1 = A Brown-out Reset has not occurred

0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

6. DATA EEPROM MEMORY

The data EEPROM is readable and writable during normal operation over the entire VDD range. The data memory is not directly mapped in the register file space. Instead, it is indirectly addressed through the Special Function Registers (SFR). There are four SFRs used to read and write the program and data EEPROM memory. These registers are:

- EECON1
- EECON2
- EEDATA
- EEADRL
- EEADRH

The EEPROM data memory allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and EEADR holds the address of the EEPROM location being accessed. The SE1100 devices have 512 bytes of data EEPROM with an address range from 00h to FFh. The EEPROM data memory is rated for high erase/write cycles. A byte write automatically erases the location and writes the new data (erase-before-write). The write time is controlled by an on-chip timer. The write time will vary with voltage and temperature, as well as from chip-to-chip. Please refer to the specifications for exact limits.

6.1 EEADRH and EEADRL Register

The address register can address up to a maximum of 512 bytes of data EEPROM.

6.2 EECON1 and EECON2 Registers

EECON1 is the control register for EEPROM memory accesses. EECON2 is not a physical register. Reading EECON2 will read all ‘0’s. The EECON2 register is used exclusively in the EEPROM write sequence. Control bits, RD and WR, initiate read and write operations, respectively. These bits cannot be cleared, only set, in software. They are cleared in hardware at the completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation. The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a RESETn Reset, or a WDT Time-out Reset, during normal operation. In these situations, the user can check the WRERR bit and rewrite the location. It is necessary to reload the data and address registers (EEDATA and EEADR) due to the Reset condition forcing the contents of the registers to zero.

Note: Interrupt flag bit, EEIF in the PIR2 register, is set when write is complete. It must be cleared

in software.

REGISTER 6-1: EECON1: EEPROM CONTROL REGISTER 1

Read/Write	R/W-x	R/W-x	R/W-0	R/W-0	R/W-x	R/W-0	R/S-x	R/S-x	
(0xFA6)	EEPGD	CFGS	SIGS	FREE	WRERR	WREN	$\overline{WR}$	$\overline{RD}$	EECON1
Bit	bit7								

bit 7 EEPGD: Flash Program or Data EEPROM Memory Select bit

1 = Access program Flash memory

0 = Access data EEPROM memory

bit 6 CFGS: Flash Program/Data EE or Configuration Select bit

1 = Access Configuration registers

0 = Access program Flash or data EEPROM memory

bit 5 SIGS: Signature Select bit

1 = Access Signature registers

0 = Access program Flash or data EEPROM memory

bit 4 FREE: Flash Row Erase Enable bit

1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)

0 = Perform write only

bit 3 WRERR: Write Error Flag bit

1 = A write operation is prematurely terminated (any MCLR or any WDT Reset during self-timed programming in normal operation)

0 = The write operation completed

bit 2 WREN: Write Enable bit

1 = Allows write cycles

0 = Inhibits write to the EEPROM or Flash memory

bit 1 WR: Write Control bit

1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle

(The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)

0 =Write cycle to the EEPROM is complete

bit 0 RD: Read Control bit

1 = Initiates an EEPROM read

(Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)

0 = Does not initiate an EEPROM read

6.3 Reading the Data EEPROM Memory

To read a data memory location, the user must write the address to the EEADR register, clear the EEPGD , CFGS and SIGS control bits (EECON1<7:5>) and then set control bit RD (EECON1<0>). The data is available in the very next instruction cycle of the EEDATA register;

therefore, it can be read by the next instruction. EEDATA will hold this value until another read operation or until it is written to by the user (during a write operation).

EXAMPLE 6-1: DATA EEPROM READ

```
MOVLW     DATA_EE_ADDR          ;
MOVWF     EEADR                 ; Data Memory Address to read
BCF       EECON1,    EEPGD      ; Point to Data memory
BCF       EECON1,    CFGS       ;
BCF       EECON1,    SIGS       ;
BSF       EECON1,    RD         ; EEPROM Read
MOVF      EEDATA,    W          ; W = EEDATA
```

6.4 Writing to the Data EEPROM Memory

To write an EEPROM data location, the address must first be written to the EEADR register and the data written to the EEDATA register. Then, the sequence in Example 6-2 must be followed to initiate the write cycle. The write will not initiate if the above sequence is not exactly followed (write 55h to EECON2, write 0AAh to EECON2, then set WR bit) for each byte. It is strongly recommended that interrupts be disabled during this code segment. Additionally, the WREN bit in EECON1 must be set to enable writes. This mechanism prevents accidental writes to data EEPROM due to unexpected code execution (i.e., runaway programs). The WREN bit should be kept clear at all times, except when updating the EEPROM. The WREN bit is not cleared by hardware.

After a write sequence has been initiated, clearing the WREN bit will not affect the current write cycle. The WR bit will be inhibited from being set unless the WREN bit is set. The WREN bit must be set on a previous instruction. Both WR and WREN cannot be set with the same instruction.

At the completion of the write cycle, the WR bit is cleared in hardware and the EEPROM Write Complete Interrupt Flag bit (EEIF) is set. The user may either enable this interrupt or roll this bit. EEIF must be cleared by software.

EXAMPLE 6-2: DATA EEPROM WRITE

```
            MOVLW     DATA_EE_ADDR          ;
            MOVWF     EEADR                 ; Data Memory Address to write
            MOVLW     DATA_EE_DATA          ;
            MOVWF     EEDATA                ; Data Memory Value to write
            BCF       EECON1,    EEPGD      ; Point to DATA memory
            BCF       EECON1,    CFGS       ; Access program FLASH or Data EEPROM memory
            BCF       EECON1,    SIGS       ;
            BSF       EECON1,    WREN       ; Enable writes

            BCF       INTCON,    GIE        ; Disable interrupts
Required    MOVLW     55h                   ;
Sequence    MOVWF     EECON2                ; Write 55h
            MOVLW     OAAH                  ;
            MOVWF     EECON2                ; Write aah

            BSF       EECON1,    WR         ; Set WR bit to begin write
            BSF       EECON1,    GIE        ; Enable interrupts


            .                               ; User code execution
            .
            .
            BCF       EECON1,    WREN       ; Disable writes on write complete (EEIF set)
```

6.5 Write Verify

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit. Generally, a write failure will be a bit which was written as a '1', but reads back as a '0' (due to leakage off the cell).

6.6 Protection Against Spurious Write

There are conditions when the device may not want to write to the data EEPROM memory. To protect against spurious EEPROM writes, various mechanisms have been built-in. On power-up, the WREN bit is cleared. Also, the Power-up Timer (72 ms duration) prevents EEPROM write. The write initiate sequence and the WREN bit together reduce the probability of an accidental write during brown-out, power glitch or software malfunction.

6.7 Using the Data EEPROM

The data EEPROM is a high-endurance, byte addressable array that has been optimized for the storage of frequently changing information (e.g., program variables or other data that are updated often). If this is not the case, an array refresh must be performed. For this reason, variables that change infrequently (such as constants, IDs, calibration, etc.) should be stored in Flash program memory. A simple data EEPROM refresh routine is shown in Example 6-3.

EXAMPLE 6-3: DATA EEPROM REFRESH ROUTINE

```
        CLRF    EEADR                   ; Start at address 0
        BCF     EECON1,   CFGS          ; Set for memory
        BCF     EECON1,   EEPGD         ; Set for Data EEPROM
        BCF     EECON1,   SIGS          ;
        BCF     INTCON    GIE           ; Disable interrupts
        BSF     EECON1,   WREN          ; Enable writes
Loop                                    ; Loop to refresh array
        BSF     EECON1,   RD            ; Read current address
        MOVLW   55h                     ;
        MOVWF   EECON2                  ; Write 55h
        MOVLW   OAAh                    ;
        MOVWF   EECON2                  ; Write AAh
        BSF     EECON1,   WR            ; Set WR bit to begin write
        BTFSC   EECON1,   WR            ; Wait for write to complete
        BRA     $-2

        INCFSZ  EEADR,    F             ; Increment address
        BRA     Loop                    ; Not zero, do it again

        BCF     EECON1,   WREN          ; Disable writes
        BSF     INTCON,   GIE           ; Enable interrupts
```

7. FLASH PROGRAM MEMORY

The Flash program memory is readable, writable and erasable during normal operation over the entire VDD range. A read from program memory is executed on one byte at a time. A write to program memory is executed on blocks of 8 bytes at a time. Program memory is erased in blocks of 4096 bytes at a time. A bulk erase operation may not be issued from user code. Writing or erasing program memory will cease instruction fetches until the operation is complete. The program memory cannot be accessed during the write or erase, therefore, code cannot execute. An internal programming timer terminates program memory writes and erases. A value written to program memory does not need to be a valid instruction. Executing a program memory location that forms an invalid instruction results in a NOP.

7.1 Table Reads and Table Writes

In order to read and write program memory, there are two operations that allow the processor to move bytes between the program memory space and the data RAM:

• Table Read (TBLRD)

• Table Write (TBLWT)

The program memory space is 16 bits wide, while the data RAM space is 8 bits wide. Table reads and table writes move data between these two memory spaces through an 8bit register (TABLAT). Table read operations retrieve data from program memory and place it into the data RAM space. Figure 7-1 shows the operation of a table read with program memory and data RAM. Table write operations store data from the data memory space into holding registers in program memory. The procedure to write the contents of the holding registers into program memory is detailed in Section 7.5 “Writing to Flash Program Memory”. Figure 7-2 shows the operation of a table write with program memory and data RAM. Table operations work with byte entities. A table block containing data, rather than program instructions, is not required to be word aligned. Therefore, a table block can start and end at any byte address. If a table write is being used to write executable code into program memory, program instructions will need to be word aligned.

FIGURE 7-1: TABLE READ OPERATION

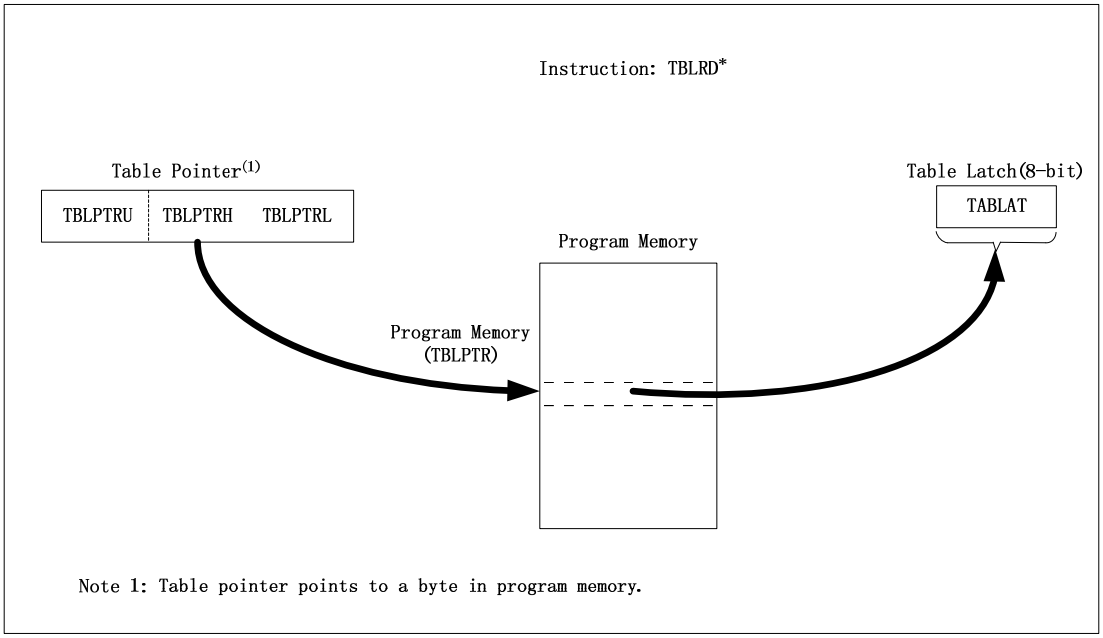
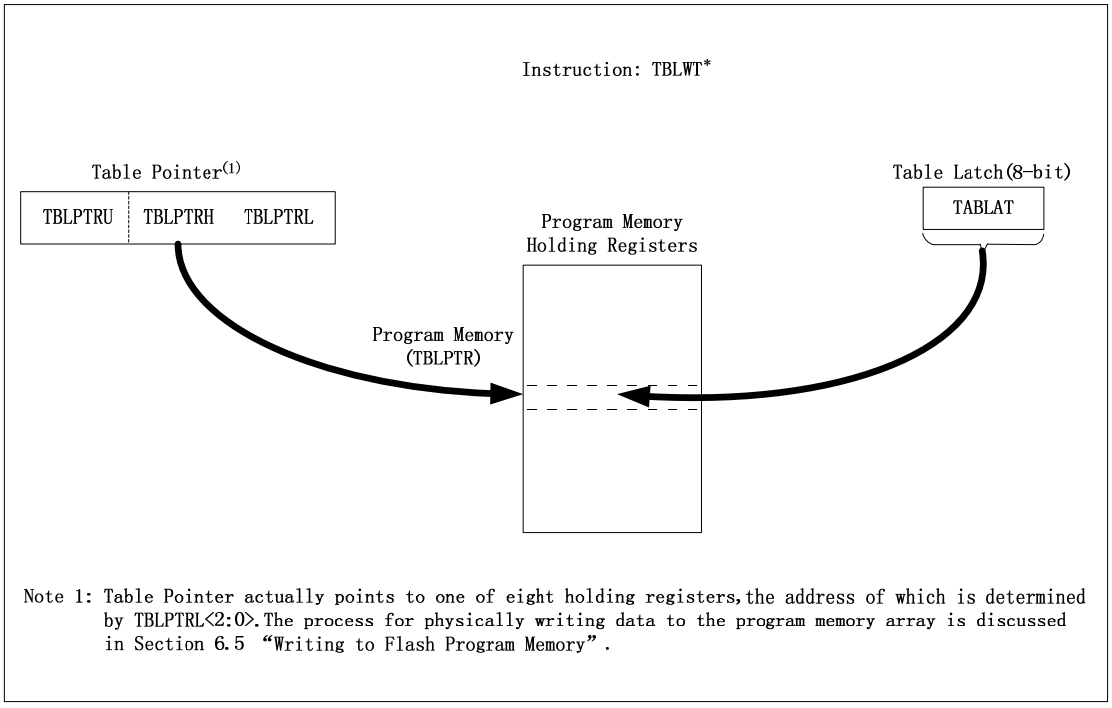


FIGURE 7-2: TABLE WRITE OPERATION



7.2 Control Registers

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include the:

- EECON1 register
- EECON2 register
- TABLAT register
- TBLPTR registers

7.2.1 EECON1 AND EECON2 REGISTERS

EECON1 is the control register for memory accesses. EECON2 is not a physical register. Reading EECON2 will read all '0's. The EECON2 register is used exclusively in the memory write and erase sequences. Control bit EEPGD determines if the access will be a program or data EEPROM memory access. When clear, any subsequent operations will operate on the data EEPROM memory. When set, any subsequent operations will operate on the program memory. Control bit CFGS determines if the access will be to the Configuration/Calibration registers or to program memory/data EEPROM memory. When set, subsequent operations will operate on Configuration registers regardless of EEPGD. When clear, memory selection access is determined by EEPGD. The FREE bit, when set, will allow a program memory erase operation. When the FREE bit is set, the erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled. The WREN bit, when set, will allow a write operation. On power-up, the WREN bit is clear. The WRERR bit is set when a write operation is interrupted by a MCLR Reset or a WDT Time-out Reset during normal operation In these situations, the user can check the WRERR bit and rewrite the location. It is necessary to reload the data and address registers (EEDATA and EEADR) due to Reset values of zero. Control bits, RD and WR, initiate read and write operations, respectively. These bits cannot be cleared, only set in software. They are cleared in hardware at the completion of the read or write operation. The inability to clear the WR bit in software prevents the accidental or premature termination of a write operation. The RD bit cannot be set when accessing program memory (EEPGD = 1).

Note: Interrupt flag bit, EEIF in the PIR2 register, is set when write is complete. It must be cleared in software.

REGISTER 7-1: EECON1: EEPROM CONTROL REGISTER 1

Read/Write (0xFA6) Bit	R/W-x	R/W-x	R/W-0	R/W-0	R/W-x	R/W-0	R/S-x	R/S-x	
	EEPGD	CFGS	SIGS	FREE	WRERR	WREN	$\overline{WR}$	$\overline{RD}$	EECON1
	bit7								

bit 7 EEPGD: Flash Program or Data EEPROM Memory Select bit

1 = Access program Flash memory

0 = Access data EEPROM memory

bit 6 CFGS: Flash Program/Data EE or Configuration Select bit

1 = Access Configuration registers

0 = Access program Flash or data EEPROM memory

bit 5 SIGS: Signature Select bit

1 = Access Signature registers

0 = Access program Flash or data EEPROM memory

bit 4 FREE: Flash Row Erase Enable bit

1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)

0 = Perform write only

bit 3 WRERR: Write Error Flag bit

1 = A write operation is prematurely terminated (any MCLR or any WDT Reset during

self-timed programming in normal operation)

0 = The write operation completed

bit 2 WREN: Write Enable bit

1 = Allows write cycles

0 = Inhibits write to the EEPROM or Flash memory

bit 1 WR: Write Control bit

1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle

(The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)

0 =Write cycle to the EEPROM is complete

bit 0 RD: Read Control bit

1 = Initiates an EEPROM read

(Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)

0 = Does not initiate an EEPROM read

7.2.2 TABLAT – TABLE LATCH REGISTER

The Table Latch (TABLAT) is an 8bit register mapped into the SFR space. The Table Latch is used to hold 8bit data during data transfers between program memory and data RAM.

7.2.3 TBLPTR – TABLE POINTER REGISTER

The Table Pointer (TBLPTR) addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22nd bit allows access to the device ID, the user ID and the configuration bits. The Table Pointer, TBLPTR, is used by the TBLRD and TBLWT instructions. These instructions can update the TBLPTR in one of four ways based on the table operation. These operations are shown in Table 7-1. These operations on the TBLPTR only affect the low-order 21 bits.

7.2.4 TABLE POINTER BOUNDARIES

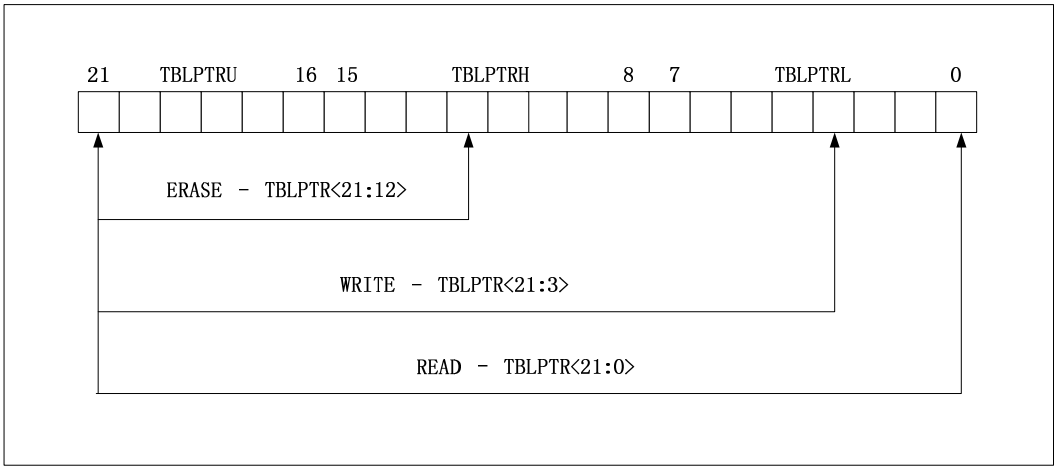
TBLPTR is used in reads, writes and erases of the Flash program memory. When a TBLRD is executed, all 22 bits of the Table Pointer determine which byte is read from program memory into TABLAT. When a TBLWT is executed, the three LSbs of the Table Pointer (TBLPTR<2:0>) determine which of the eight program memory holding registers is written to. When the timed write to program memory (long write) begins, the 19 MSbs of the Table Pointer, TBLPTR (TBLPTR<21:3>), will determine which program memory block of 8 bytes is written to. For more detail, see Section 7.5 “Writing to Flash Program Memory”. When an erase of program memory is

executed, the 16 MSbs of the Table Pointer (TBLPTR<21:12>) point to the 4096-byte block that will be erased. The Least Significant bits (TBLPTR<11:0>) are ignored. Figure 7-3 describes the relevant boundaries of TBLPTR based on Flash program memory operations.

TABLE 7-1: TABLE POINTER OPERATIONS WITH TBLRD AND TBLWT INSTRUCTIONS

Example	Operation on Table Pointer
TBLRD* TBLWT*	TBLPTR is not modified
TBLRD*+ TBLWT*+	TBLPTR is incremented after the read/write
TBLRD*- TBLWT*-	TBLPTR is decremented after the read/write
TBLRD+* TBLWT+*	TBLPTR is incremented before the read/write

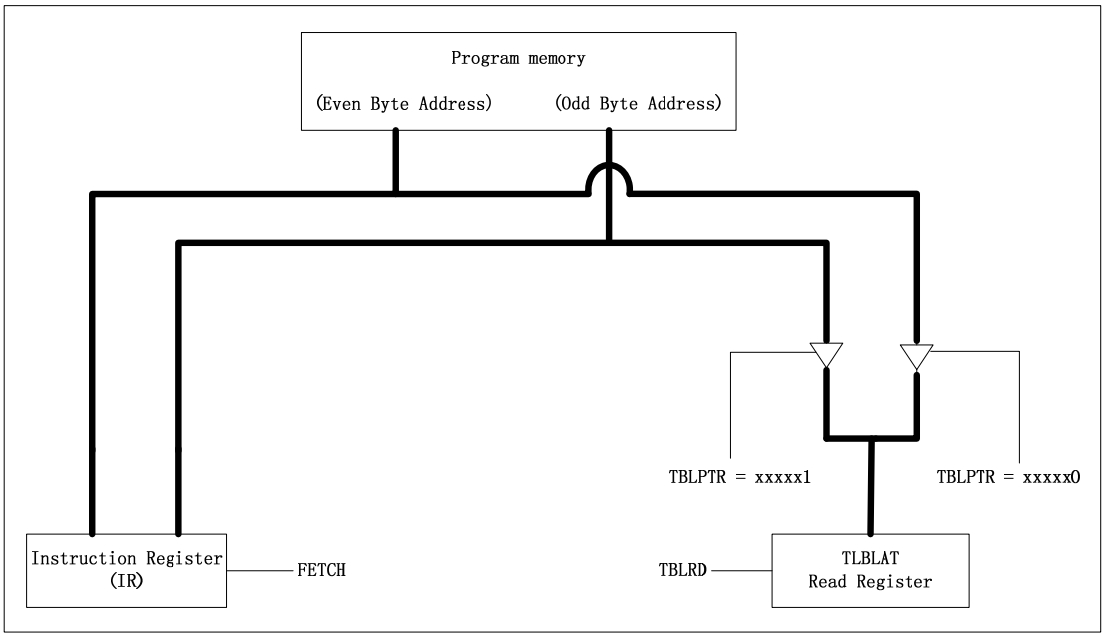
FIGURE 7-3: TABLE POINTER BOUNDARIES BASED ON OPERATION



7.3 Reading the Flash Program Memory

The TBLRD instruction is used to retrieve data from program memory and places it into data RAM. Table reads from program memory are performed one byte at a time. TBLPTR points to a byte address in program space. Executing TBLRD places the byte pointed to into TABLAT. In addition, TBLPTR can be modified automatically for the next table read operation. The internal program memory is typically organized by words. The Least Significant bit of the address selects between the high and low bytes of the word. Figure 7-4 shows the interface between the internal program memory and the TABLAT.

FIGURE 7-4: READS FROM FLASH PROGRAM MEMORY



EXAMPLE 7-1: READING A FLASH PROGRAM MEMORY WORD

```
        MOVLW   CODE_ADDR_UPPER         ; Load TBLPTR with the base
        MOVWF   TBLPTRU                 ; Address of the word
        MOVLW   CODE_ADDR_HIGH
        MOVWF   TBLPTRH
        MOVLW   CODE_ADDR_LOW
        MOVWF   TBLPTRL
READ_WORD
        TBLRD*+                         ; Read into TABLAT and increment
        MOVF    TABLAT,  W              ; Get data
        MOVWF   WORD_LSB
        TBLRD*+                         ; Read into TABLAT and increment
        MOVF    TABLAT,  W              ; Get data
        MOVWF   WORD_MSB
```

7.4 Erasing Flash Program Memory

The minimum erase block is 2048 words or 4096 bytes. When initiating an erase sequence from the microcontroller itself, a block of 4096 bytes of program memory is erased. The Most Significant 16 bits of the TBLPTR<21:12> point to the block being erased. TBLPTR<11:0> are ignored. The EECON1 register commands the erase operation. The EEPGD bit must be set to point to the Flash program memory. The WREN bit must be set to enable write operations. The FREE bit is set to select an erase operation. For protection, the write initiate sequence for

EECON2 must be used. A long write is necessary for erasing the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer.

7.4.1 FLASH PROGRAM MEMORY ERASE SEQUENCE

The sequence of events for erasing a block of internal program memory location is:

1. Load Table Pointer with address of row being erased.
2. Set the EECON1 register for the erase operation:
• set the EEPGD bit to point to program memory;
• clear the CFGS bit to access program memory;
• clear the SIGS bit to access program memory;
• set the WREN bit to enable writes;
• set the FREE bit to enable the erase.
3. Disable interrupts.
4. Write 55h to EECON2.
5. Write 0AAh to EECON2.
6. Set the WR bit. This will begin the row erase cycle.
7. The CPU will stall for duration of the erase (about 2 ms using internal timer).
8. Re-enable interrupts.

EXAMPLE 7-2: ERASING A FLASH PROGRAM MEMORY ROW

```
          MOVLW     upper (CODE_ADDR)    ; Load TBLPTR with the base
          MOVWF     TBLPTRU              ; Address of the memory block
          MOVLW     high (CODE_ADDR)
          MOVWF     TBLPTRH
          MOVLW     low (CODE_ADDR)
          MOVWF     TBLPTRL
ERASE_ROW
          BSF       EECON1,    EEPGD     ; Point to FLASH program memory
          BCF       EECON1,    CFGS      ; Access FLASH program memory
          BCF       EECON1,    SIGS
          BSF       EECON1,    WREN      ; Enable write to memory
          BSF       EECON1,    FREE      ; Enable Row Erase operation
          BCF       INTCON,    GIE       ; Disable interrupts
          MOVLW     55h                  ;
          MOVWF     EECON2               ; Write 55h
Required  MOVLW     0AAH                 ;
Sequence  MOVWF     EECON2               ; Write 0aah
          BSF       EECON1,    WR        ; Start erase (CPU stall)
          NOP                            ; NOP needed for proper code execution
          BSF       INTCON,    GIE       ; Re-enable interrupts
```

7.5 Writing to Flash Program Memory

The minimum programming block is 4 words or 8 bytes. Word or byte programming is not supported. Table writes are used internally to load the holding registers needed to program the

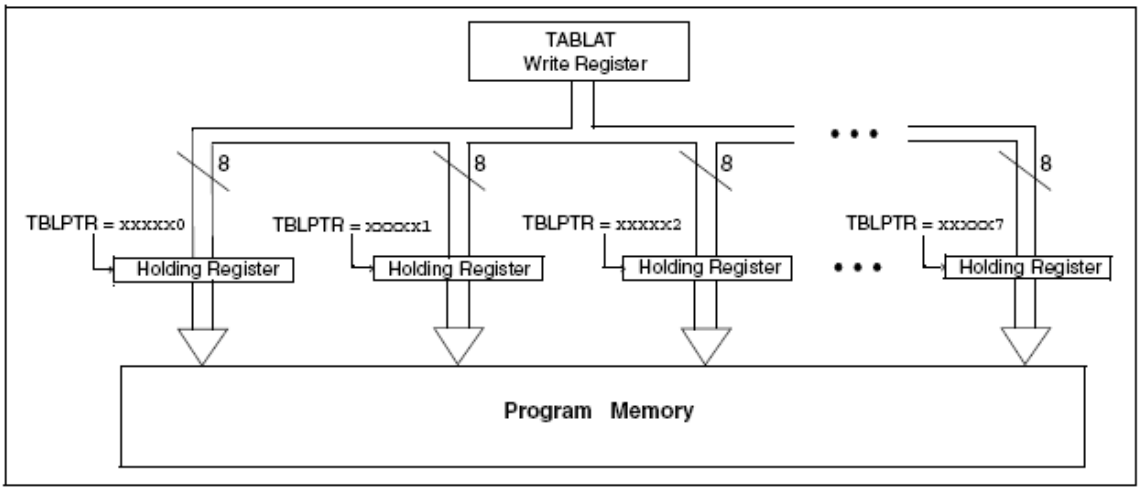
Flash memory. There are 8 holding registers used by the table writes for programming. Since the Table Latch (TABLAT) is only a single byte, the TBLWT instruction has to be executed 8 times for each programming operation. All of the table write operations will essentially be short writes, because only the holding registers are written. At the end of updating 8 registers, the EECON1 register must be written to, to start the programming operation with a long write. The long write is necessary for programming the internal Flash. Instruction execution is halted while in a long write cycle. The long write will be terminated by the internal programming timer. The EEPROM on-chip timer controls the write time. The write/erase voltages are generated by an on-chip charge pump rated to operate over the voltage range of the device for byte or word operations.

7.5.1 FLASH PROGRAM MEMORY WRITESE QUENCE

The sequence of events for programming an internal program memory location should be:
1. Read 1024 bytes into RAM.
2. Update data values in RAM as necessary.
3. Load Table Pointer with address being erased.
4. Do the row erase procedure.
5. Load Table Pointer with address of first byte being written.
6. Write the first 8 bytes into the holding registers using the TBLWT instruction, auto-increment may be used.
7. Set the EECON1 register for the write operation:
• set the EEPGD bit to point to program memory;
• clear the CFGS bit to access program memory;
• clear the SIGS bit to access program memory;
• set the WREN to enable byte writes.
8. Disable interrupts.
9. Write 55h to EECON2.
10. Write AAh to EECON2.
11. Set the WR bit. This will begin the write cycle.
12. The CPU will stall for duration of the write (about 2 ms using internal timer).
13. Re-enable interrupts.
14. Repeat steps 7-14 seven times to write 64 bytes.
15. Verify the memory (table read). This procedure will require about 18 ms to update one row of 64 bytes of memory. An example of the required code is given in Example 7-3.

```
Note:  Before setting the WR bit, the Table
       Pointer address needs to be within the
       intended address range of the 8 bytes
       in the holding registers.
```

FIGURE 7-5: TABLE WRITES TO FLASH PROGRAM MEMORY



EXAMPLE 7-3: WRITING TO FLASH PROGRAM MEMORY

```
          MOVLW     D’64                ; Number of bytes in erase block
          MOVWF     COUNTER
          MOVLW     high (BUFFER_ADDR)  ; Point to buffer
          MOVWF     FSR0H
          MOVLW     low (BUFFER_ADDR)
          MOVWF     FSR0L
          MOVLW     upper (CODE_ADDR)   ; Load TBLPTR with the base
          MOVWF     TBLPTRU             ; Address of the memory block
          MOVLW     high (CODE_ADDR)
          MOVWF     TBLPTRH
          MOVLW     low (CODE_ADDR)
          MOVWF     TBLPTRL
READ_BLOCK
          TBLRD*+                       ; Read into TABLAT, and inc
          MOVF      TABLAT,    W        ; Get data
          MOVWF     POSTINC0            ; Store data
          DECFSZ    COUNTER             ; Done?
          BRA       READ_BLOCK          ; Repeat
MODIFY_WORD
          MOVLW     DATA_ADDR_HIGH      ; Point to buffer
          MOVWF     FSR0H
          MOVLW     DATA_ADDR_LOW
          MOVWF     FSR0L
          MOVLW     NEW_DATA_LOW        ; Update buffer word
          MOVWF     POSTINC0
          MOVLW     NEW_DATA_HIGH
          MOVWF     INDF0
ERASE_BLOCK
          MOVLW     upper (CODE_ADDR)   ; Load TBLPTR with the base
          MOVWF     TBLPTRU             ; Address of the memory block
          MOVLW     high (CODE_ADDR)
          MOVWF     TBLPTRH
          MOVLW     low (CODE_ADDR)
          MOVWF     TBLPTRL
          BSF       EECON1,    EEPGD    ; Point to FLASH program memory
          BCF       EECON1,    CFGS     ; Access program FLASH or Data EEPROM memory
          BCF       EECON1,    SIGS     ;
          BSF       EECON1,    WREN     ; Enable write to memory
          BSF       EECON1,    FREE     ; Enable Row Erase operation
          BCF       INTCON,    GIE      ; Disable interrupts
Required  MOVLW     55h                 ;
Sequence  MOVWF     EECON2              ; Write 55h
          MOVLW     0AAH                ;
          MOVWF     EECON2              ; Write aah
          BSF       EECON1,    WR       ; Start erase (CPU stall)
          NOP
          BSF       EECON1,    GIE      ; Re-enable interrupts
          TBLRD*-                       ; Dummy read decrement
WRITE_BUFFER_BACK
          MOVLW     8                   ; Number of write buffer groups of 8 bytes
          MOVWF     COUNTER_HI
          MOVLW     high (BUFFER_ADDR)  ; Point to buffer
          MOVWF     FSR0H
          MOVLW     low (BUFFER_ADDR)
          MOVWF     FSR0L
PROGRAM_LOOP
          MOVLW     8                   ; Number of bytes in holding register
          MOVWF     COUNTER
WRITE_WORD_TO_HREGS
          MOVFW     POSTINC0,  W        ; Get low byte of buffer data
          MOVWF     TABLAT              ; Present data to table latch
          TBLWT+*                       ; Write data< perform a short write
                                        ; to internal TBLWT holding register.
          DECFSZ    COUNTER             ; Loop until buffers are full
          BRA       WRITE_WORD_TO_HREGS
PROGRAM_MEMORY
          BSF       EECON1,    EEPGD    ; Point to FLASH program memory
          BCF       EECON1,    CFGS     ; Access program FLASH or Data EEPROM memory
          BCF       EECON1,    SIGS     ;
          BSF       EECON1,    WREN     ; Enable write to memory
          BCF       INTCON,    GIE      ; Disable interrupts
Required  MOVLW     55h                 ;
Sequence  MOVWF     EECON2              ; Write 55h
          MOVLW     0AAH                ;
          MOVWF     EECON2              ; Write aah
          BSF       EECON1,    WR       ; Start erase (CPU stall)
          NOP
          BSF       INTCON,    GIE      ; Re-enable interrupts
          DECFSZ    COUNTER_HI          ; Loop until done
          BRA       PROGRAM_LOOP
          BCF       EECON1,    WREN     ; Disable write to memory
```

7.5.2 WRITE VERIFY

Depending on the application, good programming practice may dictate that the value written to the memory should be verified against the original value. This should be used in applications where excessive writes can stress bits near the specification limit.

7.5.3 UNEXPECTED TERMINATION OF WRITE OPERATION

If a write is terminated by an unplanned event, such as loss of power or an unexpected Reset, the memory location just programmed should be verified and reprogrammed if needed. The WRERR bit is set when a write operation is interrupted by a RESETN Reset or a WDT Time-out Reset during normal operation. In these situations, users can check the WRERR bit and rewrite the location.

7.5.4 PROTECTION AGAINST SPURIOUS WRITES

To reduce the probability against spurious writes to Flash program memory, the write initiate sequence must also be followed.

7.6 Flash Program Operation During Code Protection

TABLE 7-2: REGISTERS ASSOCIATED WITH PROGRAM FLASH MEMORY

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other Resets
TBLPTRU	—	—	bit 21	Program Memory Table Pointer Upper Byte (TBLPTR<20:16>)					--00 0000	--00 0000
TBPLTRH	Program Memory Table Pointer High Byte (TBLPTR<15:8>)								0000 0000	0000 0000
TBLPTRL	Program Memory Table Pointer Low Byte (TBLPTR<7:0>)								0000 0000	0000 0000
TABLAT	Program Memory Table Latch								0000 0000	0000 0000
INTCON	GIE/GIEH	PEIE/ GIEL	TMR0IE	INTE	RBIE	TMR0IF	INTF	RBIF	0000 000x	0000 000u
EECON2	EEPROM Control Register 2 (not a physical register)								—	—
EECON1	EEPGD	CFGS	SIGS	FREE	WRERR	WREN	WR	RD	xx00 x000	uu00 u000
IPR2	—	—	—	EEIP	BCLIP	—	—	—	---1 1---	---1 1---
PIR2	—	—	—	EEIF	BCLIF	—	—	—	---0 0---	---0 0---
PIE2	—	—	—	EEIE	BCLIE	—	—	—	---0 0---	---0 0---

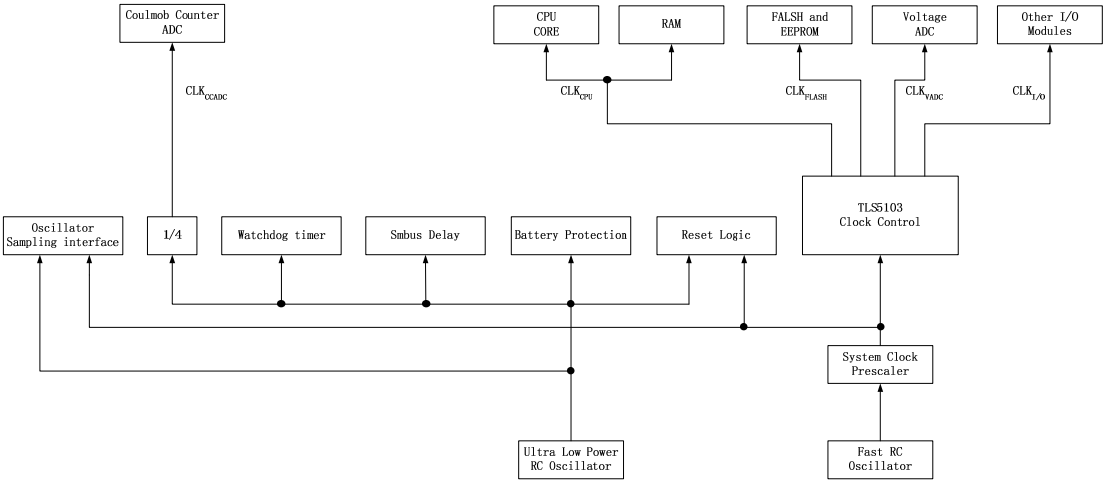
Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'.
Shaded cells are not used during Flash/EEPROM access.

8. System Clock and Clock Options

8.1 Clock Systems and their Distribution

Figure 8-1 presents the principal clock systems in the SE1100. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes. The clock systems are detailed below.

Figure 8-1. Clock Distribution



8.1.1 CPU Clock - CLKcpu

The CPU clock is routed to parts of the system concerned with operation of the SE1100 core. Examples of such modules are the Register File, the Status Register and the data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

8.1.2 I/O Clock - CLKi/o

The I/O clock is used by the majority of the I/O modules. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted.

8.1.3 Flash Clock – CLKflash

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

8.1.4 Voltage ADC Clock-CLKvadc

The Voltage ADC is provided with a dedicated clock domain. The VADC clock is automatically prescaled relative to the System Clock, giving a fixed VADC clock at 1 MHz.

8.1.5 Coulomb Counter ADC Clock - CLKccadc

The Coulomb Counter ADC is provided with a dedicated clock domain. This allows operating the Coulomb Counter ADC in low power modes like Power-save for continuous current measurements.

8.1.6 Watchdog Timer and Battery Protection Clock

The Watchdog Timer and Battery Protection are provided with a dedicated clock domain. This allows operation in all modes except Power-off. It also allows very low power operation by utilizing an Ultra Low Power RC Oscillator dedicated to this purpose.

8.2 Clock Sources

The following section describes the clock sources available in the device. The clocks are input to the clock generator, and routed to the appropriate modules.
The SE1100 has 2 on-board oscillator used to clock the internal logic. Table 8-1 shows the clock sources and their usage.

Table 8-2. Available Clock Sources.

Clock Source	Usage
Calibrated Fast RC Oscillator	The clock source for the CPU, I/O, Flash, and Voltage ADC.
Ultra Low Power RC Oscillator	The clock source for Coulomb Counter ADC, Bandgap Buffer Short Circuit Detector, and SMBus Connect/Disconnect.

8.2.1 Calibrated Fast RC Oscillator

The calibrated Fast RC Oscillator by default provides a 8.0 MHz clock. The frequency is nominal value at 25C. This clock will operate with no external components. During reset, hardware loads the calibration byte into the FOSCCAL Register and thereby automatically calibrates the Fast RC Oscillator. At 25C, this calibration gives a frequency of 8 MHz± 1%. The oscillator can be calibrated to any frequency in the range 7.3 - 8.1 MHz by changing the FOSCCAL register. Note that the frequency of the system clock is given by the “System Clock Prescaler” .

8.2.3 Ultra Low Power RC Oscillator

The Ultra Low Power RC Oscillator (ULP Oscillator) provides a 128 kHz clock (typical value. This oscillator provides the clock for the Watchdog Timer and Battery Protection modules. The actual ULP Oscillator frequency depends on process variations and temperature. The Oscillator is automatically enabled in all operational modes. It is also enabled during reset. There are two alternative methods for determining the actual clock period of the ULP Oscillator:

1.To determine the accurate clock period as a function of die temperature, if needed by the application, the Oscillator Sampling Interface should be used.

2.To determine a fixed value for the actual clock period independent of the die temperature, for example to determine the best setting of the Battery Protection timing, use the calibration byte ULP_RC_FRQ stored in the signature address space.

8.3 Clock Startup Sequence

8.4 Clock Output

The CPU clock divided by 4 can be output to the PB1 pin. The CPU can enable the clock output function by setting the CKOE bit in the RCON Control Register. The clock will not run in any sleep modes.

8.5 System Clock Prescaler

The SE1100 has a System Clock Prescaler, used to prescale the Calibrated Fast RC Oscillator. The system clock can be divided by setting the CLKPR Register, and this enables the user to decrease or increase the system clock frequency as the requirement for power consumption and processing power changes. This system clock will affect the clock frequency of the CPU and all synchronous peripherals.

When switching between prescaler settings, the System Clock Prescaler ensures that no glitches occurs in the clock system. It also ensures that no intermediate frequency is higher than neither the clock frequency corresponding to the previous setting, nor the clock frequency corresponding to the new setting. The ripple counter that implements the prescaler runs at the frequency of the undivided clock, and may be faster than the CPU's clock frequency. It is not possible to determine the state of the prescaler, and the exact time it takes to switch from one clock division to the other cannot be exactly predicted. From the time the CLKPS values are written, it takes between T1 + T2 and T1+ 2*T2 before the new clock frequency is active. In this interval, two active clock edges are produced. Here, T1 is the previous clock period, and T2 is the period corresponding to the new prescaler setting.

To avoid unintentional changes of clock frequency, a special write procedure must be followed to change the CLKPS bits:

1.Write the Clock Prescaler Change Enable (CLKPCE) bit to one and all other bits in CLKPR to zero.

2.Within four cycles, write the desired value to CLKPS while writing a zero to CLKPCE. Interrupts must be disabled when changing prescaler setting to make sure the write procedure is not interrupted.

8.6 OSI - Oscillator Sampling Interface

8.6.1 Features

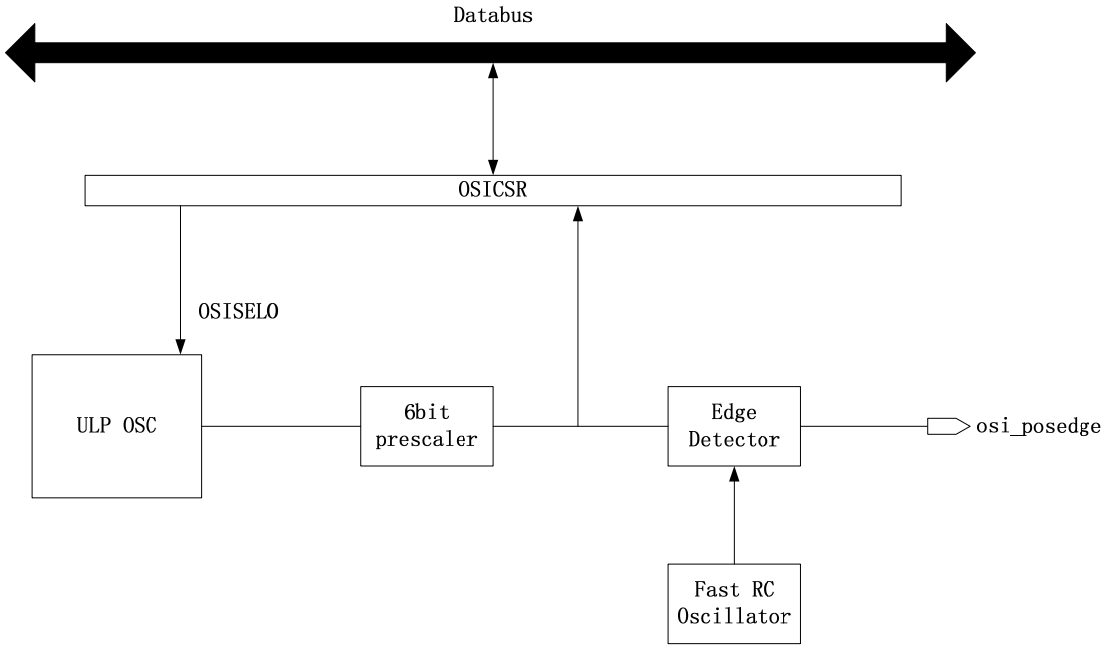
- ✧ Runtime selectable oscillator input ULP RC Oscillator
- ✧ 6 bit prescaling of the selected oscillator
- ✧ Software read access to the phase of the prescaled clock
- ✧ Input capture trigger source for CCP

8.6.2 Overview

The Oscillator Sampling Interface (OSI) enables sampling of the Ultra Low Power RC (ULP) oscillators in SE1100. OSI can be used to calibrate the Fast RC Oscillator runtime with high accuracy. OSI can also provide an accurate reference for compensating the ULP Oscillator frequency drift. This is useful for modules with high-precision requirements, such as the Coulomb Counter.

The prescaled oscillator phase can be continuously read by the CPU through the OSICSR register. In addition, the input capture function of CCP can be set up to trigger on the rising edge of the prescaled clock. This enables accurate measurements of the oscillator frequencies relative to the Fast RC Oscillator.

A simplified block diagram of the Oscillator Sampling Interface is shown in Figure 8-2
Figure 8-2. Oscillator Sampling Interface Block Diagram



The osi_posedge signal pulses on each rising edge of the prescaled clock. This signal is not directly accessible by the CPU, but can be used to trigger the input capture function of Timer/Counter1. Using OSI in combination with the input capture function of Timer/Counter0 facilitates accurate measurement of the oscillator frequencies with a minimum of CPU calculation. Refer to CCP module for details on how to enable the Input Capture function.

8.6.3 Usage

The ULP RC oscillator represents a highly predictable and accurate clock source over the entire temperature range and provides an excellent reference for calibrating the Fast RC oscillator runtime. Typically, runtime calibration is needed to provide an accurate Fast RC frequency for asynchronous serial communication in the complete temperature range.

8.7 Register Description

8.7.1 FOSCCAL – Fast RC Oscillator Calibration Register

Bit	7	6	5	4	3	2	1	0	
(0XE5C)	FCAL7	FCAL6	FCAL5	FCAL4	FCAL3	FCAL2	FCAL1	FCAL0	FOSCCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	Device Specific Calibration Value								

• Bits 7:0 – FCAL7:0: Fast RC Oscillator Calibration Value

The Fast RC Oscillator Calibration Register is used to trim the Fast RC Oscillator to remove process variations from the oscillator frequency. The factory-calibrated value is automatically

written to this register during chip reset, giving an oscillator frequency of approximately 8.0 MHz at 25°C. The application software can write this register to change the oscillator frequency. The oscillator can be run-time calibrated to any frequency in the range 3.8 - 17.8 MHz. Calibration outside that range is not guaranteed. Note that this oscillator is used to time EEPROM and Flash write accesses, and these write times will be affected accordingly. If the EEPROM or Flash are written, do not calibrate to morethan 8.1 MHz. Otherwise, the EEPROM or Flash write may fail.

The FCAL[7:5] bits determine the range of operation for the oscillator. Setting these bits to 3'b000 gives the lowest frequency range, setting these bit to 3'b111 gives the highest frequency range. The frequency ranges are overlapping. A setting of for instance FOSCCAL = 0x1F gives a higher frequency than FOSCCAL = 0x20.

The FCAL[4:0] bits are used to tune the frequency within the selected range. A setting of 0x00 gives the lowest frequency in that range, and a setting of 0x1F gives the highest frequency in the range. Incrementing FCAL[4:0] by 1 will give a frequency increment of less than 1% in the frequency range 3.8 - 17.8 MHz. With an accurate time reference, an oscillator accuracy of ±0.5% can be achieved after calibration. The frequency will drift with temperature, so run-time calibration will be required to maintain the accuracy.

8.7.2 RCON

BIT	7	6	5	4	3	2	1	0	
(0XFD0)	IPEN	CKOE	-	RI	TO	PD	POR	BOR	RCON
Read/Write	R/W	R/W	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	1	1	1	0	0	

• Bit 5 – CKOE: Clock Output

When this bit is written to one, the CPU clock divided by 4 is output on the PB1 pin.

8.7.3 CLKPR – Clock Prescale Register

Bit	7	6	5	4	3	2	1	0	
(0XFB9)	CLKPCE	–	–	–	–	–	CLKPS1	CLKPS0	CLKPR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 – CLKPCE: Clock Prescaler Change Enable

The CLKPCE bit must be written to logic one to enable change of the CLKPS bits. The CLKPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero. CLKPCE is cleared by hardware four cycles after it is written or when CLKPS bits are written. Rewriting the CLKPCE bit within this time-out period does neither extend the time-out period, or clear the CLKPCE bit.

• Bit 1:0 – CLKPS1:0: Clock Prescaler Select Bit 1..0

These bits define the division factor between the selected clock source and the internal system clock. These bits can be written run-time to vary the clock frequency to suit the application requirements. As the divider divides the master clock input to the MCU, the speed of all synchronous peripherals is reduced when a division factor is used.

Table 7-4 System Clock Prescaler Select

CLKPS1	CLKPS0	Clock Division Factor
0	0	1
0	1	2
1	0	4
1	1	8

8.7.4 OSICSR – Oscillator Sampling Interface Control and Status Register

Bit	7	6	5	4	3	2	1	0	
(0XFB8)	–	–	–	OSISEL0	–	–	OSIST	OSIEN	OSICSR
Read/Write	R	R	R	R/W	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7:5,3:2 – RES: Reserved bits

These bits are reserved bits in the SE1100 and will always read as zero.

• Bit 4 - OSISEL0: Oscillator Sampling Interface Select 0

The bit select the ULP OSC as the clock source.

• Bit 1 – OSIST: Oscillator Sampling Interface Status

This bit continuously displays the phase of the prescaled clock. This bit can be polled by the CPU to determine the rising and falling edges of the prescaled clock.

• Bit 0 – OSIEN: Oscillator Sampling Interface Enable

Setting this bit enables the Oscillator Sampling Interface. When this bit is cleared, the Oscillator Sampling Interface is disabled.

9. Power Management and Sleep Modes

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The SE1100 provides various sleep modes allowing the user to tailor the power consumption to the application's requirements.

9.1 Sleep Modes

The different sleep modes and their wake up sources is summarized in Table 9-1, and shows a sleep mode state diagram.

Table 9-1 Wake-up Sources for Sleep Modes

Mode	Wake-up sources											
	Bandgap Buffer Short Circuit Detection	Wake-up on Regular Current	Battery Protection Interrupts	External Interrupts	WDT	SPM/EEPROM Ready	CC-ADC	V-ADC	Other I/O	Charger Detector(1)(2)	SMBus Address match and Bus Connect/Disconnect	Voltage Regulator warning
Idle	X	X	X	X	X	X	X	X	X	X	X	X
ADC Noise Reduction	X	X	X	X	X	X	X	X		X	X	X
Power-save	X	X	X	X	X		X			X	X	
Power-off										X		

Notes: 1. Discharge FET must be switched off for Charge Detect to be active.
2. When Waking from Power-Off the Charger Detect will generate a Power-on Reset(POR). From other sleep modes a charger detect interrupt will wake-up chip.

To enter any of the sleep modes, the SE bit in SMCR, must be written to logic one and a SLEEP instruction must be executed. The SM2..0 bits in the SMCR Register select which sleep mode will be activated by the SLEEP instruction. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the register file and SRAM are unaltered when the device wakes up from any sleep mode except Power-off. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

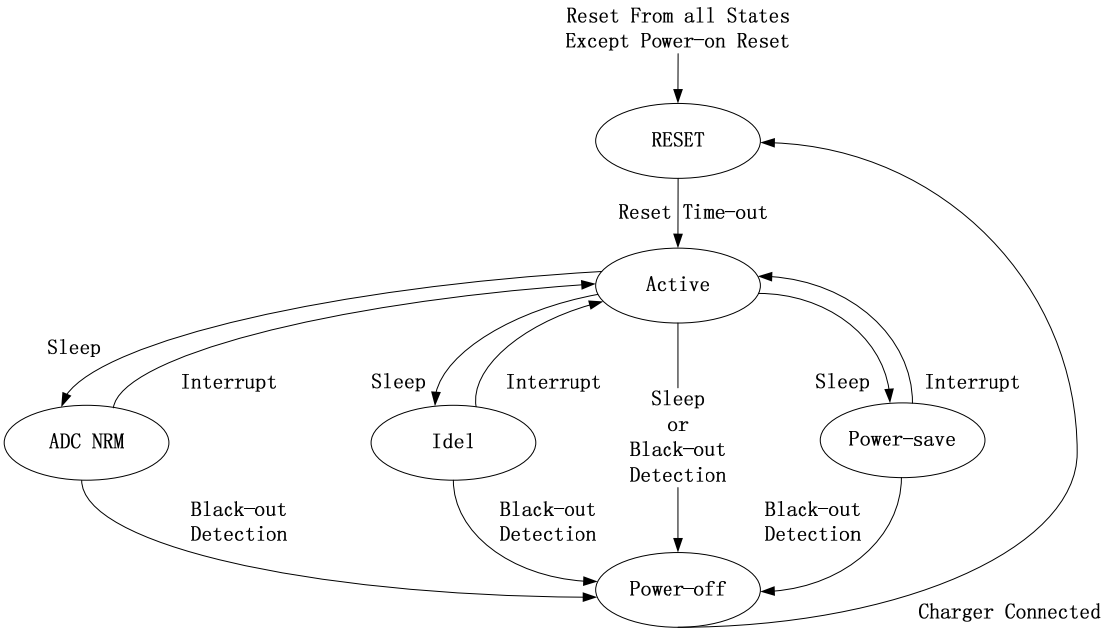


Table 9-2. Active modules in different Sleep Modes

Module	Mode				
	Active	Idle	ADC Noise Reduction	Power-save	Power-off
RCOSC_FAST	X	X			
RCOSC_ULP	X	X	X	X	
OSI	X	X			
CPU	X				
FLASH	X				
EEPROM	X				
Timer0/2	X	X			
Timer1	X	X	X	X	
SMBUS	X	X	X(3)	X(3)	
V-ADC	X	X	X		
CC-ADC	X	X	X	X	
External Interrupts	X	X	X	X	
CBP	X	X	X	X	
DUV	X	X	X	X	
WDT	X	X	X	X	
VREG	X	X	X	X	
VREGMON	X	X	X		
Bandgap Reference	X	X	X	X	
FET Driver	X	X	X	X	
Charge_Detect(4)	X	X	X	X	X

9.2 Idle Mode

When the SM2..0 bits are written to 000, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing all peripheral functions to continue operating. This sleep mode basically halts clkCPU and clkFLASH, while allowing the other clocks to run. Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow interrupt.

9.3 ADC Noise Reduction

When the SM2:0 bits are written to 001, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode, stopping the CPU but allowing the Voltage ADC (V-ADC), Voltage Regulator Monitor (VREGMON), Watchdog Timer (WDT), Coulomb Counter (CC), Current Battery Protection (CBP), and the Ultra Low Power RC Oscillator (RCOSC_ULP) to continue operating. This sleep mode basically halts clkI/O, clkCPU, and clkFLASH, while allowing the other clocks to run. This improves the noise environment for the Voltage ADC, enabling higher accuracy on measurements.

9.4 Power-save Mode

When the SM2..0 bits are written to 011, the SLEEP instruction makes the MCU enter Power-save mode. In this mode, the internal Fast RC Oscillator (RCOSC_FAST) is stopped, while Watchdog Timer (WDT), Coulomb Counter (CC), Current Battery Protection (CBP) and the Ultra Low Power RC Oscillator (RCOSC_ULP) continue operating. This mode will be the default mode when application software does not require operation of CPU, Flash or any of the peripheral units running at the Fast internal Oscillator (RCOSC_FAST). If the current through the sense resistor is so small that the Coulomb Counter cannot measure it accurately, Regular Current detection should be enabled to reduce power consumption. The WDT keeps accurately track of the time so that battery self discharge can be calculated.

Note that if a level triggered interrupt is used for wake-up from Power-save mode, the changed level must be held for some time to wake up the MCU. When waking up from Power-save mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped.

9.5 Power-off Mode

When the SM2..0 bits are written to 100 and the SE bit is set, the SLEEP instruction makes the CPU shut down the Voltage Regulator, leaving only the Charger Detect Circuitry operational. To ensure that the MCU enters Power-off mode only when intended, the SLEEP instruction must be executed within 4 clock cycles after the SM2..0 bits are written. The MCU will reset when returning from Power-off mode.

Note: Before entering Power-off sleep mode, interrupts should be disabled by software. Otherwise interrupts may prevent the SLEEP instruction from being executed within the time limit.

9.6 Power Reduction Register

The Power Reduction Register (PRR), provides a method to stop the clock to individual peripherals to reduce power consumption. The current state of the peripheral is frozen and the I/O registers can not be read or written. Resources used by the peripheral when stopping the clock will remain occupied, hence the peripheral should in most cases be disabled before stopping the clock. Waking up a module, which is done by clearing the bit in PRR, puts the module in the same state as before shutdown. Module shutdown can be used in Idle mode and Active mode to significantly reduce the overall power consumption. In all other sleep modes, the clock is already stopped.

9.7 Minimizing Power Consumption

There are several issues to consider when trying to minimize the power consumption in an SE1100 controlled system. In general, sleep modes should be used as much as possible, and the sleep mode should be selected so that as few as possible of the device's functions are operating. All functions not needed should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

9.7.1 Watchdog Timer

If the Watchdog Timer is not needed in the application, the module should be turned off. If the Watchdog Timer is enabled, it will be enabled in all sleep modes except Power-off. The Watchdog Timer current consumption is significant only in Power-save mode.

9.7.2 Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. The most important is then to ensure that no pins drive resistive loads. In sleep modes where both the I/O clock (clkI/O) and the ADC clock (clkADC) are stopped, the input buffers of the device will be disabled. This ensures that no power is consumed by the input logic when not needed. In some cases, the input logic is needed for detecting wake-up conditions, and it will then be enabled. If the input buffer is enabled and the input signal is left floating or have an analog signal level close to VREG/2, the input buffer will use excessive power.

For analog input pins, the digital input buffer should be disabled at all times. An analog signal level close to VREG/2 on an input pin can cause significant current even in active mode. Digital input buffers can be disabled by writing to the Digital Input Disable Register.

9.7.3 On-chip Debug System

A programmed JTAGEN Fuse enables some parts of the clock system to be running in all sleep modes. This will increase the power consumption while in sleep. Thus, the JTAGEN Fuse should be disabled when debugwire is not used.

9.7.4 Battery Protection

If one of the Battery Protection features is not needed by the application, this feature should be disabled. The current consumption in the Battery Protection circuitry is only significant in Power-save mode. Disabling both FETs will automatically disable the Battery Protection module in order to save power. The bandgap reference should always be enabled whenever Battery Protection is enabled.

9.7.5 Voltage ADC

If enabled, the V-ADC will consume power independent of sleep mode. To save power, the VADC should be disabled when not used, and before entering Power-save sleep mode.

9.7.6 Coulomb Counter

If enabled, the CC-ADC will consume power independent of sleep mode. To save power, the CC-ADC should be disabled when not used, or set in Regular Current detection mode.

9.7.7 Bandgap Voltage Reference

If enabled, the Bandgap reference will consume power independent of sleep mode. To save power, the Bandgap reference should be disabled when not used as reference for the Voltage ADC, the Coloumb Counter or Battery Protection.

9.8 Register Description

9.8.1 SMCR – Sleep Mode Control Register

The Sleep Mode Control Register contains control bits for power management.

Bit	7	6	5	4	3	2	1	0	
(0XFB7)	–	–	–	–	SM2	SM1	SM0	SE	SMCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7:4 – Res: Reserved Bits

These bits are reserved bits in the SE1100 , and will always read as zero.

• Bits 3:1 – SM2:0: Sleep Mode Select Bits 2, 1 and 0

These bits select between the four available sleep modes as shown in Table 9-3.

Table 9-3. Sleep Mode

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Reserved
0	1	1	Power-save
1	0	0	Power-off
1	0	1	Reserved
1	1	0	Reserved
1	1	1	Reserved

• Bit 0 – SE: Sleep Enable

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmer's purpose, it is recommended to write the Sleep Enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.

9.8.2 PRR0 – Power Reduction Register 0

bit	7	6	5	4	3	2	1	0	
(0xfb6)	PRI2CM	PRI2CS	PREEP	PRTMR0	PRTMR1	PRTMR2	PRUART	PRVADC	PRR0
Read/Write	r/w	r/w	r/w	r/w	r/w	r/w	r/w	r/w	
initial	1	1	1	1	1	1	1	1	

• Bit 7-PRI2CM: Power Reduction I2C Master

Writing a logic zero to this bit shuts down the I2C Master by stopping the clock to the module. When waking up the I2C Master again, the I2C Master should be re initialized to ensure proper operation.

• Bit 7-PRI2CS: Power Reduction I2C Slave

Writing a logic zero to this bit shuts down the I2C Slave by stopping the clock to the module. When waking up the I2C Slave again, the I2C Slave should be re initialized to ensure proper operation.

• Bit 5 - PREEP: Power Reduction EEPROM

Writing a logic zero to this bit shuts down the EEPROM and FlASH by stopping the clock to the module. When waking up the EEPROM and FlASH again, the EEPROM and FlASH should be re initialized to ensure proper operation.

• Bit 3 – PRTMR0: Power Reduction TIMER0 Interface

Writing a logic zero to this bit shuts down the Timer0 module. When the Timer0 is enabled, operation will continue like before the shutdown.

• Bit 2 – PRTMR1: Power Reduction TIMER1 Interface

Writing a logic zero to this bit shuts down the Timer1 module. When the Timer1 is enabled, operation will continue like before the shutdown.

• Bit 1 – PRTMR2: Power Reduction TIMER2 Interface

Writing a logic zero to this bit shuts down the Timer2 module. When the Timer2 is enabled, operation will continue like before the shutdown.

• Bit 0 - PRVADC: Power Reduction V-ADC

Writing a logic zero to this bit shuts down the V-ADC. Before writing the PRVADC bit, make sure that the VADEN bit is cleared to minimize the power consumption.

9.8.2 PRR1 – Power Reduction Register1

bit	7	6	5	4	3	2	1	0	PRR0
(0xfb6)	-	-	-	-	-	-	-	PRCCP	
Read/Write	r	r	r	r	r	r	r	r/w	
initial	0	0	0	0	0	0	0	1	

• Bit 7, 1 - Res: Reserved bits

These bits are reserved for future use. For compatibility with future devices, these bits must be written to zero when PRR1 is written.

• Bit 0-PRCCP: Power Reduction CCP

Writing a logic zero to this bit shuts down the CCP by stopping the clock to the module. When waking up the CCP again, the CCP should be re initialized to ensure proper operation.

10. System Control and Reset

10.1 Resetting the SE1100

During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. The I/O ports of the SE1100 are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running. After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the voltage regulator to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the SUT Fuses.

10.2 Reset Sources

The SE1100 has five sources of reset:

• The Power-on Reset module generates a Power-on Reset when the Voltage Regulator starts up.

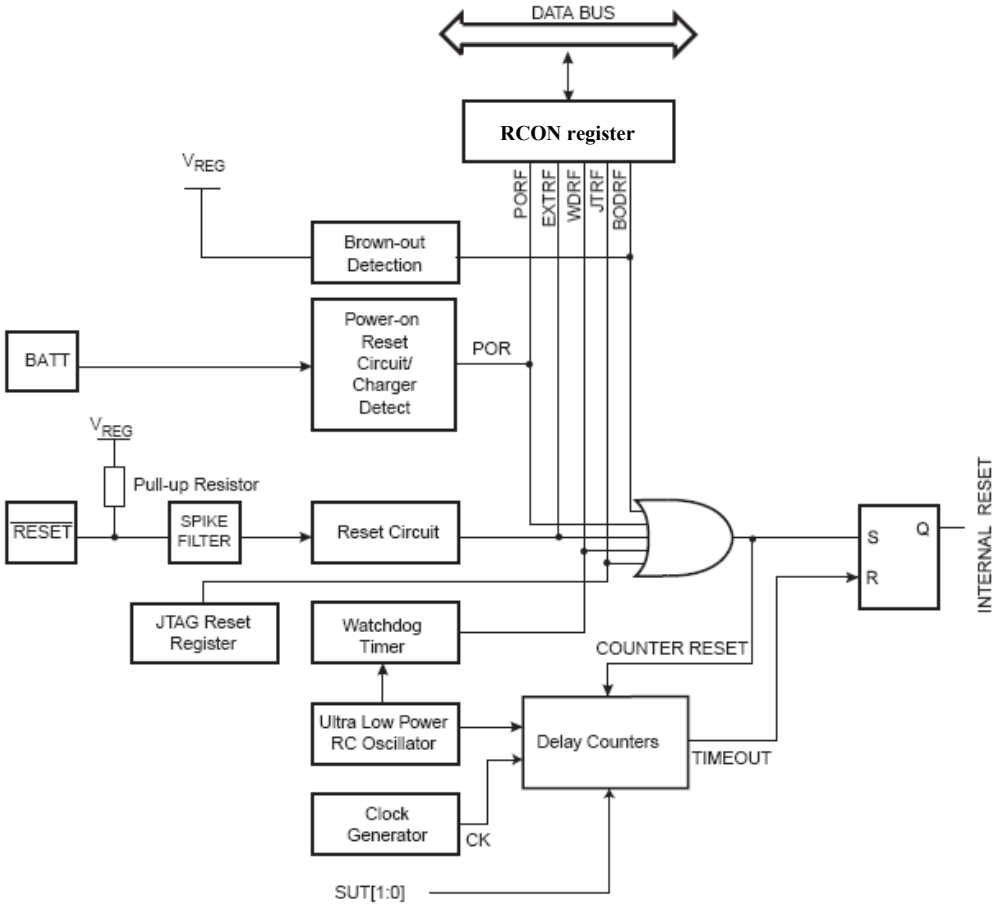
• External Reset. The MCU is reset when a low level is present on the RESET pin for longer than the minimum pulse length.

• Watchdog Reset. The MCU is reset when the Watchdog Timer period expires and the Watchdog is enabled.

• Brown-out Reset. The MCU is reset when VREG is below the Brown-out Reset Threshold, VBOT.

• debugWIRE Reset. In On-chip Debug mode, the debugWIRE resets the MCU when giving the Reset command.

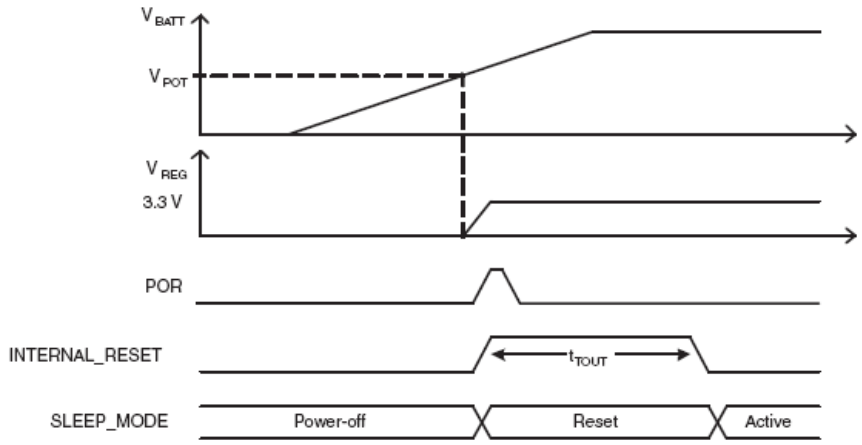
Figure 10-1. Reset Logic



10.2.1 Power-on Reset and Charger Connect

The Voltage Regulator will not start up until the Charger Detect module has enabled it. Before this happens the chip will be in Power-off mode and only the Charger Detect module is enabled. In order for the Charger Detect module to enable the Voltage Regulator, the voltage at the BATT pin must exceed the Power-On Threshold Voltage, VPOT. When the voltage at the BATT pin exceeds VPOT, the Voltage Regulator starts up and the chip enters RESET mode. When the Delay Counter times out, the chip will enter Active mode.

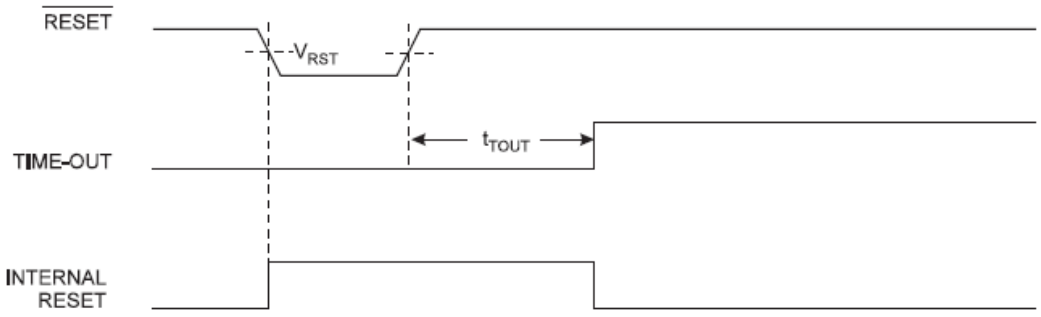
Figure 10-2. Powering up SE1100



10.2.2 External Reset

An External Reset is generated by a low level on the RESET pin. Reset pulses longer than the minimum pulse width will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage – VRST – on its positive edge, the delay counter starts the MCU after the Time-out period – tTOUT – has expired.

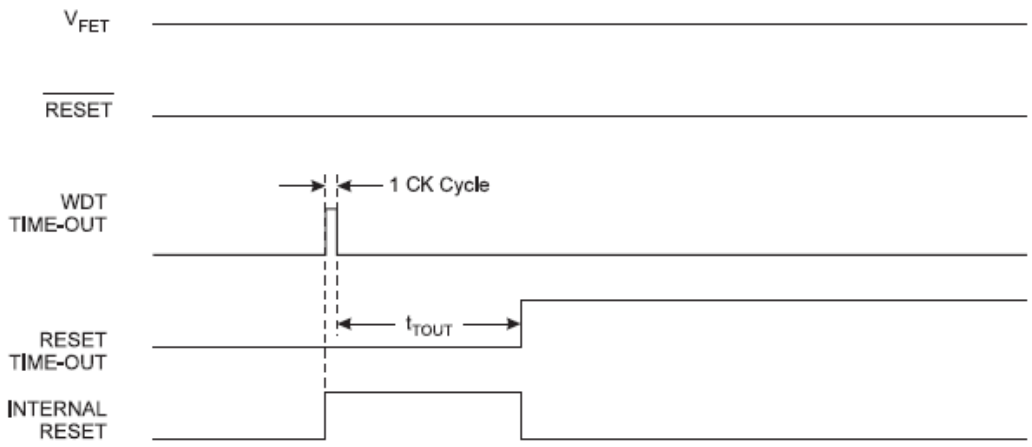
Figure 10-3. External Reset During Operation



10.2.3 Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period tTOUT.

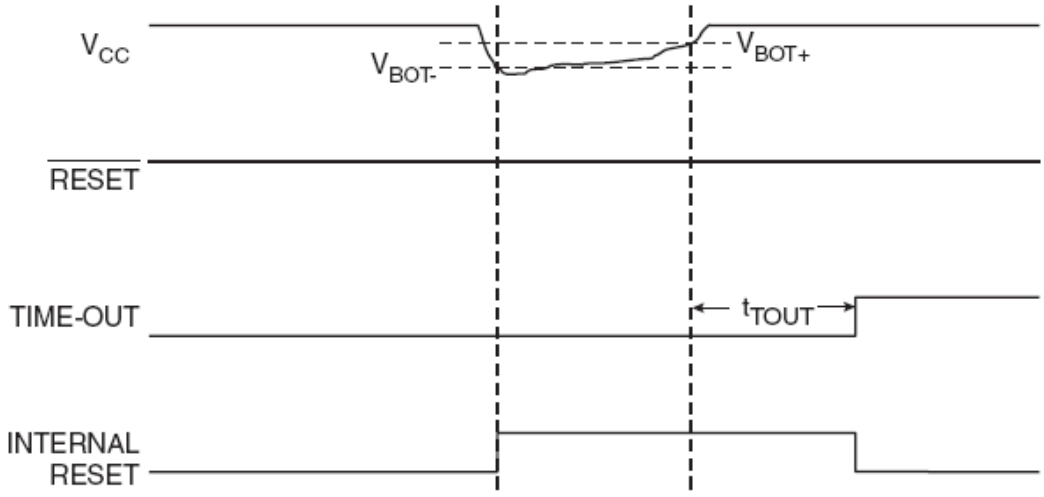
Figure 10-4. Watchdog Reset During Operation



10.2.4 Brown-out Detection

SE1100 has an On-chip Brown-out Detection (BOD) circuit for monitoring the VCC level during operation by comparing it to a fixed trigger level VBOT. The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as VBOT = VBOT + VHYST/2 and VBOT-= VBOT - VHYST/2. The BOD is automatically enabled in all modes of operation, except in Power-off mode. When the BOD is enabled, and VCC decreases to a value below the trigger level (VBOT- in Figure 10-5), the Brown-out Reset is immediately activated. When VCC increases above the trigger level (VBOT+ in Figure 10-5), the delay counter starts the MCU after the Time-out period tTOUT has expired.

Figure 10-5. Brown-out Reset During Operation



10.3 Reset and the Voltage Reference

The Voltage Reference is important for the performance of the system as the VREF voltage will be used as reference voltage for several modules. It is therefore important to notice that after a reset condition the Voltage Reference needs calibration and settling before the VREF voltage is accurate. For details on Voltage Reference calibration and settling time.

11. 8 x 8 HARDWARE MULTIPLIER

11.1 Introduction

An 8 x 8 hardware multiplier is included in the ALU of the SE1100 devices. By making the multiply a hardware operation, it completes in a single instruction cycle. This is an unsigned multiply that gives a 16-bit result. The result is stored in the 16-bit product register pair (PRODH:PRODL). The multiplier does not affect any flags in the ALUSTA register. Making the 8 x 8 multiplier execute in a single cycle gives the following advantages:

- Higher computational throughput
- Reduces code size requirements for multiply algorithms

The performance increase allows the device to be used in applications previously reserved for Digital Signal Processors. Table11-1 shows a performance comparison between Enhanced devices using the single-cycle hardware multiply and performing the same function without the hardware multiply.

11.2 Operation

Example 11-1 shows the sequence to do an 8 x 8 unsigned multiply. Only one instruction is required when one argument of the multiply is already loaded in the WREG register. Example 11-2 shows the sequence to do an 8 x 8 signed multiply. To account for the sign bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 11-1: 8 x 8 UNSIGNEDMULTIPLY ROUTINE

```
MOVF     ARG1, W      ;
MULWF    ARG2         ; ARG1 * ARG2 ->
                      ;    PRODH:PRODL
```

EXAMPLE 11-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
MOVF     ARG1, W
MULWF    ARG2         ; ARG1 * ARG2 ->
                      ; PRODH:PRODL
BTFSC    ARG2, SB     ; Test Sign Bit
SUBWF    PRODH        ; PRODH = PRODH
                      ;          - ARG1
MOVF     ARG2, W
BTFSC    ARG1, SB     ; Test Sign Bit
SUBWF    PRODH        ; PRODH = PRODH
                      ;          - ARG2
```

TABLE 11-1: PERFORMANCE COMPARISON

Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time		
				@ 40 MHz	@ 10 MHz	@ 4 MHz
8 x 8 unsigned	Without hardware multiply	13	69	6.9 µs	27.6 µs	69 µs
	Hardware multiply	1	1	100 ns	400 ns	1 µs
8 x 8 signed	Without hardware multiply	33	91	9.1 µs	36.4 µs	91 µs
	Hardware multiply	6	6	600 ns	2.4 µs	6 µs
16 x 16 unsigned	Without hardware multiply	21	242	24.2 µs	96.8 µs	242 µs
	Hardware multiply	24	24	2.4 µs	9.6 µs	24 µs
16 x 16 signed	Without hardware multiply	52	254	25.4 µs	102.6 µs	254 µs
	Hardware multiply	36	36	3.6 µs	14.4 µs	36 µs

Example 11-3 shows the sequence to do a 16 x 16 unsigned multiply. Equation 11-1 shows the algorithm that is used. The 32-bit result is stored in four registers, RES3:RES0.

EQUATION 11-1: 16 x 16 UNSIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
RES3{:}RES0 &= ARG1H{:}ARG1L \bullet ARG2H{:}ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L)
\end{aligned}
$$

EQUATION 11-2: 16 x 16 SIGNED MULTIPLICATION ALGORITHM

$$
\begin{aligned}
&RES3{:}RES0 \\
&= ARG1H{:}ARG1L \bullet ARG2H{:}ARG2L \\
&= (ARG1H \bullet ARG2H \bullet 2^{16}) + \\
&\quad (ARG1H \bullet ARG2L \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2H \bullet 2^{8}) + \\
&\quad (ARG1L \bullet ARG2L) + \\
&\quad (-1 \bullet ARG2H\langle 7\rangle \bullet ARG1H{:}ARG1L \bullet 2^{16}) + \\
&\quad (-1 \bullet ARG1H\langle 7\rangle \bullet ARG2H{:}ARG2L \bullet 2^{16})
\end{aligned}
$$

EXAMPLE 11-3: 16 x 16 UNSIGNED MULTIPLY ROUTINE

```
    MOVF    ARG1L, W
    MULWF   ARG2L          ; ARG1L * ARG2L ->
                           ; PRODH:PRODL
    MOVFF   PRODH, RES1    ;
    MOVFF   PRODL, RES0    ;
;
    MOVF    ARG1H, W
    MULWF   ARG2H          ; ARG1H * ARG2H ->
                           ; PRODH:PRODL
    MOVFF   PRODH, RES3    ;
    MOVFF   PRODL, RES2    ;
;
    MOVF    ARG1L, W
    MULWF   ARG2H          ; ARG1L * ARG2H ->
                           ; PRODH:PRODL
    MOVF    PRODL, W       ;
    ADDWF   RES1           ; Add cross
    MOVF    PRODH, W       ; products
    ADDWFC  RES2           ;
    CLRF    WREG           ;
    ADDWFC  RES3           ;
;
    MOVF    ARG1H, W       ;
    MULWF   ARG2L          ; ARG1H * ARG2L ->
                           ; PRODH:PRODL
    MOVF    PRODL, W       ;
    ADDWF   RES1           ; Add cross
    MOVF    PRODH, W       ; products
    ADDWFC  RES2           ;
    CLRF    WREG           ;
    ADDWFC  RES3           ;
```

Example 11-4 shows the sequence to do a 16 x 16 signed multiply. Equation 11-2 shows the algorithm used. The 32-bit result is stored in four registers, RES3:RES0. To account for the sign bits of the arguments, each argument pair's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 11-4: 16 x 16 SIGNED MULTIPLY ROUTINE

```
        MOVF    ARG1L,    W
        MULWF   ARG2L                   ; ARG1L * ARG2L -> PRODH:PRODL
        MOVFF   PRODH,    RES1          ;
        MOVFF   PRODL,    RES0          ;
;
        MOVF    ARG1H,    W
        MULWF   ARG2H                   ; ARG1H * ARG2H -> PRODH:PRODL
        MOVFF   PRODH,    RES3          ;
        MOVFF   PRODL,    RES2          ;
;
        MOVF    ARG1L,    W
        MULWF   ARG2H                   ; ARG1L * ARG2H -> PRODH:PRODL
        MOVF    PRODL,    W             ;
        ADDWF   RES1                    ; Add cross
        MOVF    PRODH,    W             ; Products
        ADDWFC  RES2                    ;
        CLRF    WREG                    ;
        ADDWFC  RES3                    ;
;
        MOVF    ARG1H,    W
        MULWF   ARG2L                   ; ARG1H * ARG2L -> PRODH:PRODL
        MOVF    PRODL,    W             ;
        ADDWF   RES1                    ; Add cross
        MOVF    PRODH,    W             ; Products
        ADDWFC  RES2                    ;
        CLRF    WREG                    ;
        ADDWFC  RES3
;
        BTFSS   ARG2H,    7             ; ARG2H:ARG2L neg?
        BRA     SIGH_ARG1               ; No, check ARG1
        MOVF    ARG1L,    W             ;
        SUBWF   RES2                    ;
        MOVF    ARG1H,    W             ;
        SUBWFB  RES3
;
SIGH_ARG1
        BTFSS   ARG1H,    7             ; ARG1H:ARG1L neg?
        BRA     CONT_CODE               ; No, done
        MOVF    ARG2L,    W             ;
        SUBWF   RES2                    ;
        MOVF    ARG2H,    W             ;
        SUBWFB  RES3
;
CONT_CODE
            :
```

12. INTERRUPTS

The SE1100 devices have multiple interrupt sources and an interrupt priority feature that allows each interrupt source to be assigned a high priority level or a low priority level. The high priority interrupt vector is at 000008h and the low priority interrupt vector is at 000018h. High priority interrupt events will override any low priority interrupts that may be in progress. There are 13 registers that are used to control interrupt operation. These registers are:

- RCON
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3
- PIE1, PIE2, PIE3
- IPR1, IPR2, IPR3

It is recommended that the SE1100 header files be used for the symbolic bit names in these registers. This allows the assembler/ compiler to automatically take care of the placement of these bits within the specified register. Each interrupt source has three bits to control its operation. The functions of these bits are:

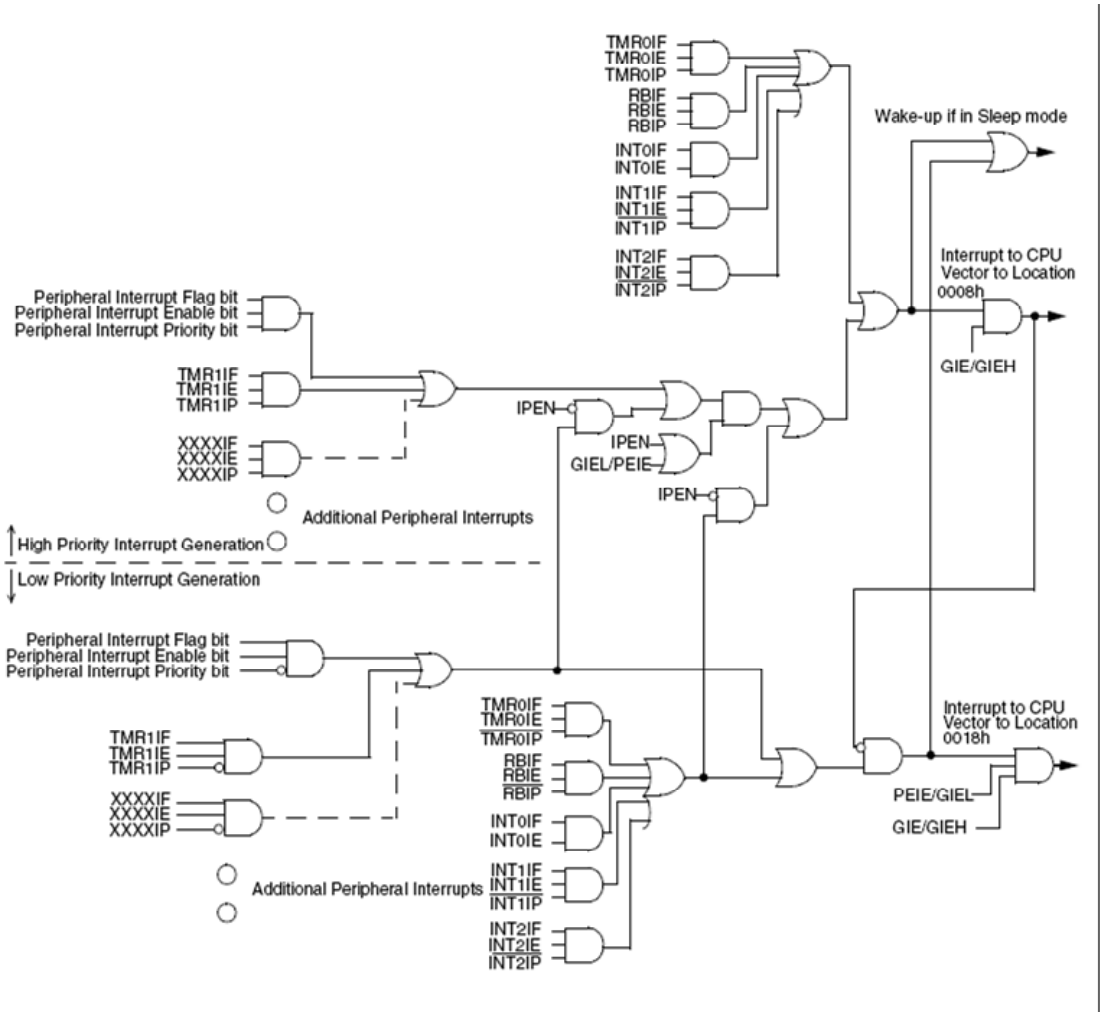
- Flag bit to indicate that an interrupt event occurred
- Enable bit that allows program execution to branch to the interrupt vector address when the flag bit is set
- Priority bit to select high priority or low priority

The interrupt priority feature is enabled by setting the IPEN bit (RCON register). When interrupt priority is enabled, there are two bits that enable interrupts globally. Setting the GIEH bit (INTCON<7>) enables all interrupts. Setting the GIEL bit (INTCON register) enables all interrupts that have the priority bit cleared. When the interrupt flag, enable bit and appropriate global interrupt enable bit are set, the interrupt will vector immediately to address 000008h or 000018h, depending on the priority level. Individual interrupts can be disabled through their corresponding enable bits. When the IPEN bit is cleared (default state), the interrupt priority feature is disabled. In Compatibility mode, the interrupt priority bits for each source have no effect. The PEIE bit (INTCON register) enables/disables all peripheral interrupt sources. The GIE bit (INTCON register) enables/disables all interrupt sources. All interrupts branch to address 000008h in Compatibility mode. When an interrupt is responded to, the global interrupt enable bit is cleared to disable further interrupts. If the IPEN bit is cleared, this is the GIE bit. If interrupt priority levels are used, this will be either the GIEH or GIEL bit. High priority interrupt sources can interrupt a low priority interrupt. The return address is pushed onto the stack and the PC is loaded with the interrupt vector address (000008h or 000018h). Once in the Interrupt Service Routine, the source(s) of the interrupt can be determined by polling the interrupt flag bits. The interrupt flag bits must be cleared in software before re-enabling interrupts to avoid recursive interrupts. The “return from interrupt” instruction, RETFIE, exits the interrupt routine and sets the GIE bit (GIEH

or GIEL if priority levels are used), which re-enables interrupts. For external interrupt events, such as the INT pins or the PORTB input change interrupt, the interrupt latency will be three to four instruction cycles. The exact latency is the same for one or two-cycle instructions. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit.

Note: Do not use the MOVFF instruction to modify any of the interrupt control registers while **any** interrupt is enabled. Doing so may cause erratic microcontroller behavior.

FIGURE 12-1: INTERRUPT LOGIC



12.1 INTCON Registers

The INTCON registers are readable and writable registers which contain various enable, priority and flag bits. Because of the number of interrupts to be controlled, SE1100 devices have three INTCON registers. They are detailed in Register 12-1 through Register 12-3.

Note: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the global interrupt enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt. This feature allows software polling.

REGISTER 12-1: INTCON: INTERRUPT CONTROL REGISTER

Read/Write (0xFF2) Bit	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0	
	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	—	TMR0IF	INT0IF	—	INTCON
	bit7							bit0	

bit 7 **GIE/GIEH:** Global Interrupt Enable bit
When IPEN (RCON<7>) = 0:
1 = Enables all unmasked interrupts
0 = Disables all interrupts
When IPEN (RCON<7>) = 1:
1 = Enables all high priority interrupts
0 = Disables all priority interrupts

bit 6 **PEIE/GIEL:** Peripheral Interrupt Enable bit
When IPEN (RCON<7>) = 0:
1 = Enables all unmasked peripheral interrupts
0 = Disables all peripheral interrupts
When IPEN (RCON<7>) = 1:
1 = Enables all low priority peripheral interrupts
0 = Disables all low priority peripheral interrupts

bit 5 **TMR0IE:** TMR0 Overflow Interrupt Enable bit
1 = Enables the TMR0 overflow interrupt
0 = Disables the TMR0 overflow interrupt

bit 4 **INT0IE:** INT0 External Interrupt Enable bit
1 = Enables the INT0 external interrupt
0 = Disables the INT0 external interrupt

bit 3 Unimplemented: Read as '0'

bit 2 **TMR0IF:** TMR0 Overflow Interrupt Flag bit
1 = TMR0 register has overflowed (must be cleared in software)
0 = TMR0 register did not overflow

bit 1 **INT0IF:** INT0 External Interrupt Flag bit
1 = The INT0 external interrupt occurred (must be cleared in software by reading PORTB)
0 = The INT0 external interrupt did not occur

bit 0 Unimplemented: Read as '0'

REGISTER 12-2: INTCON2: INTERRUPT CONTROL REGISTER 2

Read/Write (0xFF1) Bit	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0	
	$\overline{\text{PUD}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	—	INTCON2
	bit7							bit0	

bit 7 **PUD:** PORTB Pull-up Enable bit
1 = All PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG0:** External Interrupt 0 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 5 **INTEDG1:** External Interrupt 1 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 4 **INTEDG2:** External Interrupt 2 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 3 **INTEDG3:** External Interrupt 3 Edge Select bit
1 = Interrupt on rising edge
0 = Interrupt on falling edge

bit 2 **TMR0IP:** TMR0 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

bit 1 **INT3IP:** External Interrupt 3 Priority bit
1 = High priority
0 = Low priority

bit 0 **Unimplemented:** Read as '0'

REGISTER 12-3: INTCON3: INTERRUPT CONTROL REGISTER 3

Read/Write (0xFF0) Bit	R/W-1	R/W-1	R/W-1	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	INT2IP	INT1IP	INT3IE	INT2IE	INT1IE	INT3IF	INT2IF	INT1IF	INTCON3
	bit7							bit0	

bit 7 **INT2IP:** INT2 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **INT1IP:** INT1 External Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **INT3IE:** INT3 External Interrupt Enable bit
1 = Enables the INT3 external interrupt
0 = Disables the INT3 external interrupt

bit 4 **INT2IE:** INT2 External Interrupt Enable bit
1 = Enables the INT2 external interrupt
0 = Disables the INT2 external interrupt

bit 3 **INT1IE:** INT1 External Interrupt Enable bit
1 = Enables the INT1 external interrupt
0 = Disables the INT1 external interrupt

bit 1 **INT3IF:** INT3 External Interrupt Flag bit
1 = The INT3 external interrupt occurred (must be cleared in software)
0 = The INT3 external interrupt did not occur

bit 1 **INT2IF:** INT2 External Interrupt Flag bit

1 = The INT2 external interrupt occurred (must be cleared in software)
0 = The INT2 external interrupt did not occur

bit 0 **INT1IF:** INT1 External Interrupt Flag bit
1 = The INT1 external interrupt occurred (must be cleared in software)
0 = The INT1 external interrupt did not occur

12.2 PIR Registers

The Peripheral Interrupt Request (PIR) registers contain the individual flag bits for the peripheral interrupts (Register 12-4 through Register 12-6). Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Request (Flag) registers (PIR1, PIR2, PIR3).

Note 1: Interrupt flag bits are set when an interrupt condition occurs regardless of the state of its corresponding enable bit or the Global Interrupt Enable bit, GIE (INTCON register).

2: User software should ensure the appropriate interrupt flag bits are cleared prior to enabling an interrupt and after servicing that interrupt.

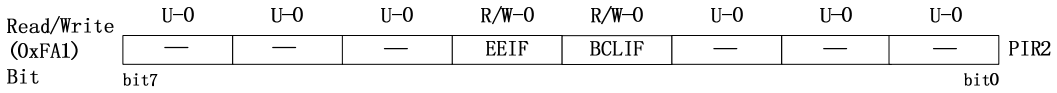
REGISTER 12-4: PIR1: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 1

Read/Write	U-0	U-0	R-0	R-0	U-0	R/W-0	R/W-0	R/W-0	
(0xF9E)	—	—	RCIF	TXIF	—	CCP1IF	TMR2IF	TMR1IF	PIR1
Bit	bit7							bit0	

bit 7 Unimplemented: Read as '0'
bit 6 Unimplemented: Read as '0'
bit 5 **RCIF:** USART Receive Interrupt Flag bit
1 = The USART receive buffer, RCREG, is full (cleared when RCREG is read)
0 = The USART receive buffer is empty
bit 4 **TXIF:** USART Transmit Interrupt Flag bit
1 = The USART transmit buffer, TXREG, is empty (cleared when TXREG is written)
0 = The USART transmit buffer is full
bit 3 Unimplemented: Read as '0'
bit 2 **CCP1IF:** CCP1 Interrupt Flag bit
Capture mode:
1 = A TMR1 register capture occurred (must be cleared in software)
0 = No TMR1 register capture occurred
Compare mode:
1 = A TMR1 register compare match occurred (must be cleared in software)
0 = No TMR1 register compare match occurred
PWM mode:
Unused in this mode.
bit 1 **TMR2IF:** TMR2 to PR2 Match Interrupt Flag bit
1 = TMR2 to PR2 match occurred (must be cleared in software)
0 = No TMR2 to PR2 match occurred

bit 0 **TMR1IF:** TMR1 Overflow Interrupt Flag bit
1 = TMR1 register overflowed (must be cleared in software)
0 = TMR1 register did not overflow

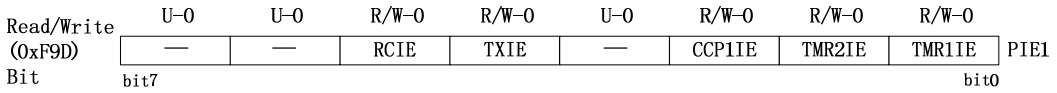
REGISTER 12-5: PIR2: PERIPHERAL INTERRUPT REQUEST (FLAG) REGISTER 2



bit 7-5 **Unimplemented:** Read as '0''
bit 4 **EEIF:** EEPROM Write Operation Interrupt Flag bit
1 =Write operation is complete (must be cleared in software)
0 =Write operation is not complete
bit 3 **BCLIF:** Bus Collision Interrupt Flag bit
1 = A bus collision occurred (must be cleared in software)
0 = No bus collision occurred
bit 2-0 **Unimplemented:** Read as '0''

12.3 PIE Registers

The Peripheral Interrupt Enable (PIE) registers contain the individual enable bits for the peripheral interrupts (Register 12-7 through Register 12-9). Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Enable registers (PIE1, PIE2). When IPEN is clear, the PEIE bit must be set to enable any of these peripheral interrupts.
REGISTER 12-7: PIE1: PERIPHERAL INTERRUPT ENABLE REGISTER 1



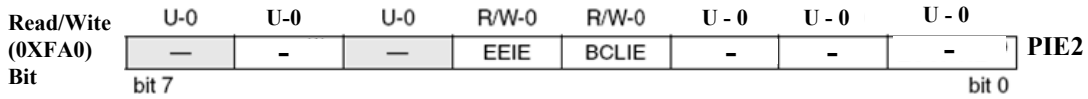
bit 7-6 '0'' **Unimplemented:** Read as
bit 5 **RCIE:** USART Receive Interrupt Enable bit
1 = Enables the USART receive interrupt
0 = Disables the USART receive interrupt
bit 4 **TXIE:** USART Transmit Interrupt Enable bit
1 = Enables the USART transmit interrupt
0 = Disables the USART transmit interrupt
bit 3 **Unimplemented:** Read as '0''
bit 2 **CCP1IE:** CCP1 Interrupt Enable bit
1 = Enables the CCP1 interrupt

0 = Disables the CCP1 interrupt

bit 1 **TMR2IE:** TMR2 to PR2 Match Interrupt Enable bit
1 = Enables the TMR2 to PR2 match interrupt
0 = Disables the TMR2 to PR2 match interrupt

bit 0 **TMR1IE:** TMR1 Overflow Interrupt Enable bit
1 = Enables the TMR1 overflow interrupt
0 = Disables the TMR1 overflow interrupt

REGISTER 12-8: PIE2: PERIPHERAL INTERRUPT ENABLE REGISTER 2



bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EEIE:** EEPROM Write Interrupt Enable bit
1 = Enabled
0 = Disabled

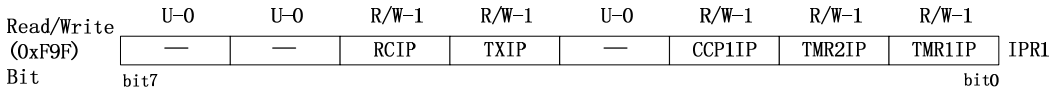
bit 3 **BCLIE:** Bus Collision Interrupt Enable bit
1 = Enabled
0 = Disabled

bit 2-0 **Unimplemented:** Read as '0'

12.4 IPR Registers

The Interrupt Priority (IPR) registers contain the individual priority bits for the peripheral interrupts. Due to the number of peripheral interrupt sources, there are three Peripheral Interrupt Priority registers (IPR1, IPR2). The operation of the priority bits requires that the Interrupt Priority Enable bit (IPEN) be set.

REGISTER 12-10: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1



bit 7-6 **Unimplemented:** Read as '0'

bit 5 **RCIP:** USART Receive Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **TXIP:** USART Transmit Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **Unimplemented:** Read as '0'

bit 2 **CCP1IP:** CCP1 Interrupt Priority bit
1 = High priority

0 = Low priority

bit 1 **TMR2IP:** TMR2 to PR2 Match Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **TMR1IP:** TMR1 Overflow Interrupt Priority bit
1 = High priority
0 = Low priority

REGISTER 12-11: IPR2: PERIPHERAL INTERRUPT PRIORITY REGISTER 2

Read/Write	U-0	U-0	U-0	R/W-1	R/W-1	U-0	U-0	U-0	
(0xFA2)	—	—	—	EEIP	BCLIP	—	—	—	IPR2
Bit	bit7							bit0	

bit 7-5 **Unimplemented:** Read as '0'

bit 4 **EEIP:** EEPROM Write Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **BCLIP:** Bus Collision Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2-0 **Unimplemented:** Read as '0'

12.5 RCON Register

The Reset Control (RCON) register contains the IPEN bit which is used to enable prioritized interrupts.

BIT	7	6	5	4	3	2	1	0	
(0XFD0)	IPEN	CKOE	-	RI	TO	PD	POR	BOR	RCON
Read/Write	R/W	R/W	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	1	1	1	0	0	

bit 7 **IPEN:** Interrupt Priority Enable bit
1 = Enable priority levels on interrupts
0 = Disable priority levels on interrupts (PIC16CXXX Compatibility mode)

bit 6 **CKOE:** Clock Output Enable bit
When this bit is written to one, the CPU clock divided by 4 is output on the PB1 pin.

bit 5 **Unimplemented:** Read as '0'

bit 4 **RI:** RESET Instruction Flag bit
1 = The RESET instruction was not executed
0 = The RESET instruction was executed causing a device Reset
(must be set in software after a Brown-out Reset occurs)

bit 3 **TO:** Watchdog Time-out Flag bit

1 = After power-up, CLRWDT instruction or SLEEP instruction

0 = A WDT time-out occurred

bit 2 **PD:** Power-down Detection Flag bit

1 = After power-up or by the CLRWDT instruction

0 = By execution of the SLEEP instruction

bit 1 **POR:** Power-on Reset Status bit

1 = A Power-on Reset has not occurred

0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **BOR:** Brown-out Reset Status bit

1 = A Brown-out Reset has not occurred

0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

12.6 INT Interrupts

External interrupts on the PA4/INT0, PA5/INT1 , PA6/INT2 and PA7/INT3 pins are edge triggered: either rising if the corresponding INTEDGx bit is set in the INTCON2 register, or falling if the INTEDGx bit is clear. When a valid edge appears on the PAx/INTx pin, the corresponding flag bit INTxIF is set. This interrupt can be disabled by clearing the corresponding enable bit INTxIE. Flag bit INTxIF must be cleared in software in the Interrupt Service Routine before re-enabling the interrupt. All external interrupts (INT0, INT1, INT2 and INT3) can wake-up the processor from Sleep if bit INTxIE was set prior to going into Sleep. If the Global Interrupt Enable bit, GIE, is set, the processor will branch to the interrupt vector following wake-up. Interrupt priority for INT1 and INT2 is determined by the value contained in the interrupt priority bits INT1IP (INTCON3<6>) ,INT2IP (INTCON3<7>) and INT3IP(INTCON2<1>). There is no priority bit associated with INT0; it is always a high priority interrupt source.

12.7 TMR0 Interrupt

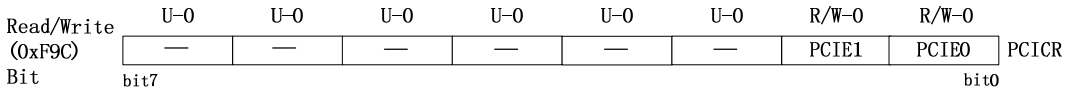
In 8-bit mode (which is the default), an overflow (FFh → 00h) in the TMR0 register will set flag bit TMR0IF. In 16-bit mode, an overflow (FFFFh →0000h) in the TMR0H:TMR0L registers will set flag bit TMR0IF. The interrupt can be enabled/disabled by setting/clearing enable bit TMR0IE (INTCON register). Interrupt priority for Timer0 is determined by the value contained in the interrupt priority bit TMR0IP (INTCON2 register).

12.8 PORTA and PORTB Interrupt-on-Change

An input change on PA<7:0> and PB<7:0> sets flag bit PCIF0 and PCIF1 (PCIFR register). The

interrupt can be enabled/ disabled by setting/clearing enable bit PCIE0 and PCIE1 (PCICR register). Each PCINT15:0-bit selects whether pin change interrupt is enabled on the corresponding I/O pin by the PCMASK0 and PCMASK1 register. There is no priority bit associated with PCINT15:0; it is always a high priority interrupt source.

12.8.1 PCICR– Pin Change Interrupt Control Register



• Bit 7:2 - Res: Reserved Bits

These bits are reserved bits in the SE1100, and will always read as zero.

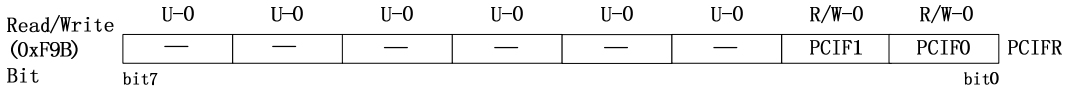
• Bit 1 - PCIE1: Pin Change Interrupt Enable 1

When the PCIE1 bit is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT15:8 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI1 Interrupt Vector. PCINT15:8 pins are enabled individually by the PCMSK1 Register.

• Bit 0 - PCIE0: Pin Change Interrupt Enable 0

When the PCIE0 bit is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT7:0 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI0 Interrupt Vector. PCINT7:0 pins are enabled individually by the PCMSK0 Register.

12.8.2 PCIFR – Pin Change Interrupt Flag Register



• Bit 7:2 - Res: Reserved Bits

These bits are reserved bits in the ATmega406, and will always read as zero.

• Bit 1 - PCIF1: Pin Change Interrupt Flag 1

When a logic change on any PCINT15:8 pin triggers an interrupt request, PCIF1 becomes set (one). The flag can be cleared by writing a logical zero to it.

• Bit 0 - PCIF0: Pin Change Interrupt Flag 0

When a logic change on any PCINT7:0 pin triggers an interrupt request, PCIF0 becomes set (one). The flag can be cleared by writing a logical zero to it.

12.8.3 PCMSK1 – Pin Change Mask Register 1

Read/Write (0xF9A) Bit	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	PCINT15	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
	bit7							bit0	

• Bit 7:0 – PCINT15:8: Pin Change Enable Mask 15:8

Each PCINT15:11-bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT15:8 is set and the PCIE1 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT15:8 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

12.8.4 PCMSK0 – Pin Change Mask Register 0

Read/Write (0xF99) Bit	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
	bit7							bit0	

• Bit 7:0 – PCINT7:0: Pin Change Enable Mask 7:0

Each PCINT7:0 bit selects whether pin change interrupt is enabled on the corresponding I/O pin. If PCINT7:0 is set and the PCIE0 bit in PCICR is set, pin change interrupt is enabled on the corresponding I/O pin. If PCINT7:0 is cleared, pin change interrupt on the corresponding I/O pin is disabled.

12.9 Context Saving During Interrupts

During an interrupt, the return PC value is saved on the stack. Additionally, the WREG, Status and BSR registers are saved on the fast return stack. If a fast return from interrupt is not used, the user may need to save the WREG, Status and BSR registers in software. Depending on the user's application, other registers may also need to be saved. Example 11-1 saves and restores the WREG, Status and BSR registers during an Interrupt Service Routine.

EXAMPLE 11-1: SAVING STATUS, WREG AND BSR REGISTERS IN RAM

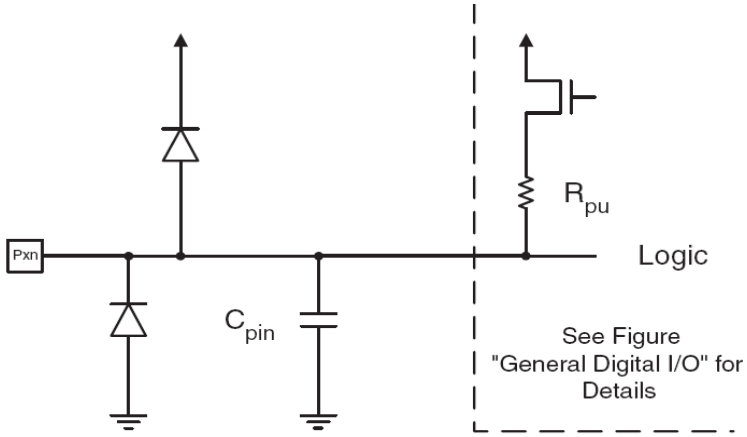
```
MOVWF       W_TEMP                  ; W_TEMP is in Low Access bank
MOVFF       STATUS, STATUS_TEMP     ; STATUS_TEMP located anywhere
MOVFF       BSR, BSR_TEMP           ; BSR located anywhere
;
; USER ISR CODE
;
MOVFF       BSR_TEMP, BSR           ; Restore BSR
MOVF        W_TEMP, W               ; Restore WREG
MOVFF       STATUS_TEMP, STATUS     ; Restore STATUS
```

13. Low Voltage I/O-Ports

13.1 Introduction

All low voltage SE1100 ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). All low voltage port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both VREG and Ground as indicated in Figure 12-1 on page 60.

Figure 12-1. Low Voltage I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn.

Three I/O memory address locations are allocated for each low voltage port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. However, writing a logic one to a bit in the PINx Register, will result in a toggle in the corresponding bit in the Data Register. In addition, the Pull-up Disable – PUD bit in INT2CON disables the pull-up function for all low voltage pins in all ports when set.

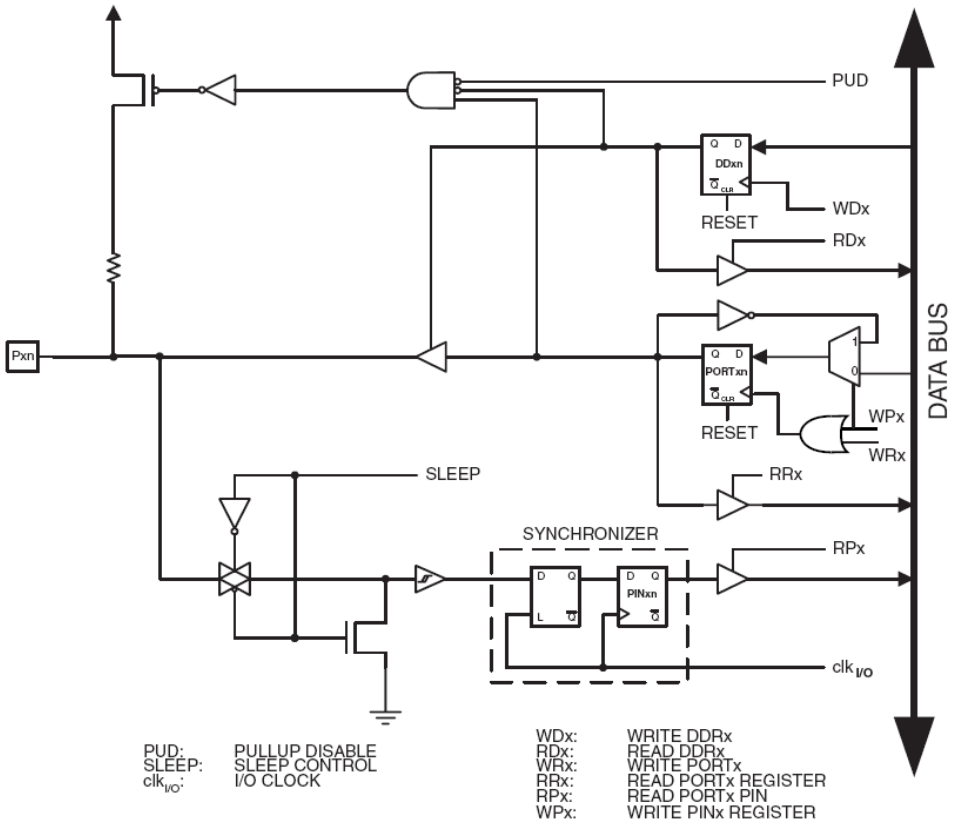
Using the I/O port as General Digital I/O is described in ”Low Voltage Ports as General Digital I/O”. Many low voltage port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in ”Alternate Port Functions”. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

13.2 Low Voltage Ports as General Digital I/O

The low voltage ports are bi-directional I/O ports with optional internal pull-ups. Figure 12-2 shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 12-2. General Low Voltage Digital I/O



Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. clkI/O, SLEEP, and PUD are common to all ports.

13.2.1 Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. The DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is

activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

13.2.3 Switching Between Input and Output

When switching between tri-state ({DDxn, PORTxn} = 0b00) and output high ({DDxn, PORTxn} = 0b11), an intermediate state with either pull-up enabled {DDxn, PORTxn} = 0b01) or output low ({DDxn, PORTxn} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedant environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b11) as an intermediate step.
Table 12-1 summarizes the control signals for the pin value.
Table 12-1. Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

13.2.4 Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register bit. The PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid metastability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay.

13.2.5 Digital Input Enable and Sleep Modes

As shown in Figure 12-2, the digital input signal can be clamped to ground at the input of the schmitt-trigger. The signal denoted SLEEP in the figure, is set by the MCU Sleep Controller in

Power-down mode and Power-save mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to VREG/2.

SLEEP is overridden for port pins enabled as external interrupt pins. If the external interrupt request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in ”Alternate Port Functions” .

If a logic high level (“one”) is present on an asynchronous external interrupt pin configured as “Interrupt on Rising Edge, Falling Edge, or Any Logic Change on Pin” while the external interrupt is *not* enabled, the corresponding External Interrupt Flag will be set when resuming from the above mentioned Sleep mode, as the clamping in these sleep mode produces the requested logic change.

13.2.6 Unconnected Pins

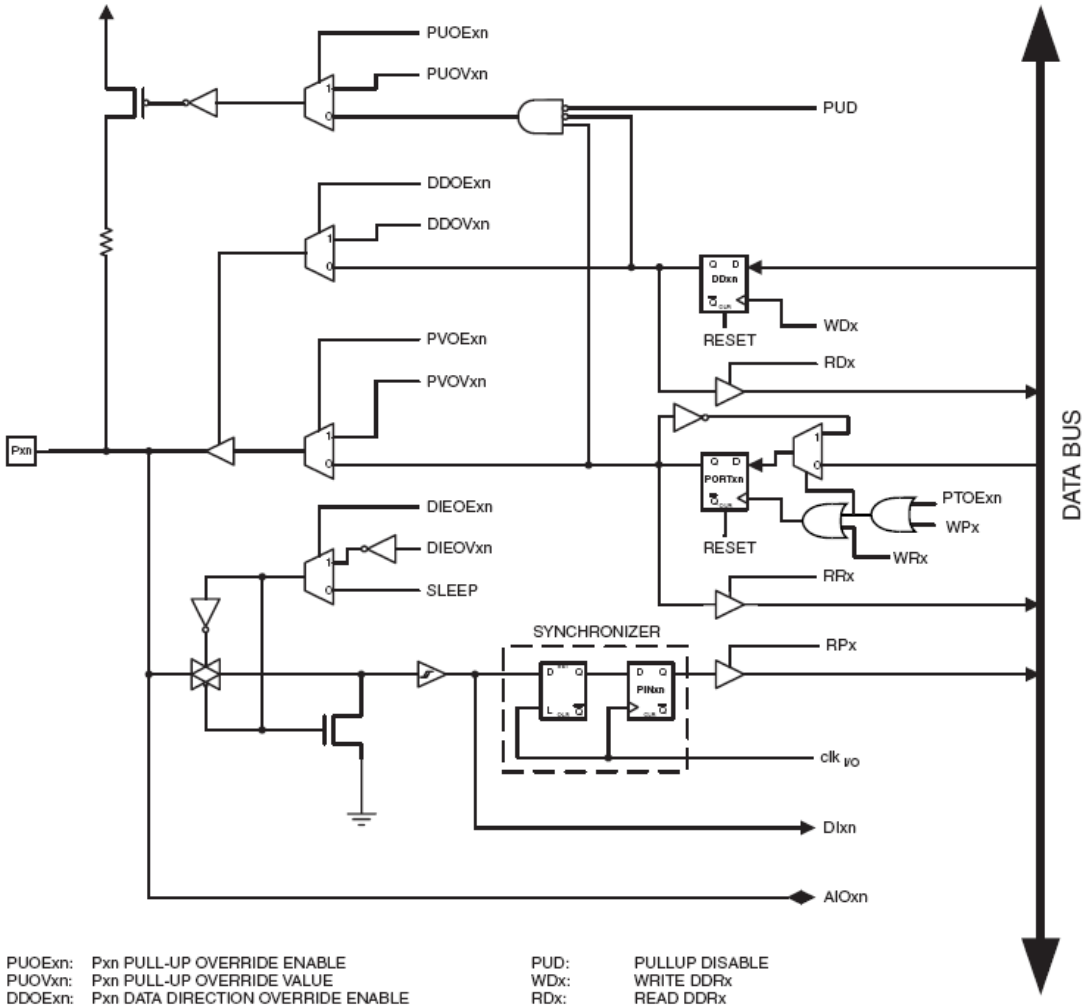
If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset, Active mode and Idle mode).

The simplest method to ensure a defined level of an unused pin, is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to VCC or GND is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

13.3 Alternate Port Functions

Many low voltage port pins have alternate functions in addition to being general digital I/Os. Figure 12-5 shows how the port pin control signals from the simplified Figure 12-2 can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the figure serves as a generic description applicable to all port pins in the SE1100.

Figure 12-5. Alternate Port Functions



PUOExn:	Pxn PULL-UP OVERRIDE ENABLE	PUD:	PULLUP DISABLE
PUOVxn:	Pxn PULL-UP OVERRIDE VALUE	WDx:	WRITE DDRx
DDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE	RDx:	READ DDRx
DDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE	RRx:	READ PORTx REGISTER
PVOExn:	Pxn PORT VALUE OVERRIDE ENABLE	WRx:	WRITE PORTx
PVOVxn:	Pxn PORT VALUE OVERRIDE VALUE	RPx:	READ PORTx PIN
DIEOExn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE ENABLE	WPx:	WRITE PINx
DIEOVxn:	Pxn DIGITAL INPUT-ENABLE OVERRIDE VALUE	$clk_{I/O}$:	I/O CLOCK
SLEEP:	SLEEP CONTROL	DIxn:	DIGITAL INPUT PIN n ON PORTx
PTOExn:	Pxn, PORT TOGGLE OVERRIDE ENABLE	AIOxn:	ANALOG INPUT/OUTPUT PIN n ON PORTx

Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. clkI/O,
SLEEP, and PUD are common to all ports. All other signals are unique for each pin.

Table 12-2 summarizes the function of the overriding signals. The pin and port indexes from Figure 12-5 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

Table 12-2. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUOE	Pull-up Override Enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-up Override Value	If PUOE is set, the pull-up is enabled/disabled when PUOV is set/cleared, regardless of the setting of the DDxn, PORTxn, and PUD Register bits.
DDOE	Data Direction Override Enable	If this signal is set, the Output Driver Enable is controlled by the DDOV signal. If this signal is cleared, the Output driver is enabled by the DDxn Register bit.
DDOV	Data Direction Override Value	If DDOE is set, the Output Driver is enabled/disabled when DDOV is set/cleared, regardless of the setting of the DDxn Register bit.
PVOE	Port Value Override Enable	If this signal is set and the Output Driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the Output Driver is enabled, the port Value is controlled by the PORTxn Register bit.
PVOV	Port Value Override Value	If PVOE is set, the port value is set to PVOV, regardless of the setting of the PORTxn Register bit.
PTOE	Port Toggle Override Enable	If PTOE is set, the PORTxn Register bit is inverted.
DIEOE	Digital Input Enable Override Enable	If this bit is set, the Digital Input Enable is controlled by the DIEOV signal. If this signal is cleared, the Digital Input Enable is determined by MCU state (Normal mode, sleep mode).
DIEOV	Digital Input Enable Override Value	If DIEOE is set, the Digital Input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state (Normal mode, sleep mode).
DI	Digital Input	This is the Digital Input to alternate functions. In the figure, the signal is connected to the output of the schmitt trigger but before the synchronizer. Unless the Digital Input is used as a clock source, the module with the alternate function will use its own synchronizer.
AIO	Analog Input/Output	This is the Analog Input/output to/from alternate functions. The signal is connected directly to the pad, and can be used bi-directionally.

The following subsections shortly describe the alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

13.3.1 Alternate Functions of Port A

The Port A has an alternate function as input pins to the Voltage ADC.

Table 12-3. Port A Pins Alternate Functions

Port Pin	Alternate Function
PA7	INT3 (External Interrupt 3) PCINT7 (Pin Change Interrupt 7)
PA6	INT2 (External Interrupt 2) PCINT6 (Pin Change Interrupt 6)
PA5	INT1 (External Interrupt 1) PCINT5 (Pin Change Interrupt 5)
PA4	ADC4 (ADC Input Channel 4) INT0 (External Interrupt 0) PCINT4 (Pin Change Interrupt 4)
PA3	ADC3 (ADC Input Channel 3) PCINT3 (Pin Change Interrupt 3)
PA2	ADC2 (ADC Input Channel 2) PCINT2 (Pin Change Interrupt 2)
PA1	ADC1 (ADC Input Channel 1) PCINT1 (Pin Change Interrupt 1)
PA0	ADC0 (ADC Input Channel 0) PCINT0 (Pin Change Interrupt 0)

The alternate pin configuration is as follows:

• **ADC4/INT3:0/PCINT7:4 – Port A, Bit 7:4**

Analog to Digital Converter, Channel 4.

INT3 - INT0, External Interrupt Sources 3:0. The PA7:4 pins can serve as external interrupt sources to the MCU.

PCINT7 - PCINT4, Pin Change Interrupt Sources 7:4. The PA7:4 pins can serve as external interrupt sources to the MCU.

• **ADC3:0/PCINT3:0 – Port A, Bit 3:0**

Analog to Digital Converter, Channels 3:0.

PCINT3 - PCINT0, Pin Change Interrupt Sources 3:0. The PA3:0 pins can serve as external interrupt sources to the MCU.

Table 12-4 and Table 12-5 relates the alternate functions of Port A to the overriding signals shown in Figure 12-5 on page 66.

Table 12-4. Overriding Signals for Alternate Functions in PA7:PA4

Signal Name	PA7/INT3/ PCINT7	PA6/INT2/ PCINT6	PA5/INT1/ PCINT5	PA4/ADC4 INT0/PCINT4
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
PTOE	–	–	–	–
DIEOE	INT3 ENABLE	INT2 ENABLE	INT1 ENABLE	INT0 ENABLE
DIEOV	INT3 ENABLE	INT2 ENABLE	INT1 ENABLE	INT0 ENABLE
DI	INT3 INPUT/ PCINT7 INPUT	INT2 INPUT/ PCINT6 INPUT	INT1 INPUT/ PCINT5 INPUT	INT0 INPUT/ PCINT4 INPUT
AIO	–	–	–	ADC4 INPUT

Table 12-5. Overriding Signals for Alternate Functions in PA3:PA0

Signal Name	PA3/ADC3/ PCINT3	PA2/ADC2/ PCINT2	PA1/ADC1/ PCINT1	PA0/ADC0/ PCINT0
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	0	0
PVOV	0	0	0	0
PTOE	–	–	–	–
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	PCINT3 INPUT	PCINT2 INPUT	PCINT1 INPUT	PCINT0 INPUT
AIO	ADC3 INPUT	ADC2 INPUT	ADC1 INPUT	ADC0 INPUT

13.3.2 Alternate Functions of Port B

The Port B pins with alternate functions are shown in Table 12-6.

Table 12-6. Port B Pins Alternate Functions

Port Pin	**Alternate Functions**
PB7	PCINT15 (Pin Change Interrupt 15)
PB6	CCP1 (Output CCP module) PCINT14 (Pin Change Interrupt 14)
PB5	RX(UART receive signal) PCINT13 (Pin Change Interrupt 13)
PB4	TX(UART transmit signal) PCINT12 (Pin Change Interrupt 12)
PB3	TCK (JTAG Test Clock) PCINT11 (Pin Change Interrupt 11)
PB2	TMS (JTAG Test Mode Select) PCINT10 (Pin Change Interrupt 10)
PB1	TDI (JTAG Test Data Input/) PCINT9 (Pin Change Interrupt 9)
PB0	TDO (JTAG Test Data Output) PCINT8 (Pin Change Interrupt 8)

The alternate pin configuration is as follows:

• PCINT15 – Port B, Bit 7

PCINT15, Pin Change Interrupt Source 15. The PB7 pin can serve as external interrupt source to the MCU.

• CCP1/PCINT14 – Port B, Bit 6

CCP1 Output Compare Match A output: The PB6 pin can serve as an external output for the CCP Output Compare. The pin has to be configured as an output (DDB6 set (one)) to serve this function. The CCP1 pin is also the output pin for the PWM mode timer function.

PCINT14, Pin Change Interrupt Source 13. The PB6 pin can serve as external interrupt source to the MCU.

• **RX/PCINT13 – Port B, Bit 5**

RX is input in UART: The PB5 pin can serve as an external input for the UART. The pin has to be configured as an input (DDB5 clear (zero)) to serve this function.

PCINT13, Pin Change Interrupt Source 12. The PB5 pin can serve as external interrupt source to the MCU.

• **TX/PCINT12 – Port B, Bit 4**

TX is input in UART: The PB4 pin can serve as an external output for the UART. The pin has to be configured as an output (DDB4 set (one)) to serve this function.

PCINT12, Pin Change Interrupt Source 12. The PB4 pin can serve as external interrupt source to the MCU.

• **TCK/PCINT11 – Port B, Bit 3**

TCK, JTAG Test Clock: JTAG operation is synchronous to TCK. When the JTAG Interface is enabled, this pin can not be used as an I/O pin.

PCINT11, Pin Change Interrupt Source 11. The PB3 pin can serve as external interrupt source to the MCU.

• **TMS/PCINT10 – Port B, Bit 2**

TMS, JTAG Test Mode Select: This pin is used for navigating through the TAP-controller state machine. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

PCINT10, Pin Change Interrupt Source 10. The PB2 pin can serve as external interrupt source to the MCU.

• **TDI/PCINT9 – Port B, Bit 1**

TDI, JTAG Test Data Input: Serial input data to be shifted in the Instruction Register or Data Register (scan chains). When the JTAG Interface is enabled, this pin can not be used as I/O pin.

PCINT9, Pin Change Interrupt Source 9. The PB1 pin can serve as external interrupt source to the MCU.

• **TDO/PCINT8 – Port B, Bit 0**

TDO, JTAG Test Data Output: Serial output data from Instruction Register or Data Register. When the JTAG Interface is enabled, this pin can not be used as an I/O pin.

PCINT8, Pin Change Interrupt Source 8. The PB0 pin can serve as external interrupt source to the MCU.

Table 12-7 and Table 12-8 relate the alternate functions of Port B to the overriding signals shown in Figure 12-5

Table 12-7. Overriding Signals for Alternate Functions in PB7:PB4

Signal Name	PB7/OCOB/ PCINT15	PB6/OCOA/ PCINT14	PB5/ PCINT13	PB4/ PCINT12
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	SPEN	SPEN
DDOV	0	0	0	SPEN
PVOE	0	CCP1 Enable	0	SPEN
PVOV	0	CCP1	0	TX
PTOE	–	–	–	–
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	PCINT15 INPUT	PCINT14 INPUT	PCINT13 INPUT/RX	PCINT12 INPUT
AIO	–	–	–	–

Table 12-8. Overriding Signals for Alternate Functions in PB3:PB0

Signal Name	PB3/TCK/ PCINT11	PB2/TMS/ PCINT10	PB1/TDI/ PCINT9	PB0/TDO/ PCINT8
PUOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
PUOV	1	1	1	0
DDOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
DDOV	0	0	0	SHIFT_IR + SHIFT_DR
PVOE	0	0	0	JTAGEN
PVOV	0	0	0	TDO
PTOE	–	–	–	–
DIEOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
DIEOV	0	0	0	0
DI	TCK/PCINT11 INPUT	TMS/PCINT10 INPUT	TDI/PCINT9 INPUT	PCINT8 INPUT
AIO	–	–	–	–

13.3.3 Alternate Functions of Port D

The Port D pins with alternate functions are shown in Table 12-9.

Table 12-9. Port D Pins Alternate Functions

Port Pin	**Alternate Functions**
PD1	T1CKI (Timer1 Clock Input)
PD0	T0CKI (Timer0 Clock Input)

• T1CKI – Port D, Bit 1

T1CKI, Time1 Counter Source.

• T0CKI – Port D, Bit 0

T0CKI, Timer0 Counter Source.

Table 12-10 on page 73 relates the alternate functions of Port D to the overriding signals shown in Figure 12-5 .

Table 12-10. Overriding Signals for Alternate Functions in PD1:PD0

Signal Name	PD1/T1	PD0/T0
PUOE	0	0
PUOV	0	0
DDOE	T1OSCEN	T1OSCEN
DDOV	0	0
PVOE	0	0
PVOV	0	0
PTOE	–	–
DIEOE	0	0
DIEOV	0	0
DI	T1CKI	T0CKI
AIO	–	–

13.4 Register Description

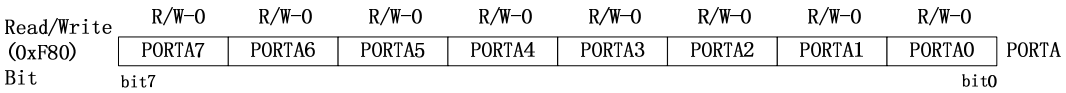
13.4.1 INTCON2 – INTERRUPT CONTROL REGISTER

Read/Write	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	U-0	
(0xFF1)	$\overline{\text{PUD}}$	INTEDG0	INTEDG1	INTEDG2	INTEDG3	TMR0IP	INT3IP	—	INTCON2
Bit	bit7							bit0	

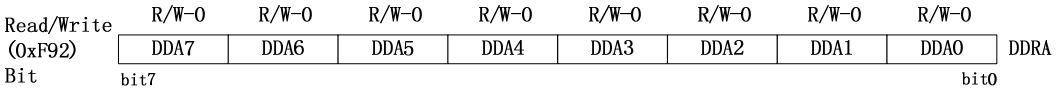
• Bit 7 – PUD: Pull-up Disable

When this bit is written to one, the pull-ups in the I/O ports are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-ups ({DDxn, PORTxn} = 0b01). S

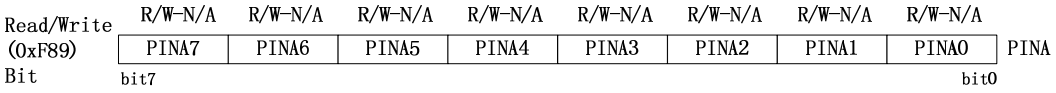
13.4.2 PORTA – Port A Data Register



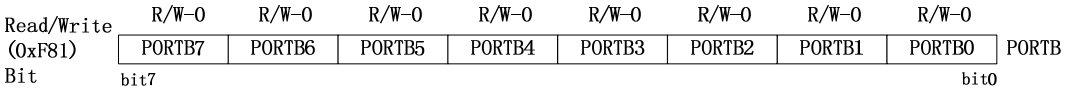
13.4.3 DDRA – Port A Data Direction Register



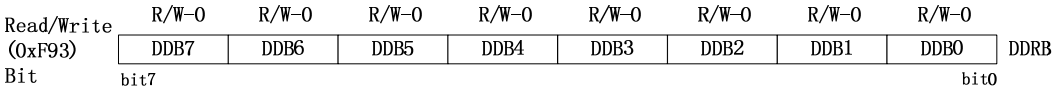
13.4.4 PINA – Port A Input Pins Address



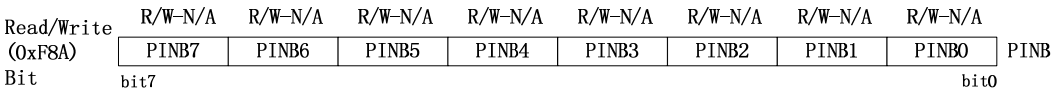
13.4.5 PORTB – Port B Data Register



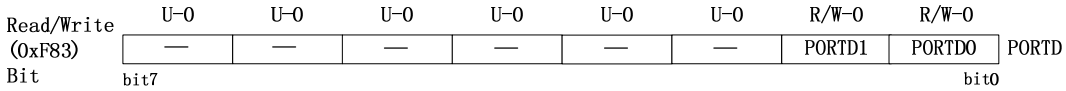
13.4.6 DDRB – Port B Data Direction Register



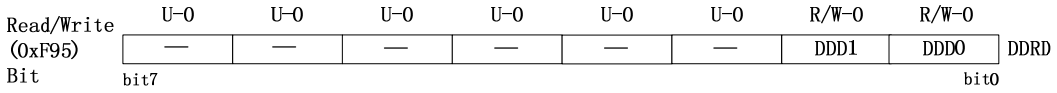
13.4.7 PINB – Port B Input Pins Address



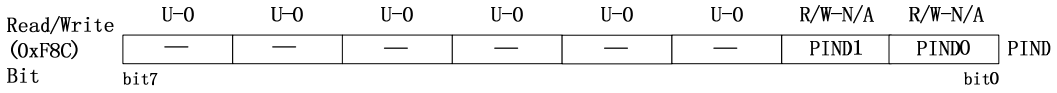
13.4.8 PORTD – Port D Data Register



13.4.9 DDRD – Port D Data Direction Register



13.4.10 PIND – Port D Input Pins Address



13.4.11 PORTE – Port E Data Register

BIT	7	6	5	4	3	2	1	0	
(0XF84)	PORTE7	PORTE6	-	-	-	-	-	-	PORTE
Read/Write	R/W	R/W	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	1	1	1	0	0	

13.4.12 DDRE – Port E Data Direction Register

BIT	7	6	5	4	3	2	1	0	
(0XF96)	DDRE7	DDRE6	-	-	-	-	-	-	DDRE
Read/Write	R/W	R/W	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	1	1	1	0	0	

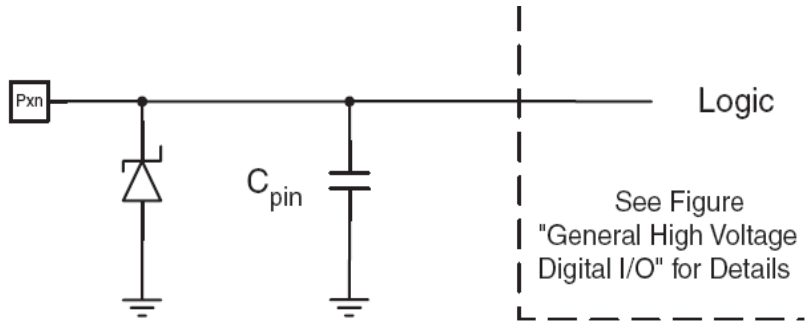
13.4.13 PINE – Port E Input Pins Address

BIT	7	6	5	4	3	2	1	0	
(0XF84)	PINE7	PINE6	-	-	-	-	-	-	PINE
Read/Write	R/W	R/W	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	1	1	1	0	0	

14. High Voltage I/O Ports

All high voltage SE1100 ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the state of one port pin can be changed without unintentionally changing the state of any other pin. All high voltage I/O pins have protection Zener diodes to Ground as indicated in Figure 14-1.

Figure 14-1. High Voltage I/O Pin Equivalent Schematic



All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTC3 for bit number three in Port C, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in ”Register Description for High Voltage Output Ports” .

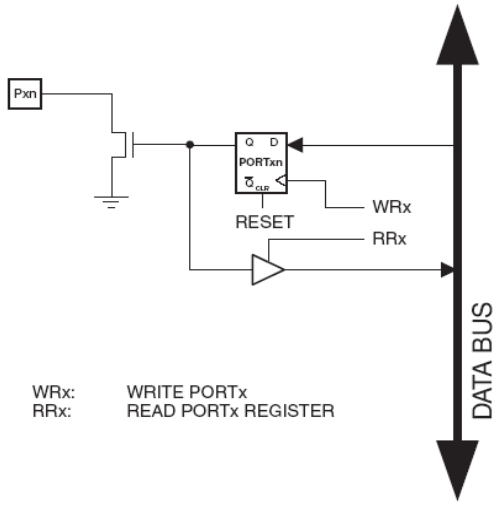
One I/O Memory address location is allocated for each high voltage port, the Data Register – PORTx. The Data Register is read/write.

Using the I/O port as General Digital Output is described in ”High Voltage Ports as General Digital Outputs” .

14.1 High Voltage Ports as General Digital Outputs

The high voltage ports are high voltage tolerant open collector output ports. Figure 14-2 shows a functional description of one output port pin, here generically called Pxn.

Figure 14-2. General High Voltage Digital I/O



Note: 1. WRx and RRx are common to all pins within the same port.

14.2 Configuring the Pin

Each port pin has one register bit: PORTxn. As shown in "Register Description for High Voltage Output Ports", the PORTxn bits are accessed at the PORTx I/O address. If PORTxn is written logic one, the port pin is driven low (zero). If PORTxn is written logic zero, the port pin is tri-stated. The port pins are tri-stated when a reset condition becomes active, even if no clocks are running.

14.3 Register Description for High Voltage Output Ports

14.3.1 PORTC – Port C Data Register

Read/Write	U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	
(0xF82)	—	—	—	—	—	—	—	PORTC0	PORTC
Bit	bit7							bit0	

15. TIMER0 MODULE

The Timer0 module has the following features:

• Software selectable as an 8-bit or 16-bit timer/counter
• Readable and writable
• Dedicated 8-bit software programmable prescaler
• Clock source selectable to be external or internal
• Interrupt-on-overflow from FFh to 00h in 8-bit mode and FFFFh to 0000h in 16-bit mode
• Edge select for external clock

Register 15-1 shows the Timer0 Control register (T0CON). Figure 15-1 shows a simplified block diagram of the Timer0 module in 8-bit mode and Figure 15-2 shows a simplified block diagram of the Timer0 module in 16-bit mode. The T0CON register is a readable and writable register that controls all the aspects of Timer0, including the prescale selection.

REGISTER 15-1: T0CON: TIMER0 CONTROL REGISTER

Read/Wite (0XFD5) Bit	R/W-0	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	
	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	**T0CON**
	bit 7							bit 0	

bit 7 **TMR0ON:** Timer0 On/Off Control bit
1 = Enables Timer0
0 = Stops Timer0

bit 6 **T08BIT:** Timer0 8-bit/16-bit Control bit
1 = Timer0 is configured as an 8-bit timer/counter
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS:** Timer0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKO)

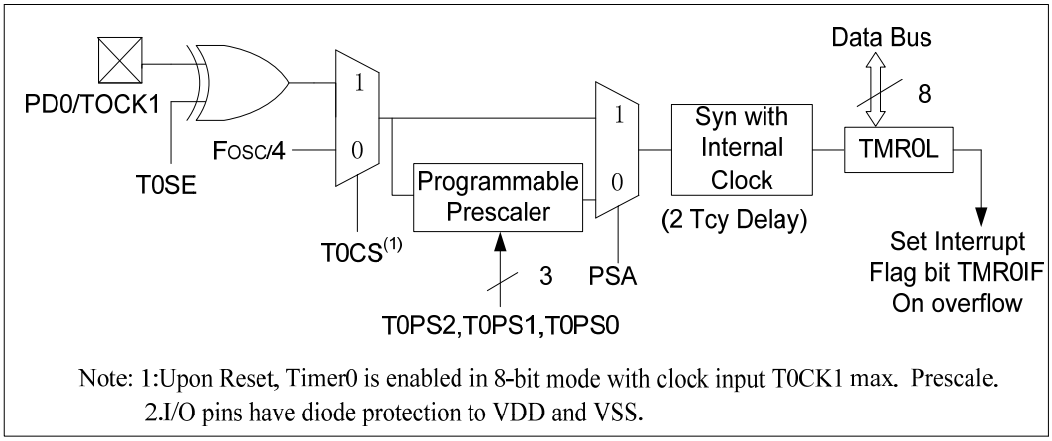
bit 4 **T0SE:** Timer0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Timer0 Prescaler Assignment bit
1 = TImer0 prescaler is not assigned. Timer0 clock input bypasses prescaler.
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS2:**T0PS0: Timer0 Prescaler Select bits
111 = 1:256 Prescale value
110 = 1:128 Prescale value
101 = 1:64 Prescale value
100 = 1:32 Prescale value

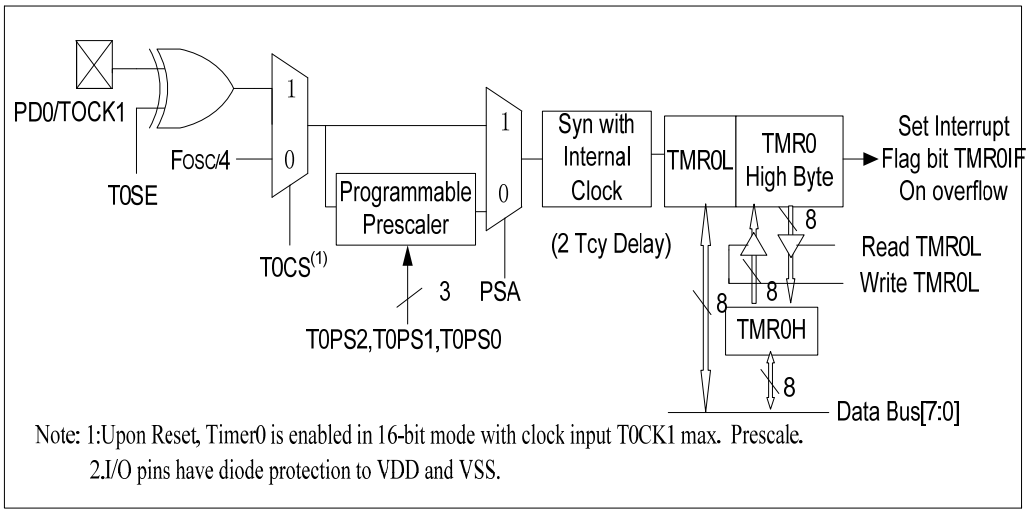
011 = 1:16 Prescale value
010 = 1:8 Prescale value
001 = 1:4 Prescale value
000 = 1:2 Prescale value

FIGURE 15-1: TIMER0 BLOCK DIAGRAM IN 8-BIT MODE



Note: 1:Upon Reset, Timer0 is enabled in 8-bit mode with clock input T0CK1 max. Prescale.
2.I/O pins have diode protection to VDD and VSS.

FIGURE 15-2: TIMER0 BLOCK DIAGRAM IN 16-BIT MODE



Note: 1:Upon Reset, Timer0 is enabled in 16-bit mode with clock input T0CK1 max. Prescale.
2.I/O pins have diode protection to VDD and VSS.

15.1 Timer0 Operation

Timer0 can operate as a timer or as a counter. Timer mode is selected by clearing the T0CS bit. In Timer mode, the Timer0 module will increment every instruction cycle (without prescaler). If the TMR0L register is written, the increment is inhibited for the following two instruction cycles. The user can work around this by writing an adjusted value to the TMR0L register. Counter mode is selected by setting the T0CS bit. In Counter mode, Timer0 will increment either on every rising or falling edge of pin RA4/T0CKI. The incrementing edge is determined by the Timer0 Source Edge Select bit (T0SE). Clearing the T0SE bit selects the rising edge. Restrictions on the external clock input are discussed below. When an external clock input is used for Timer0, it must meet certain

requirements. The requirements ensure the external clock can be synchronized with the internal phase clock (TOSC). Also, there is a delay in the actual incrementing of Timer0 after synchronization.

15.2 Prescaler

An 8-bit counter is available as a prescaler for the Timer0 module. The prescaler is not readable or writable.
The PSA and T0PS2:T0PS0 bits determine the prescaler assignment and prescale ratio. Clearing bit PSA will assign the prescaler to the Timer0 module. When the prescaler is assigned to the Timer0 module, prescale values of 1:2, 1:4, ..., 1:256 are selectable. When assigned to the Timer0 module, all instructions writing to the TMR0 register (e.g., CLRF TMR0, MOVWF TMR0, BSF TMR0, x.... etc.) will clear the prescaler count.

15.2.1 SWITCHING PRESCALER ASSIGNMENT

The prescaler assignment is fully under software control (i.e., it can be changed “on-the-fly” during program execution).

15.3 Timer0 Interrupt

The TMR0 interrupt is generated when the TMR0 register overflows from FFh to 00h in 8-bit mode or FFFFh to 0000h in 16-bit mode. This overflow sets the TMR0IF bit. The interrupt can be masked by clearing the TMR0IE bit. The TMR0IF bit must be cleared in software by the Timer0 module Interrupt Service Routine before re-enabling this interrupt. The TMR0 interrupt cannot awaken the processor from Sleep since the timer is shut-off during Sleep.

15.4 16-Bit Mode Timer Reads and Writes

Timer0 can be set in 16-bit mode by clearing the T08BIT in T0CON. Registers TMR0H and TMR0L are used to access the 16-bit timer value. TMR0H is not the high byte of the timer/counter in 16-bit mode, but is actually a buffered version of the high byte of Timer0 (refer to Figure 15-2). The high byte of the Timer0 timer/counter is not directly readable nor writable. TMR0H is updated with the contents of the high byte of Timer0 during a read of TMR0L. This provides the ability to read all 16 bits of Timer0 without having to verify that the read of the high and low byte were valid, due to a rollover between successive reads of the high and low byte. A write to the high byte of Timer0 must also take place through the TMR0H Buffer register. Timer0 high byte is updated with the contents of the buffered value of TMR0H when a write occurs to TMR0L. This allows all 16 bits of Timer0 to be updated at once.

Note: Writing to TMR0 when the prescaler is assigned to Timer0 will clear the prescaler count but will not change the prescaler assignment.

TABLE 15-1: REGISTERS ASSOCIATED WITH TIMER0

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
TMR0L	Timer0 Module Low Byte Register								xxxx xxxx	uuuu uuuu
TMR0H	Timer0 Module High Byte Register								0000 0000	0000 0000
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	1111 1111	1111 1111
TRISA	—	PORTA Data Direction Register(1)							-111 1111	-111 1111

16. TIMER1 MODULE

The Timer1 module timer/counter has the following features:

• 16-bit timer/counter(two 8-bit registers: TMR1H and TMR1L)

• Readable and writable (both registers)

• Internal or external clock select

• Interrupt-on-overflow from FFFFh to 0000h

• Reset from CCP module special event trigger

Register 16-1 shows the Timer1 Control register. This register controls the operating mode of the Timer1 module, as well as contains the Timer1 Oscillator Enable bit (T1OSCEN). Timer1 can be enabled/disabled by setting/clearing control bit, TMR1ON(T1CON register).

Figure 16-1 is a simplified block diagram of the Timer1 module.

Note: Timer1 is disabled on POR.

REGISTER 16-1: T1CON: TIMER1 CONTROL REGISTER

Read/Wite (0XFCD) Bit	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	**T1CON**
	bit 7							bit 0	

bit 7 **RD16**: 16-bit Read/Write Mode Enable bit

1 = Enables register read/write of Timer1 in one 16-bit operation

0 = Enables register read/write of Timer1 in two 8-bit operations

bit 6 Unimplemented: Read as '0'

bit 5-4 **T1CKPS1**:T1CKPS0: Timer1 Input Clock Prescale Select bits

11 = 1:8 Prescale value

10 = 1:4 Prescale value

01 = 1:2 Prescale value

00 = 1:1 Prescale value

bit 3 **T1OSCEN**: Timer1 Oscillator Enable bit

1 = Timer1 128K ULP OSC is enabled

0 = Timer1 128K ULP OSC is shutoff

bit 2 **T1SYNC**: Timer1 External Clock Input Synchronization Select bit

When TMR1CS = 1:

1 = Do not synchronize external clock input

0 = Synchronize external clock input

When TMR1CS = 0:

This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.

bit 1 **TMR1CS:** Timer1 Clock Source Select bit

1 = External clock from pin RC0/T1OSO/T1CKI (on the rising edge)

0 = Internal clock (FOSC/4)

bit 0 **TMR1ON:** Timer1 On bit

1 = Enables Timer1

0 = Stops Timer1

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit ,read as '0'	
-n = Value at POR	'1' =Bit is set	'0' = Bit is cleared	x= Bit is unknown

16.1 Timer1 Operation

Timer1 can operate in one of these modes:

• As a timer

• As a synchronous counter

• As an asynchronous counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON register).

When TMR1CS is clear, Timer1 increments every instruction cycle. When TMR1CS is set, Timer1 increments on every rising edge of the external clock input or the Timer1 oscillator, if enabled. When the Timer1 oscillator is enabled (T1OSCEN is set), the RC1/T1OSI and RC0/T1OSO/T1CKI pins become inputs. That is, the TRISC<1:0> value is ignored.

Timer1 also has an internal "Reset input". This Reset can be generated by the CCP module

FIGURE 16-1: TIMER1 BLOCK DIAGRAM

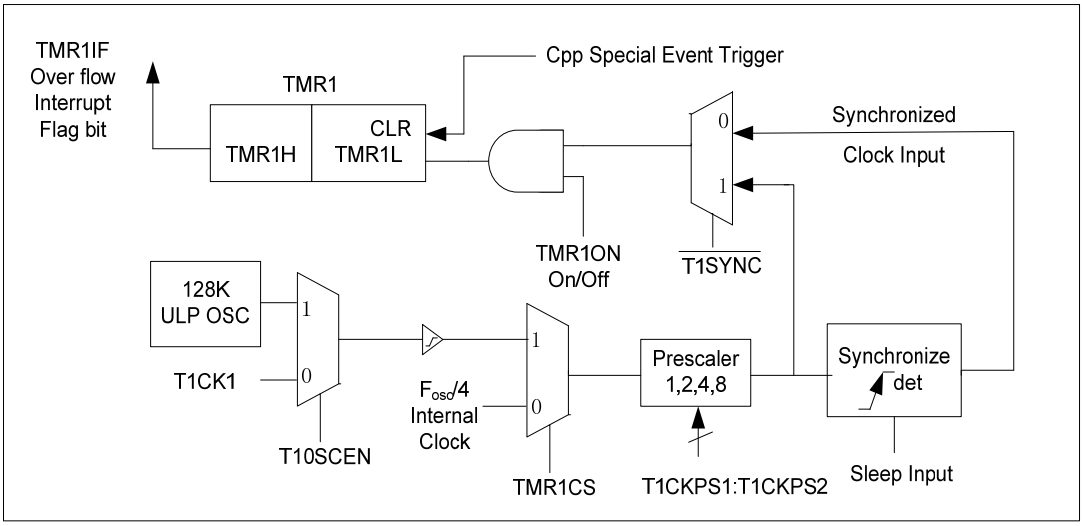
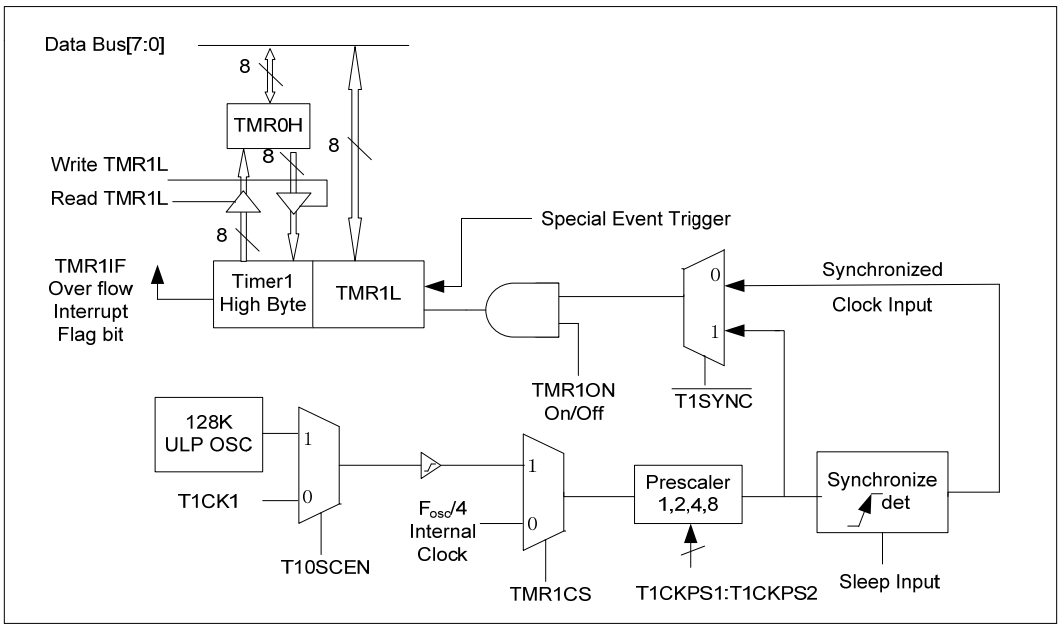


FIGURE 16-2: TIMER1 BLOCK DIAGRAM: 16-BIT READ/WRITE MODE



16.2 Timer1 Oscillator

The ULP OSC circuit is connected the Timer1. It is enabled by setting control bit T1OSCEN (T1CON register). The oscillator is a low-power oscillator rated up to 128 kHz. It will continue to run during Sleep.

16.3 Timer1 Interrupt

The TMR1 register pair (TMR1H:TMR1L) increments from 0000h to FFFFh and rolls over to 0000h. The TMR1 Interrupt, if enabled, is generated on overflow which is latched in interrupt flag bit, TMR1IF (PIR registers). This interrupt can be enabled/disabled by setting/clearing TMR1 Interrupt Enable bit, TMR1IE (PIE registers).

16.4 Resetting Timer1 Using a CCP Trigger Output

If the CCP module is configured in Compare mode to generate a “special event trigger” (CCP1M3:CCP1M0 = 1011), this signal will reset Timer1 and start an A/D conversion (if the A/D module is enabled).

Note: The special event triggers from the CPP1 module will not set Interrupt flag bit, TMR1IF(PIR registers)

Timer1 must be configured for either Timer or Synchronized Counter mode to take advantage of this feature. If Timer1 is running in Asynchronous Counter mode, this Reset operation may not work. In the event that a write to Timer1 coincides with a special event trigger from CCP1, the write will take precedence. In this mode of operation, the CCPR1H:CCPR1L register pair effectively becomes the period register for Timer1.

16.5 Timer1 16-Bit Read/Write Mode

Timer1 can be configured for 16-bit reads and writes (see Figure 16-2). When the RD16 control bit (T1CON register) is set, the address for TMR1H is mapped to a buffer register for the high byte of Timer1. A read from TMR1L will load the contents of the high byte of Timer1 into the Timer1 High Byte Buffer register. This provides the user with the ability to accurately read all 16 bits of Timer1 without having to determine whether a read of the high byte, followed by a read of the low byte, is valid due to a rollover between reads.

A write to the high byte of Timer1 must also take place through the TMR1H Buffer register. Timer1 high byte is updated with the contents of TMR1H when a write occurs to TMR1L. This allows a user to write all 16 bits to both the high and low bytes of Timer1 at once.

The high byte of Timer1 is not directly readable or writable in this mode. All reads and writes must take place through the Timer1 High Byte Buffer register. Writes to TMR1H do not clear the Timer1 prescaler. The prescaler is only cleared on writes to TMR1L.

TABLE 12-2: REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0-00 0000	u-uu uuuu

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.
Note 1: These registers or register bits are not implemented on the PIC18F248 and PIC18F258 and read as '0's.

17. TIMER2 MODULE

The Timer2 module timer has the following features:

• 8-bit timer (TMR2 register)

• 8-bit period register (PR2)

• Readable and writable (both registers)

• Software programmable prescaler (1:1, 1:4, 1:16)

• Software programmable postscaler (1:1 to 1:16)

• Interrupt on TMR2 match of PR2

• SSP module optional use of TMR2 output to generate clock shift

Register 17-1 shows the Timer2 Control register.Timer2 can be shut-off by clearing control bit TMR2ON(T2CON register) to minimize power consumption. Figure 17-1 is a simplified block diagram of the Timer2 module. The prescaler and postscaler selection of Timer2 are controlled by this register.

17.1 Timer2 Operation

Timer2 can be used as the PWM time base for the PWM mode of the CCP module. The TMR2 register is readable and writable and is cleared on any device Reset. The input clock (FOSC/4) has a prescale option of 1:1, 1:4 or 1:16, selected by control bits T2CKPS1:T2CKPS0 (T2CON register). The match output of TMR2 goes through a 4-bit postscaler (which gives a 1:1 to 1:16 scaling inclusive) to generate a TMR2 interrupt (latched in flag bit TMR2IF, PIR registers).

The prescaler and postscaler counters are cleared when any of the following occurs:

• A write to the TMR2 register

• A write to the T2CON register

• Any device Reset (Power-on Reset, MCLR Reset, Watchdog Timer Reset or Brown-out Reset) TMR2 is not cleared when T2CON is written.

Note: Timer2 is disabled on POR.

REGISTER 17-1: T2CON: TIMER2 CONTROL REGISTER

Read/Wite (0XFCA) Bit	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	**T2CON**
	bit 7							bit 0	

bit 7 Unimplemented: Read as ‘0’

bit 6-3 TOUTPS3:TOUTPS0: Timer2 Output Postscale Select bits

0000 = 1:1 Postscale

0001 = 1:2 Postscale

•

•

•

1111 =1:16 Postscale

bit 2 TMR2ON: Timer2 On bit

1 = Timer2 is on

0 = Timer2 is off

bit 1-0 T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit ,read as '0'	
-n = Value at POR	'1' =Bit is set	'0' = Bit is cleared	x= Bit is unknown

17.2 Timer2 Interrupt

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon Reset.

17.3 Output of TMR2

The output of TMR2 (before the postscaler) is a clock input to the Synchronous Serial Port module which optionally uses it to generate the shift clock.

FIGURE 17-1: TIMER2 BLOCK DIAG

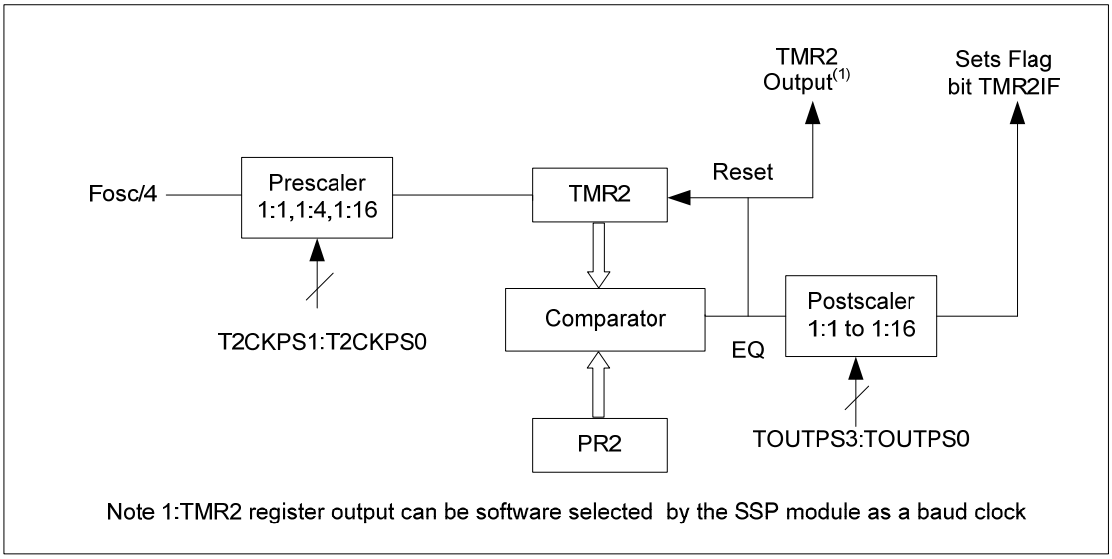


TABLE 17-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TMR2	Timer2 Module Register								0000 0000	0000 0000
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
PR2	Timer2 Period Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

18. CAPTURE/COMPARE/PWM(CCP) MODULES

The CCP (Capture/Compare/PWM) module contains a 16-bit register that can operate as a 16-bit Capture register, as a 16-bit Compare register or as a PWM Duty Cycle register.

The control register for the CCP module is shown in Register 18-1. Table 18-2 (following page) details the interactions of the CCP modules.

REGISTER 18-1: CCP1CON: CCP1 CONTROL REGISTER

Read/Wite (0XFBD) Bit	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	**CCP1CON**
	bit 7							bit 0	

bit 7-6 Unimplemented: Read as '0'

bit 5-4 **DCxB1:DCxB0**: PWM Duty Cycle bit 1 and bit 0 Capture mode:

Capture mode : Unused.

Compare mode : Unused.

PWM mode : These bits are the two LSbs (bit 1 and bit 0) of the 10-bit PWM duty cycle. The upper eight bits (DCx9:DCx2) of the duty cycle are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0**: CCPx Mode Select bits

0000 = Capture/Compare/PWM off (resets CCPx module)

0001 = Reserved

0010 = Compare mode, toggle output on match (CCPxIF bit is set)

0011 = Capture mode, CAN message received (CCP1 only)

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, initialize CCP pin low, on compare match force CCP pin high (CCPIF bit is set)

1001 = Compare mode, initialize CCP pin high, on compare match force CCP pin low (CCPIF bit is set)

1010 = Compare mode, CCP pin is unaffected (CCPIF bit is set)

1011 = Compare mode, trigger special event (CCP1IF bit is set; CCP resets TMR1 or TMR3 and starts an A/D conversion if the A/D module is enabled)

11xx =PWM mode

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit ,read as '0'	
-n = Value at POR	'1' =Bit is set	'0' = Bit is cleared	x= Bit is unknown

18.1 CCP1 Module

Capture/Compare/PWM Register1 (CCPR1) is comprised of two 8-bit registers: CCPR1L (low byte) and CCPR1H (high byte). The CCP1CON register controls the operation of CCP1. All are readable and writable.

Table 18-1 shows the timer resources of the CCP module modes.

TABLE 18-1: CCP1 MODE – TIMER RESOURCE

CCP1 Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

18.2 Capture Mode

In Capture mode, CCPR1H:CCPR1L captures the 16-bit value of the TMR1 register when an event occurs on pin RC2/CCP1. An event is defined as:

• every falling edge
• every rising edge
• every 4th rising edge
• every 16th rising edge

An event is selected by control bits CCP1M3:CCP1M0 (CCP1CON<3:0>). When a capture is made, the interrupt request flag bit, CCP1IF (PIR registers), is set. It must be cleared in software. If another capture occurs before the value in register CCPR1 is read, the old captured value will be lost.

18.2.1 CCP PIN CONFIGURATION

In Capture mode, the PB6/CCP1 pin should be configured as an input by setting the DDRB<6> bit.

18.2.2 TIMER1 MODE SELECTION

The timers used with the capture feature must be running in Timer mode or Synchronized Counter mode. In Asynchronous Counter mode, the capture operation may not work.

18.2.3 SOFTWARE INTERRUPT

When the Capture mode is changed, a false capture interrupt may be generated. The user should keep bit CCP1IE (PIE registers) clear to avoid false interrupts and should clear the flag bit CCP1IF, following any such change in operating mode.

18.2.4 CCP1 PRESCALER

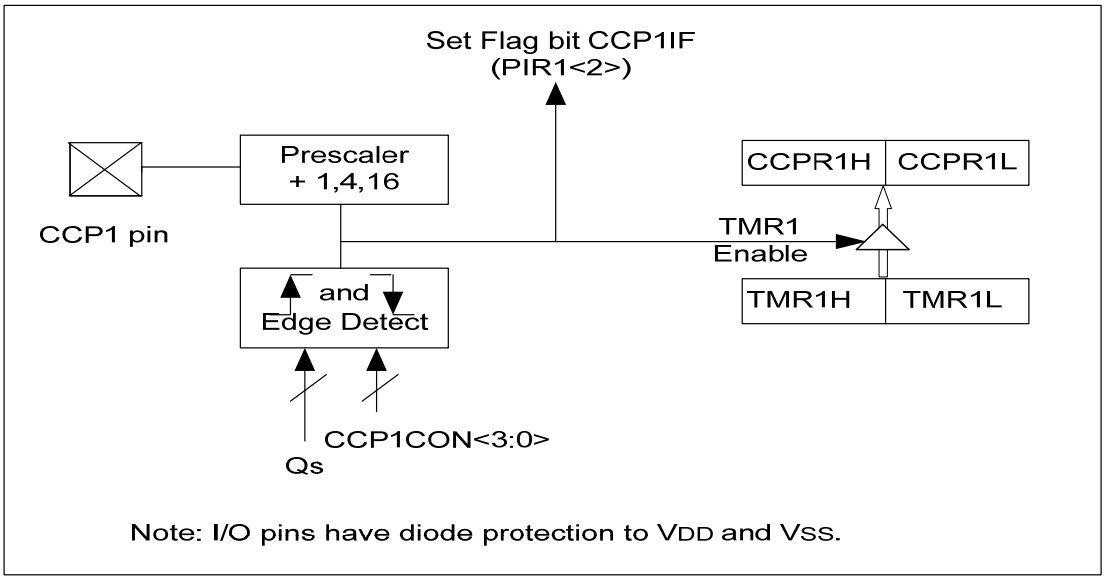
There are four prescaler settings specified by bits CCP1M3:CCP1M0. Whenever the CCP1 module is turned off, or the CCP1 module is not in Capture mode, the prescaler counter is cleared. This means that any Reset will clear the prescaler counter.

Switching from one capture prescaler to another may generate an interrupt. Also, the prescaler counter will not be cleared; therefore, the first capture may be from a non-zero prescaler. Example 18-1 shows the recommended method for switching between capture prescalers. This example also clears the prescaler counter and will not generate the "false" interrupt.

EXAMPLE 18-1: CHANGING BETWEEN CAPTURE PRESCALERS

```
CLRF    CCP1CON, F    ;  Turn CCP module off
MOVLW   NEW_CAPT_PS   ;  Load WREG with the
                      ;  new prescaler mode
MOVWF   CCP1CON       ;  Load CCP1CON with
                      ;  this value
```

FIGURE 18-1: CAPTURE MODE OPERATION BLOCK DIAGRAM



18.3 Compare Mode

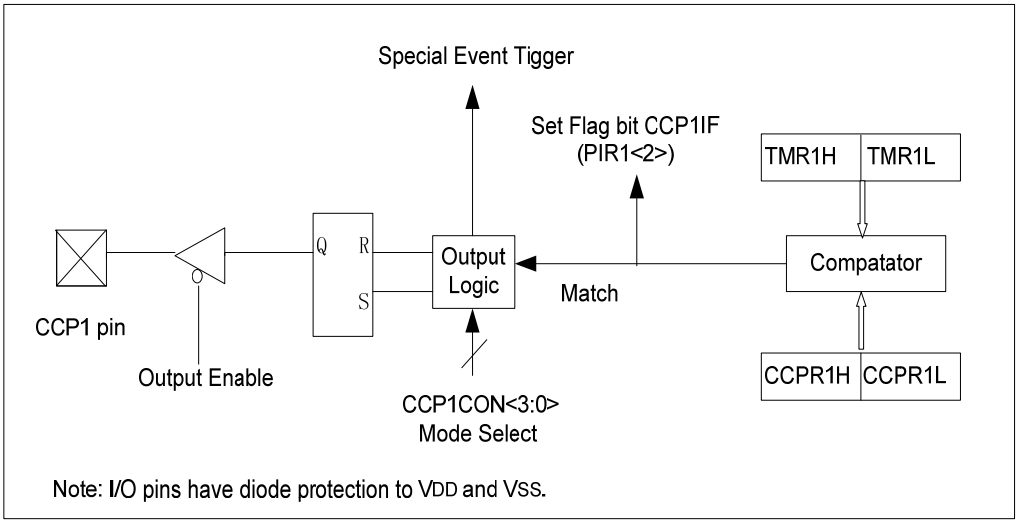
In Compare mode, the 16-bit CCPR1 register value is constantly compared against either the TMR1 register pair value. When a match occurs, the CCP1 pin can have one of the following actions:

- Driven high
- Driven low
- Toggle output (high-to-low or low-to-high)

• Remains unchanged

The action on the pin is based on the value of control bits CCP1M3:CCP1M0. At the same time, interrupt flag bit CCP1IF is set.

FIGURE 18-2: COMPARE MODE OPERATION BLOCK DIAGRAM



18.3.1 CCP1 PIN CONFIGURATION

The user must configure the CCP1 pin as an output by clearing the appropriate DDRB bit.

Note: Clearing the CCP1CON register will force the CCP1 compare output latch to the default low level .This is not the data latch.

18.3.2 TIMER1 MODE SELECTION

Timer1 and must be running in Timer mode, or Synchronized Counter mode, if the CCP module is using the compare feature. In Asynchronous Counter mode, the compare operation may not work.

18.3.3 SOFTWARE INTERRUPT MODE

When generate software interrupt is chosen, the CCP1 pin is not affected. Only a CCP interrupt is generated (if enabled).

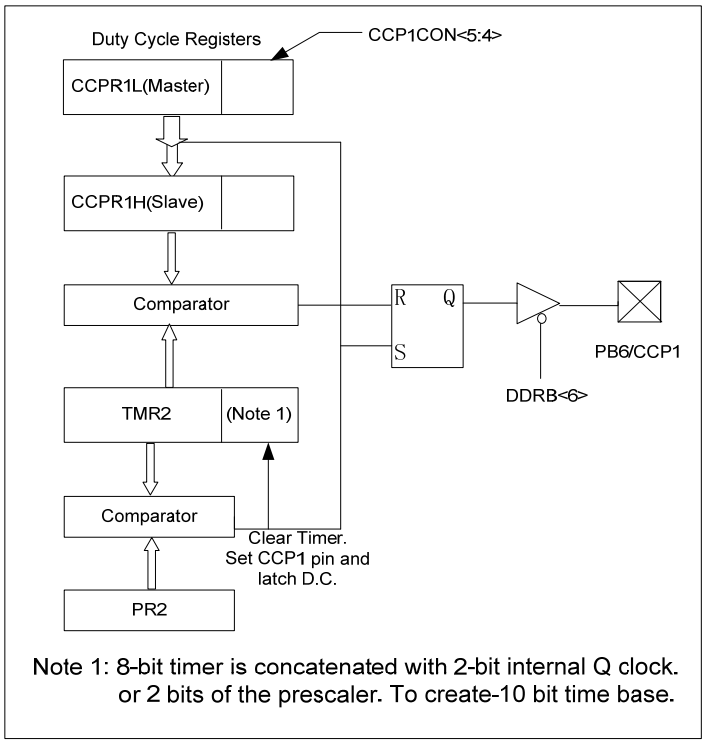
TABLE 18-3: REGISTERS ASSOCIATED WITH CAPTURE, COMPARE, TIMER1

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TRISD	PORTD Data Direction Register								1111 1111	1111 1111
TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register								xxxx xxxx	uuuu uuuu
T1CON	RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	$\overline{\text{T1SYNC}}$	TMR1CS	TMR1ON	0-00 0000	u-uu uuuu
CCPR1L	Capture/Compare/PWM Register 1 (LSB)								xxxx xxxx	uuuu uuuu
CCPR1H	Capture/Compare/PWM Register 1 (MSB)								xxxx xxxx	uuuu uuuu
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000
PIR2	—	CMIF	—	EEIF	BCLIF	LVDIF	TMR3IF	ECCP1IF	-0-0 0000	-0-0 0000
PIE2	—	CMIE	—	EEIE	BCLIE	LVDIE	TMR3IE	ECCP1IE	-0-0 0000	-0-0 0000
IPR2	—	CMIP	—	EEIP	BCLIP	LVDIP	TMR3IP	ECCP1IP	-1-1 1111	-1-1 1111

18.4 PWM Mode

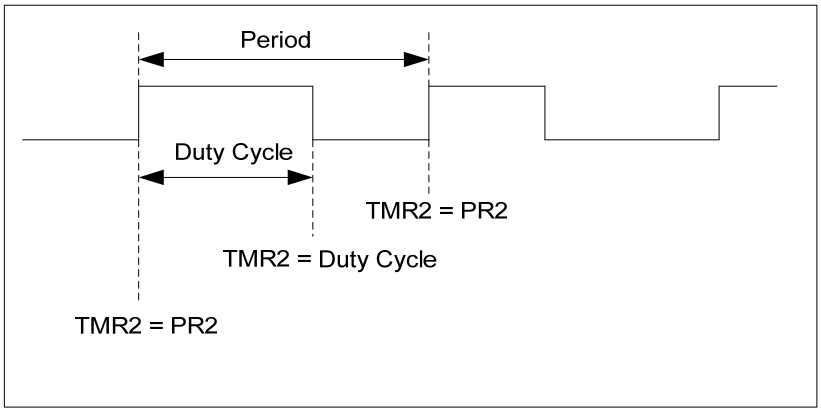
In Pulse-Width Modulation (PWM) mode, the CCP1 pin produces up to a 10-bit resolution PWM output. Since the CCP1 pin is multiplexed with the PORTB data latch, the DDRB<6> bit must be set to make the CCP1 pin an output.

Figure 18-3 shows a simplified block diagram of the CCP module in PWM mode.
For a step-by-step procedure on how to set up the CCP module for PWM operation, see Section 18.4.3“Setup for PWM Operation”.
FIGURE 18-3: SIMPLIFIED PWM BLOCK DIAGRAM



A PWM output (Figure 18-4) has a time base (period) and a time that the output stays high (duty cycle). The frequency of the PWM is the inverse of the period (1/period).

FIGURE 18-4: PWM OUTPUT



18.4.1 PWM PERIOD

The PWM period is specified by writing to the PR2 register. The PWM period can be calculated using the following formula.

EQUATION 18-2:

PWM Period = [(PR2)+1] • 4 •TOSC • (TMR2 Prescale Value)

PWM frequency is defined as 1/[PWM period]. When TMR2 is equal to PR2, the following three events occur on the next increment cycle:

•TMR2 is cleared

• The CCP1 pin is set (exception: if PWM duty cycle = 0%, the CCP1 pin will not be set)
• The PWM duty cycle is latched from CCPR1L into CCPR1H

18.4.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the CCPR1L register and to the CCP1CON<5:4> bits. Up to 10-bit resolution is available. The CCPR1L contains the eight MSbs and the CCP1CON<5:4> contains the two LSbs. This 10-bit value is represented by CCPR1L:CCP1CON<5:4>. The following equation is used to calculate the PWM duty cycle in time.

EQUATION 18-2:

PWM Duty Cycle = (CCPR1L:CCP1CON<5:4>) •TOSC • (TMR2 Prescale Value)

CCPR1L and CCP1CON<5:4> can be written to at any time, but the duty cycle value is not latched into CCPR1H until after a match between PR2 and TMR2 occurs (i.e., the period is complete). In PWM mode, CCPR1H is a read-only register.The CCPR1H register and a 2-bit internal latch are used to double-buffer the PWM duty cycle. This double-buffering is essential for glitchless PWM operation.When the CCPR1H and 2-bit latch match TMR2, concatenated with an internal 2-bit Q clock or 2 bits of the TMR2 prescaler, the CCP1 pin is cleared.The maximum PWM resolution (bits) for a given PWM frequency is given by the following equation.

EQUATION 18-3:

$$\text{PWM Resolution(max)} = \frac{\text{Log}\left(\frac{Fosc}{FPWM}\right)}{\text{Log}(2)}$$

18.4.3 SETUP FOR PWM OPERATION

The following steps should be taken when configuring the CCP module for PWM operation:
1. Set the PWM period by writing to the PR2 register.
2. Set the PWM duty cycle by writing to the CCPR1L register and CCP1CON<5:4> bits.
3. Make the CCP1 pin an output by clearing the TRISC<2> bit.
4. Set the TMR2 prescale value and enable Timer2 by writing to T2CON.
5. Configure the CCP1 module for PWM operation.

TABLE 18-4: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 40 MHz

PWM Frequency	2.44 kHz	9.76 kHz	39.06 kHz	156.3 kHz	312.5 kHz	416.6 kHz
Timer Prescaler (1, 4, 16)	16	4	1	1	1	1
PR2 Value	0FFh	0FFh	0FFh	3Fh	1Fh	17h
Maximum Resolution (bits)	10	10	10	8	7	5.5

TABLE 18-5: REGISTERS ASSOCIATED WITH PWM AND TIMER2

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/ GIEH	PEIE/ GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TRISD	PORTD Data Direction Register								1111 1111	1111 1111
TMR2	Timer2 Module Register								0000 0000	0000 0000
PR2	Timer2 Module Period Register								1111 1111	1111 1111
T2CON	—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
CCPR1L	Capture/Compare/PWM Register1 (LSB)								xxxx xxxx	uuuu uuuu
CCPR1H	Capture/Compare/PWM Register1 (MSB)								xxxx xxxx	uuuu uuuu
CCP1CON	—	—	DC1B1	DC1B0	CCP1M3	CCP1M2	CCP1M1	CCP1M0	--00 0000	--00 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PWM and Timer2.

19. MASTER SYNCHRONOUS SERIAL PORT (MSSP) MODULE

19.1 Master SSP (MSSP) Module Overview

The Master Synchronous Serial Port (MSSP) module is a serial interface useful for communicating with other peripheral or microcontroller devices. These peripheral devices may be serial EEPROMs, shift registers, display drivers, A/D converters, etc.

The I^2C interface supports the following modes in hardware:

• Master mode

• Multi-Master mode

• Slave mode

19.2 Control Registers

The MSSP module has three associated registers. These include a status register (SSPSTAT) and two control registers (SSPCON1 and SSPCON2).

19.4 I^2C Mode

The MSSP module in I^2C mode fully implements all master and slave functions (including general

call support) and provides interrupts on Start and Stop bits in hardware to determine a free bus (multi-master function). The MSSP module implements the standard mode specifications, as well as 7-bit and 10-bit addressing.

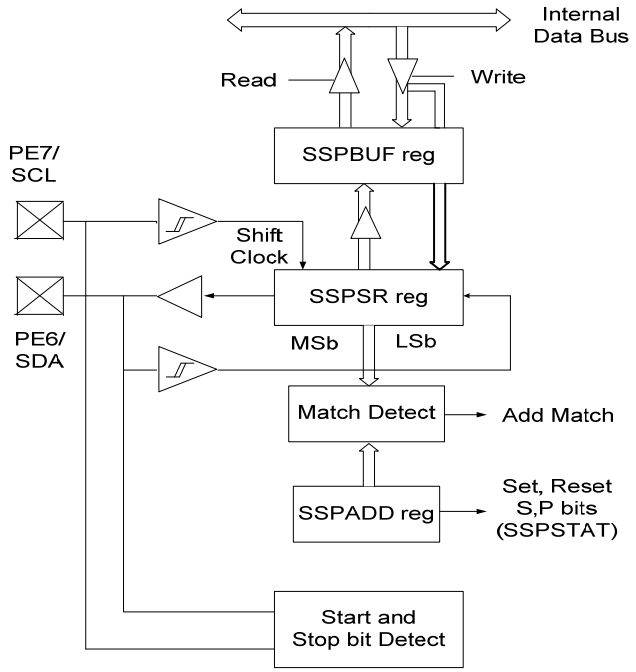
Two pins are used for data transfer:

• Serial clock (SCL) – PE7/SCL

• Serial data (SDA) – PE6/SDA

The user must configure these pins as inputs or outputs through the DDRE<7:6> bits.

FIGURE 19-7: MSSP BLOCK DIAGRAM (I2C™ MODE)



19.4.1 REGISTERS

The MSSP module has six registers for I^2C operation. These are:

• MSSP Control Register 1 (SSPCON1)

• MSSP Control Register 2 (SSPCON2)

• MSSP Status Register (SSPSTAT)

• Serial Receive/Transmit Buffer (SSPBUF)

• MSSP Shift Register (SSPSR) – Not directly accessible

• MSSP Address Register (SSPADD)

SSPCON1, SSPCON2 and SSPSTAT are the control and status registers in I^2C mode operation. The SSPCON1 and SSPCON2 registers are readable and writable. The lower 6 bits of the SSPSTAT are read-only. The upper two bits of the SSPSTAT are read/write. SSPSR is the shift register used for shifting data in or out. SSPBUF is the buffer register to which data bytes are written to or read from.

SSPADD register holds the slave device address when the SSP is configured in I^2C Slave mode. When the SSP is configured in Master mode, the lower seven bits of SSPADD act as the Baud

Rate Generator reload value.

In receive operations, SSPSR and SSPBUF together create a double-buffered receiver. When SSPSR receives a complete byte, it is transferred to SSPBUF and the SSPIF interrupt is set.

During transmission, the SSPBUF is not double-buffered. A write to SSPBUF will write to both SSPBUF and SSPSR.

REGISTER 19-3: SSPSTAT: MSSP STATUS REGISTER (I^2C MODE)

Read/Wite (0XFC7) Bit	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R-0	R-0	R-0	
	SMP	CKE	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF	SSPSTAT
	bit 7							bit 0	

bit 7 **SMP:** Slew Rate Control bit
In Master or Slave mode:
1 = Slew rate control disabled for Standard Speed mode (100 kHz and 1 MHz)
0 = Slew rate control enabled for High-Speed mode (400 kHz)

bit 6 **CKE:** SMBus Select bit
In Master or Slave mode:
1 = Enable SMBus specific inputs
0 = Disable SMBus specific inputs

bit 5 **D/A:** Data/Address bit
In Master mode:
Reserved.
In Slave mode:
1 = Indicates that the last byte received or transmitted was data
0 = Indicates that the last byte received or transmitted was address

bit 4 **P:** Stop bit
1 = Indicates that a Stop bit has been detected last
0 = Stop bit was not detected last
Note: This bit is cleared on Reset and when SSPEN is cleared.

bit 3 **S:** Start bit
1 = Indicates that a Start bit has been detected last
0 = Start bit was not detected last
Note: This bit is cleared on Reset and when SSPEN is cleared.

bit 2 **R/W:** Read/Write Information bit (I^2C mode only)
In Slave mode:
1 = Read
0 = Write
Note: This bit holds the R/W bit information following the last address match. This bit is only valid from the address match to the next Start bit, Stop bit or not ACK bit.
In Master mode:
1 = Transmit is in progress
0 = Transmit is not in progress

Note: ORing this bit with SEN, RSEN, PEN, RCEN or ACKEN will indicate if the MSSP is in Idle mode.

bit 1 **UA:** Update Address bit (10-bit Slave mode only)

1 = Indicates that the user needs to update the address in the SSPADD register

0 = Address does not need to be updated

bit 0 **BF:** Buffer Full Status bit

In Transmit mode:

1 = Data transmit in progress (does not include the ACK and Stop bits), SSPBUF is full

0 = Data transmit complete (does not include the ACK and Stop bits), SSPBUF is empty

In Receive mode:

1 = Receive complete, SSPBUF is full

0 = Receive not complete, SSPBUF is empty

REGISTER 19-4: SSPCON1: MSSP CONTROL REGISTER 1 (I^2C MODE)

Read/Wite (0XFC6) Bit	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	SSPCON1
	bit 7							bit 0	

bit 7 **WCOL**: Write Collision Detect bit

In Master Transmit mode:

1 = A write to the SSPBUF register was attempted while the I^2C conditions were not valid for a transmission to be started (must be cleared in software)

0 = No collision

In Slave Transmit mode:

1 = The SSPBUF register is written while it is still transmitting the previous word (must be cleared in software)

0 = No collision

In Receive mode (Master or Slave modes):

This is a “don’t care” bit.

bit 6 **SSPOV**: Receive Overflow Indicator bit

In Receive mode:

1 = A byte is received while the SSPBUF register is still holding the previous byte (must be cleared in software), and will send a NACK if the SSPOV is logic one.

0 = No overflow

In Transmit mode:

This is a “don’t care” bit in Transmit mode.

bit 5 **SSPEN**: Synchronous Serial Port Enable bit

1 = Enables the serial port and configures the SDA and SCL pins as the serial port pins

0 = Disables serial port and configures these pins as I/O port pins

Note: When enabled, the SDA and SCL pins must be properly configured as input or output.

bit 4 **CKP**: SCK Release Control bit

In Slave mode:

1 = Release clock

0 = Holds clock low (clock stretch), used to ensure data setup time

In Master mode:

Unused in this mode.

bit 3-0 **SSPM3**:**SSPM0**: Synchronous Serial Port Mode Select bits

1111 = I^2C Slave mode, 10-bit address with Start and Stop bit interrupts enabled

1110 = I^2C Slave mode, 7-bit address with Start and Stop bit interrupts enabled

1011 = I^2C Firmware Controlled Master mode (Slave Idle)

1000 = I^2C Master mode, clock = FOSC/(4 * (SSPADD + 1))

0111 = I^2C Slave mode, 10-bit address

0110 = I^2C Slave mode, 7-bit address

Note: Bit combinations not specifically listed here are either reserved or implemented in SPI mode only.

REGISTER 19-5: SSPCON2: MSSP CONTROL REGISTER 2 (I^2C MODE)

Read/Wite (0XFC5) Bit	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	SSPCON2
	bit 7							bit 0	

bit 7 **GCEN:** General Call Enable bit (Slave mode only)

1 = Enable interrupt when a general call address (0000h) is received in the SSPSR

0 = General call address disabled

bit 6 **ACKSTAT:** Acknowledge Status bit (Master Transmit mode only)

1 = Acknowledge was not received from slave

0 = Acknowledge was received from slave

bit 5 **ACKDT:** Acknowledge Data bit (Master Receive mode only)

1 = Not Acknowledge

0 = Acknowledge

Note: Value that will be transmitted when the user initiates an Acknowledge sequence at the end of a receive.

bit 4 **ACKEN:** Acknowledge Sequence Enable bit (Master Receive mode only)

1 = Initiate Acknowledge sequence on SDA and SCL pins and transmit ACKDT data bit. Automatically cleared by hardware.

0 = Acknowledge sequence Idle

bit 3 **RCEN:** Receive Enable bit (Master Mode only)

1 = Enables Receive mode for I^2C

0 = Receive Idle

bit 2 **PEN:** Stop Condition Enable bit

In Master mode :

1 = Initiate Stop condition on SDA and SCL pins. Automatically cleared by hardware.

0 = Stop condition Idle

In Slave mode:

1 = Enable Stop Interrupt when receive a Stop condition on SDA and SCL pins.

0 = Disable Stop Interrupt

bit 1 **RSEN:** Repeated Start Condition Enable bit

In Master mode :

1 = Initiate Repeated Start condition on SDA and SCL pins. Automatically cleared by

hardware.
0 = Repeated Start condition Idle
In Slave mode:
1 = Enable Stop Interrupt when receive a Start condition on SDA and SCL pins.
0 = Disable Stop Interrupt

bit 0 **SEN:** Start Condition Enable/Stretch Enable bit
In Master mode:
1 = Initiate Start condition on SDA and SCL pins. Automatically cleared by hardware.
0 = Start condition Idle
In Slave mode:
1 = Clock stretching is enabled for both slave transmit and slave receive (stretch enabled)
0 = Clock stretching is enabled for slave transmit only (Legacy mode)

19.4.2 OPERATION

The MSSP module functions are enabled by setting MSSP Enable bit, SSPEN (SSPCON1<5>).
The SSPCON1 register allows control of the I^2C operation. Four mode selection bits (SSPCON1<3:0>) allow one of the following I^2C modes to be selected:
•I^2C Master mode, clock = OSC/4 (SSPADD +1)
•I^2C Slave mode (7-bit address)
•I^2C Slave mode (10-bit address)
•I^2C Slave mode (7-bit address) with Start and Stop bit interrupts enabled
•I^2C Slave mode (10-bit address) with Start and Stop bit interrupts enabled
•I^2C Firmware Controlled Master mode, slave is Idle
Selection of any I^2C mode with the SSPEN bit set forces the SCL and SDA pins to be open-drain, provided these pins are programmed to inputs by setting the appropriate TRISC bits. To ensure proper operation of the module, pull-up resistors must be provided externally to the SCL and SDA pins.

19.4.3 SLAVE MODE

In Slave mode, the SCL and SDA pins must be configured as inputs (TRISC<4:3> set). The MSSP module will override the input state with the output data when required (slave-transmitter).

The I^2C Slave mode hardware will always generate an interrupt on an address match. Through the mode select bits, the user can also choose to interrupt on Start and Stop bits. When an address is matched, or the data transfer after an address match is received, the hardware automatically will generate the Acknowledge (ACK) pulse and load the SSPBUF register with the received value currently in the SSPSR register.
Any combination of the following conditions will cause the MSSP module not to give this ACK pulse:
• The Buffer Full bit, BF (SSPSTAT<0>), was set before the transfer was received.

• The overflow bit, SSPOV (SSPCON1<6>), was set before the transfer was received.

In this case, the SSPSR register value is not loaded into the SSPBUF, but bit SSPIF (PIR1<3>) is set. The BF bit is cleared by reading the SSPBUF register, while bit SSPOV is cleared through software.

The SCL clock input must have a minimum high and low for proper operation. The high and low times of the I^2C specification, as well as the requirement of the MSSP module, are shown in timing parameter #100 and parameter #101.

19.4.3.1 Addressing

Once the MSSP module has been enabled, it waits for a Start condition to occur. Following the Start condition, the 8 bits are shifted into the SSPSR register. All incoming bits are sampled with the rising edge of the clock (SCL) line. The value of register SSPSR<7:1> is compared to the value of the SSPADD register. The address is compared on the falling edge of the eighth clock (SCL) pulse. If the addresses match and the BF and SSPOV bits are clear, the following events occur:

1. The SSPSR register value is loaded into the SSPBUF register.
2. The Buffer Full bit BF is set.
3. An ACK pulse is generated.
4. MSSP Interrupt Flag bit, SSPIF (PIR1<3>), is set (interrupt is generated if enabled) on the falling edge of the ninth SCL pulse. In 10-bit Address mode, two address bytes need to be received by the slave. The five Most Significant bits (MSbs) of the first address byte specify if this is a 10-bit address. Bit R/W (SSPSTAT<2>) must specify a write so the slave device will receive the second address byte. For a 10-bit address, the first byte would equal‘11110 A9 A8 0’, where ‘A9’ and ‘A8’ are the two MSbs of the address. The sequence of events for 10-bit address is as follows, with steps 7 through 9 for the slave-transmitter:

1. Receive first (high) byte of address (bits SSPIF, BF and bit UA (SSPSTAT<1>) are set).
2. Update the SSPADD register with second (low) byte of address (clears bit UA and releases the SCL line).
3. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
4. Receive second (low) byte of address (bits SSPIF, BF and UA are set).
5. Update the SSPADD register with the first (high) byte of address. If match releases SCL line, this will clear bit UA.
6. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.
7. Receive Repeated Start condition.
8. Receive first (high) byte of address (bits SSPIF and BF are set).
9. Read the SSPBUF register (clears bit BF) and clear flag bit SSPIF.

19.4.3.2 Reception

When the R/W bit of the address byte is clear and an address match occurs, the R/W bit of the

SSPSTAT register is cleared. The received address is loaded into the SSPBUF register and the SDA line is held low (ACK).

When the address byte overflow condition exists, then the no Acknowledge (ACK) pulse is given. An overflow condition is defined as either bit BF (SSPSTAT<0>) is set or bit SSPOV (SSPCON1<6>) is set.

An MSSP interrupt is generated for each data transfer byte. Flag bit SSPIF (PIR1<3>) must be cleared in software. The SSPSTAT register is used to determine the status of the byte. If SEN is enabled (SSPCON2<0> = 1), RC3/SCK/SCL will be held low (clock stretch) following each data transfer. The clock must be released by setting bit CKP (SSPCON1<4>). See Section 19.4.4 “Clock Stretching” for more detail.

19.4.3.3 Transmission

When the R/W bit of the incoming address byte is set and an address match occurs, the R/W bit of the SSPSTAT register is set. The received address is loaded into the SSPBUF register. The ACK pulse will be sent on the ninth bit and pin RC3/SCK/SCL is held low regardless of SEN (see Section 19.4.4 “Clock Stretching” for more detail). By stretching the clock, the master will be unable to assert another clock pulse until the slave is done preparing the transmit data. The transmit data must be loaded into the SSPBUF register, which also loads the SSPSR register. Then, pin RC3/SCK/SCL should be enabled by setting bit CKP (SSPCON1<4>). The eight data bits are shifted out on the falling edge of the SCL input. This ensures that the SDA signal is valid during the SCL high time (Figure 19-9).

The ACK pulse from the master-receiver is latched on the rising edge of the ninth SCL input pulse. If the SDA line is high (not ACK), then the data transfer is complete. In this case, when the ACK is latched by the slave, the slave logic is reset (resets SSPSTAT register) and the slave monitors for another occurrence of the Start bit. If the SDA line was low (ACK), the next transmit data must be loaded into the SSPBUF register. Again, pin RC3/SCK/SCL must be enabled by setting bit CKP.

An MSSP interrupt is generated for each data transfer byte. The SSPIF bit must be cleared in software and the SSPSTAT register is used to determine the status of the byte. The SSPIF bit is set on the falling edge of the ninth clock pulse.

FIGURE 19-8: I2C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 7-BIT ADDRESS)

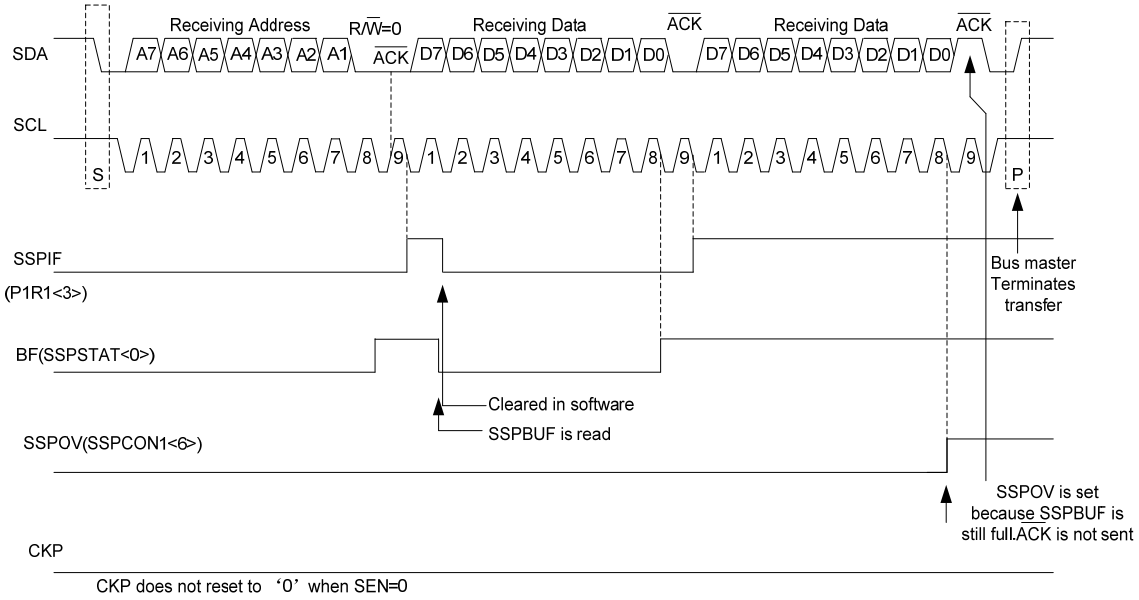


FIGURE 19-9: I2C™ SLAVE MODE TIMING (TRANSMISSION, 7-BIT ADDRESS)

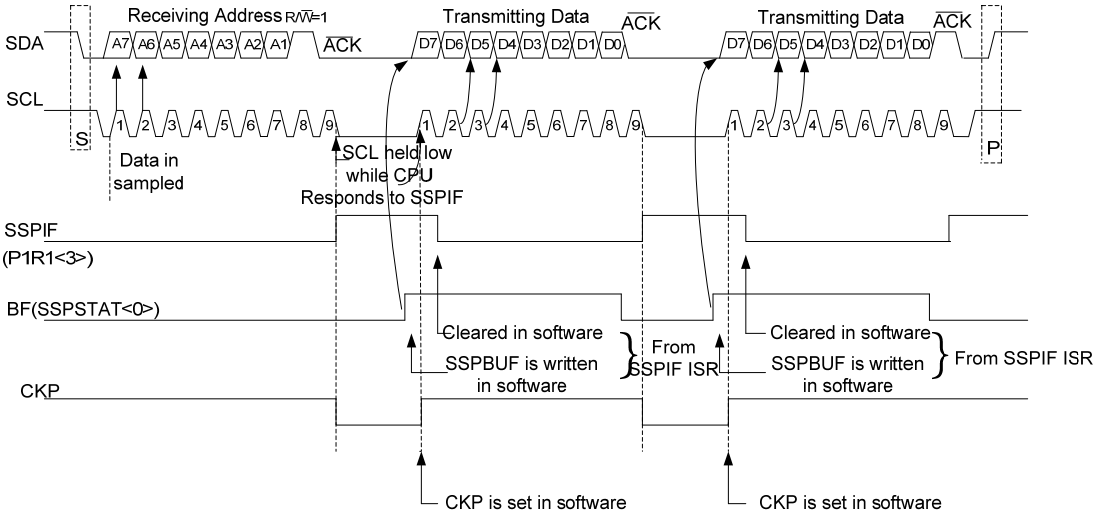


FIGURE 19-10: I2C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)

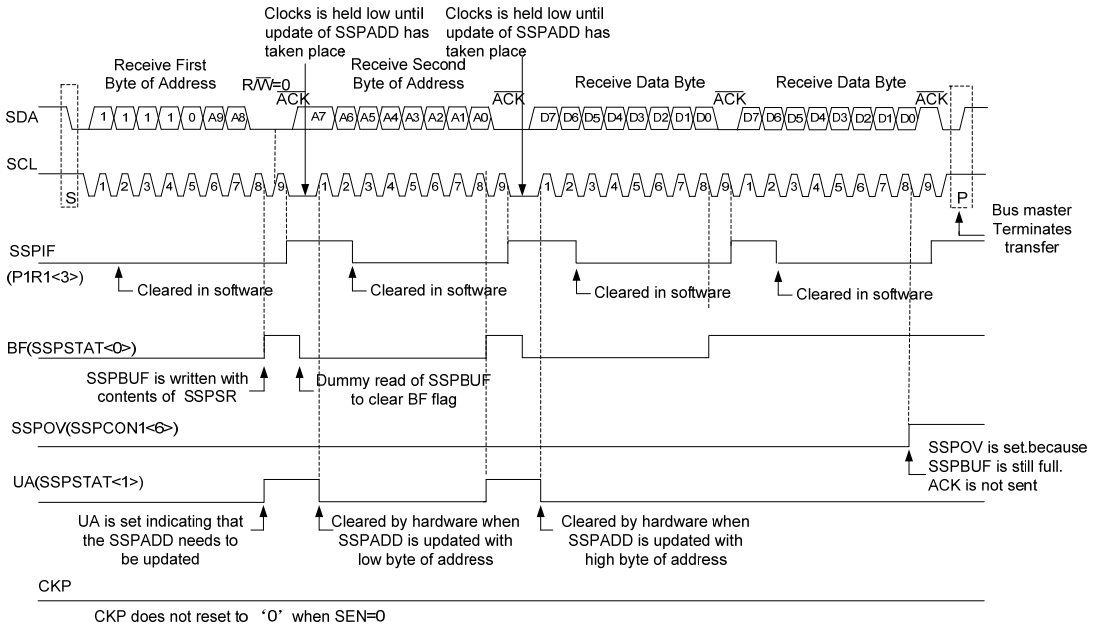
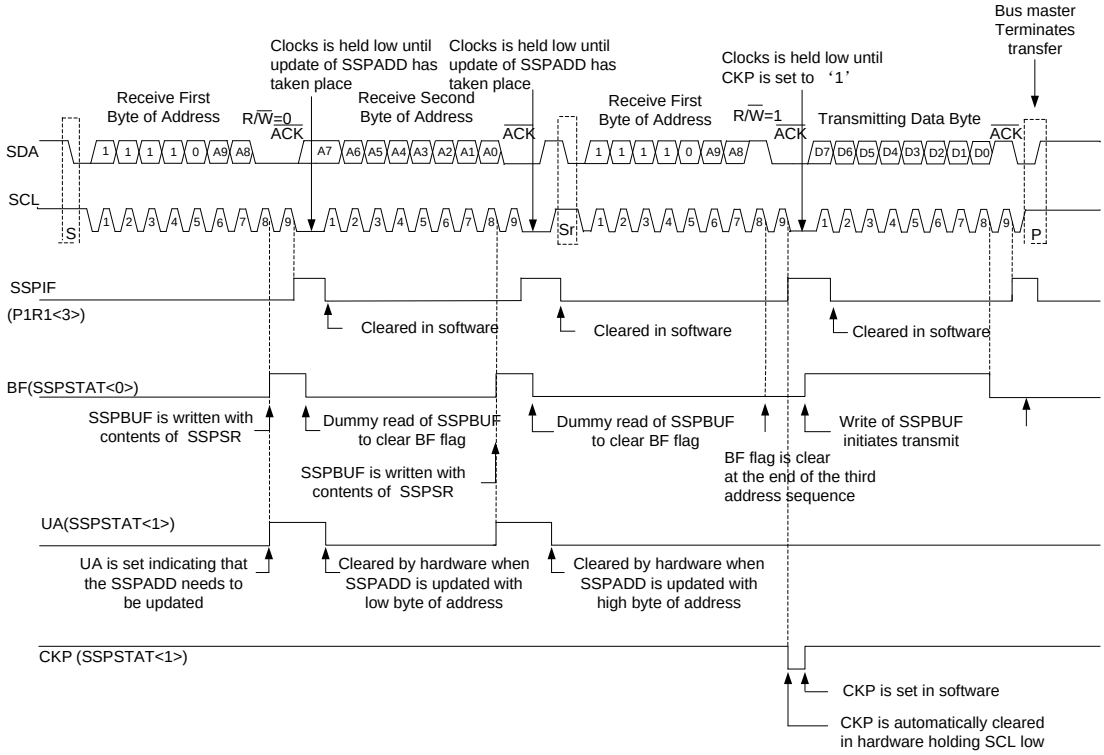


FIGURE 19-11: I2C™ SLAVE MODE TIMING (TRANSMISSION, 10-BIT ADDRESS)



19.4.4 CLOCK STRETCHING

Both 7 and 10-bit Slave modes implement automatic clock stretching during a transmit sequence. The SEN bit (SSPCON2<0>) allows clock stretching to be enabled during receives. Setting SEN will cause the SCL pin to be held low at the end of each data receive sequence.

19.4.4.1 Clock Stretching for 7-bit Slave Receive Mode (SEN = 1)

In 7-bit Slave Receive mode, on the falling edge of the ninth clock at the end of the ACK sequence, if the BF bit is set, the CKP bit in the SSPCON1 register is automatically cleared, forcing the SCL output to be held low. The CKP being cleared to '0' will assert the SCL line low. The CKP bit must be set in the user's ISR before reception is allowed to continue. By holding the SCL line low, the user has time to service the ISR and read the contents of the SSPBUF before the master device can initiate another receive sequence. This will prevent buffer overruns from occurring.

Note 1: If the user reads the contents of the SSPBUF before the falling edge of the ninth clock, thus clearing the BF bit, the CKP bit will not be cleared and clock stretching will not occur.

2: The CKP bit can be set in software regardless of the state of the BF bit. The user should be careful to clear the BF bit in the ISR before the next receive sequence in order to prevent an overflow condition.

19.4.4.2 Clock Stretching for 10-bit Slave Receive Mode (SEN = 1)

In 10-bit Slave Receive mode, during the address sequence, clock stretching automatically takes place but CKP is not cleared. During this time, if the UA bit is set after the ninth clock, clock stretching is initiated. The UA bit is set after receiving the upper byte of the 10-bit address and following the receive of the second byte of the 10-bit address with the R/W bit cleared to'0'. The release of the clock line occurs upon updating SSPADD. Clock stretching will occur on each data receive sequence as described in 7-bit mode.

Note: If the user polls the UA bit and clears it by updating the SSPADD register before the falling edge of the ninth clock occurs and if the user hasn't cleared the BF bit by reading the SSPBUF register before that time, then the CKP bit will still NOT be asserted low. Clock stretching on the basis of the state of the BF bit only occurs during a data sequence, not an address sequence.

19.4.4.3 Clock Stretching for 7-bit Slave Transmit Mode

7-bit Slave Transmit mode implements clock stretching by clearing the CKP bit after the falling edge of the ninth clock if the BF bit is clear. This occurs regardless of the state of the SEN bit. The user's ISR must set the CKP bit before transmission is allowed to continue. By holding the SCL line low, the user has time to service the ISR and load the contents of the SSPBUF before the master device can initiate another transmit sequence (see Figure 19-9).

Note 1:	If the user loads the contents of SSPBUF, setting the BF bit before the falling edge of the ninth clock, the CKP bit will not be cleared and clock stretching will not occur.
2:	The CKP bit can be set in software regardless of the state of the BF bit.

19.4.4.4 Clock Stretching for 10-bit Slave Transmit Mode

In 10-bit Slave Transmit mode, clock stretching is controlled during the first two address sequences by the state of the UA bit, just as it is in 10-bit Slave Receive mode. The first two addresses are followed by a third address sequence which contains the high-order bits of the 10-bit address and the R/W bit set to '1'. After the third address sequence is performed, the UA bit is not set, the module is now configured in Transmit mode and clock stretching is controlled by the BF flag as in 7-bit Slave Transmit mode (see Figure 19-11).

19.4.4.5 Clock Synchronization and The CKP bit

If a user clears the CKP bit, the SCL output is forced to'0'. Setting the CKP bit will not assert the SCL output low until the SCL output is already sampled low. If the user attempts to drive SCL low, the CKP bit will not assert the SCL line until an external I^2C master device has already asserted the SCL line. The SCL output will remain low until the CKP bit is set and all other devices on the I^2C bus have deasserted SCL. This ensures that a write to the CKP bit will not violate the minimum high time requirement for SCL (see Figure 19-12).

FIGURE 19-12:CLOCK SYNCHRONIZATION TIMING

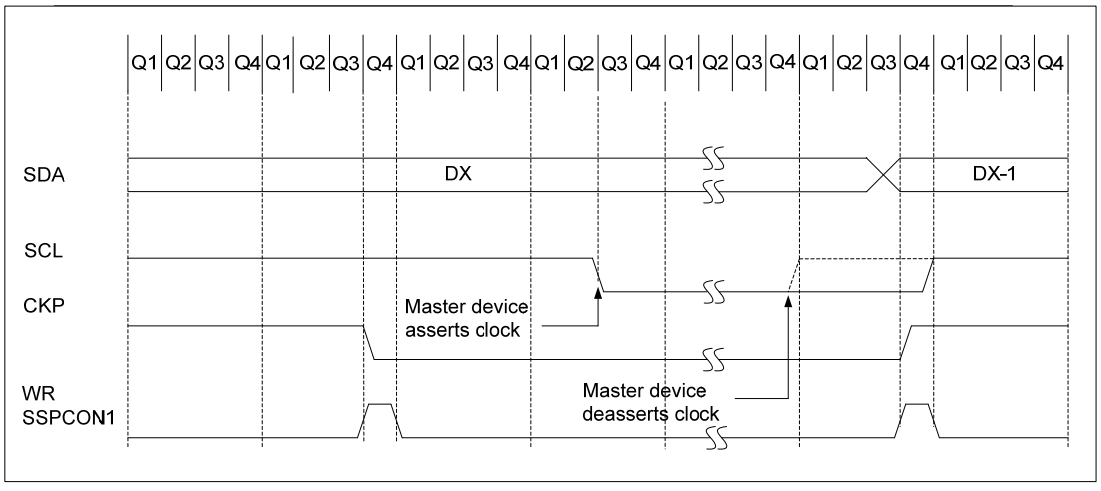


FIGURE 19-13: I2C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 7-BIT ADDRESS)

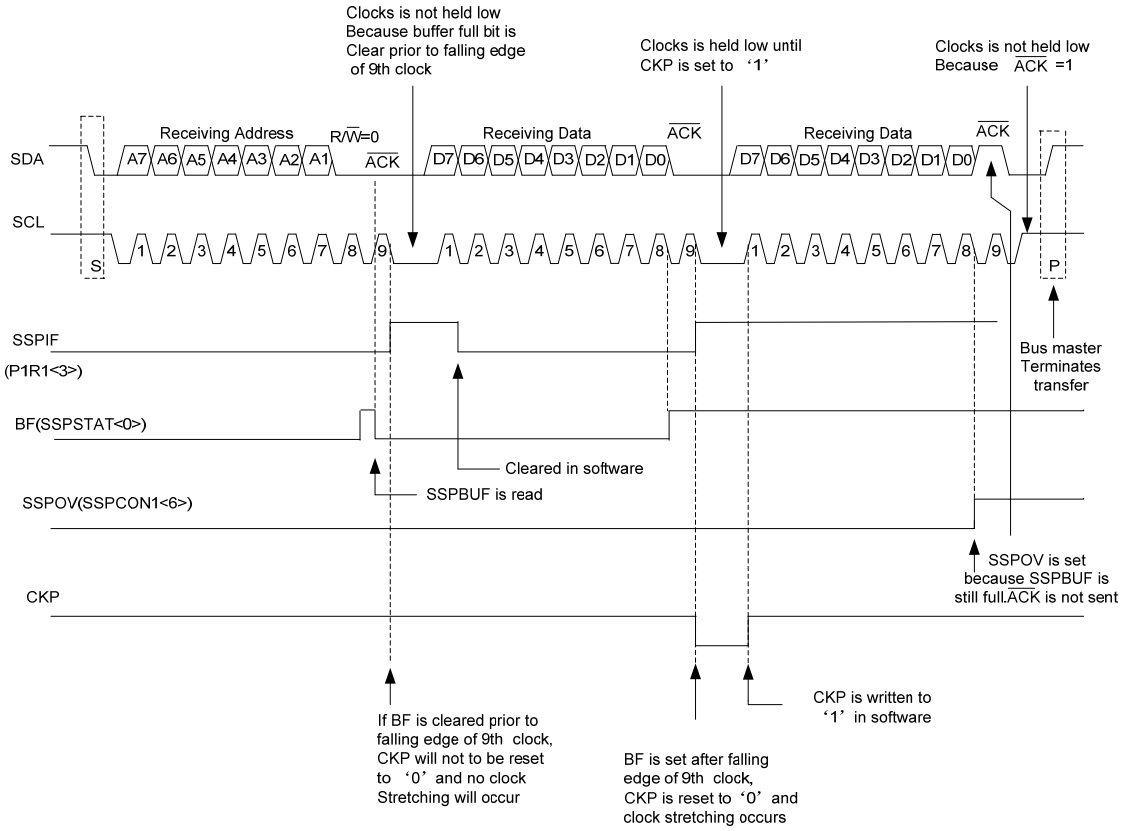
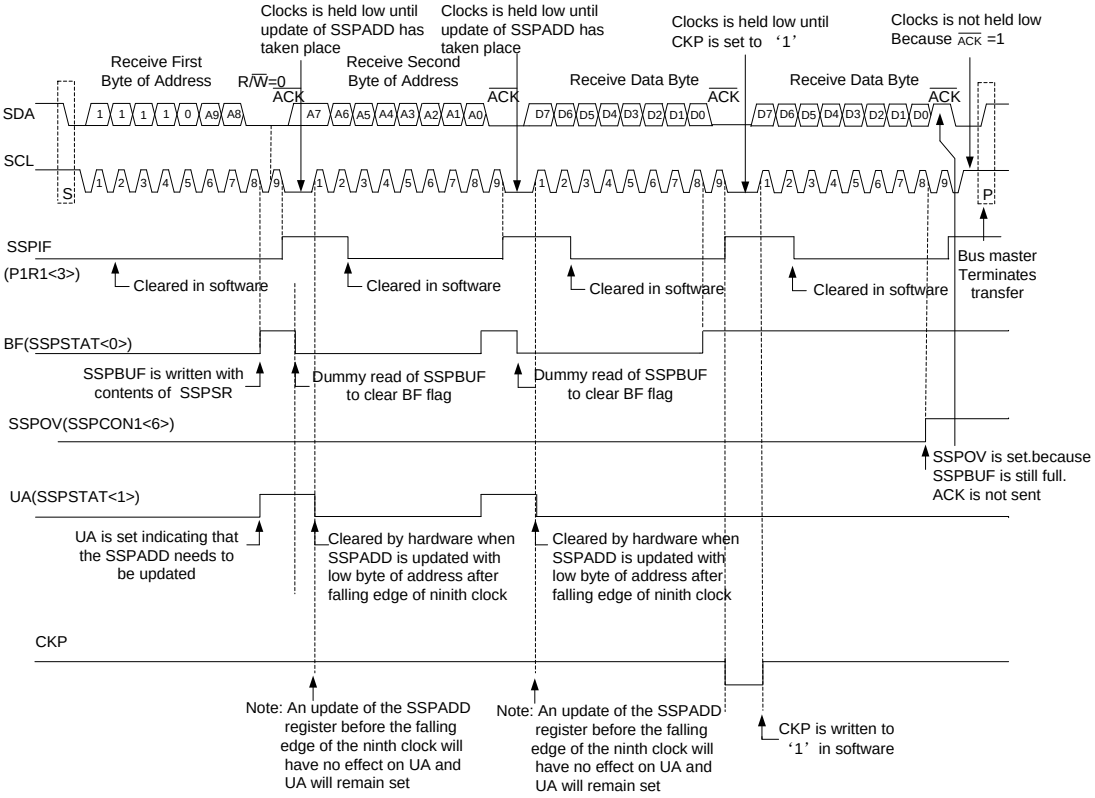


FIGURE 19-14: I2C™ SLAVE MODE TIMING WITH SEN = 1 (RECEPTION, 10-BIT ADDRESS)



19.4.5 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an Acknowledge.

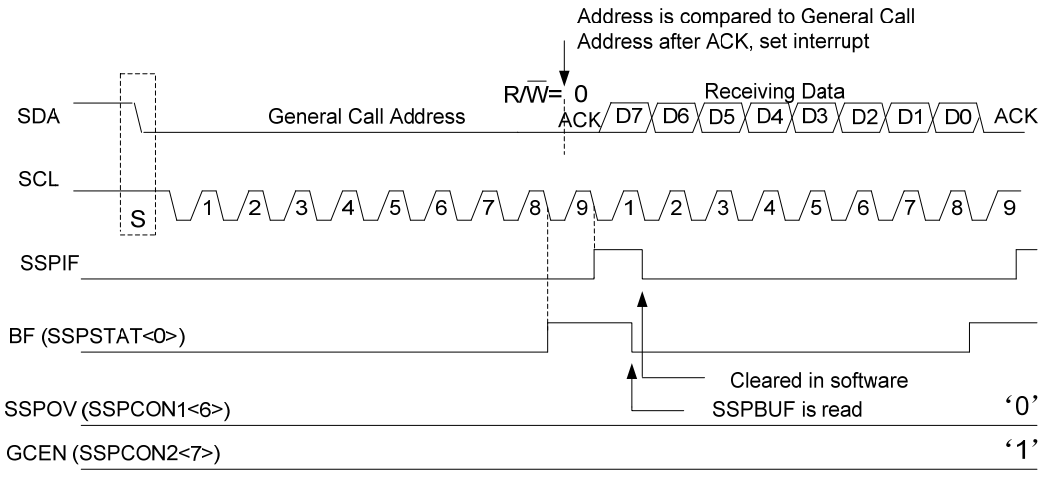
The general call address is one of eight addresses reserved for specific purposes by the I^2C protocol. It consists of all '0's with R/W = 0.

The general call address is recognized when the General Call Enable bit (GCEN) is enabled (SSPCON2<7>set). Following a Start bit detect, 8 bits are shifted into the SSPSR and the address is compared against the SSPADD. It is also compared to the general call address and fixed in hardware.

If the general call address matches, the SSPSR is transferred to the SSPBUF, the BF flag bit is set (eighth bit) and on the falling edge of the ninth bit (ACK bit), the SSPIF interrupt flag bit is set.

When the interrupt is serviced, the source for the interrupt can be checked by reading the contents of the SSPBUF. The value can be used to determine if the address was device specific or a general call address. In 10-bit mode, the SSPADD is required to be updated for the second half of the address to match and the UA bit is set (SSPSTAT<1>). If the general call address is sampled when the GCEN bit is set, while the slave is configured in 10-bit Address mode, then the second half of the address is not necessary, the UA bit will not be set and the slave will begin receiving data after the Acknowledge (Figure 19-15).

FIGURE 19-15: SLAVE MODE GENERAL CALL ADDRESS SEQUENCE (7 OR 10-BIT ADDRESS MODE)



19.4.6 MASTER MODE

Master mode is enabled by setting and clearing the appropriate SSPM bits in SSPCON1 and by setting the SSPEN bit. In Master mode, the SCL and SDA lines are manipulated by the MSSP hardware.

Master mode of operation is supported by interrupt generation on the detection of the Start and Stop conditions. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP

module is disabled. Control of the I²C bus may be taken when the P bit is set or the bus is Idle, with both the S and P bits clear.

In Firmware Controlled Master mode, user code conducts all I²C bus operations based on Start and Stop bit conditions.

Once Master mode is enabled, the user has six options.

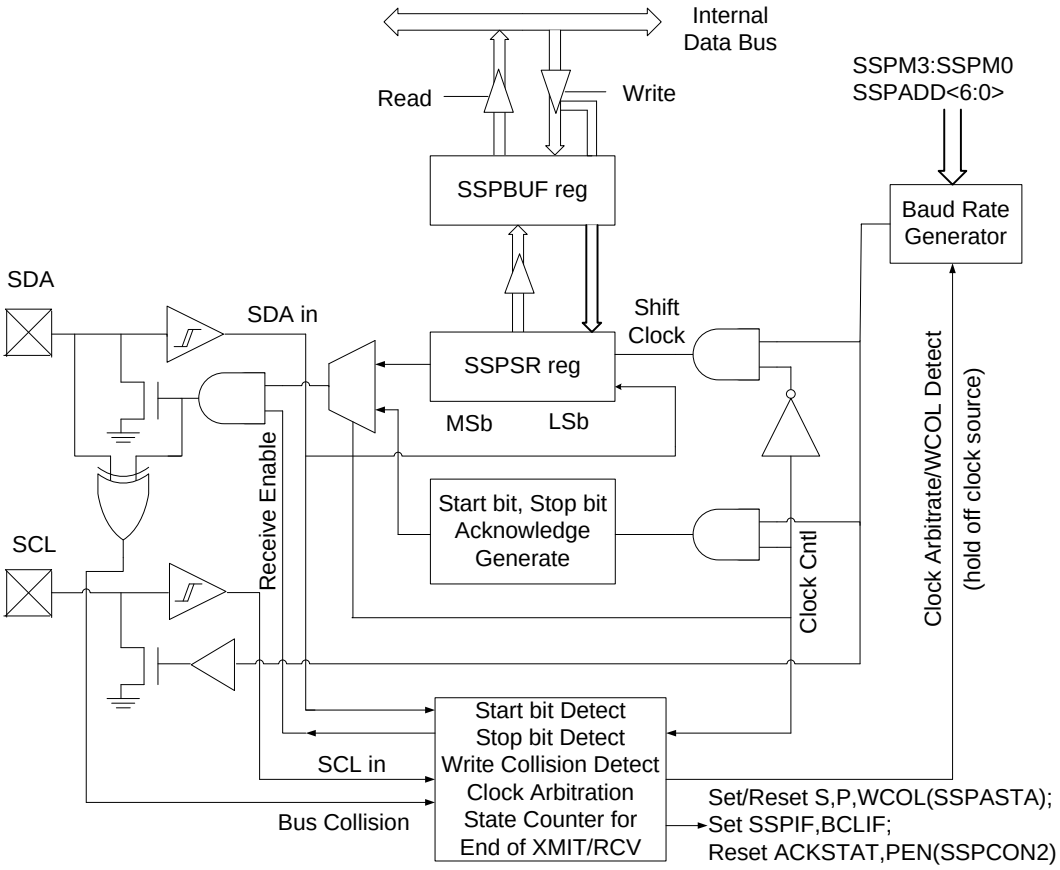
1. Assert a Start condition on SDA and SCL.
2. Assert a Repeated Start condition on SDA and SCL.
3. Write to the SSPBUF register initiating transmission of data/address.
4. Configure the I²C port to receive data.
5. Generate an Acknowledge condition at the end of a received byte of data.
6. Generate a Stop condition on SDA and SCL.

Note:	The MSSP module, when configured in I²C Master mode, does not allow queueing of events. For instance, the user is not allowed to initiate a Start condition and immediately write the SSPBUF register to initiate transmission before the Start condition is complete. In this case, the SSPBUF will not be written to and the WCOL bit will be set, indicating that a write to the SSPBUF did not occur.

The following events will cause SSP Interrupt Flag bit, SSPIF, to be set (SSP interrupt if enabled):

- Start condition
- Stop condition
- Data transfer byte transmitted/received
- Acknowledge transmit
- Repeated Start

FIGURE 19-16: MSSP BLOCK DIAGRAM (I2C™ MASTER MODE)



19.4.6.1 I²C Master Mode Operation

The master device generates all of the serial clock pulses and the Start and Stop conditions. A transfer is ended with a Stop condition, or with a Repeated Start condition. Since the Repeated Start condition is also the beginning of the next serial transfer, the I²C bus will not be released.

In Master Transmitter mode, serial data is output through SDA while SCL outputs the serial clock. The first byte transmitted contains the slave address of the receiving device (7 bits) and the Read/Write (R/W) bit. In this case, the R/W bit will be logic '0'. Serial data is transmitted 8 bits at a time. After each byte is transmitted, an Acknowledge bit is received. Start and Stop conditions are output to indicate the beginning and the end of a serial transfer.

In Master Receive mode, the first byte transmitted contains the slave address of the transmitting device (7 bits) and the R/W bit. In this case, the R/W bit will be logic '1'. Thus, the first byte transmitted is a 7-bit slave address followed by a '1' to indicate receive bit. Serial data is received via SDA while SCL outputs the serial clock. Serial data is received 8 bits at a time. After each byte is received, an Acknowledge bit is transmitted. Start and Stop conditions indicate the beginning and end of transmission.

The Baud Rate Generator used for the SPI mode operation is used to set the SCL clock frequency for either 100 kHz, 400 kHz or 1 MHz I²C operation. See Section 19.4.7 "Baud Rate Generator" for more details.

A typical transmit sequence would go as follows:

1. The user generates a Start condition by setting the Start Enable bit, SEN (SSPCON2<0>).
2. SSPIF is set. The MSSP module will wait the required start time before any other operation

takes place.

3. The user loads the SSPBUF with the slave address to transmit.
4. Address is shifted out the SDA pin until all 8 bits are transmitted.
5. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
6. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
7. The user loads the SSPBUF with eight bits of data.
8. Data is shifted out the SDA pin until all 8 bits are transmitted.
9. The MSSP module shifts in the ACK bit from the slave device and writes its value into the SSPCON2 register (SSPCON2<6>).
10. The MSSP module generates an interrupt at the end of the ninth clock cycle by setting the SSPIF bit.
11. The user generates a Stop condition by setting the Stop Enable bit PEN (SSPCON2<2>).
12. Interrupt is generated once the Stop condition is complete.

19.4.7 BAUD RATE GENERATOR

In I^2C Master mode, the Baud Rate Generator (BRG) reload value is placed in the lower 7 bits of the SSPADD register (Figure 19-17). When a write occurs to SSPBUF, the Baud Rate Generator will automatically begin counting. The BRG counts down to 0 and stops until another reload has taken place. The BRG count is decremented twice per instruction cycle (TCY) on the Q2 and Q4 clocks. In I^2C Master mode, the BRG is reloaded automatically.

Once the given operation is complete (i.e., transmission of the last data bit is followed by ACK), the internal clock will automatically stop counting and the SCL pin will remain in its last state.

Table 19-3 demonstrates clock rates based on instruction cycles and the BRG value loaded into SSPADD.

FIGURE 19-17: BAUD RATE GENERATOR BLOCK DIAGRAM

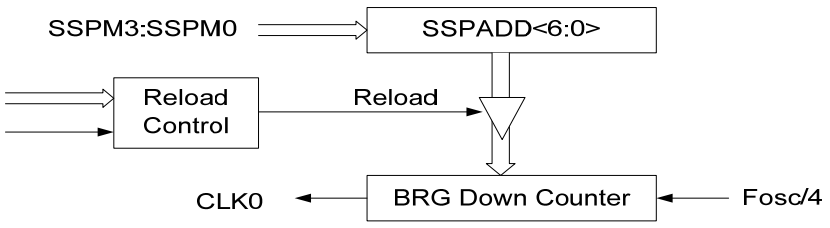


TABLE 19-3: I2C™ CLOCK RATE w/BRG

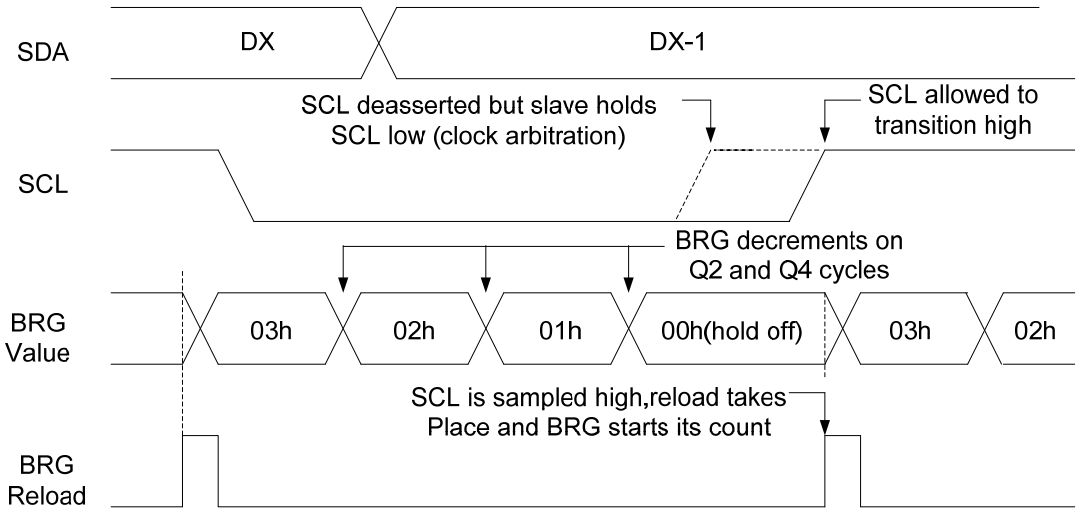
FOSC	FCY	FCY * 2	BRG Value	FSCL (2 Rollovers of BRG)
40 MHz	10 MHz	20 MHz	18h	400 kHz(1)
40 MHz	10 MHz	20 MHz	1Fh	312.5 kHz
40 MHz	10 MHz	20 MHz	63h	100 kHz
16 MHz	4 MHz	8 MHz	09h	400 kHz(1)
16 MHz	4 MHz	8 MHz	0Ch	308 kHz
16 MHz	4 MHz	8 MHz	27h	100 kHz
4 MHz	1 MHz	2 MHz	02h	333 kHz(1)
4 MHz	1 MHz	2 MHz	09h	100kHz
4 MHz	1 MHz	2 MHz	00h	1 MHz(1)

Note 1: The I²C™ interface does not conform to the 400 kHz I²C specification (which applies to rates greater than 100 kHz) in all details, but may be used with care where higher rates are required by the application.

19.4.7.1 Clock Arbitration

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, deasserts the SCL pin (SCL allowed to float high). When the SCL pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCL pin is actually sampled high. When the SCL pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. This ensures that the SCL high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 19-18).

FIGURE 19-18: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION



19.4.8 I²C MASTER MODE START CONDITION TIMING

To initiate a Start condition, the user sets the Start condition enable bit, SEN (SSPCON2<0>). If the SDA and SCL pins are sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0>and starts its count. If SCL and SDA are both sampled high when the Baud Rate Generator times out (TBRG), the SDA pin is driven low. The action of the SDA being driven low,

while SCL is high, is the Start condition and causes the S bit (SSPSTAT<3>) to be set. Following this, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and resumes its count. When the Baud Rate Generator times out (TBRG), the SEN bit (SSPCON2<0>) will be automatically cleared by hardware, the Baud Rate Generator is suspended, leaving the SDA line held low and the Start condition is complete.

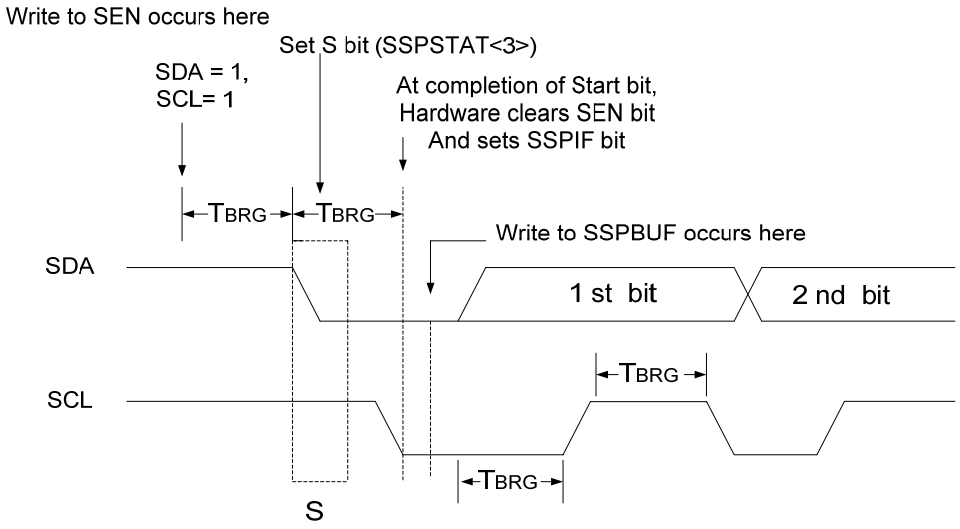
Note:	If, at the beginning of the Start condition, the SDA and SCL pins are already sampled low, or if during the Start condition, the SCL line is sampled low before the SDA line is driven low, a bus collision occurs; the Bus Collision Interrupt Flag, BCLIF, is set, the Start condition is aborted and the I^2C module is reset into its Idle state.

19.4.8.1 WCOL Status Flag

If the user writes the SSPBUF when a Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note:	Because queueing of events is not allowed, writing to the lower 5 bits of SSPCON2 is disabled until the Start condition is complete.

FIGURE 19-19: FIRST START BIT TIMING



19.4.9 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition occurs when the RSEN bit (SSPCON2<1>) is programmed high and the I²C logic module is in the Idle state. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded with the contents of SSPADD<5:0> and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded with the contents of SSPADD<6:0> and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. Following this, the RSEN bit (SSPCON2<1>) will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit (SSPSTAT<3>) will be set. The SSPIF bit will not be set until the Baud Rate Generator has timed out.

Note 1: If RSEN is programmed while any other event is in progress, it will not take effect.

2: A bus collision during the Repeated Start condition occurs if:

- SDA is sampled low when SCL goes from low-to-high.
- SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

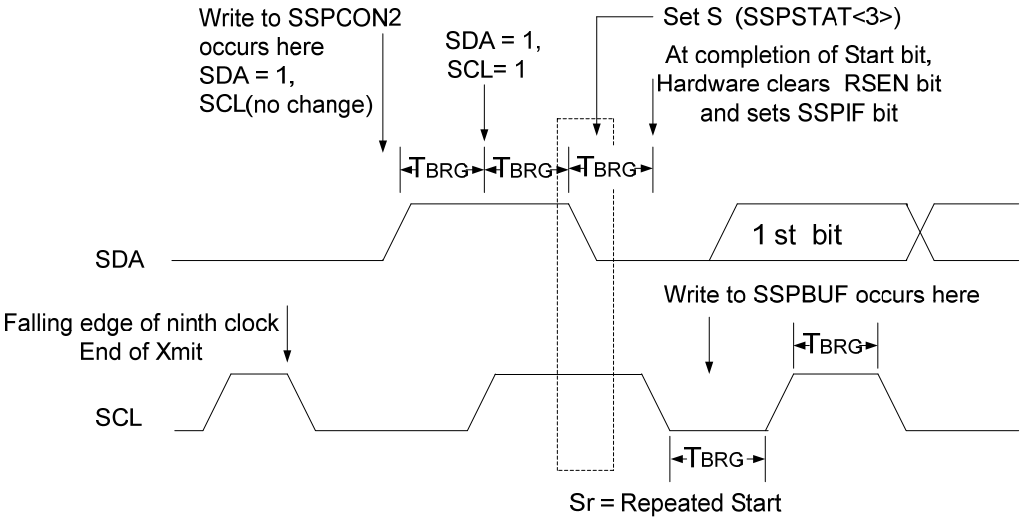
Immediately following the SSPIF bit getting set, the user may write the SSPBUF with the 7-bit address in 7-bit mode, or the default first address in 10-bit mode. After the first eight bits are transmitted and an ACK is received, the user may then transmit an additional eight bits of address (10-bit mode) or eight bits of data (7-bit mode).

19.4.9.1 WCOL Status Flag

If the user writes the SSPBUF when a Repeated Start sequence is in progress, the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

Note: Because queueing of events is not allowed, writing of the lower 5 bits of SSPCON2 is disabled until the Repeated Start condition is complete.

FIGURE 19-20: REPEATED START CONDITION WAVEFORM



19.4.10 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPBUF register. This action will set the Buffer Full flag bit BF and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted (see data hold time specification parameter #106). SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high (see data setup time specification parameter #107). When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an ACK bit during the ninth bit time, if an address match occurred, or if data was received properly. The status of ACK is written into the ACKDT bit on the falling edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPBUF, leaving SCL low and SDA unchanged (Figure 19-21).

After the write to the SSPBUF, each bit of address will be shifted out on the falling edge of SCL until all seven address bits and the R/W bit are completed. On the falling edge of the eighth clock, the master will deassert the SDA pin, allowing the slave to respond with an Acknowledge. On the falling edge of the ninth clock, the master will sample the SDA pin to see if the address was recognized by a slave. The status of the ACK bit is loaded into the ACKSTAT status bit (SSPCON2<6>). Following the falling edge of the ninth clock transmission of the address, the SSPIF bit is set, the BF flag is cleared and the Baud Rate Generator is turned off until another write to the SSPBUF takes place, holding SCL low and allowing SDA to float.

19.4.10.1 BF Status Flag

In Transmit mode, the BF bit (SSPSTAT<0>) is set when the CPU writes to SSPBUF and is cleared when all 8 bits are shifted out.

19.4.10.2 WCOL Status Flag

If the user writes the SSPBUF when a transmit is already in progress (i.e., SSPSR is still shifting out a data byte), the WCOL is set and the contents of the buffer are unchanged (the write doesn't occur). WCOL must be cleared in software.

19.4.10.3 ACKSTAT Status Flag

In Transmit mode, the ACKSTAT bit (SSPCON2<6>) is cleared when the slave has sent an Acknowledge (ACK = 0) and is set when the slave does not Acknowledge (ACK = 1). A slave sends an Acknowledge when it has recognized its address (including a general call) or when the slave has properly received its data.

19.4.11 I²C MASTER MODE RECEPTION

Master mode reception is enabled by programming the Receive Enable bit, RCEN (SSPCON2<3>).

Note:	The RCEN bit should be set after the $\overline{ACK}$ sequence is complete or the RCEN bit will be disregarded.

The Baud Rate Generator begins counting and on each rollover, the state of the SCL pin changes (high-to-low/low-to-high) and data is shifted into the SSPSR. After the falling edge of the eighth clock, the receive enable flag is automatically cleared, the contents of the SSPSR are loaded into the SSPBUF, the BF flag bit is set, the SSPIF flag bit is set and the Baud Rate Generator is suspended from counting, holding SCL low. The MSSP is now in Idle state awaiting the next command. When the buffer is read by the CPU, the BF flag bit is automatically cleared. The user can then send an Acknowledge bit at the end of reception by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>).

19.4.11.1 BF Status Flag

In receive operation, the BF bit is set when an address or data byte is loaded into SSPBUF from SSPSR. It is cleared when the SSPBUF register is read.

19.4.11.2 SSPOV Status Flag

In receive operation, the SSPOV bit is set when 8 bits are received into the SSPSR and the BF flag

bit is already set from a previous reception.

19.4.11.3 WCOL Status Flag

If the user writes the SSPBUF when a receive is already in progress (i.e., SSPSR is still shifting in a data byte), the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

FIGURE 19-21: I2C™ MASTER MODE WAVEFORM (TRANSMISSION, 7 OR 10-BIT ADDRESS)

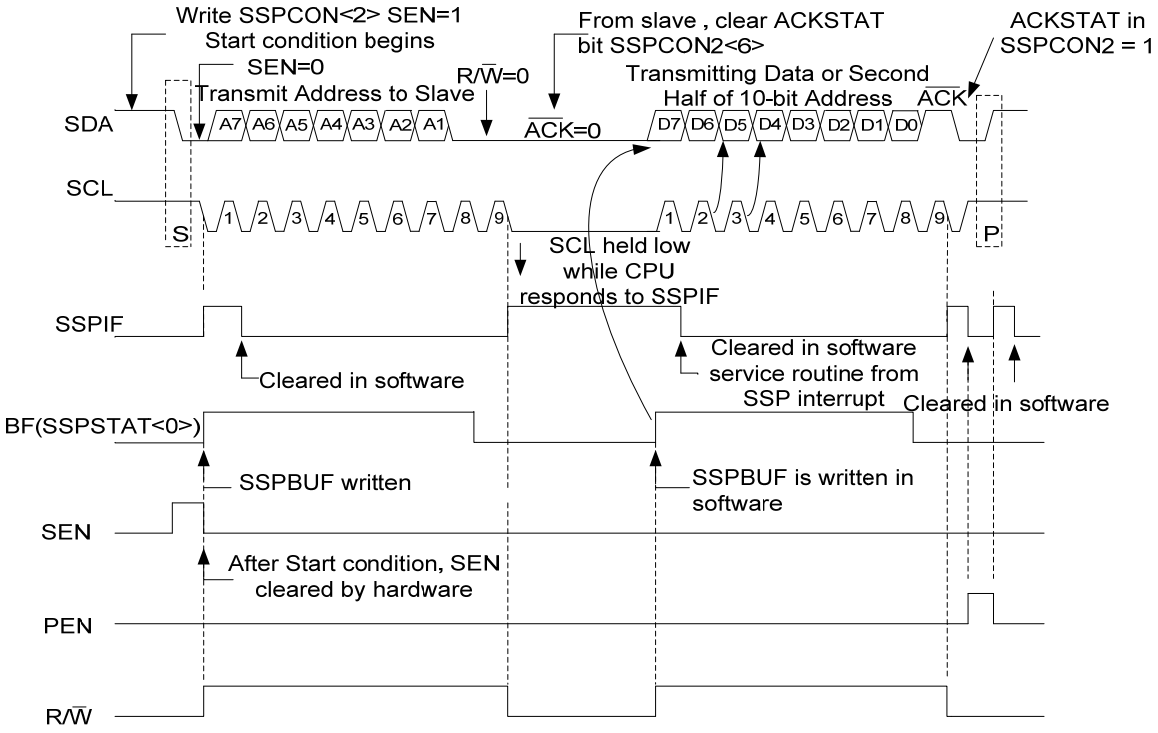
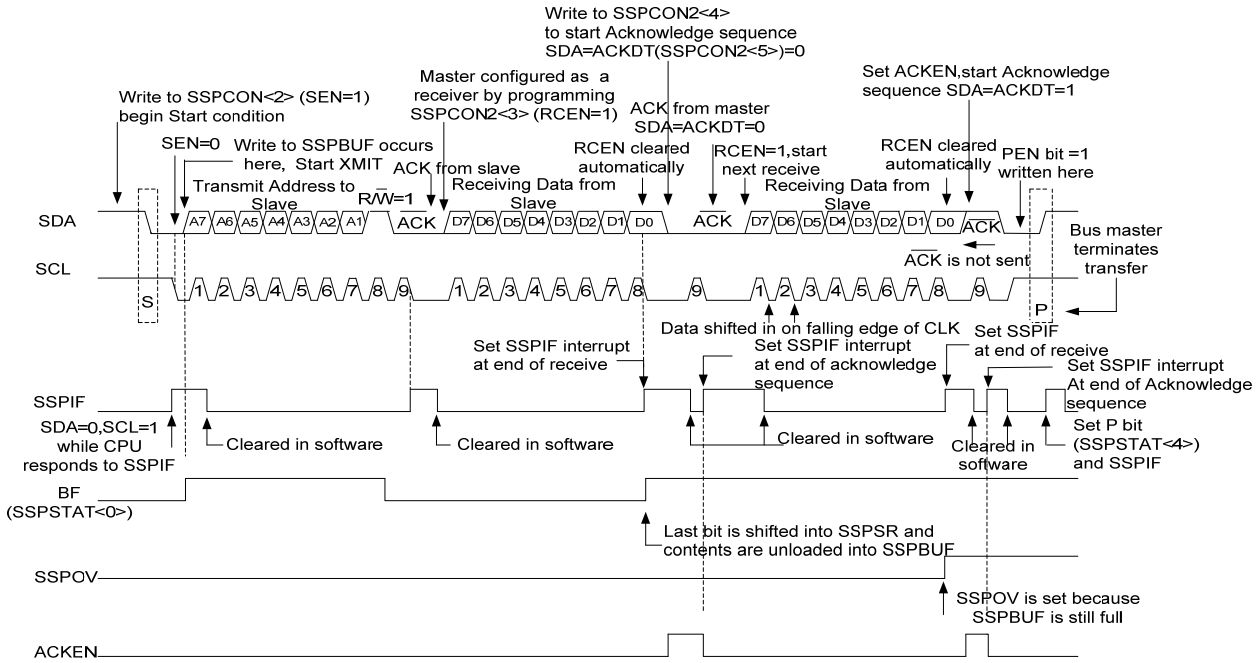


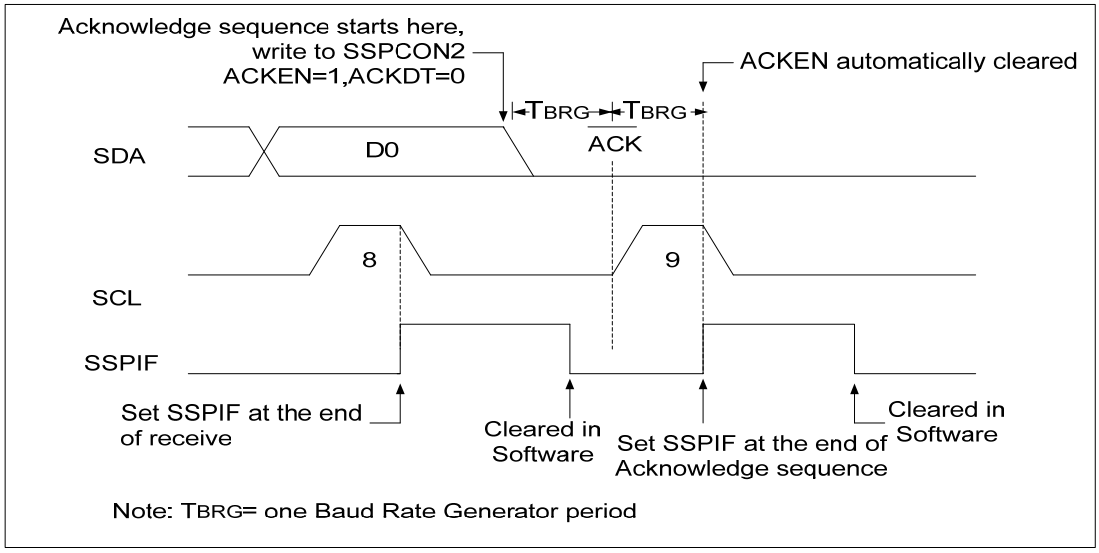
FIGURE 19-22: I2C™ MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



19.4.12 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled by setting the Acknowledge Sequence Enable bit, ACKEN (SSPCON2<4>). When this bit is set, the SCL pin is pulled low and the contents of the Acknowledge data bit are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The Baud Rate Generator then counts for one rollover period (TBRG) and the SCL pin is deasserted (pulled high). When the SCL pin is sampled high (clock arbitration), the Baud Rate Generator counts for TBRG. The SCL pin is then pulled low. Following this, the ACKEN bit is automatically cleared, the Baud Rate Generator is turned off and the MSSP module then goes into Idle mode (Figure 19-23).

FIGURE 19-23: ACKNOWLEDGE SEQUENCE WAVEFORM



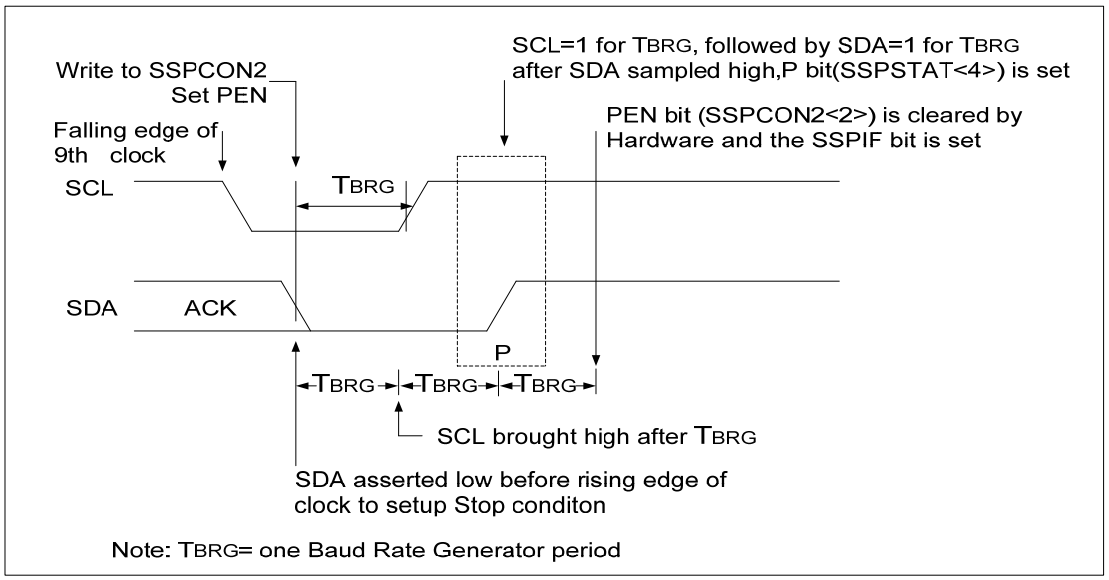
19.4.12.1 WCOL Status Flag

If the user writes the SSPBUF when an Acknowledge sequence is in progress, then WCOL is set and the contents of the buffer are unchanged (the write doesn't occur).

19.4.13 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of a receive/transmit by setting the Stop Sequence Enable bit, PEN (SSPCON2<2>). At the end of a receive/transmit, the SCL line is held low after the falling edge of the ninth clock. When the PEN bit is set, the master will assert the SDA line low. When the SDA line is sampled low, the Baud Rate Generator is reloaded and counts down to 0. When the Baud Rate Generator times out, the SCL pin will be brought high and one TBRG (Baud Rate Generator rollover count) later, the SDA pin will be deasserted. When the SDA pin is sampled high while SCL is high, the P bit (SSPSTAT<4>) is set. A TBRG later, the PEN bit is cleared and the SSPIF bit is set (Figure 19-24).

FIGURE 19-24: STOP CONDITION RECEIVE OR TRANSMIT MODE



19.4.13.1 WCOL Status Flag

If the user writes the SSPBUF when a Stop sequence is in progress, then the WCOL bit is set and the contents of the buffer are unchanged (the write doesn't occur).

19.4.14 SLEEP OPERATION

While in Sleep mode, the I^2C module can receive addresses or data and when an address match or complete byte transfer occurs, wake the processor from Sleep (if the MSSP interrupt is enabled).

19.4.15 EFFECT OF A RESET

A Reset disables the MSSP module and terminates the current transfer.

19.4.16 MULTI-MASTER MODE

In Multi-Master mode, the interrupt generation on the detection of the Start and Stop conditions allows the determination of when the bus is free. The Stop (P) and Start (S) bits are cleared from a Reset or when the MSSP module is disabled. Control of the I^2C bus may be taken when the P bit (SSPSTAT<4>) is set, or the bus is Idle, with both the S and P bits clear. When the bus is busy, enabling the SSP interrupt will generate the interrupt when the Stop condition occurs.

In multi-master operation, the SDA line must be monitored for arbitration to see if the signal level is the expected output level. This check is performed in hardware with the result placed in the BCLIF bit.

The states where arbitration can be lost are:

• Address Transfer

• Data Transfer
• A Start Condition
• A Repeated Start Condition
• An Acknowledge Condition

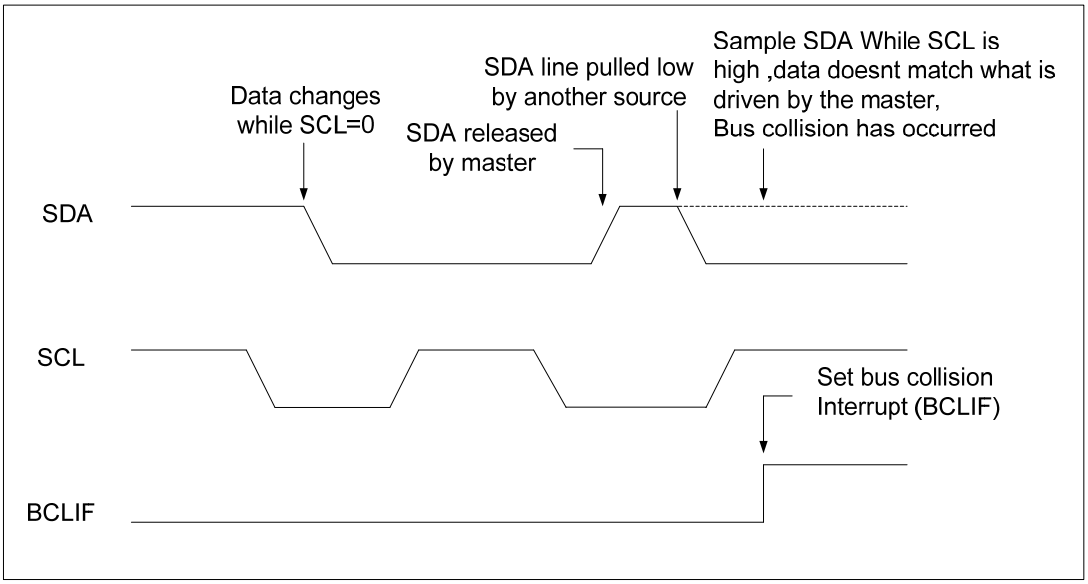
19.4.17 MULTI–MASTER COMMUNICATION, BUS COLLISION AND BUS ARBITRATION

Multi-Master mode support is achieved by bus arbitration. When the master outputs address/data bits onto the SDA pin, arbitration takes place when the master outputs a '1' on SDA by letting SDA float high and another master asserts a '0'. When the SCL pin floats high, data should be stable. If the expected data on SDA is a '1' and the data sampled on the SDA pin = 0, then a bus collision has taken place. The master will set the Bus Collision Interrupt Flag BCLIF and reset the I^2C port to its Idle state (Figure 19-25). If a transmit was in progress when the bus collision occurred, the transmission is halted, the BF flag is cleared, the SDA and SCL lines are deasserted and the SSPBUF can be written to. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition. If a Start, Repeated Start, Stop or Acknowledge condition was in progress when the bus collision occurred, the condition is aborted, the SDA and SCL lines are deasserted and the respective control bits in the SSPCON2 register are cleared. When the user services the bus collision Interrupt Service Routine and if the I^2C bus is free, the user can resume communication by asserting a Start condition.

The master will continue to monitor the SDA and SCL pins. If a Stop condition occurs, the SSPIF bit will be set. A write to the SSPBUF will start the transmission of data at the first data bit regardless of where the transmitter left off when the bus collision occurred.

In Multi-Master mode, the interrupt generation on the detection of Start and Stop conditions allows the determination of when the bus is free. Control of the I^2C bus can be taken when the P bit is set in the SSPSTAT register or the bus is Idle and the S and P bits are cleared.

FIGURE 19-25: BUS COLLISION TIMING FOR TRANSMIT AND ACKNOWLEDGE



19.4.17.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:
a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 19-26).
b) SCL is sampled low before SDA is asserted low (Figure 19-27).
During a Start condition, both the SDA and the SCL pins are monitored.
If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:
• the Start condition is aborted,
• the BCLIF flag is set and
• the MSSP module is reset to its Idle state
(Figure 19-26).
The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded from SSPADD<6:0>and counts down to 0. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.
If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 19-28). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to 0 and during this time, if the SCL pins are sampled as '0', a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note:	The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.

FIGURE 19-26: BUS COLLISION DURING START CONDITION (SDA ONLY)

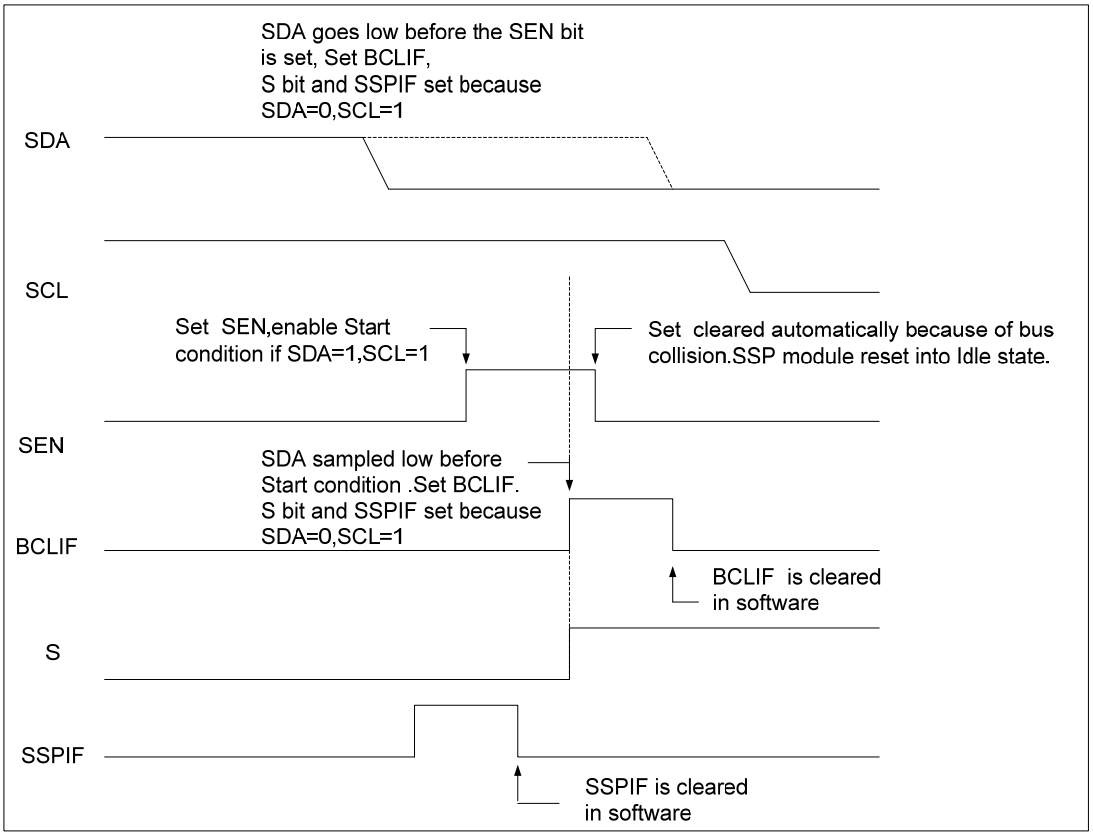


FIGURE 19-27: BUS COLLISION DURING START CONDITION (SCL = 0)

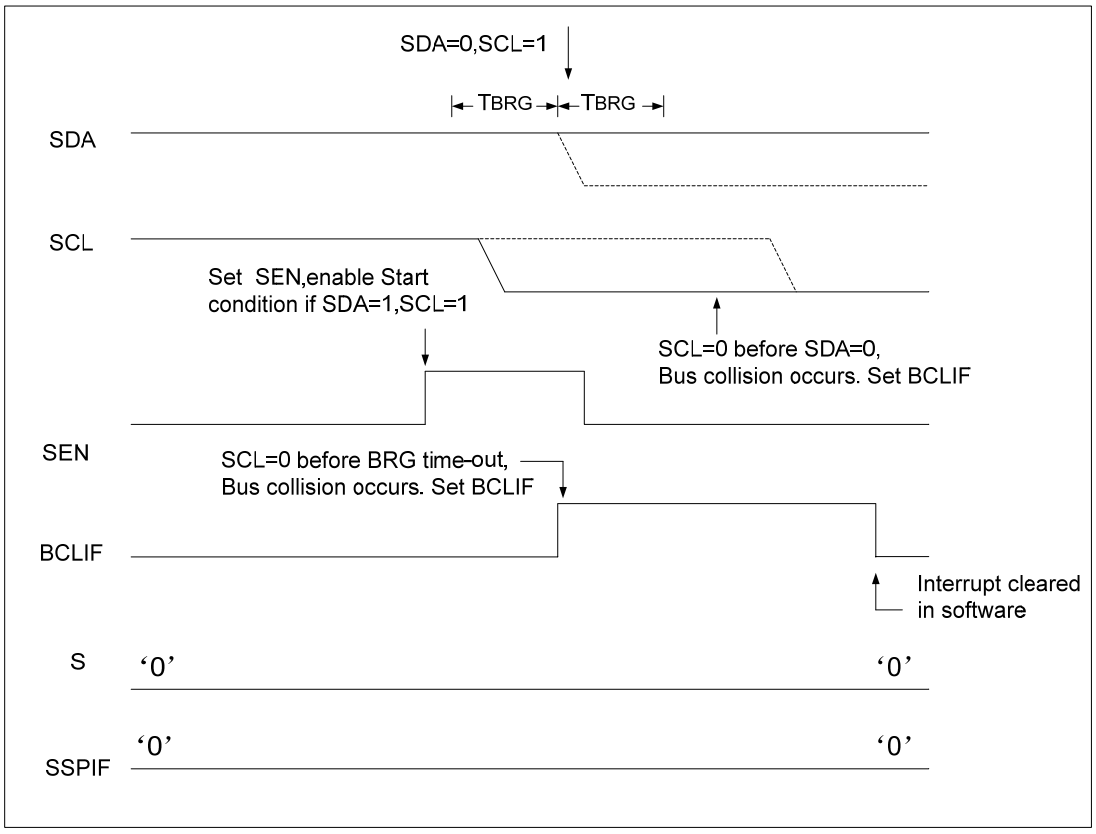
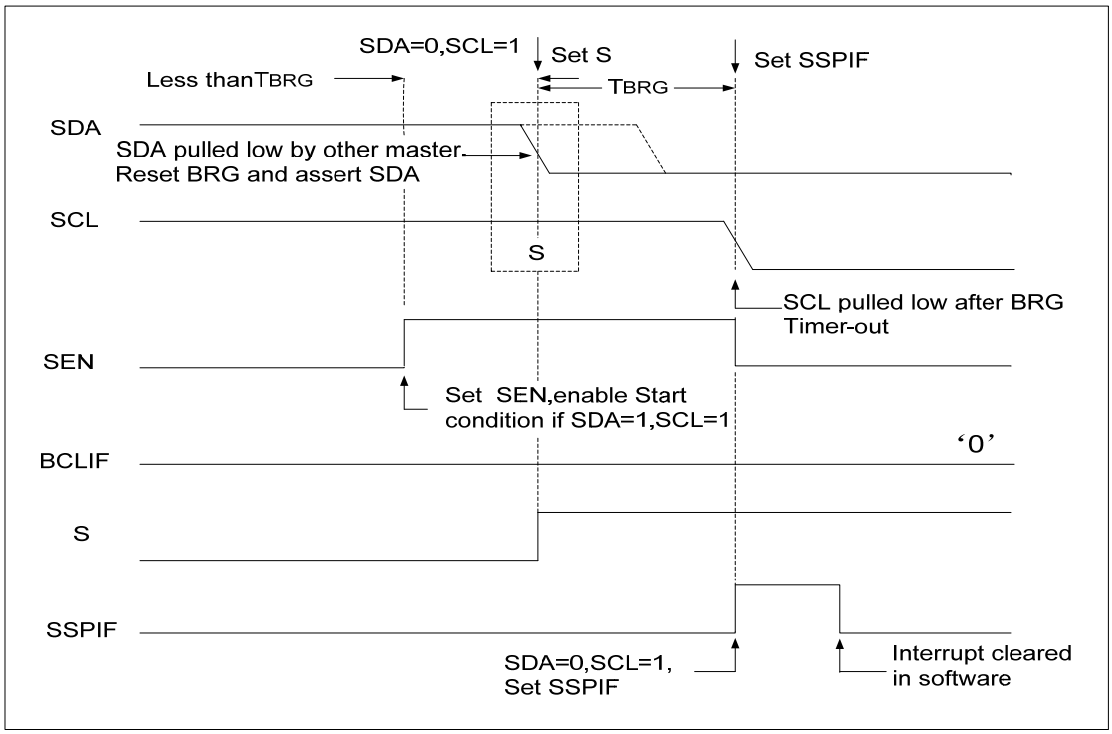


FIGURE 19-28: BRG RESET DUE TO SDA ARBITRATION DURING START CONDITION



19.4.17.2 Bus Collision During a Repeated Start Condition

During a Repeated Start condition, a bus collision occurs if:

a) A low level is sampled on SDA when SCL goes from low level to high level.

b) SCL goes low before SDA is asserted low, indicating that another master is attempting to transmit a data '1'.

When the user deasserts SDA and the pin is allowed to float high, the BRG is loaded with SSPADD<6:0> and counts down to 0. The SCL pin is then deasserted and when sampled high, the SDA pin is sampled.

If SDA is low, a bus collision has occurred (i.e., another master is attempting to transmit a data '0', Figure 19-29). If SDA is sampled high, the BRG is reloaded and begins

counting. If SDA goes from high-to-low before the BRG times out, no bus collision occurs because no two masters can assert SDA at exactly the same time.If SCL goes from high-to-low before the BRG times out and SDA has not already been asserted, a bus collision occurs. In this case, another master is attempting to transmit a data '1' during the Repeated Start condition (Figure 19-30).

If, at the end of the BRG time-out, both SCL and SDA are still high, the SDA pin is driven low and the BRG is reloaded and begins counting. At the end of the count, regardless of the status of the SCL pin, the SCL pin is driven low and the Repeated Start condition is complete.

FIGURE 19-29: BUS COLLISION DURING A REPEATED START CONDITION (CASE 1)

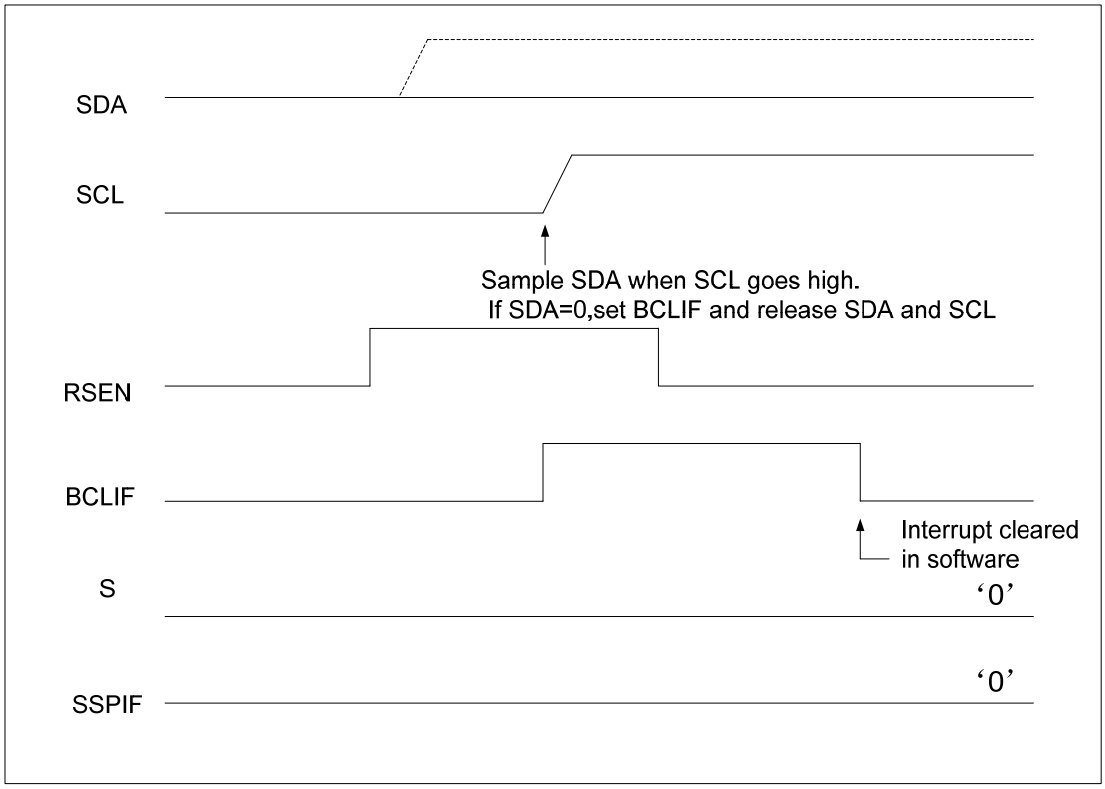
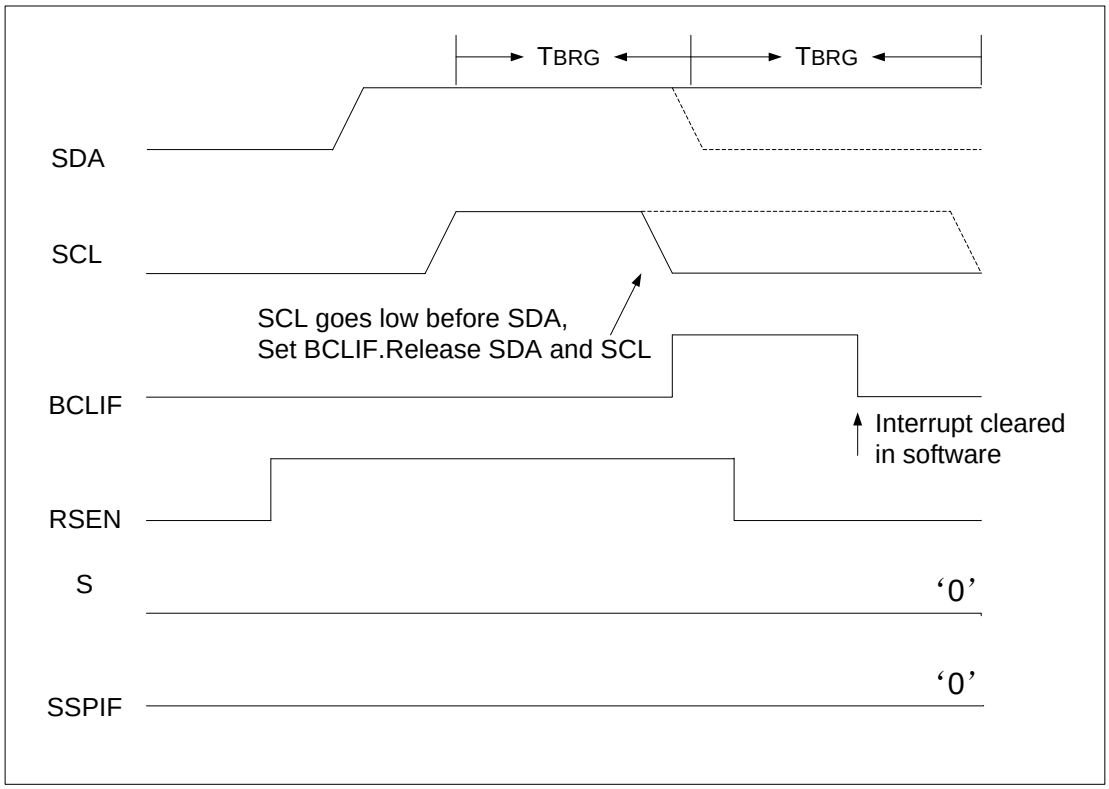


FIGURE 19-30: BUS COLLISION DURING A REPEATED START CONDITION (CASE 2)



19.4.17.3 Bus Collision During a Stop Condition

Bus collision occurs during a Stop condition if:

a) After the SDA pin has been deasserted and allowed to float high, SDA is sampled low after the BRG has timed out.

b) After the SCL pin is deasserted, SCL is sampled low before SDA goes high.

The Stop condition begins with SDA asserted low. When SDA is sampled low, the SCL pin is allowed to float. When the pin is sampled high (clock arbitration), the Baud Rate Generator is loaded with SSPADD<6:0>and counts down to 0. After the BRG times out, SDA is sampled. If SDA is sampled low, a bus collision has occurred. This is due to another master attempting to drive a data '0' (Figure 19-31). If the SCL pin is sampled low before SDA is allowed to float high, a bus collision occurs. This is another case of another master attempting to drive a data '0' (Figure 19-32).

FIGURE 19-31: BUS COLLISION DURING A STOP CONDITION (CASE 1)

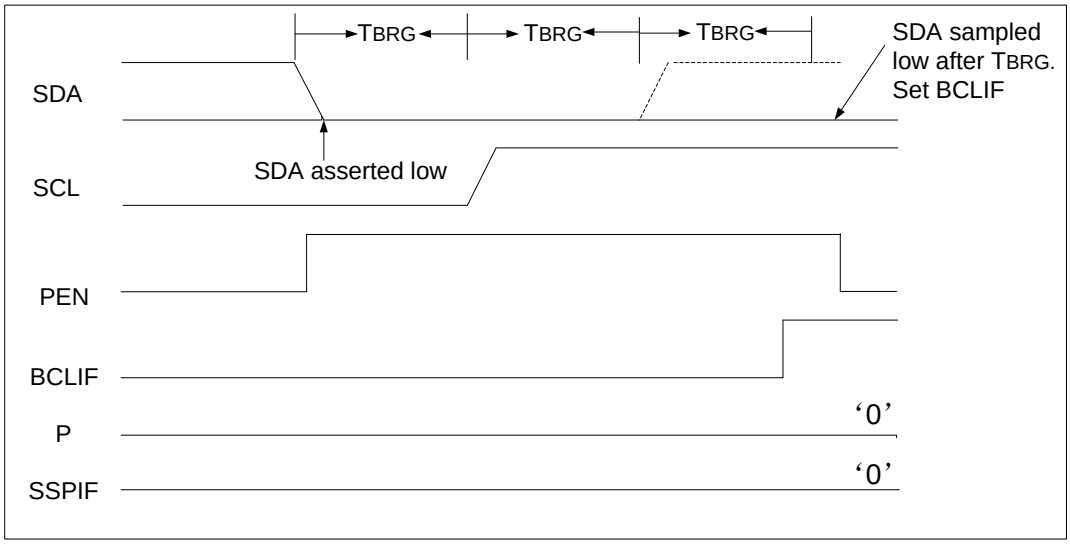
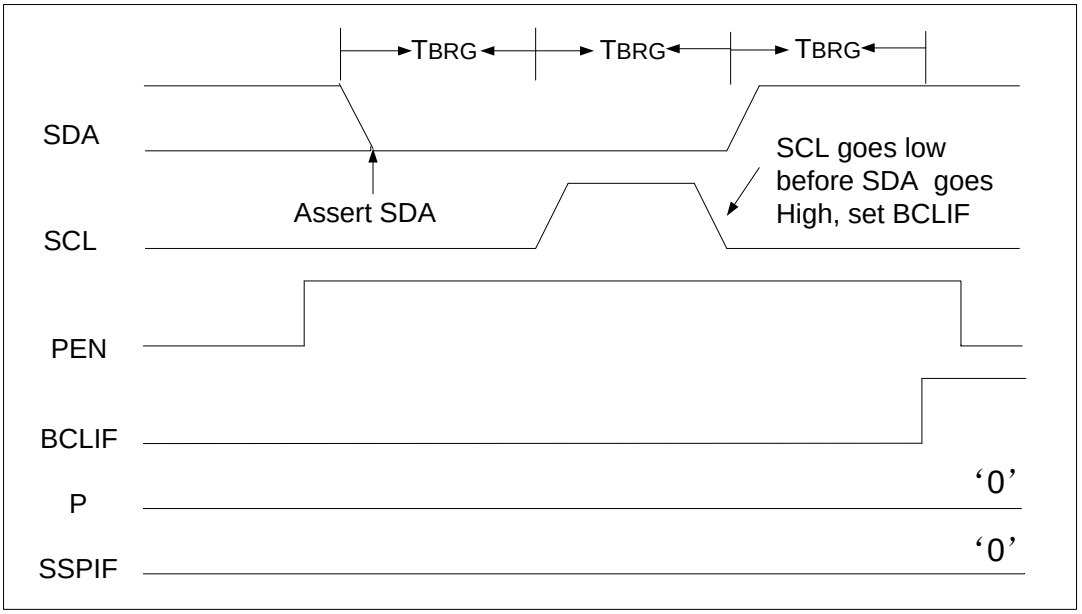


FIGURE 19-32: BUS COLLISION DURING A STOP CONDITION (CASE 2)



20. UNIVERSAL ASYNCHRONOUS RECEIVER TRANSMITTER (UART)

The Universal Asynchronous Receiver Transmitter (UART) be configured as a full-duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers.

The UART can be configured in the following modes:

• Asynchronous (full-duplex)

The SPEN (RCSTA register) and the DDRB<4> bits have to be set and the DDRB<5> bit must be cleared in order to configure pins RB4/TX and RB5/RX as the Universal Asynchronous Receiver Transmitter.

Register 20-1 shows the Transmit Status and Control register (TXSTA) and Register 20-2 shows the Receive Status and Control register (RCSTA).

REGISTER 20-1: TXSTA: TRANSMIT STATUS AND CONTROL REGISTER

Read/Wite (0XFAC)	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0	
	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	TXSTA
Bit	bit 7							bit 0	

bit 7 **CSRC**: Clock Source Select bit
Asynchronous mode:
Don't care.
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9**: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN**: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
Note: SREN/CREN overrides TXEN in Sync mode.

bit 4 **SYNC**: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode

bit 3 Unimplemented: Read as '0'

bit 2 **BRGH**: High Baud Rate Select bit
Asynchronous mode:
1 = High speed

0 = Low speed

Synchronous mode:

Unused in this mode.

bit 1 **TRMT**: Transmit Shift Register Status bit

1 = TSR empty

0 = TSR full

bit 0 **TX9D**: 9th bit of Transmit Data

Can be address/data bit or a parity bit.

REGISTER 20-2: RCSTA: RECEIVE STATUS AND CONTROL REGISTER

Read/Wite (0XFAB) Bit	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x	
	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	RCSTA
	bit 7							bit 0	

bit 7 **SPEN:** Serial Port Enable bit

1 = Serial port enabled (configures RX/DT and TX/CK pins as serial port pins)

0 = Serial port disabled

bit 6 **RX9:** 9-bit Receive Enable bit

1 = Selects 9-bit reception

0 = Selects 8-bit reception

bit 5 **SREN:** Single Receive Enable bit

Asynchronous mode:

Don't care.

Synchronous mode – Master:

1 = Enables single receive

0 = Disables single receive (this bit is cleared after reception is complete)

Synchronous mode – Slave:

Unused in this mode.

bit 4 **CREN:** Continuous Receive Enable bit

Asynchronous mode:

1 = Enables continuous receive

0 = Disables continuous receive

Synchronous mode:

1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)

0 = Disables continuous receive

bit 3 **ADDEN:** Address Detect Enable bit

Asynchronous mode 9-bit (RX9 = 1):

1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8>is set

0 = Disables address detection, all bytes are received and ninth bit can be used as parity bit

bit 2 **FERR:** Framing Error bit

1 = Framing error (can be updated by reading RCREG register and receive next valid byte)

0 = No framing error

bit 1 **OERR:** Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error
bit 0 **RX9D:** 9th bit of Received Data
Can be address/data bit or a parity bit.

20.1 UART Baud Rate Generator (BRG)

The BRG supports both the Asynchronous and Synchronous modes of the USART. It is a dedicated 8-bit Baud Rate Generator. The SPBRG register controls the period of a free running, 8-bit timer. In Asynchronous mode, bit BRGH (TXSTA register) also controls the baud rate. In Synchronous mode, bit BRGH is ignored. Table 20-1 shows the formula for computation of the baud rate for different UART modes which only apply in Master mode (internal clock).
Given the desired baud rate and FOSC, the nearest integer value for the SPBRG register can be calculated using the formula in Table 20-1. From this, the error in baud rate can be determined.
Example 20-1 shows the calculation of the baud rate error for the following conditions:

FOSC = 16 MHz
Desired Baud Rate = 9600
BRGH = 0
SYNC = 0

It may be advantageous to use the high baud rate (BRGH = 1) even for slower baud clocks. This is because the FOSC/(16(X + 1)) equation can reduce the baud rate error in some cases.
Writing a new value to the SPBRG register causes the BRG timer to be reset (or cleared). This ensures the BRG does not wait for a timer overflow before outputting the new baud rate.

20.1.1 SAMPLING

The data on the RC7/RX/DT pin is sampled three times by a majority detect circuit to determine if a high or a low level is present at the RX pin.

EXAMPLE 20-1: CALCULATING BAUD RATE ERROR

Desired Baud Rate	=	Fosc/(64 (X + 1))
Solving for X:		
X	=	((Fosc/Desired Baud Rate)/64) – 1
X	=	((16000000/9600)/64) – 1
X	=	[25.042] = 25
Calculated Baud Rate	=	16000000/(64 (25 + 1))
	=	9615
Error	=	(Calculated Baud Rate – Desired Baud Rate) / Desired Baud Rate
	=	(9615 – 9600)/9600
	=	0.16%

TABLE 20-1: BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = Fosc/(64 (X + 1))	Baud Rate = Fosc/(16 (X + 1))
1	(Synchronous) Baud Rate = Fosc/(4 (X + 1))	NA

Legend: X = value in SPBRG (0 to 255)

TABLE 20-2: REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000u
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.

TABLE 20-3: BAUD RATES FOR SYNCHRONOUS MODE

BAUD RATE (Kbps)	Fosc = 40 MHz		SPBRG value (decimal)	33 MHz		SPBRG value (decimal)	25 MHz		SPBRG value (decimal)	20 MHz		SPBRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	NA	-	-	NA	-	-
19.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
76.8	76.92	+0.16	129	77.10	+0.39	106	77.16	+0.47	80	76.92	+0.16	64
96	96.15	+0.16	103	95.93	-0.07	85	96.15	+0.16	64	96.15	+0.16	51
300	303.03	+1.01	32	294.64	-1.79	27	297.62	-0.79	20	294.12	-1.96	16
500	500	0	19	485.30	-2.94	16	480.77	-3.85	12	500	0	9
HIGH	10000	-	0	8250	-	0	6250	-	0	5000	-	0
LOW	39.06	-	255	32.23	-	255	24.41	-	255	19.53	-	255

BAUD RATE (Kbps)	Fosc = 16 MHz		SPBRG value (decimal)	10 MHz		SPBRG value (decimal)	7.15909 MHz		SPBRG value (decimal)	5.0688 MHz		SPBRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	NA	-	-	9.62	+0.23	185	9.60	0	131
19.2	19.23	+0.16	207	19.23	+0.16	129	19.24	+0.23	92	19.20	0	65
76.8	76.92	+0.16	51	75.76	-1.36	32	77.82	+1.32	22	74.54	-2.94	16
96	95.24	-0.79	41	96.15	+0.16	25	94.20	-1.88	18	97.48	+1.54	12
300	307.70	+2.56	12	312.50	+4.17	7	298.35	-0.57	5	316.80	+5.60	3
500	500	0	7	500	0	4	447.44	-10.51	3	422.40	-15.52	2
HIGH	4000	-	0	2500	-	0	1789.80	-	0	1267.20	-	0
LOW	15.63	-	255	9.77	-	255	6.99	-	255	4.95	-	255

BAUD RATE (Kbps)	Fosc = 4 MHz		SPBRG value (decimal)	3.579545 MHz		SPBRG value (decimal)	1 MHz		SPBRG value (decimal)	32.768 kHz		SPBRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	NA	-	-	NA	-	-	NA	-	-	0.30	+1.14	26
1.2	NA	-	-	NA	-	-	1.20	+0.16	207	1.17	-2.48	6
2.4	NA	-	-	NA	-	-	2.40	+0.16	103	2.73	+13.78	2
9.6	9.62	+0.16	103	9.62	+0.23	92	9.62	+0.16	25	8.20	-14.67	0
19.2	19.23	+0.16	51	19.04	-0.83	46	19.23	+0.16	12	NA	-	-
76.8	76.92	+0.16	12	74.57	-2.90	11	83.33	+8.51	2	NA	-	-
96	1000	+4.17	9	99.43	+3.57	8	83.33	-13.19	2	NA	-	-
300	333.33	+11.11	2	298.30	-0.57	2	250	-16.67	0	NA	-	-
500	500	0	1	447.44	-10.51	1	NA	-	-	NA	-	-
HIGH	1000	-	0	894.89	-	0	250	-	0	8.20	-	0
LOW	3.91	-	255	3.50	-	255	0.98	-	255	0.03	-	255

TABLE 20-4: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (Kbps)	Fosc = 40 MHz		SPBRG value (decimal)	33 MHz		SPBRG value (decimal)	25 MHz		SPBRG value (decimal)	20 MHz		SPBRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	2.40	-0.07	214	2.40	-0.15	162	2.40	+0.16	129
9.6	9.62	+0.16	64	9.55	-0.54	53	9.53	-0.76	40	9.47	-1.36	32
19.2	18.94	-1.36	32	19.10	-0.54	26	19.53	+1.73	19	19.53	+1.73	15
76.8	78.13	+1.73	7	73.66	-4.09	6	78.13	+1.73	4	78.13	+1.73	3
96	89.29	-6.99	6	103.13	+7.42	4	97.66	+1.73	3	104.17	+8.51	2
300	312.50	+4.17	1	257.81	-14.06	1	NA	-	-	312.50	+4.17	0
500	625	+25.00	0	NA	-	-	NA	-	-	NA	-	-
HIGH	625	-	0	515.63	-	0	390.63	-	0	312.50	-	0
LOW	2.44	-	255	2.01	-	255	1.53	-	255	1.22	-	255

BAUD RATE (Kbps)	Fosc = 16 MHz		SPBRG value (decimal)	10 MHz		SPBRG value (decimal)	7.15909 MHz		SPBRG value (decimal)	5.0688 MHz		SPBRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	1.20	+0.16	207	1.20	+0.16	129	1.20	+0.23	92	1.20	0	65
2.4	2.40	+0.16	103	2.40	+0.16	64	2.38	-0.83	46	2.40	0	32
9.6	9.62	+0.16	25	9.77	+1.73	15	9.32	-2.90	11	9.90	+3.13	7
19.2	19.23	+0.16	12	19.53	+1.73	7	18.64	-2.90	5	19.80	+3.13	3
76.8	83.33	+8.51	2	78.13	+1.73	1	111.86	+45.65	0	79.20	+3.13	0
96	83.33	-13.19	2	78.13	-18.62	1	NA	-	-	NA	-	-
300	250	-16.67	0	156.25	-47.92	0	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	250	-	0	156.25	-	0	111.86	-	0	79.20	-	0
LOW	0.98	-	255	0.61	-	255	0.44	-	255	0.31	-	255

BAUD RATE (Kbps)	Fosc = 4 MHz		SPBRG value (decimal)	3.579545 MHz		SPBRG value (decimal)	1 MHz		SPBRG value (decimal)	32.768 kHz		SPBRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	0.30	-0.16	207	0.30	+0.23	185	0.30	+0.16	51	0.26	-14.67	1
1.2	1.20	+1.67	51	1.19	-0.83	46	1.20	+0.16	12	NA	-	-
2.4	2.40	+1.67	25	2.43	+1.32	22	2.23	-6.99	6	NA	-	-
9.6	8.93	-6.99	6	9.32	-2.90	5	7.81	-18.62	1	NA	-	-
19.2	20.83	+8.51	2	18.64	-2.90	2	15.63	-18.62	0	NA	-	-
76.8	62.50	-18.62	0	55.93	-27.17	0	NA	-	-	NA	-	-
96	NA	-	-	NA	-	-	NA	-	-	NA	-	-
300	NA	-	-	NA	-	-	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	62.50	-	0	55.93	-	0	15.63	-	0	0.51	-	0
LOW	0.24	-	255	0.22	-	255	0.06	-	255	0.002	-	255

TABLE 20-5: BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (Kbps)	Fosc = 40 MHz		SPBRG value (decimal)	33 MHz		SPBRG value (decimal)	25 MHz		SPBRG value (decimal)	20 MHz		SPBRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	NA	-	-	NA	-	-
9.6	NA	-	-	9.60	-0.07	214	9.59	-0.15	162	9.62	+0.16	129
19.2	19.23	+0.16	129	19.28	+0.39	106	19.30	+0.47	80	19.23	+0.16	64
76.8	75.76	-1.36	32	76.39	-0.54	26	78.13	+1.73	19	78.13	+1.73	15
96	96.15	+0.16	25	98.21	+2.31	20	97.66	+1.73	15	96.15	+0.16	12
300	312.50	+4.17	7	294.64	-1.79	6	312.50	+4.17	4	312.50	+4.17	3
500	500	0	4	515.63	+3.13	3	520.83	+4.17	2	416.67	-16.67	2
HIGH	2500	-	0	2062.50	-	0	1562.50	-	0	1250	-	0
LOW	9.77	-	255	8.06	-	255	6.10	-	255	4.88	-	255

BAUD RATE (Kbps)	Fosc = 16 MHz		SPBRG value (decimal)	10 MHz		SPBRG value (decimal)	7.15909 MHz		SPBRG value (decimal)	5.0688 MHz		SPBRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	NA	-	-	NA	-	-	NA	-	-	NA	-	-
1.2	NA	-	-	NA	-	-	NA	-	-	NA	-	-
2.4	NA	-	-	NA	-	-	2.41	+0.23	185	2.40	0	131
9.6	9.62	+0.16	103	9.62	+0.16	64	9.52	-0.83	46	9.60	0	32
19.2	19.23	+0.16	51	18.94	-1.36	32	19.45	+1.32	22	18.64	-2.94	16
76.8	76.92	+0.16	12	78.13	+1.73	7	74.57	-2.90	5	79.20	+3.13	3
96	100	+4.17	9	89.29	-6.99	6	89.49	-6.78	4	105.60	+10.00	2
300	333.33	+11.11	2	312.50	+4.17	1	447.44	+49.15	0	316.80	+5.60	0
500	500	0	1	625	+25.00	0	447.44	-10.51	0	NA	-	-
HIGH	1000	-	0	625	-	0	447.44	-	0	316.80	-	0
LOW	3.91	-	255	2.44	-	255	1.75	-	255	1.24	-	255

BAUD RATE (Kbps)	Fosc = 4 MHz		SPBRG value (decimal)	3.579545 MHz		SPBRG value (decimal)	1 MHz		SPBRG value (decimal)	32.768 kHz		SPBRG value (decimal)
	KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR		KBAUD	% ERROR	
0.3	NA	-	-	NA	-	-	0.30	+0.16	207	0.29	-2.48	6
1.2	1.20	+0.16	207	1.20	+0.23	185	1.20	+0.16	51	1.02	-14.67	1
2.4	2.40	+0.16	103	2.41	+0.23	92	2.40	+0.16	25	2.05	-14.67	0
9.6	9.62	+0.16	25	9.73	+1.32	22	8.93	-6.99	6	NA	-	-
19.2	19.23	+0.16	12	18.64	-2.90	11	20.83	+8.51	2	NA	-	-
76.8	NA	-	-	74.57	-2.90	2	62.50	-18.62	0	NA	-	-
96	NA	-	-	111.86	+16.52	1	NA	-	-	NA	-	-
300	NA	-	-	223.72	-25.43	0	NA	-	-	NA	-	-
500	NA	-	-	NA	-	-	NA	-	-	NA	-	-
HIGH	250	-	0	55.93	-	0	62.50	-	0	2.05	-	0
LOW	0.98	-	255	0.22	-	255	0.24	-	255	0.008	-	255

20.2 UART Asynchronous Mode

In this mode, the UART uses standard Non-Return-to-Zero (NRZ) format (one Start bit, eight or nine data bits and one Stop bit). The most common data format is 8 bits. An on-chip dedicated 8-bit Baud Rate Generator can be used to derive standard baud rate frequencies from the oscillator. The UART transmits and receives the LSb first. The UART's transmitter and receiver are functionally independent but use the same data format and baud rate. The Baud Rate Generator produces a clock, either x16 or x64 of the bit shift rate, depending on the BRGH bit (TXSTA register). Parity is not supported by the hardware but can be implemented in software (and stored as the ninth data bit). Asynchronous mode is stopped during Sleep.

Asynchronous mode is selected by clearing the SYNC bit (TXSTA register).

The USART Asynchronous module consists of the following important elements:

• Baud Rate Generator
• Sampling Circuit
• Asynchronous Transmitter
• Asynchronous Receiver.

20.2.1 UART ASYNCHRONOUS TRANSMITTER

The UART transmitter block diagram is shown in Figure 20-1. The heart of the transmitter is the Transmit (Serial) Shift Register (TSR). The TSR register obtains its data from the Read/Write Transmit Buffer register (TXREG). The TXREG register is loaded with data in software. The TSR

register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the TSR is loaded with new data from the TXREG register (if available). Once the TXREG register transfers the data to the TSR register (occurs in one TCY), the TXREG register is empty and flag bit TXIF (PIR1 register) is set. This interrupt can be enabled/disabled by setting/clearing enable bit TXIE (PIE1 register). Flag bit TXIF will be set regardless of the state of enable bit TXIE and cannot be cleared in software. It will reset only when new data is loaded into the TXREG register. While flag bit, TXIF, indicated the status of the TXREG register, another bit, TRMT (TXSTA register), shows the status of the TSR register. Status bit TRMT is a read-only bit which is set when the TSR register is empty. No interrupt logic is tied to this bit, so the user has to poll this bit in order to determine if the TSR register is empty.

Note 1: The TSR register is not mapped in data memory, so it is not available to the user.
2: Flag bit TXIF is set when enable bit TXENis set.

Steps to follow when setting up an Asynchronous Transmission:
1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH (Section 20.1 “USART Baud Rate Generator (BRG)”).
2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit TXIE.
4. If 9-bit transmission is desired, set transmit bit TX9. Can be used as address/data bit.
5. Enable the transmission by setting bit TXEN which will also set bit TXIF.
6. If 9-bit transmission is selected, the ninth bit should be loaded in bit TX9D.
7. Load data to the TXREG register (starts transmission).
Note: TXIF is not cleared immediately upon loading data into the transmit buffer TXREG. The flag bit becomes valid in the second instruction cycle following the load instruction.

FIGURE 20-1: USART TRANSMIT BLOCK DIAGRAM

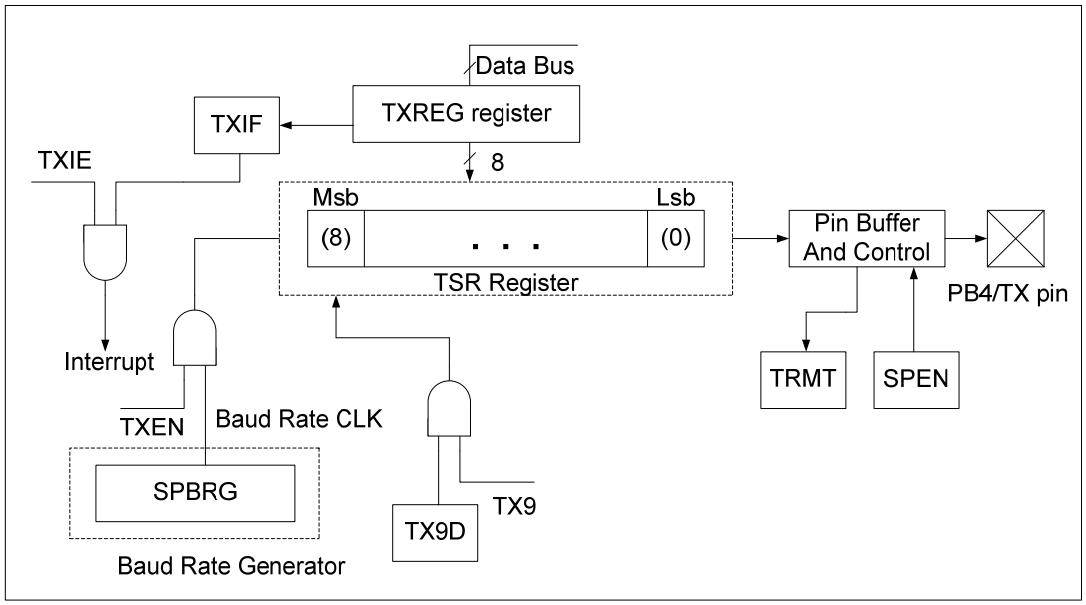


FIGURE 20-2: ASYNCHRONOUS TRANSMISSION

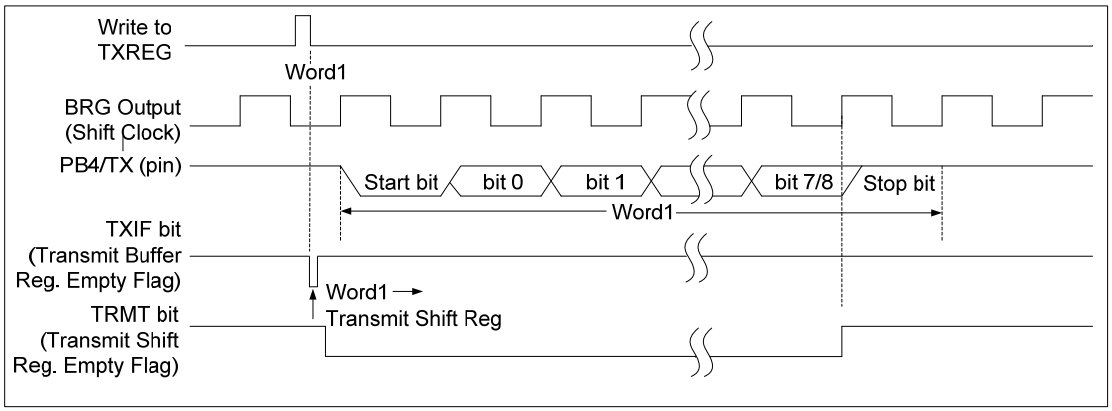
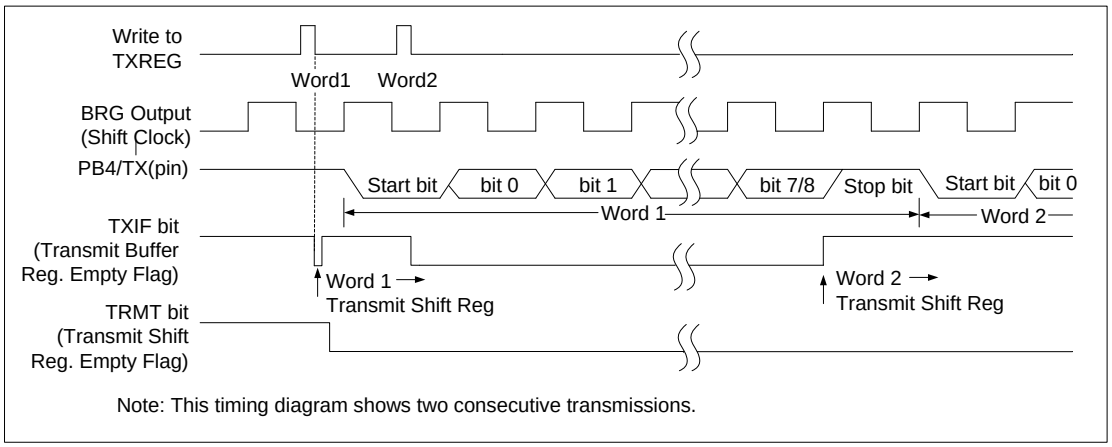


FIGURE 20-3: ASYNCHRONOUS TRANSMISSION (BACK TO BACK)



Note: This timing diagram shows two consecutive transmissions.

TABLE 20-6: REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000u
TXREG	USART Transmit Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

20.2.2 USART ASYNCHRONOUS RECEIVER

The receiver block diagram is shown in Figure 20-4. The data is received on the PB5/RX pin and drives the data recovery block. The data recovery block is actually a high-speed shifter, operating at x16 times the baud rate, whereas the main receive serial shifter operates at the bit rate or at FOSC. This mode would typically be used in RS-232 systems.

Steps to follow when setting up an Asynchronous Reception:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is desired, set bit BRGH.

2. Enable the asynchronous serial port by clearing bit SYNC and setting bit SPEN.
3. If interrupts are desired, set enable bit RCIE.
4. If 9-bit reception is desired, set bit RX9.
5. Enable the reception by setting bit CREN.
6. Flag bit RCIF will be set when reception is complete and an interrupt will be generated if enable bit RCIE was set.
7. Read the RCSTA register to get the ninth bit (if enabled) and determine if any error occurred during reception.
8. Read the 8-bit received data by reading the RCREG register.
9. If any error occurred, clear the error by clearing enable bit CREN.

20.2.3 SETTING UP 9-BIT MODE WITH ADDRESS DETECT

This mode would typically be used in RS-485 systems. Steps to follow when setting up an Asynchronous Reception with Address Detect Enable:

1. Initialize the SPBRG register for the appropriate baud rate. If a high-speed baud rate is required, set the BRGH bit.
2. Enable the asynchronous serial port by clearing the SYNC bit and setting the SPEN bit.
3. If interrupts are required, set the RCEN bit and select the desired priority level with the RCIP bit.
4. Set the RX9 bit to enable 9-bit reception.
5. Set the ADDEN bit to enable address detect.
6. Enable reception by setting the CREN bit.
7. The RCIF bit will be set when reception is complete. The interrupt will be Acknowledged if the RCIE and GIE bits are set.
8. Read the RCSTA register to determine if any error occurred during reception, as well as read bit 9 of data (if applicable).
9. Read RCREG to determine if the device is being addressed.
10. If any error occurred, clear the CREN bit.
11. If the device has been addressed, clear the ADDEN bit to allow all received data into the receive buffer and interrupt the CPU.

FIGURE 20-4: USART RECEIVE BLOCK DIAGRAM

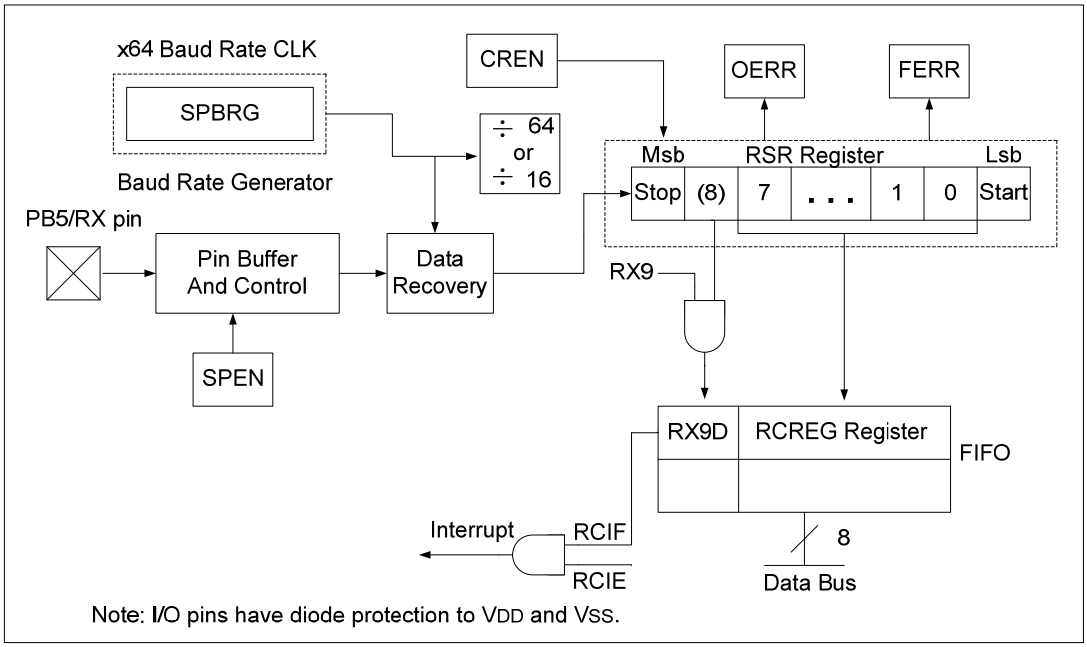


FIGURE 20-5: ASYNCHRONOUS RECEPTION

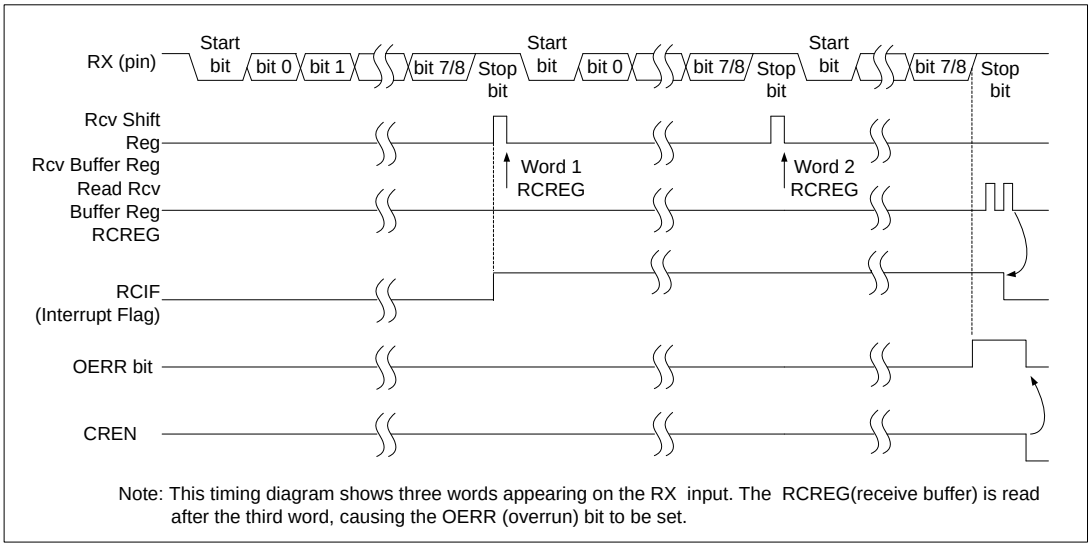


TABLE 20-7: REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RCIP	TXIP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000u
RCREG	USART Receive Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

21. Coulomb Counter - Dedicated Fuel Gauging Sigma-delta ADC

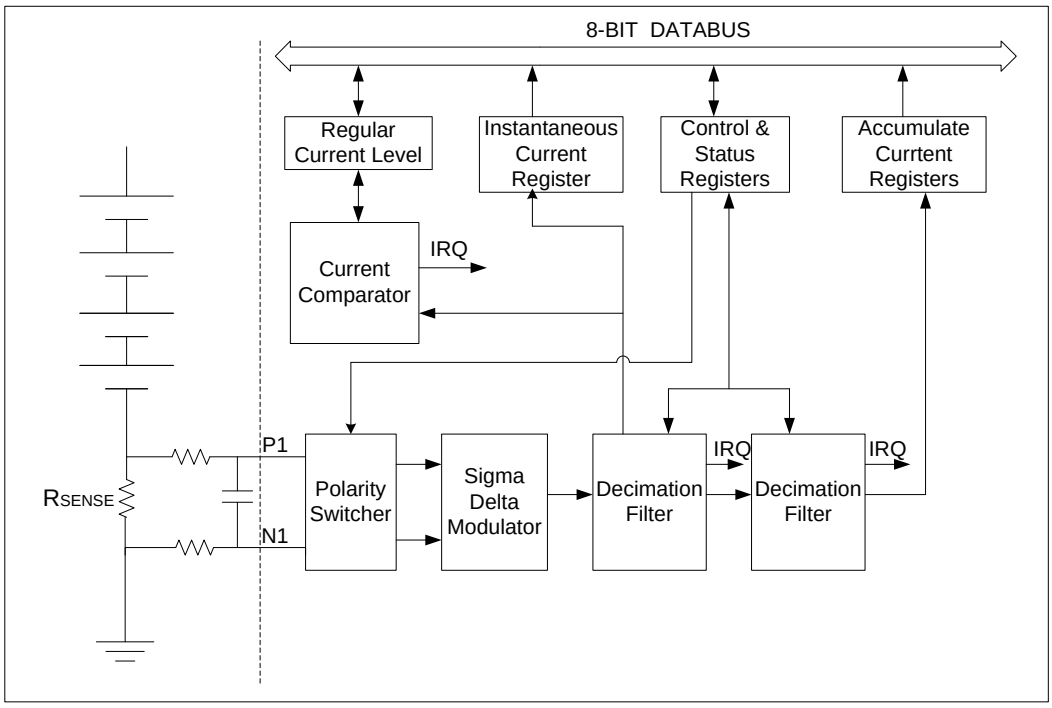
21.1 Features

• Sampled System Coulomb Counter
• Low Power Sigma-Delta ADC Optimized for Coulomb Counting
• Instantaneous Current Output with 3.9 ms Conversion Time
– 13 bit Resolution (including sign bit)
– Interrupt on Instantaneous Current Conversion Complete
• Accumulate Current Output
– Programmable Conversion Time: 128/256/512/1000ms
– 18-bit Resolution (including sign bit)
– Interrupt on Accumulation Current Conversion Complete
• Regular Current Detection Mode
– Programmable Sampling Interval: 256/512/1000/2000 ms
• Programmable Input Voltage Range ± 100/200mV
– Allowing Measurement of ± 20/40A @ 5 mΩ
• Offset canceling by input polarity switching

21.2 Overview

SE1100 features a dedicated Sigma-Delta ADC (CC-ADC) optimized for Coulomb Counting. By sampling the charge or discharge current flowing through an external sense resistor RSENSE, the CC-ADC is used to track the flow of current going into and out of the battery cells.

Figure 21-1. Coulomb Counter Block Diagram



The CC-ADC has a programmable voltage range allowing trade-off to be made between resolution,dynamic range and external sense resistor RSENSE.
In normal conversion mode two different output values are provided, Instantaneous Current and Accumulate Current. The Instantaneous Current Output has a short conversion time at the cost of lower resolution. The Accumulate Current Output provides a highly accurate current measurement for Coulomb Counting.

The CC-ADC also provides a special Regular Current detection mode. This allows ultra-low power operation in Power-save mode when small charge or discharge currents are flowing.
For offset cancellation the polarity of the input signal could be switched during run time. Using this feature, the internal CC-ADC offset could be removed.

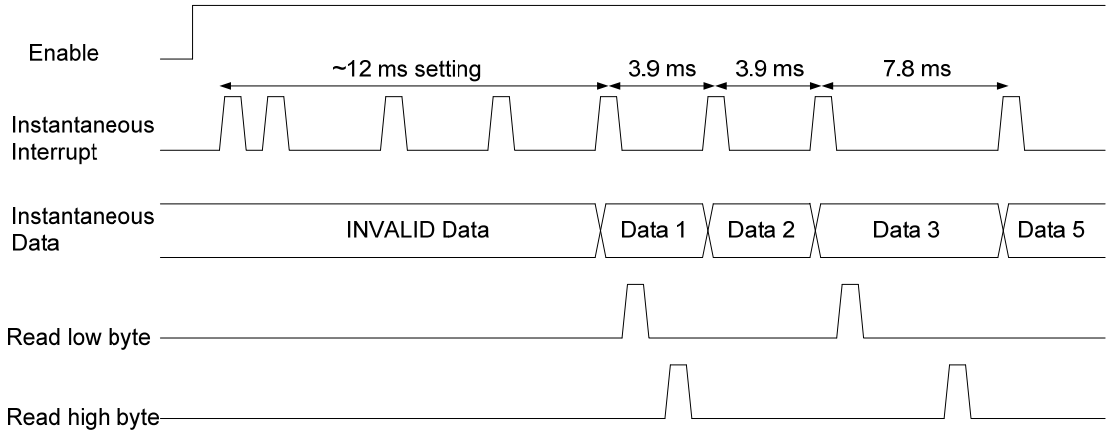
21.3 Normal Operation

When enabled the CC-ADC continuously measures the current over the external sense resistor RSENSE. Running in normal conversion mode, two data conversion outputs are provided.
• Instantaneous Conversion Result
• Accumulation Conversion Result

The Instantaneous Current conversion time is fixed to 3.9 ms (typical value) allowing the output value to closely follow the input. After each Instantaneous Current conversion an interrupt is generated if the interrupt is enabled. Data from conversion will be updated in the Instantaneous Current registers CADICL and CADICH simultaneously as the interrupt is given. To avoid losing conversion data, both the low and high byte must be read within a 3.9 ms timing window after the corresponding interrupt is given. When the low byte register is read, updating of the Instantaneous

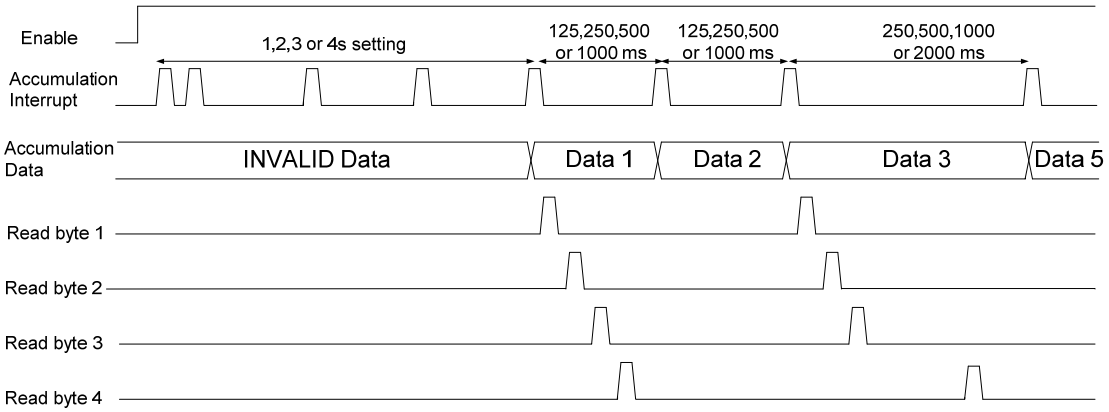
Current registers and interrupts will be stopped until the high byte is read. Figure 21-2 shows an Instantaneous Current conversion diagram, where DATA4 will be lost because DATA3 reading is not completed within the limited period.

Figure 21-2. Instantaneous Current Conversions



The Accumulate Current output is a high-resolution, high accuracy output with programmable conversion time selected by the CADAS bits in CADCSRA. The converted value is an accurate measurement of the average current flow during one conversion period. The CC-ADC generates an interrupt each time a new Accumulate Current conversion has finished if the interrupt is enabled. Data from conversion will be updated in the Accumulation Current registers - CADAC0,CADAC1, CADAC2 and CADAC3 simultaneously as the interrupt is given. To avoid losing conversion data, all bytes must be read within the selected conversion period. When the lower byte registers are read, updating of the Accumulation Current registers and interrupts will be stopped until the highest byte is read. Figure 21-3 shows an Accumulation Current conversion example, where DATA4 will be lost because DATA3 reading is not completed within the limited period.

Figure 21-3. Accumulation Current Conversions

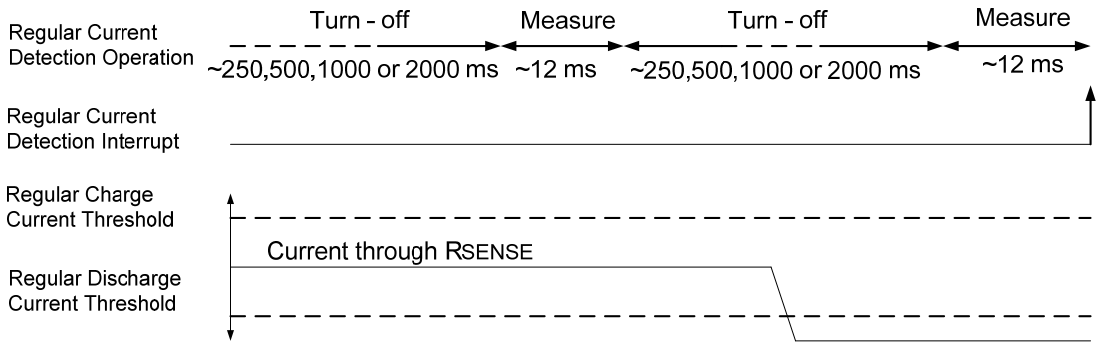


21.4 Regular Current Detection Operation

By setting the CADSE bit in CADCSRA the CC-ADC will enter a special Regular Current Detection Sampling Mode. In this mode the CC-ADC will do one Instantaneous Current

Conversion on regular sampling intervals while updating of the Accumulation Current Register is automatically disabled. The sampling interval is controlled by writing to the CADSI bits in CADCSRA. Each time a conversion is completed the result is compared with Regular Charge/Discharge Threshold levels specified in the CADRCC/CADRDC registers. If interrupt is enabled and the voltage is above/below the specified limit a Regular Current Detection Interrupt will be issued. Figure 21-4 illustrates the Regular Current Detection Mode.

Figure 21-4. Regular Current Detection Mode (CADSE=1)



21.5 Offset Canceling by Polarity Switching

The CC-ADC offers Polarity Switching for internal offset canceling. By switching the polarity of the sampled input signal at selected time intervals, the internal voltage offset of the CC-ADC will cancel at the output. This feature prevents the CC-ADC from accumulating an offset error over time.

21.6 Configuration and Usage

While the CC-ADC is converting, the CPU can enter sleep mode and wait for an interrupt. After adding the conversion data for the Coulomb Counting, the CPU can go back to sleep again. This reduces the CPU workload, and allows more time spent in low power modes, reducing power consumption. To use the CC-ADC the bandgap voltage reference must be enabled separately.The CC-ADC will not consume power when CADEN is cleared. It is therefore recommended to switch off the CC-ADC whenever the Coulomb Counter or Regular Current Detection functions are not used. The CC-ADC is automatically disabled in Power-off mode.
After the CC-ADC is enabled by setting the CADEN bit, the first four conversions do not contain useful data and should be ignored. This also applies after clearing the CADSE bit, or after changing the CADPOL or CADVSE bits. The conversion times and sampling intervals are controlled by the Ultra Low Power RC Oscillator, and will depend on its actual frequency. To obtain accurate coulomb counting results, the actual conversion time should be calculated.

21.7 Register Description

21.7.1 CADCSRA – CC-ADC Control and Status Register A

Bit	7	6	5	4	3	2	1	0	
(0XE6D)	CADEN	CADPOL	CADUB	CADAS1	CADAS0	CADSI1	CADSI0	CADSE	CADCSRA
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 – CADEN: CC-ADC Enable

When the CADEN bit is cleared (zero), the CC-ADC is disabled, and any ongoing conversions will be terminated. When the CADEN bit is set (one), the CC-ADC will continuously measure the voltage drop over the external sense resistor RSENSE. In Power-off, the CC-ADC is always disabled. Note that the bandgap voltage reference must be enabled separately.

• Bit 6 – CADPOL: CC-ADC Polarity

The CADPOL bit is used to change input sampling polarity in the Sigma Delta Modulator. Writing this bit to one, the polaritiy will be negative. When the bit is zero, the polarity will be positive.

• Bit 5 - CADUB: CC-ADC Update Busy

The CC-ADC operates in a different clock domain than the CPU. Whenever a new value is written to CADCSRA, CADCSRC, CADRCC or CADRDC, this value must be synchronized to the CC-ADC clock domain. Subsequent writes to these registers will be blocked during this synchronization. Synchronization of one of the registers will block updating of all the others. The CADUB bit will be read as one while any of these registers is being synchronized, and will be read as zero when neither register is being synchronized.

• Bits 4:3 – CADAS1:0: CC-ADC Accumulate Current Select

The CADAS bits select the conversion time for the Accumulate Current output as shown in the Table 21-1.

Table 21-1. CC-ADC Accumulate Current Conversion Time

CADAS1:0	CC-ADC Accumulate Current Conversion Time(1)	Number of CC-ADC clock Cycles
00	128 ms	4096
01	256 ms	8192
10	512 ms	16384
11	1 s	32768

• Bits 2:1 – CADSI1:0: CC-ADC Current Sampling Interval

The CADSI bits determine the current sampling interval for the Regular Current detection as shown in the Table 21-2.

Table 21-2. CC-ADC Regular Current Sampling Interval

CADSI1:0	CC-ADC Regular Current Sampling Interval[1][2]	Number of CC-ADC clock Cycles
00	256 ms (+ sampling time)	8192 (+ sampling time)
01	512 ms (+ sampling time)	16384 (+ sampling time)
10	1 s (+ sampling time)	32768 (+ sampling time)
11	2 s (+ sampling time)	65536 (+ sampling time)

• Bit 0 – CADSE: CC-ADC Sampling Enable

When the CADSE bit is written to one, the ongoing CC-ADC conversion is aborted and the CCADC enters Regular Current detection mode.

21.7.2 CADCSRB – CC-ADC Control and Status Register B

Bit	7	6	5	4	3	2	1	0	
(0XE6E)	–	CADACIE	CADRCIE	CADICIE	–	CADACIF	CADRCIF	CADICIF	CADCSRB
Read/Write	R	R/W	R	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7, 3 – Res: Reserved

These bits are reserved bits in the TLS5103 and will always read as zero.

• Bit 6 – CADACIE: CC-ADC Accumulate Current Interrupt Enable

When the CADACIE bit is set (one), and the I-bit in the Status Register is set (one), the CC-ADC Accumulate Current Interrupt is enabled.

• Bit 5 – CADRCIE: CC-ADC Regular Current Interrupt Enable

When the CADRCIE bit is set (one), and the I-bit in the Status Register is set (one), the CC-ADC Regular Current Interrupt is enabled.

• Bit 4 – CADICIE: CC-ADC Instantaneous Current Interrupt Enable

When the CADICIE bit is set (one), and the I-bit in the Status Register is set (one), the CC-ADC Instantaneous Current Interrupt is enabled.

• Bit 2 – CADACIF: CC-ADC Accumulate Current Interrupt Flag

The CADACIF bit is set (one) after the Accumulate Current conversion has completed. The CCADC Accumulate Current Interrupt is executed if the CADACIE bit is set (one). CADACIF is cleared by by writing a logic zero to the flag.

• Bit 1 – CADRCIF: CC-ADC Regular Current Interrupt Flag

The CADRCIF bit is set (one) when the absolute value of the result of the last CC-ADC conversion is greater than, or equal to, the compare values set by the CC-ADC Regular Charge/Discharge Current Level Registers. A positive value is compared to the Regular Charge Current Level, and a negative value is compared to the Regular Discharge Current Level. The

CC-ADC Regular Current Interrupt is executed if the CADRCIE bit are set (one). CADRCIF is cleared by writing a logic zero to the flag.

• Bit 0 – CADICIF: CC-ADC Instantaneous Current Interrupt Flag

The CADICIF bit is set (one) when a CC-ADC Instantaneous Current conversion is completed. The CC-ADC Instantaneous Current Interrupt is executed if the CADICIE bit are set (one). CADICIF is cleared by writing a logic zero to the flag.

21.7.3 CADCSRC – CC-ADC Control and Status Register C

Bit	7	6	5	4	3	2	1	0	
(0XE6F)	–	–	–	–	–	–	–	CADVSE	CADCSRC
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7:1 – Res: Reserved

These bits are reserved bits and will always read as zero.

• Bit 0 – CADVSE: CC-ADC Voltage Scaling Enable

Setting this bit enables the internal Voltage Scaling. When enabling the internal Voltage Scaling the internal CC-ADC reference will be divided by 2, affecting the Input Voltage Range and the resulting step-size. Table 21-3 shows the Input Voltage Range and the conversion value stepsize for the CADVSE settings.

Table 21-3. Input Voltage Range and the conversion value step-size for the CADVSE settings.

CADVSE	Voltage Range	Step-size CADAC	Step-size CADIC
0	± 200 mV	1.67 µV	53.7 µV
1	± 100 mV	0.84 µV	26.9 µV

21.7.4 CADICH and CADICL – CC-ADC Instantaneous Current

Bit	15	14	13	12	11	10	9	8	
(0XE6C)	CADIC[15:8]								CADICH
(0XE6B)	CADIC[7:0]								CADICL
Bit	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When a CC-ADC Instantaneous Current conversion is complete, the result is found in these two registers. CADIC15:0 represents the converted result in 2's complement format. Bits 12:0 are the 13-bit ADC result (including sign), while bit 15:13 are the sign extension bits.

When CADICL is read, the CC-ADC Instantaneous Current register is not updated until CADCH

is read. Reading the registers in the sequence CADICL, CADICH will ensure that consistent values are read. When a conversion is completed, both registers must be read before the next conversion is completed, otherwise data will be lost.

21.7.5 CADAC3, CADAC2, CADAC1 and CADAC0 – CC-ADC Accumulate Current

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	
(0XE6A)	CADAC[31:24]								CADAC3
(0XE69)	CADAC[23:16]								CADAC2
(0XE68)	CADAC[15:8]								CADAC1
(0XE67)	CADAC[7:0]								CADAC0
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The CADAC3, CADAC2, CADAC1 and CADAC0 Registers contain the Accumulate Current measurements in 2's complement format. Bits 17:0 are the 18-bit ADC result (including sign), while bit 31:18 are the sign extension bits. When CADAC0 is read, the CC-ADC Accumulate Current register is not updated until CADAC3 is read. Reading the registers in the sequence CADAC0, CADAC1, CADAC2, CADAC3 will ensure that consistent values are read. When a conversion is completed, all four registers must be read before the next conversion is completed, otherwise data will be lost.

21.7.6 CADRCC – CC-ADC Regular Charge Current

Bit	7	6	5	4	3	2	1	0	
(0XE70)	CADRCC[7:0]								CADRCC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The CC-ADC Regular Charge Current Register determines the threshold level for the Regular Charge Current detection. When the result of a CC-ADC Instantaneous Current conversion is positive with a value greater than, or equal to, the Regular Charge Current level, the CC-ADC Regular Current Interrupt Flag is set. The value in this register defines the eight least significant bits of the Regular Charge Current level in 2's complement format, where the most significant bits of the Regular Charge Current level are always zero. The programmable range for the Regular Charge Current level is given in the Table 21-4.

Table 21-4. Programmable Range for the Regular Charge Current Level(1)

		Minimum	Maximum	Step Size
Voltage (µV)		0	13696/6848	53.7/26.9
Current (mA)	R_{SENSE} = 1 mΩ	0	13696/6848	53.7/26.9
	R_{SENSE} = 5 mΩ	0	2740/1370	10.7/5.4
	R_{SENSE} = 10 mΩ	0	1370/685	5.3/2.7

Note: 1. Values in the table are shown with the CADVSE set to both 0 and 1.
The CC-ADC Regular Charge Current Register does not affect the setting of the CC-ADC Conversion Complete Interrupt Flag.

21.7.7 CADRDC – CC-ADC Regular Discharge Current

Bit	7	6	5	4	3	2	1	0	
(0XE71)	CADRDC[7:0]								CADRDC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The CC-ADC Regular Discharge Current Register determines the threshold level for the Regular Discharge Current detection. When the result of a CC-ADC Instantaneous Current conversion is negative with an absolute value greater than, or equal to, the Regular Discharge Current level, the CC-ADC Regular Current Interrupt Flag is set.

The value in this register defines the eight least significant bits of the Regular Discharge Current level in 2's complement format, where the most significant bits of the Regular Discharge Current level are always one. The programmable range for the Regular Discharge Current level is given in the Table 21-5.

Table 21-5. Programmable Range for the Regular Discharge Current Level

		Minimum	Maximum	Step Size
Voltage (µV)		0	13696/6848	53.7/26.9
Current (mA)	R_{SENSE} = 1 mΩ	0	13696/6848	53.7/26.9
	R_{SENSE} = 5 mΩ	0	2740/1370	10.7/5.4
	R_{SENSE} = 10 mΩ	0	1370/685	5.3/2.7

Note: 1. Values in the table are shown with the CADVSE set to both 0 and 1.
The CC-ADC Regular Discharge Current Register does not affect the setting of the CC-ADC Conversion Complete Interrupt Flag.

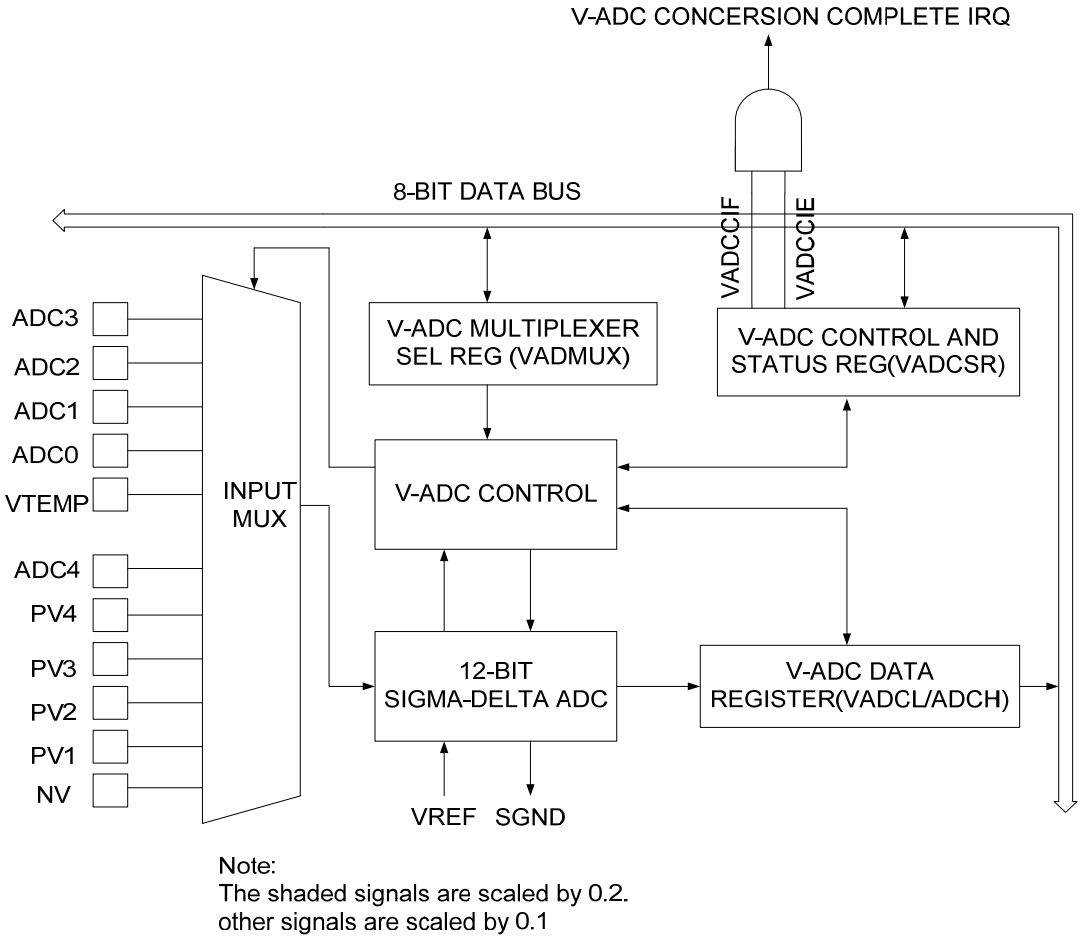
23. Voltage ADC – 10-channel General Purpose 12-bit Sigma-Delta ADC

23.1 Features

- **12-bit Resolution**
- **519µs Conversion Time @ 1 MHz clkVADC**
- **Four Differential Input Channels for Cell Voltage Measurements**
- **Three Single Ended Input Channels**
- **0.2x Pre-scaling of Cell Voltages**
- **Interrupt on V-ADC Conversion Complete**

The SE1100 features a 12-bit Sigma-Delta ADC. Automatic offset cancellation technique reduces the input offset voltage to less than 0.5 mV. The Voltage ADC (V-ADC) is connected to ten different sources through the Input Multiplexer. There are four differential channels for Cell Voltage measurements. These channels are scaled 0.2x to comply with the Full Scale range of the V-ADC. In addition there are six single ended channels referenced to SGND. One channel is for measuring the internal temperature sensor VPTAT and five channels for measuring the ADC input pins at Port A. ADC3:0 are not scaled, meaning that full-scale reading corresponds to 1.1 V. ADC4 is scaled by 0.2x, meaning that fullscale reading corresponds to 5.5 V. The ADC4 input can be used to measure the voltage at the PA4 pin when this pin is used to supply an external thermistor. To obtain a total absolute accuracy better than ± 0.25% for the cell voltage measurements, calibration registers for the individual cell voltage gain in the analog front-end is provided. A factory calibration value is stored in the signature row. The V-ADC conversion of a cell voltage must be scaled with the corresponding calibration value by software to correct for gain error in the analog front-end.

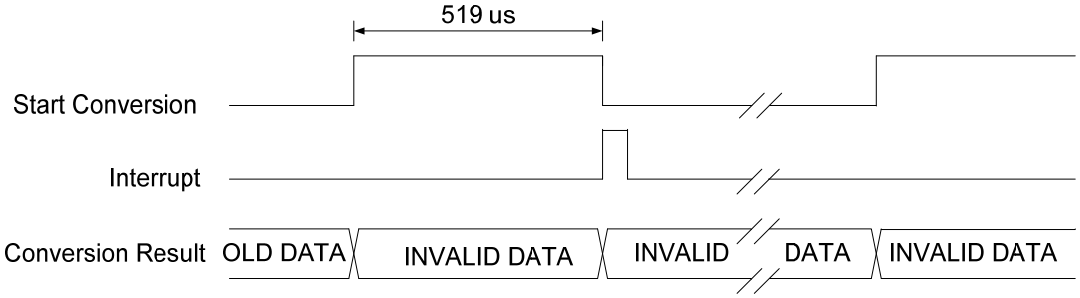
Figure 23-1. Voltage ADC Block Schematic



23.2 Operation

To enable V-ADC conversions, the V-ADC Enable bit, VADEN, in V-ADC Control and Status Register – VADCSR must be set. If this bit is cleared, the V-ADC will be switched off, and any ongoing conversions will be terminated. The V-ADC is automatically halted in Power-save, Power-down and Power-off mode. Note that the bandgap voltage reference must be enabled and disabled separately, see “BGCCR – Bandgap Calibration C Register”.

Figure 23-2. Voltage ADC Conversion Diagram



To perform a V-ADC conversion, the analog input channel must first be selected by writing to the

VADMUX bits in VADMUX. When a logical one is written to the V-ADC Start Conversion bit VADSC, a conversion of the selected channel will start. The VADSC bit stays high as long as the conversion is in progress and will be cleared by hardware when the conversion is completed. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change. When a conversion is finished the VADC Conversion Complete Interrupt Flag – VADCCIF is set. One 12-bit conversion takes 519 µs to complete from the start bit is set to the interrupt flag is set. To ensure that correct data is read, both high and low byte data registers should be read before starting a new conversion.

23.3 Register Description

23.3.1 VADMUX – V-ADC Multiplexer Selection Register

Bit	7	6	5	4	3	2	1	0	
(0XE61)	–	–	–	–	VADMUX3	VADMUX2	VADMUX1	VADMUX0	VADMUX
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7:4 – RES: Reserved Bits

These bits are reserved bits in the SE1100 and will always read as zero.

• Bit 3:0 – VADMUX3:0: V-ADC Channel Selection Bits

The VADMUX bits determine the V-ADC channel selection. See Table 23-1.

Table 23-1. VADMUX channel selection

VADMUX3:0	Channel Selected	Scale
0001	CELL 1	0.2
0010	CELL 2	0.2
0011	CELL 3	0.2
0100	CELL 4	0.2
0101	ADC4	0.2
0110	VTEMP	1.0
0111	ADC0	1.0
1000	ADC1	1.0
1001	ADC2	1.0
1010	ADC3	1.0

23.3.2 VADCSR – V-ADC Control and Status Register

Bit	7	6	5	4	3	2	1	0	
(0XE60)	VADUB	–	–	–	VADEN	VADSC	VADCCIF	VADCCIE	VADCSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 - VADUB: VADC Update Busy

The VADC operates in a different clock domain than the CPU. Whenever a new value is written to DIDR0,VADCSR or VADMUX, this value must be synchronized to the VADC clock domain. Subsequent writes to these registers will be blocked during this synchronization. Synchronization of one of the registers will block updating of all the others. The VADUB bit will be read as one while any of these registers is being synchronized, and will be read as zero when neither register is being synchronized.

• Bit 6:4 – RES: Reserved Bits

These bits are reserved bits in the SE1100 and will always read as zero.

• Bit 3 – VADEN: V-ADC Enable

Writing this bit to one enables V-ADC conversion. By writing it to zero, the V-ADC is turned off. Turning the V-ADC off while a conversion is in progress will terminate this conversion. Note that the bandgap voltage reference must be enabled separately, see “BGCCR – Bandgap Calibration C Register”.

• Bit 2 – VADSC: Voltage ADC Start Conversion

Write this bit to one to start a new conversion of the selected channel.

VADSC will read as one as long as the conversion is not finished. When the conversion is complete, it returns to zero. Writing zero to this bit has no effect. VADSC will automatically be cleared when the VADEN bit is written to zero.

• Bit 1 – VADCCIF: V-ADC Conversion Complete Interrupt Flag

This bit is set when a V-ADC conversion completes and the data registers are updated. The VADC Conversion Complete Interrupt is executed if the VADCCIE bit are set. VADCCIF is cleared by writing a logical zero to the flag.

• Bit 0 – VADCCIE: V-ADC Conversion Complete Interrupt Enable

When this bit is written to one , the V-ADC Conversion Complete Interrupt is activated.

23.3.3 VADCL and VADCH – The V-ADC Data Register

Bit	15	14	13	12	11	10	9	8	
(0XE5F)	–	–	–	–	VADC[11:8]				VADCH
(0XE5E)	VADC[7:0]								VADCL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

When a V-ADC conversion is complete, the result is found in these two registers. To ensure that correct data is read, the data registers must be read before starting a new conversion.

• VADC11:0: V-ADC Conversion Result

These bits represent the result from the conversion.

To obtain the best absolute accuracy for the cell voltage measurements, gain and offset compensation is required. Factory calibration values are stored in the device signature row, refer to section ”Reading the Signature Row from Software” for details. The cell voltage in mV is given by:

$$\text{Cell}_n\text{voltage}(mV) = \frac{\text{cell}_n\ \text{result} \cdot \text{cell}_n\text{gain calibration word}}{\text{TBD}} - \text{cell}_n\text{offset calibration word}$$

When performing a Vtemp conversion, the result must be adjusted by the factory calibration value stored in the signature row, refer to section ”Reading the Signature Row from Software” for details. The absolute temperature in Kelvin is given by:

$$\text{T(K)} = \frac{V_{temp}\text{result} \cdot \text{VPTAT calibration word}}{\text{TBD}}$$

23.3.4 DIDR0 – Digital Input Disable Register 0

Bit	7	6	5	4	3	2	1	0	
(0XE5D)	–	–	–	–	VADC3D	VADC2D	VADC1D	VADC0D	DIDR0
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7:4 – Res: Reserved Bits

These bits are reserved for future use. To ensure compatibility with future devices, these bits

must be written to zero when DIDR0 is written.

• Bit 3:0 – VADC3D:VADC0D: V-ADC3:0 Digital Input Disable

When this bit is written logic one, the digital input buffer on the corresponding V-ADC pin is disabled. The corresponding PIN Register bit will always read as zero when this bit is set. When an analog signal is applied to the VADC3:0 pin and the digital input from this pin is not needed, this bit should be written logic one to reduce power consumption in the digital input buffer.

24. Voltage Reference and Temperature Sensor

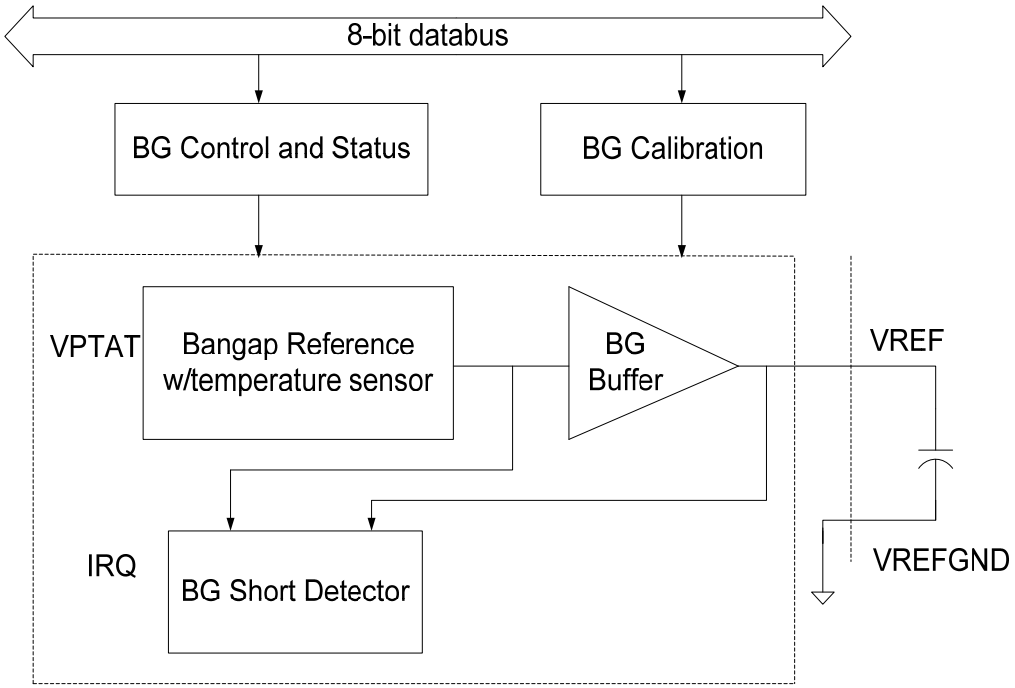
24.1 Features

- **Accurate Voltage Reference of 1.100V**
- **Voltage Reference Calibration Interface**
- **Internal Temperature Sensor**
- **External Decoupling for Optimum Noise Performance**
- **Short Circuit Detection on the External Decoupling pin**
- **Low Power Consumption**

24.2 Overview

SE1100 features a highly accurate low power On-chip Bandgap Reference Voltage, VREF of 1.100V. This reference voltage is used as reference for the On-chip Voltage Regulator, the Brown-out Detector, the internal Cell Balancing, the Battery Protection, the VADC and the CC-ADC. Figure 24-1 shows an overview of the On-chip voltage reference.

Figure 24-1. Bandgap Reference Voltage



In addition to the Bandgap Reference Voltage, the Voltage Reference includes a Calibration Unit that enables run time calibration of the Reference Voltage, an On-chip temperature sensor for monitoring the die temperature, and a Bandgap Short Detector to detect short circuit conditions on

the external VREF pin for highest safety.

24.3 Operation

When the device is in power-off state, the Voltage Reference will be switched off. After a Poweron reset condition the Voltage Reference will automatically be enabled. By default the Bandgap Buffer will be enabled as the buffered reference voltage is used as reference for the Battery Protection, the internal Cell Balancing, the V-ADC and the CC-ADC. If any of these modules are enabled the Bandgap Buffer must be enabled, otherwise it is recommended to disable the buffer by writing to the BGD bit in BGCSR to save power. Note that the Bandgap Buffer needs settling time before the voltage is stable. For details on settling time, see ”Bandgap Buffer Settling Time”.

To ensure that the battery protection has safe operation condition, the Voltage Reference includes a Short-circuit Detector. The Bandgap Short Detector continuously monitors the internal 1.100V reference voltage against the VREF pin voltage to detect potentially external short conditions to VCC or GND. If an external short condition is detected, the Bandgap Short Circuit Detection is capable of interrupting and waking up the CPU from any sleep mode. If a Bandgap Short-circuit condition occurs software should immediately disable the C-FET and D-FET. If no external protection is provided to detect such a condition it is recommended to always enable this feature, by setting the BGSCDE bit in the Bandgap Control and Status Register.

The Temperature Sensor generates a voltage Proportional-To-Absolute-Temperature, VPTAT. This voltage is connected to the multiplexer at the V-ADC input and it can be used for runtime compensation of temperature drift in both the voltage reference and the On-chip Oscillator. To get the absolute temperature in degrees Kelvin, the measured VPTAT voltage must be scaled with the VPTAT factory calibration value stored in the signature row. See ”Reading the Signature Row from Software” for details.

24.4 Bandgap Calibration

To guarantee ultra low temperature drift the Voltage Reference includes two calibration registers that could be changed run-time by software. Changing values to the BGCCR IO register will change the nominal value of the Bandgap Reference Voltage, while changing values to the BGCRR IO register trims the temperature gradient of the bandgap reference. When the calibration registers are changed it will affect both the Voltage Regulator output and BOD-level. The BOD will react quickly to new detection levels, while the regulator will adjust the voltage more slowly, depending on the size of the external decoupling capacitor. To avoid that a BOD-reset is issued when calibration is done, it is recommended to change the values of the BGCC and BGCR bits stepwise, with a step size of 1, and with a hold-off time between each step. See ”Electrical Characteristics” for details on hold-off time. Changing VREF will influence the conversion results for the V-ADC and CC-ADC. It is therefore not recommended to do V-ADC and CC-ADC conversions while calibrating the bandgap. To guarantee ultra low temperature drift it is

recommended to perform factory calibration using a two-step calibration algorithm. By default, Atmel factory calibration is performed at hot temperature, and the result is stored in the signature row. See ”Reading the Signature Row from Software” for details. The customer can easily implement the second calibration step in their test flow.

24.5 Bandgap Buffer Settling Time

After the Voltage Reference have been enabled it needs a settling time before the voltage is stable. The settling time depends on the size of the external decoupling capacitor. With 1uF external capacitor a minimum settling time of 2ms should be used. Until settling is done it is not recommended to enable the NFET driver (OC/OD or enter DUVR operation), Battery Protection, V-ADC or CC-ADC. Settling time is needed when the Buffer is enabled by software, or after a reset condition where the buffer is automatically enabled.

24.6 Register Description

24.6.1 BGCCR - Bandgap Calibration C Register

Bit	7	6	5	4	3	2	1	0	
(0XE63)	-	-	BGCC5	BGCC4	BGCC3	BGCC2	BGCC1	BGCC0	BGCCR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7:6 - Res: Reserved Bits

These bits are reserved bits and will always read as zero.

• Bit 5:0 - BGCC5:0: BG Calibration of PTAT Current

These bits are used for trimming of the nominal value of the bandgap reference voltage. These bits are binary coded. Minimum VREF: 000000, maximum VREF: 111111. Step size is approximately 2 mV.

24.6.2 BGCRR - Bandgap Calibration R Register

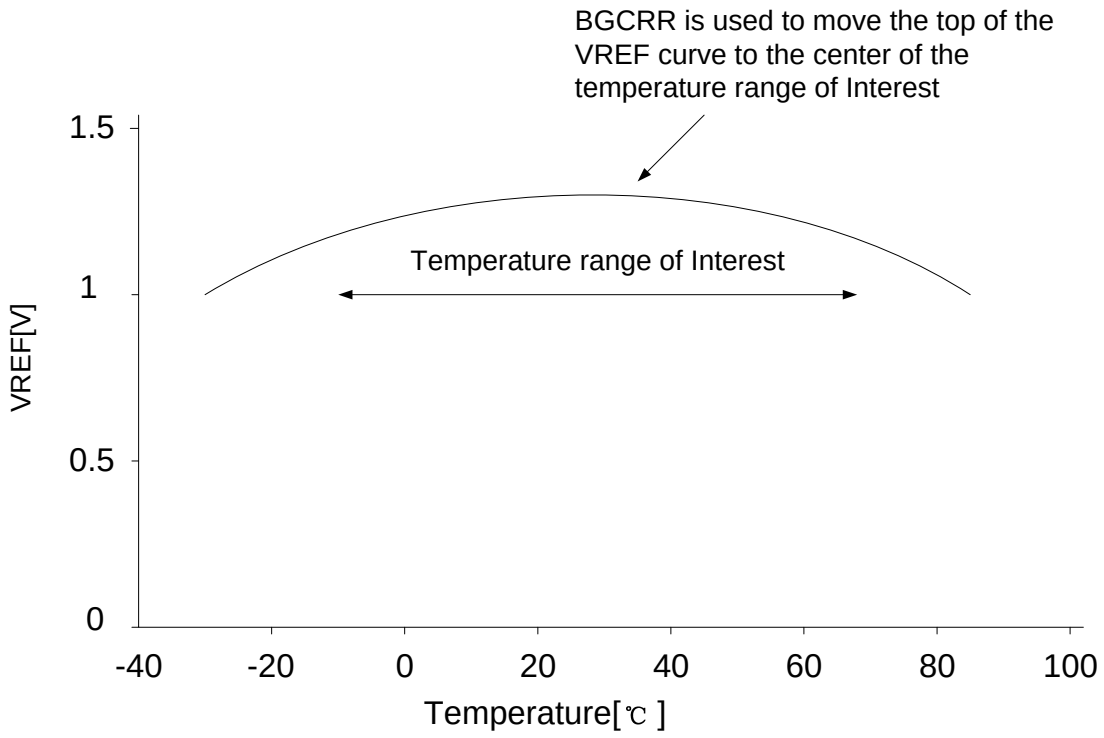
Bit	7	6	5	4	3	2	1	0	
(0XE64)	BGCR7	BGCR6	BGCR5	BGCR4	BGCR3	BGCR2	BGCR1	BGCR0	BGCRR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	1	1	1	1	

• Bit 7:0 - BGCR7:0: BG Calibration of Resistor ladder

These bits are used for temperature gradient adjustment of the bandgap reference. Figure 24-2 illustrates VREF as a function of temperature. VREF has a positive temperature coefficient at low

temperatures and negative temperature coefficient at high temperatures. Depending on the process variations, the top of the VREF curve may be located at higher or lower temperatures. To minimize the temperature drift in the temperature range of interest, BGCRR is used to adjust the top of the curve towards the centre of the temperature range of interest. The BGCRR bits are thermometer coded, resulting in 9 possible settings: 00000000, 00000001, 00000011, 00000111, … , 11111111. The value 00000000 shifts the top of the VREF curve to the highest possible temperature, and the value 11111111 shifts the top of the VREF curve to the lowest possible temperature.

Figure 24-2. Illustration of VREF as a function of temperature



24.6.3 BGCSR - Bandgap control and Status Register

Bit	7	6	5	4	3	2	1	0	
(0XE65)	-	-	BGD	BGSCDE	-	-	BGSCDIF	BGSCDIE	BGCSR
Read/Write	R	R	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7:6 - Res: Reserved Bits

These bits are reserved bits and will always read as zero.

• Bit 5 -BGD: Bandgap Disable

Setting this bit to one will disable the bandgap voltage reference. This bit must be cleared (zero) before enabling Cell Balancing, CC-ADC, V-ADC or Battery Protection, and must remain unset

(zero) while either of these modules are enabled. Note that after clearing this bit, a settling time is required before the voltage is stable, see ”Bandgap Buffer Settling Time”.

• Bit 4 - BGSCDE: Bandgap Short Circuit Detection Enabled

Setting this bit to one will enable the bandgap Short Circuit Detector. This bit should be cleared if the BGD bit in the BGCSR is set to one to avoid false setting of the BGSCDIF bit.

• Bits 3:2 - Res: Reserved Bits

These bits are reserved bits and will always read as zero.

• Bit 1 - BGSCDIF: Bandgap Short Circuit Detection Interrupt Flag

The bit is set when the Bandgap Short Circuit Detector is enabled and buffered bandgap reference is different from the unbuffered Bandgap reference. The BGSCDIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, BGSCDIF is cleared by writing a logic one to its bit position.

• Bit 0 - BGSCDIE: Bandgap Short Circuit Detection Interrupt Enable

When this bit is set, the Bandgap Short Circuit Detection Interrupt is enabled. The corresponding interrupt is executed if a short-circuit is detected on the External Decoupling Pin for the Voltage Reference.

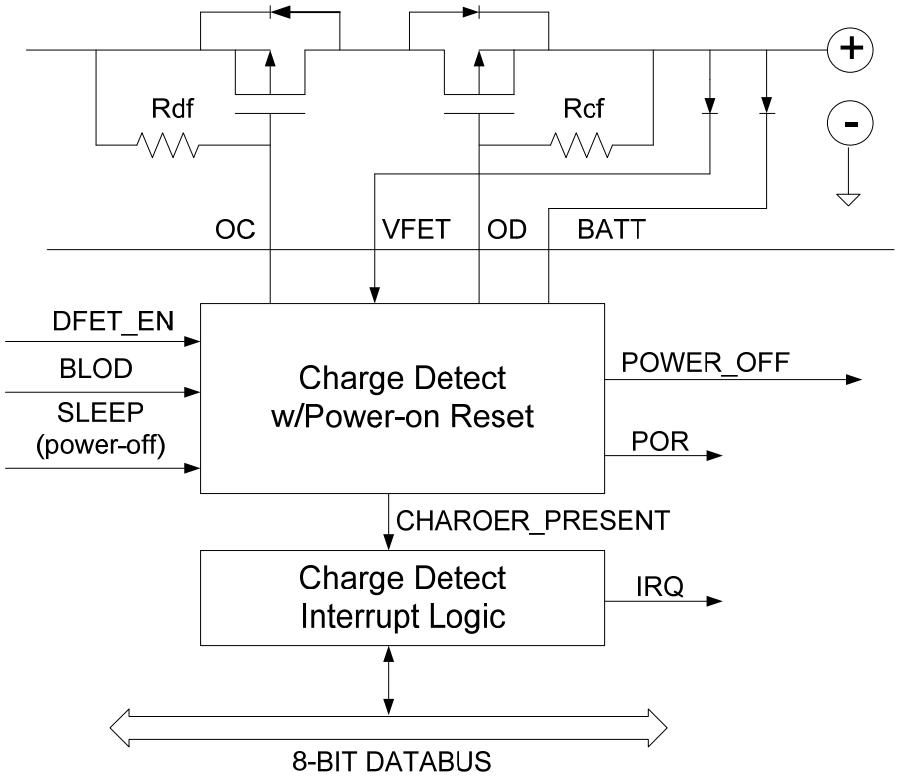
25. Charger Detect

25.1 Features

• Operates directly from VFET supply
• Detects when a charger is connected or disconnected by monitoring the BATT pin
• Controls the operation state of the device by automatically enable/disable the internal Voltage Regulator
• Automatically disabled when Discharge FET is ON.
• Interrupt wake-up from all sleep modes

25.2 Overview

The Charger Detect module has two main functions:
• Control the device operating state (Power-off or normal operation)
• Detect when a charger is connected/disconnected
Figure 25-1 shows a block diagram of the Charger Detect module.
Figure 25-1. Charger Detect block diagram.



25.3 Operation

The Charger Detect module is supplied directly from the VFET pin. When operating, the Charger Detect will monitor the voltage of the BATT pin and detect whether a charger is present or not. When the voltage on the BATT pin is above the Charger Detect Threshold level VPOT (”Electrical Characteristics”) the CHARGER_PRESENT signal will be high. This signal is edge detected to generate interrupt when a charger is connected or disconnected.

The Charger Detect module will operate as long as the Discharge FET is disabled and is able to detect a charger in all sleep modes including Power-off.

25.3.1 Device operating state

The Charger Detect module controls the operating state of the SE1100 by enabling/disabling the Voltage Regulator, which supplies the rest of the internal logic on the chip. If the chip is in Power-off state the Charger Detect will keep the Voltage Regulator disabled allowing only the Charger Detect module itself to operate. To be able to start from a Power-off state a charge voltage above VPOT has to be applied at the BATT pin. When a charger is connected the Charger Detect module will automatically start the Voltage Regulator. When the VREG voltage rises, a Power-on Reset (POR) is given and the chip enters normal operating mode after a reset delay corresponding to TTOUT. For details on POR, see ”Power-on Reset and Charger Connect”.

When the SE1100 is running in normal operation mode software or hardware could take the chip into Power-off state. For details on entering power-off by software, see ”Power-off Mode”. To protect the device against software malfunctions the SE1100 has a Black-out Detector (BLOD). When a Black-out condition occurs the Charger Detect will take the chip automatically into Power-off. For details on Black-out Detection, see ”Voltage Regulator” .

Note that after a software power-off the BATT pin voltage has to fall below the Charger Detect threshold limit, VPOT, before the chip is able to re-enable the Voltage Regulator and start-up the device.

25.3.2 Interrupt logic

When the SE1100 is running in normal operation mode, the Charger Detect is capable of giving an interrupt to the CPU if a charger is connected/disconnected or both. Interrupt is enabled/disabled by writing to the CHGDIE bit in the ”CHGDCSR - Charger Detect Control and Status Register”. Interrupt is given when a charger is connected, disconnected or both depending on interrupt sense control settings. Selecting the correct interrupt sensing is done by writing to the CHGDISC bits in the ”CHGDCSR - Charger Detect Control and Status Register”. Charger Detect interrupt works asynchronous and will wake the CPU from any sleep mode. The Charger Detect is automatically disabled/enabled when changing the state of the Discharge FET, and any interrupt

that occurs when enabling or disabling the Discharge FET has to be carefully interpreted.

• When enabling the Discharge-FET the Charger Detect module is automatically disabled.When disabling the charger detect module a charger appear to be disconnected even if a charger is present.
• When disabling the Charge-FET the Charger Detect module is automatically enabled and a charger appear to be connected.

25.4 Register Description

25.4.1 CHGDCSR - Charger Detect Control and Status Register

Bit	7	6	5	4	3	2	1	0	
(0XE66)	-	-	-	BATTPVL	CHGDISC1	CHGDISC0	CHGDIF	CHGDIE	CHGDCSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7:5 - Res: Reserved
These bits are reserved bits and will always read as zero.

• Bit 4 - BATTPVL: BATT pin Voltage Level
BATTPVL will read one as long when the Charger Detect module is enabled and the BATT pin voltage is above the VPOT level. Otherwise the BATTPVL will read zero.

• Bit 3:2 - CHGDISC1:0: Charger Detect Interrupt Sense Control
Edges in the CHARGER_PRESENT signal shown in Figure 24-1 are used to activate a Charger Detect Interrupt if the interrupt enable bit in CHGDCSR are set. By writing the CHGDISC bits to the values shown in Table 24-1 the condition generating interrupt is configured. When changing the CHGDISC bits, an interrupt can occur. Therefore, it is recommended to first disable the Interrupt by clearing its CHGDIE bit in the CHGDICSR Register. Finally, the Charger Detect interrupt flag should be cleared by writing a logical zero to CHGDIF bit before the interrupt is re-enabled.

Table 25-1. Charger Detect Interrupt Sense Control.

CHGISC1:0	Detection
00	Charger Connect
01	Charger Disconnect
10	Charger Connect/Disconnect
11	None

• Bit 1 - CHGDIF: Charger Detect Interrupt Flag
Depending on the configuration of the CHGDISC bits in the CHGDCSR, this bit is set when a charger is either connected or disconnected. The Charger Detect Interrupt is executed if the CHGDIE bit are set. This bit is cleared by hardware when writing a logical zero to the CHGDIF. It

is recommended to write this bit to zero when setting CHGDIE.

• Bit 0 - CHGDIE: Charger Detect Interrupt Enable

When the CHGDIE bit is set (one), the Charger Detect Interrupt is enabled.

26. Voltage Regulator

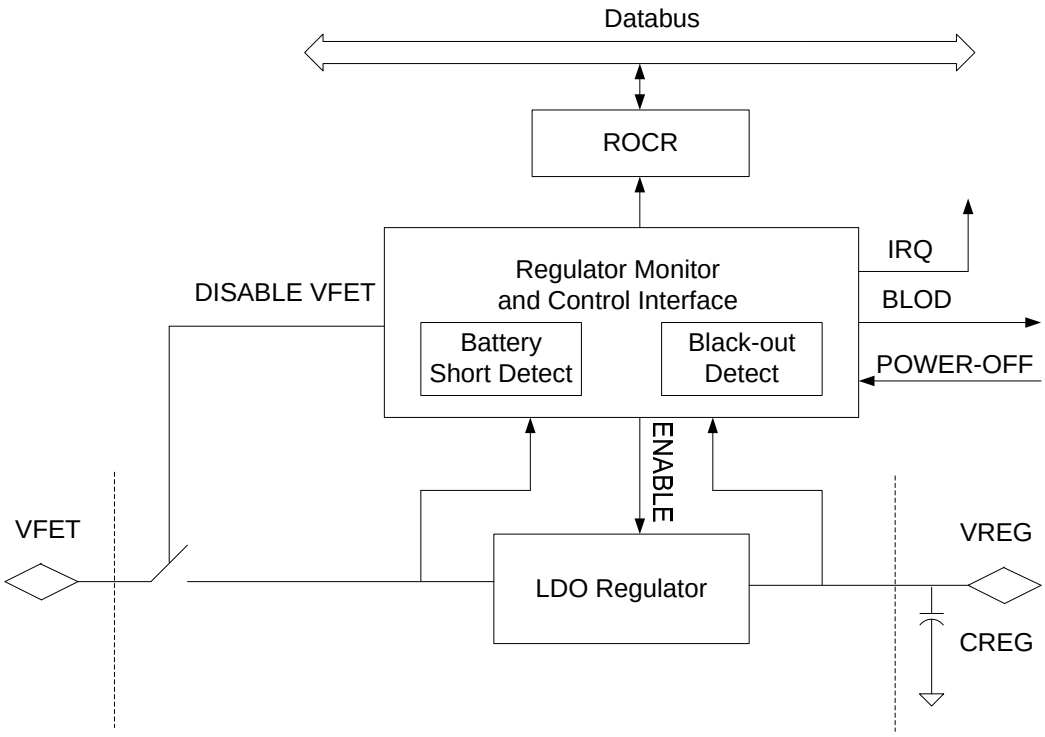
26.1 Features

- **Input voltage from 4-25V**
- **Fixed output voltage of 3.3V**
- **Battery Pack Short Detection**
- **Black-out Detection (BLOD)**

26.2 Overview

SE1100 get its voltage supply through the VFET terminal. Operating range at the VFET terminal is 4 - 25V. The on-chip LDO regulator regulates the VFET terminal down to 3.3V, which is a suitable supply voltage for the internal logic, I/O lines and analog circuitry. Figure 26- 1 illustrates the Voltage Regulator.

Figure 26-1. Voltage Regulator.



To ensure safe operating condition the Voltage Regulator has built in protection mechanisms to protect the internal circuitry if the voltage drops on either the input or output terminal. The Regulator Monitoring includes

• Battery Pack Short Detector
• Black-out Detector

An external decoupling capacitor (CREG) of minimum 1 μF is needed to ensure stable regulator operation. The same capacitor also serves as a reservoir capacitor to ensure that the chip is able to operate for some time without voltage at the VFET terminal.

26.3 Regulator Start-up

When the chip is in power-off mode the Voltage Regulator will be off and there will be no connection between VFET and VREG.

The regulator is started when the Charger Detect module detects that a charger is connected (For details on Charger Detect, see ”Charger Detect”). When starting the regulator the Voltage Regulator will stay in a force mode where the VREG output is raised against the VFET input voltage. As VREG increases to the target voltage level the regulator will automatically enter regulation mode, with a stable output voltage of nominally 3.3V. Figure 11-2, in System Control and Reset illustrates the start-up sequence from power-off.

26.4 Battery Pack Short Detection

The Voltage Regulator will continuously monitor the operating condition at the VFET terminal. If the voltage at VFET drops below the Regulator Short-circuit Level (RSCL), see ”Electrical Characteristics”, the Voltage Regulator enters the Battery Pack Short mode. In this mode, VFET is disconnected from VREG to avoid a quick drop in the voltage regulator output. When the voltage regulator enters this mode, the chip will be completely powered by the external reservoir capacitor (CREG). This allows the chip to operate a certain time without entering BOD reset, even if the VFET voltage is too low for the voltage regulator to operate.

An interrupt is issued when the regulator enters Battery Pack Short mode, if the ROCWIE bit in ROCR Register is set. This allows actions to be taken to reduce power consumption and hence prolonging the time that CREG can be used to power the chip.

In a typical short-circuit situation, VFET will drop as a consequence of high current consumption, and recover as soon as the Battery Protection module has disabled the FETs. Hence CREG should be dimensioned so that the chip can sustain operation without entering BOD reset, until the FETs are disabled either by HW or SW.

To minimize power consumption when the Voltage Regulator enters the Battery Pack Short mode, the chip should enter Power-save sleep mode as soon as possible after the ROCWIF interrupt is detected. The Watchdog Timer should be configured to wake up the CPU after a time that is considered safe. Software should then check the status of the ROC flag. If the ROCS flag is cleared, normal operation may be resumed.

26.5 Black-Out Detection

To ensure that the internal logic has safe operating condition, the Voltage Regulator has built-in Black-Out Detector (BLOD). If the voltage at the VREG pin drops below the Black-out Detection Level, VBLOT, the chip will automatically enter Power-off mode.

26.6 Register Description

26.6.1 ROCR – Regulator Operating Condition Register

Bit	7	6	5	4	3	2	1	0	
(0XE62)	ROCS	-	-	ROCD	-	-	ROCWIF	ROCWIE	ROCR
Read/Write	R	R	R	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 - ROCS: ROC Status

This bit is set when the Voltage Regulator operates in the Battery Pack Short mode, and cleared otherwise.

• Bit 6:5 - Res: Reserved Bits

These bits are reserved bits and will always read as zero.

• Bit 4 - ROCD: ROC Disable

Setting this bit will disable the Battery Pack Short Detector and VFET will never be disconnected from the LDO Regulator. Note that it is NOT recommended to disable the Battery Pack Short Detector by setting the ROCD bit unless VFET protection is implemented externally.

• Bit 3:2 - Res: reserved Bits

These bits are reserved bits and will always read as zero.

• Bit 1 - ROCWIF: ROC Warning Interrupt Flag

The ROCWIF Flag is set when the Voltage Regulator enters the Battery Pack Short mode. The flag is cleared by writing logic zero to it.

• Bit 0 - ROCWIE: ROC Warning Interrupt Enable

The ROCWIE bit enables interrupt caused by the Regulator Operating Condition Warning interrupt flag.

27. Battery Protection

27.1 Features

- **Short-circuit Protection**
- **Discharge Over-current Protection**
- **Charge Over-current Protection**
- **Discharge High-current Protection**
- **Charge High-current Protection**
- **Deep Under-voltage Protection**
- **Programmable and Lockable Detection Levels and Reaction Times**
- **Autonomous Operation Independent of CPU**

27.2 Overview

If the voltage at the VFET pin falls below the programmable Deep Under-voltage detection level, C-FET, PC-FET, and D-FET are disabled and the chip is set in Power-off mode to reduce power consumption to a minimum.

The Current Battery Protection circuitry (CBP) monitors the charge and discharge current and disables C-FET and D-FET if a Short-circuit, Over-current or High-current condition is detected. There are five different programmable detection levels: Short-circuit Detection Level, Discharge Over-current Detection Level, Charge Over-current Detection Level, Discharge High-current Detection Level, Charge High-current Detection Level. There are three different programmable delays for activating Current Battery Protection: Short-circuit Reaction Time, Over-current Reaction Time and High-current Reaction Time. After Current Battery Protection has been activated, the application software must re-enable the FETs. The Battery Protection hardware provides a hold-off time of 1 second before software can re-enable the FETs. This provides safety in case the application software should unintentionally try to re-enable the FETs too early.

The activation of a protection also issues an interrupt to the CPU. The battery protection interrupts can be individually enabled and disabled by the CPU.

Table 27-1. Effect of Battery Protection Types

Battery Protection Type	Interrupt Requests	PC-FET	C-FET	D-FET	Cell Balancing FETs	MCU
Short-circuit Protection	Entry	Operational	Disabled	Disabled	Operational	Operational
Discharge Over-current Protection	Entry	Operational	Disabled	Disabled	Operational	Operational
Charge Over-current Protection	Entry	Operational	Disabled	Disabled	Operational	Operational
Discharge High-current Protection	Entry	Operational	Disabled	Disabled	Operational	Operational
Charge High-current Protection	Entry	Operational	Disabled	Disabled	Operational	Operational
Deep Under-voltage Detected	Entry and/or Exit	Operational	Disabled	Disabled	Operational	Operational

27.3 Operation

The Current Battery Protections (CBP) monitors the cell current by sampling the shunt resistor voltage at the PPI/NNI input pins. A differential operational amplifier amplifies the voltage with a suitable gain. The output from the operational amplifier is compared to an accurate, programmable On-chip voltage reference by an Analog Comparator. If the shunt resistor voltage is above the Detection level for a time longer than the corresponding Protection Reaction Time, the chip activates Current Protection. A sampled system clocked by the internal ULP Oscillator is used for Short-circuit, Over-current, and High-current Protection. This ensures a reliable clock source, offset cancellation and low power consumption.

27.3.1 Short-circuit Protection

The Short-circuit detection is provided to enable a fast response time to very large discharge currents. If the voltage at the PPI/NNI pins is above the Short-circuit Detection Level for a period longer than Short-circuit Reaction Time, the Short-circuit Protection is activated. When the Short-circuit Protection is activated, the external D-FET and C-FET are disabled and a Current Protection Timer is started. This timer ensures that the D-FET and C-FET are disabled for at least one second. The application software must then set the DFE and CFE bits in the FET Control and Status Register to re-enable normal operation. If the D-FET is re-enabled before the cause of the short-circuit condition is removed, the Short-circuit Protection will be activated again.

27.3.2 Discharge Over-current Protection

If the voltage at the PPI/NNI pins is above the Discharge Over-current Detection level for a time longer than Over-current Protection Reaction Time, the chip activates Discharge Over-current Protection.

When the Discharge Over-current Protection is activated, the external D-FET and C-FET are disabled and a Current Protection Timer is started. This timer ensures that the FETs are disabled for at least one second. The application software must then set the DFE and CFE bits in the FET

Control and Status Register to re-enable normal operation. If the D-FET is re-enabled while the loading of the battery still is too large, the Discharge Over-current Protection will be activated again.

27.3.3 Charge Over-current Protection

If the voltage at the PPI/NNI pins is above the Charge Over-current Detection level for a time longer than Over-current Protection Reaction Time, the chip activates Charge Over-current Protection. When the Charge Over-current Protection is activated, the external D-FET and C-FET are disabled and a Current Protection Timer is started. This timer ensures that the FETs are disabled for at least one second. The application software must then set the DFE and CFE bits in the FET Control and Status Register to re-enable normal operation. If the C-FET is re-enabled and the charger continues to supply too high currents, the Charge Over-current Protection will be activated again.

27.3.4 Discharge High-current Protection

If the voltage at the PPI/NNI pins is above the Discharge High-current Detection level for a time longer than High-current Protection Reaction Time, the chip activates Discharge High-current Protection.

When the Discharge High-current Protection is activated, the external D-FET and C-FET are disabled and a Current Protection Timer is started. This timer ensures that the FETs are disabled for at least one second. The application software must then set the DFE and CFE bits in the FET Control and Status Register to re-enable normal operation. If the D-FET is re-enabled while the loading of the battery still is too large, the Discharge High-current Protection will be activated again.

27.3.5 Charge High-current Protection

If the voltage at the PPI/NNI pins is above the Charge High-current Detection level for a time longer than High-current Protection Reaction Time, the chip activates Charge High-current Protection. When the Charge High-current Protection is activated, the external D-FET and C-FET are disabled and a Current Protection Timer is started. This timer ensures that the FETs are disabled for at least one second. The application software must then set the DFE and CFE bits in the FET Control and Status Register to re-enable normal operation. If the C-FET is re-enabled and the charger continues to supply too high currents, the Charge High-current Protection will be activated again.

The Short-circuit, Over-current and High-current Protection parameters are programmable to adapt to different types of batteries. The parameters are set by writing to I/O Registers. The Parameter Registers can be locked after the initial configuration, prohibiting any further updates

until the next Hardware Reset.

27.4 Deep Under-voltage Protection

The Deep Under-voltage Protection ensures that the battery cells will not be discharged deeper than the programmable Deep Under-voltage detection level. If the voltage at the VFET pin is below this level for a time longer than the programmable delay time, C-FET, PC-FET and D-FET are automatically switched off and the chip enters Power-off mode. The Deep Under-voltage Early Warning interrupt flag (DUVIF) will be set 750 ms before the chip enters Power-off. This will give the CPU a chance to take necessary actions before the power is switched off. The device will remain in the Power-off mode until a charger is connected. When a charger is detected, a normal power-up sequence is started and the chip initializes to default state. The Deep Under-voltage delay time and Deep Under-voltage detection level are set in the Battery Protection Deep Under-voltage Register (BPDUV). The Parameter Registers can be locked after the initial configuration, prohibiting any further updates until the next Hardware Reset.

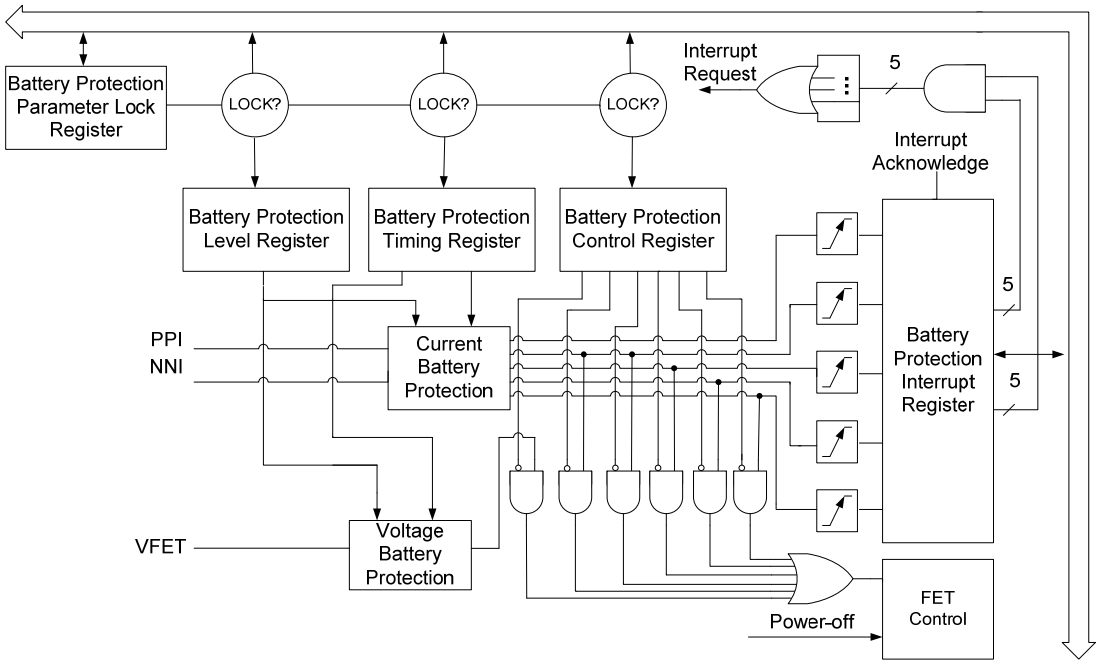
27.5 Optimizing Usage for Low Power Consumption

In order to reduce power consumption, Short-circuit, Discharge High-current and Discharge Over-current Protection are automatically deactivated when the D-FET is disabled. The Charge Over-current and Charge High-current Protection are disabled when the C-FET is disabled. Note however that Charge Over-current Protection and Charge High-current Protection are never automatically disabled when the chip is operated in DUVR mode. Also note that none of the current protections are deactivated by the External Protection Input. To save power during an External Protection event, DFE and CFE in the FCSR register should be cleared and make sure that the chip is not operating in DUVR mode.

27.6 Battery Protection CPU Interface

The Battery Protection CPU Interface is illustrated in Figure 27-2.

Figure 27-2. Battery Protection CPU Interface



Each protection originating from the Current Battery Protection module has an Interrupt Flag. Each Flag can be read and cleared by the CPU, and each flag has an individual interrupt enable. All enabled flags are combined into a single battery protection interrupt request to the CPU. This interrupt can wake up the CPU from any operation mode, except Power-off. The interrupt flags are cleared by writing a logic '0' to their bit locations from the CPU. An interrupt event for the External Protection Input can be generated by enabling the external interrupt for the input port.

27.7 Register Description

27.7.1 BPPLR – Battery Protection Parameter Lock Register

Bit	7	6	5	4	3	2	1	0	
(0XE7F)	–	–	–	–	–	CBPUB	BPPLE	BPPL	BPPLR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7:3 – Res: Reserved Bits

These bits are reserved bits and will always read as zero.

• Bit 2 - CBPUB: CBP Update Busy

The CBP operates in a different clock domain than the CPU. Whenever a new value is written to BPCR, BPHCTR, BPOCTR, BPSCTR, BPDHCD, BPCHCD, BPDOCD, BPCOCD and BPSCD, this value must be synchronized to the CBP clock domain. Subsequent writes to these registers will be blocked during this synchronization. Synchronization of one of the registers will block updating of all the others. The CBPUB bit will be read as one while any of these registers is being synchronized, and will be read as zero when neither register is being synchronized.

• Bit 1 – BPPLE: Battery Protection Parameter Lock Enable

• Bit 0 – BPPL: Battery Protection Parameter Lock

The BPCR, BPHCTR, BPOCTR, BPSCTR, BPDHCD, BPCHCD, BPDOCD, BPCOCD and BPSCD Battery Protection registers can be locked from any further software updates. Once locked, these registers cannot be accessed until the next hardware reset. This provides a safe method for protecting the registers from unintentional modification by software runaway. It is recommended that software sets these registers shortly after reset, and then protect the registers from further updates.

To lock these registers, the following algorithm must be followed:
1. In the same operation, write a logic one to BPPLE and BPPL.
2. Within the next four clock cycles, in the same operation, write a logic zero to BPPLE and a logic one to BPPL.

27.7.2 BPCR – Battery Protection Control Register

Bit	7	6	5	4	3	2	1	0	
(0XE7E)	–	–	DUVD	SCD	DOCD	COCD	DHCD	CHCD	BPCR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bits 7:6 – Res: Reserved Bits

These bits are reserved bits and will always read as zero.

• Bit 5 –DUVD: Deep Under-voltage Protection Disable

When the DUVD bit is zero, the Deep Under-voltage Protection is disabled. The Deep Under-voltage Detection will be disabled, and any Deep Under-voltage condition will be ignored.

• Bit 4 – SCD: Short Circuit Protection Disabled

When the SCD bit is zero, the Short-circuit Protection is disabled. The Short-circuit Detection will be disabled, and any Short-circuit condition will be ignored.

• Bit 3 – DOCD: Discharge Over-current Protection Disabled

When the DOCD bit is zero, the Discharge Over-current Protection is disabled. The Discharge Over-current Detection will be disabled, and any Discharge Over-current condition will be ignored.

• Bit 2 – COCD: Charge Over-current Protection Disable

When the COCD bit is zero, the Charge Over-current Protection is disabled. The Charge Over-current Detection will be disabled, and any Charge Over-current condition will be ignored.

• Bit 1 – DHCD: Discharge High-current Protection Disabled

When the DHCD bit is zero, the Discharge High-current Protection is disabled. The Discharge High-current Detection will be disabled, and any Discharge High-current condition will be

ignored.

• Bit 0 – CHCD: Charge High-current Protection Disable

When the CHCD bit is zero, the Charge High-current Protection is disabled. The Charge High-current Detection will be disabled, and any Charge High-current condition will be ignored.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPCR register is written. Any writing to the BPCR register during this period will be ignored.

27.7.3 BPSCTR – Battery Protection Short-current Timing Register

Bit	7	6	5	4	3	2	1	0	
(0XE7B)	–	SCPT[6:0]							BPSCTR
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	1	0	0	0	0	

• Bit 7 – Res: Reserved Bits

This bit is reserved in the SE1100 and will always read as zero.

• Bit 6:0 – SCPT6:0: Short-current Protection Timing

These bits control the delay of the Short-circuit Protection. The Short-circuit Timing can be set with a step size of 62.5 µs as shown in Table 27-2.

Table 27-2. Short-circuit Protection Reaction Time. SCPT[6:0] with corresponding Short-circuit Delay Time.

Short-circuit Protection Reaction Time	
SCPT[6:0]	Typ
0x00	(78.12us + 7.812us)+Td(3)
0x01	(78.12us + 7.812us)+Td(3)
0x02	((78.12*0x02)us + 7.812us)+Td(3)
0x03	((78.12*0x03)us + 7.812us)+Td(3)
…	…
0x7E	((78.12*0x7E)us + 7.812us)+Td(3)
0x7F	((78.12*0x7F)us + 7.812us)+Td(3)

Notes:

1. The actual value depends on the actual frequency of the ”Ultra Low Power RC Oscillator”. See ”Electrical Characteristics”.
2. Initial value: SCPT[0x10](1ms).
3. An additional delay Td can be expected after enabling the Discharge FET due to initialization of the protection circuit. With nomial ULP frequency this delay is maximum 86 µs.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPSCTR register is written. Any writing to the BPSCTR register during this period will be ignored.

27.7.4 BPOCTR – Battery Protection Over-current Timing Register

Bit	7	6	5	4	3	2	1	0	
(0XE7C)	–	–	OCPT[5:0]						BPOCTR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	1	0	

• Bit 7:6 – Res: Reserved Bits

These bits are reserved in the TLS5103 and will always read as zero.

• Bit 5:0 – OCPT5:0: Over-current Protection Timing

These bits control the delay of the Over-circuit Protection. The Over-current Timing can be set with a step size of 0.5 ms as shown in Table 27-3

Table 27-3. Over-current Protection Reaction Time. OCPT[5:0] with corresponding Over- current Delay Time.

Over-circuit Protection Reaction Time	
OCPT[5:0]	Typ
0x00	(0.625ms)+Td(3)
0x01	(0.625ms)+Td(3)
0x02	(0.625ms*0x02)+Td(3)
0x03	(0.625ms*0x03)+Td(3)
…	…
0x3E	(0.625ms*0x3e)+Td(3)
0x3F	(0.625ms*0x3f)+Td(3)

Notes: 1. The actual value depends on the actual frequency of the ”Ultra Low Power RC Oscillator”. See ”Electrical Characteristics”.

2. Initial value.

3. An additional delay Td can be expected after enabling the corresponding FET. This is related to the initialization of the protection circuitry. For the Discharge Over-Current protection, this applies when enabling the Discharge FET. For Charge Over-Current protection, this applies when enabling the Charge FET. With nominal ULP frequency this delay is maximum 0.1 ms.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPOCTR register is written. Any writing to the BPOCTR register during this period will be ignored.

27.7.5 BPHCTR – Battery Protection High-current Timing Register

Bit	7	6	5	4	3	2	1	0	
(0XE7D)	–	–	HCPT[5:0]						BPHCTR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	1	

• Bit 7:6 – Res: Reserved Bits

These bits are reserved in the TLS5103 and will always read as zero.

• Bit 5:0 – HCPT5:0: High-current Protection Timing

These bits control the delay of the High-circuit Protection. The High-current Timing can be set with a step size of 2 ms as shown in Table 27-4 .

Table 27-4. High-current Protection Reaction Time. HCPT[5:0] with corresponding High- current Delay Time.

Short-circuit Protection Reaction Time	
HCPT[5:0]	Typ
0x00	(2.5ms)+Td(3)
0x01	(2.5ms)+Td(3)
0x02	(2.5ms*0x02)+Td(3)
0x03	(2.5ms*0x03)+Td(3)
…	…
0x3E	(2.5ms*0x3e)+Td(3)
0x3F	(2.5ms*0x3f)+Td(3)

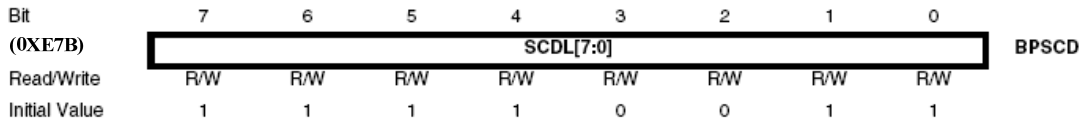
Notes: 1. The actual value depends on the actual frequency of the ”Ultra Low Power RC Oscillator”. See ”Electrical Characteristics”.

2. Initial value.

3. An additional delay Td can be expected after enabling the corresponding FET. This is related to the initialization of the protection circuitry. For the Discharge High-Current protection, this applies when enabling the Discharge FET. For Charge High-Current protection, this applies when enabling the Charge FET. With nominal ULP frequency this delay is maximum 0.2 ms.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPHCTR register is written. Any writing to the BPHCTR register during this period will be ignored.

27.7.6 BPSCD – Battery Protection Short-circuit Detection Level Register

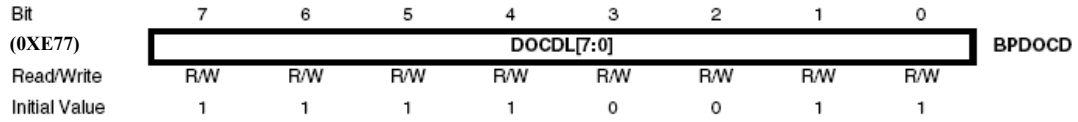


• Bits 7:0 – SCDL7:0: Short-circuit Detection Level

These bits sets the RSENSE voltage level for detection of Short-circuit in the Discharge Direction, as defined in Table 27-5-1 .

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPSCD register is written. Any writing to the BPSCD register during this period will be ignored.

27.7.7 BPDOCD – Battery Protection Discharge-Over-current Detection Level Register

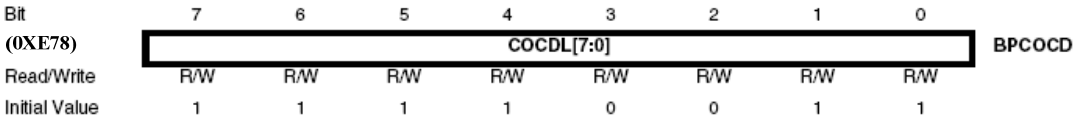


• Bits 7:0 – DOCDL7:0: Discharge Over-current Detection Level

These bits sets the RSENSE voltage level for detection of Discharge Over-current, as defined in Table 27-5-1.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPDOCD register is written. Any writing to the BPDOCD register during this period will be ignored.

27.7.8 BPCOCD – Battery Protection Charge-Over-current Detection Level Register

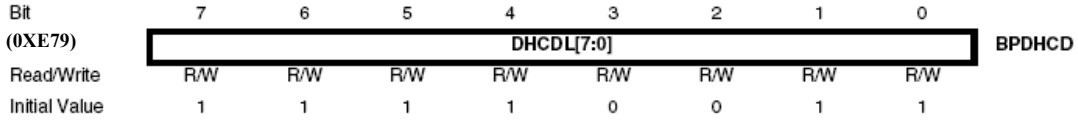


• Bits 7:0 –COCDL7:0: Charge Over-current Detection Level

These bits sets the RSENSE voltage level for detection of Charge Over-current, as defined in Table 27-5-2.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPCOCD register is written. Any writing to the BPCOCD register during this period will be ignored.

27.7.9 BPDHCD – Battery Protection Discharge-High-current Detection Level Register



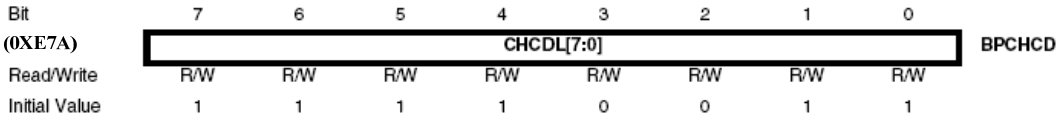
• Bits 7:0 – DHCDL7:0: Discharge High-current Detection Level

These bits sets the RSENSE voltage level for detection of DischargeHigh-current, as defined in Table 27-5-1.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP

oscillator cycles + 3 CPU clock cycles is required between each time the BPDHCD register is written. Any writing to the BPDHCD register during this period will be ignored.

27.7.10 BPCHCD – Battery Protection Charge-High-current Detection Level Register



• Bits 7:0 –CHCDL7:0: Charge High-current Detection Level

These bits sets the RSENSE voltage level for detection of Charge High-current, as defined in Table 27-5-2.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPCHCD register is written. Any writing to the BPCHCD register during this period will be ignored.

Table 27-5. DL[7:0] with corresponding RSENSE voltage levels for all Current Detection Level Registers (BPSCD, BPDOCD, BPDHCD, BPCOCD, BPCHCD).

Current Protection Detection Levels			
DL	Min	Typ.(mV)	Max.
0xf3		2.94	
0xf4		7.915	
0xf5		12.985	
0xf6		17.175	
0xf7		22.25	
0xf8		26.1	
0xf9		31.2	
0xFA		36.2	
0xFB		41.225	
0xFC		45.35	
0xFD		50.4	
0x78		54.58	
0x79		64.65	
0x7A		73.55	
0x7B		83.9	
0x7C		92.25	
0x7D		102	
0x7E		112.05	
0x7F		121.8	
0x39		129.95	
0x3A		147.5	
0x3B		167.3	

0x3C		183. 75	
0x3D		203. 3	
0x3E		222. 3	
0x3F		241. 4	
0x19		264. 8	

27.7.11 BPDUV – Battery Protection Deep Under Voltage Register

Bit	7	6	5	4	3	2	1	0	
(0XE5B)	–	–	DUVT[1:0]		DUDL[3:0]				BPDUV
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7:6 – Res: Reserved Bits

These bits are reserved bits in the ATmega406 and will always read as zero.

• Bits 5:4 – DUVT1:0: Deep Under-voltage Timing

These bits set the Deep Under-voltage Protection delay.

Table 27-6. DUVT[1:0] with Corresponding Deep Under-voltage Delay

DUVT1:0	Deep Under-voltage Delay
00	750 ms
01	1000 ms
10	1250 ms
11	1500 ms

• Bits 3:0 – DUDL3:0: Deep Under-voltage Detection Level

These bits set the Deep Under-voltage detection level.

Table 27-7. DUDL[3:0] with Corresponding Deep Under-voltage Detection Level

DUDL[3:0]	Typ	DUDL[3:0]	Typ
0000	5.4v	1000	8.6v
0001	5.8v	1001	8.9v
0010	6.2v	1010	9.2v
0011	6.6v	1011	9.5v
0100	7.1v	1100	9.9v
0101	7.5v	1101	10.3v
0110	7.9v	1110	10.7v
0111	8.2v	1111	11.1v

27.7.12 BPIMSK – Battery Protection Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0XE74)	-	-	DUVIE	SCIE	DOCIE	COCIE	DHCIE	CHCIE	BPIMSK
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7:6 – Res: Reserved Bits

These bits are reserved bits and will always read as zero.

• Bit 5 – DUVIE: Deep Under-voltage Early Warning Interrupt Enable

The DUVIE bit enables interrupt caused by the Deep Under-voltage Early Warning Interrupt Flag

• Bit 4 – SCIE: Short-circuit Protection Activated Interrupt Enable

The SCIE bit enables interrupt caused by the Short-circuit Protection Activated Interrupt.

• Bit 3 – DOCIE: Discharge Over-current Protection Activated Interrupt Enable

The DOCIE bit enables interrupt caused by the Discharge Over-current Protection Activated Interrupt.

• Bit 2 – COCIE: Charge Over-current Protection Activated Interrupt Enable

The COCIE bit enables interrupt caused by the Charge Over-current Protection Activated Interrupt.

• Bit 1 - DHCIE : Discharger High-current Protection Activated Interrupt Enable

The DHCIE bit enables interrupt caused by the Discharge High-current Protection Activated Interrupt.

• Bit 0 - CHCIE : Charger High-current Protection Activated Interrupt Enable

The CHCIE bit enables interrupt caused by the Charge High-current Protection Activated Interrupt.

27.7.13 BPIFR – Battery Protection Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
(0XE75)	-	-	DUVIF	SCIF	DOCIF	COCIF	DHCIF	CHCIF	BPIFR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7:6 – Res: Reserved Bit

These bits are reserved and will always read as zero.

• Bit 5 – DUVIF: Deep Under-voltage Early Warning Interrupt Flag

If the voltage at VFET pin is below the Deep Under-voltage detection level and only 250 ms is left of the Deep Under-voltage delay, DUVIF becomes set. The flag must be cleared by writing a

logical zero to it.

• Bit 4 – SCIF: Short-circuit Protection Activated Interrupt

Once Short-circuit violation is detected, SCIF becomes set. The flag is cleared by writing a logic zero to it.

• Bit 3 – DOCIF: Discharge Over-current Protection Activated Interrupt

Once Discharge Over-current violation is detected, DOCIF becomes set. The flag is cleared by writing a logic zero to it.

• Bit 2 – COCIF: Charge Over-current Protection Activated Interrupt

Once Charge Over-current violation is detected, COCIF becomes set. The flag is cleared by writing a logic zero to it.

• Bit 1 – DHCIF: Discharge High-current Protection Activated Interrupt

Once Discharge High-current violation is detected, DHCIF becomes set. The flag is cleared by writing a logic zero to it.

• Bit 0 – CHCIF: Charge High-current Protection Activated Interrupt

Once Charge High-current violation is detected, CHCIF becomes set. The flag is cleared by writing a logic zero to it.

28. FET Driver

28.1 Overview

In addition to the FET disable control signals from the battery protection circuitry, the CPU may disable the Charge FET (C-FET), the Discharge FET (D-FET), or both, by writing to the FET Control Register. Note that the CPU is never allowed to enable a FET that is disabled by the battery protection circuitry. The FET control is shown in Figure 28-1 .

The PWM output from the CCP, CCP1 PIN, can be configured to drive the C-FET, Precharge FET (PC-FET) or both directly. This can be useful for controlling the charging of the battery cells. The PWM is configured by the TIMER2 and CCP1 registers. Note that the CCP1 pins does not need to be configured as an output.This means that the PWM output can be used to drive the C-FET and/or the PC-FET without occupying the CCP1-pin.

If C-FET is disabled and D-FET enabled, discharge current will run through the body-drain diodeof the C-FET and vice versa. To avoid the potential heat problem from this situation, software must ensure that D-FET is not disabled when a charge current is flowing, and that C-FET is not disabled when a discharge current is flowing.

If the battery has been deeply discharged, large surge currents may result when a charger is connected. In this case, it is recommended to first pre charge the battery through a current limiting resistor. For this purpose, SE1100 provides a Precharge FET (PC-FET) control output.This output is default enabled.

If SE1100 has entered the Power-off mode, all FET control outputs will be disabled. When a charger is connected, the CPU will wake up. When waking up from Power-off mode, the C-FET and D-FET control outputs will remain disabled while PC-FET is default enabled. When the CPU detects that the cell voltages have risen enough to allow normal charging, it should enable the C-FET and D-FET control outputs and disable the PC-FET control output.

If the Current Battery Protection has been activated, the Current Protection Timer will ensure a hold-off time of 1 second before software can re-enable the external FETs.

Figure 28-1. FET Control Block Diagram

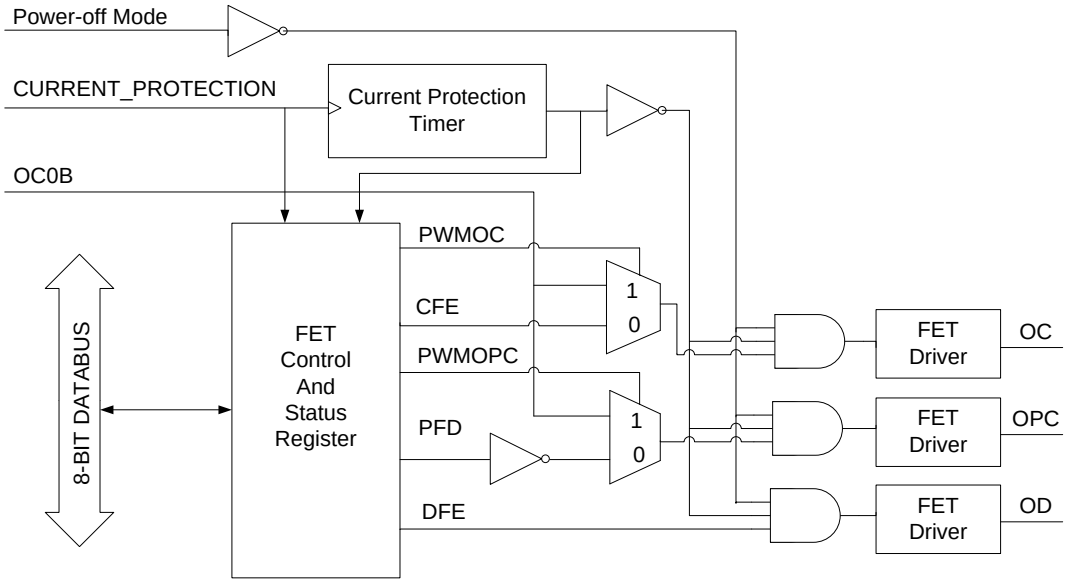
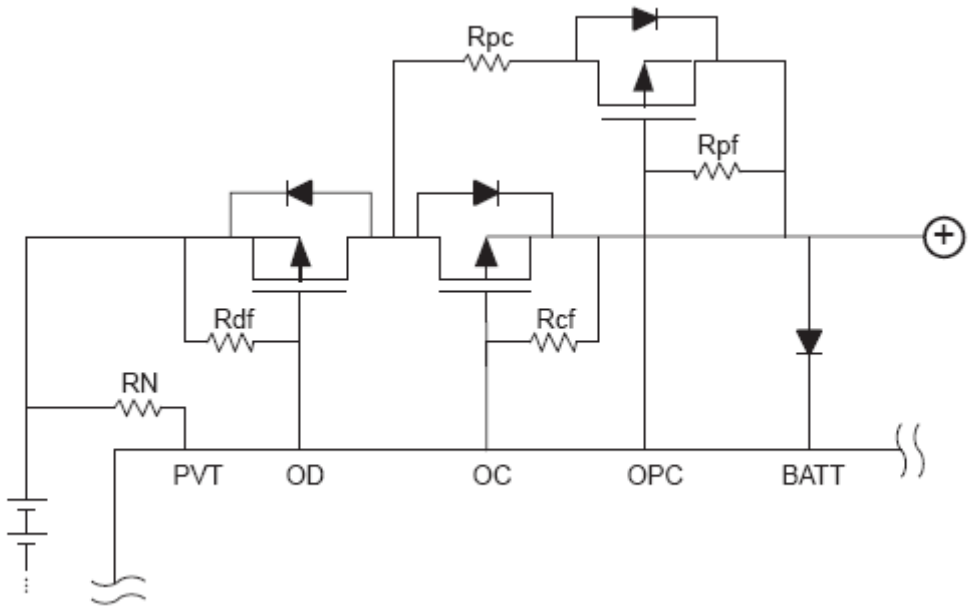


Figure 28-2. Connection of external FETs



The connection of external FETs to OD, OC, and OPC is shown in Figure 28-2. When switching on an FET, the output pulls the gate quickly low to avoid heating of the FET. When the FET is switched completely on, the output changes operation mode in order to reduce current consumption. The gate-source voltage for the FET when switched on, |VGS_ON|, is limited to 13V ± 15%.

When disabling an external FET, the FET Driver output quickly pushes the gate voltage to the source pin potential, making the gate-source voltage of the FET close to zero. This disables the FET, and the FET Driver output switches operation mode to high impedance in order to reduce current consumption. The external resistor will keep the gate-source voltage at zero until the FET is enabled again and its gate is pulled low as explained above.

28.2 Register Description for FET Control

The FET Controller operates in a different clock domain than the CPU. Whenever a new value is written to the FCSR, the value must be synchronized to the FET Controller clock domain. Subsequent writes to this register should not be made during this synchronization. Therefore, after writing to this register, a guard time of 3 ULP Oscillator cycles + 3 CPU clock cycles is required. It is recommended that software only reads the FCSR when handling a Battery Protection Interrupt (BPINT).

28.2.1 FCSR – FET Control and Status Register

Bit	7	6	5	4	3	2	1	0	
(0XE72)	FETUB	–	PWMOC	PWMOPC	CPS	DFE	CFE	PFD	FCSR
Read/Write	R	R	R/W	R/W	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7 - FETUB: FET DRIVER Update Busy

The FET DRIVER operates in a different clock domain than the CPU. Whenever a new value is written to FCSR, this value must be synchronized to the FET DIRVER clock domain. Subsequent writes to these registers will be blocked during this synchronization. Synchronization of one of the registers will block updating of all the others. The FETUB bit will be read as one while any of these registers is being synchronized, and will be read as zero when neither register is being synchronized.

• Bits6 – Res: Reserved Bits

These bits are reserved bits in the SE1100, and will always read as zero.

• Bit 5 – PWMOC: Pulse Width Modulation of OC output

When the PWMOC is cleared (zero), the CFE bit and the battery protection circuitry controls the OC output. When this bit is set (one), the OC output will be the logical AND of the PWM output from CCP1 and the inverse of CURRENT_PROTECTION from the Battery Protection circuitry.

• Bit 4 – PWMOPC: Pulse Width Modulation of OPC output

When the PWMOPC is cleared (zero), the PFD bit and the battery protection circuitry controls the OPC output. When this bit is set (one), the OPC output will be the logical AND of the PWM output from CCP1 and the inverse of CURRENT_PROTECTION from the Battery Protection circuitry.

• Bit 3 – CPS: Current Protection Status

The CPS bit shows the status of the Current Protection. This bit is set (one) when the Current Protection Timer is activated, and is cleared (zero) when the hold-off time has elapsed.

• Bit 2 – DFE: Discharge FET Enable

When the DFE bit is cleared (zero), the Discharge FET will be disabled regardless of the state of

the Battery Protection circuitry. When this bit is set (one), the Discharge FET state is determined by the Battery Protection circuitry. This bit will be cleared when CURRENT_PROTECTION is set (one).

• Bit 1 – CFE: Charge FET Enable

When the CFE bit is cleared (zero), the Charge FET will be disabled regardless of the state of the Battery Protection circuitry. When this bit is set (one), the Charge FET state is determined by the Battery Protection circuitry. This bit will be cleared when CURRENT_PROTECTION is set (one).

• Bit 0 – PFD: Precharge FET Disable

The PFD bit provides complete control of the Precharge FET. When the PFD bit is cleared (zero), the Precharge FET will be enabled. When the PFD bit is set (one), the Precharge FET will be disabled. This bit will be cleared when the CURRENT_PROTECTION is set (one)

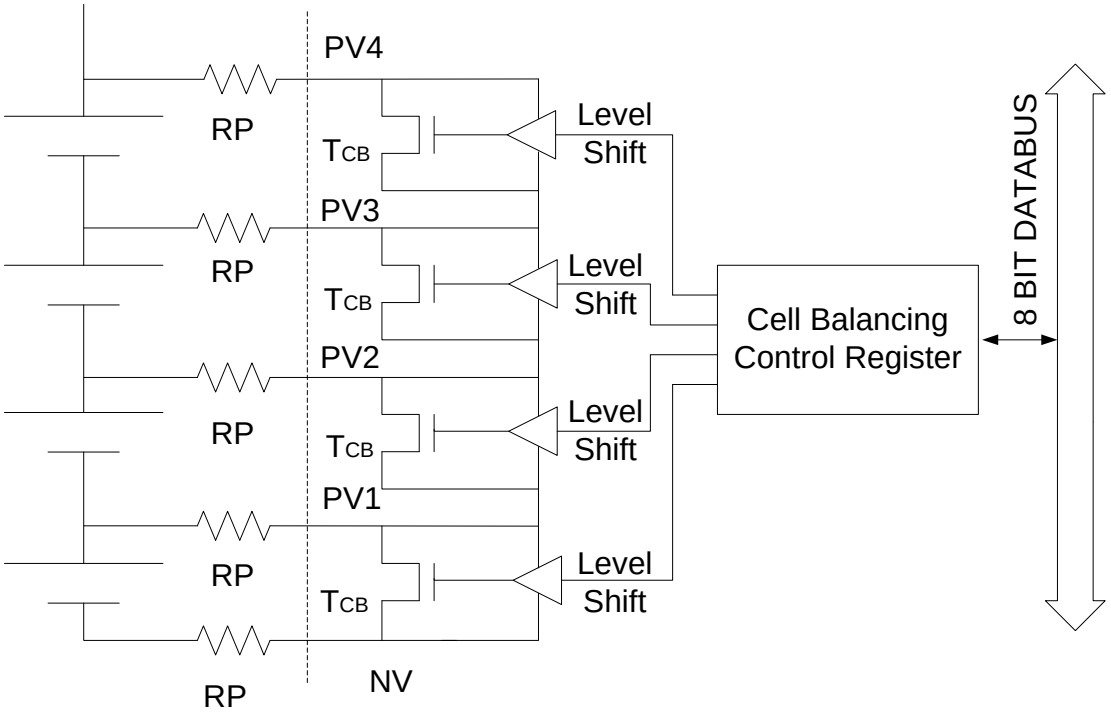
29. Cell Balancing

29.1 Overview

SE1100 incorporates cell balancing FETs. The chip provides one cell balancing FET for each battery cell in series. The FETs are directly controlled by the application software, allowing the cell balancing algorithms to be implemented in software. The FETs are connected in parallel with the individual battery cells. The cell balancing is illustrated in Figure 29-1. The figure shows a four-cell configuration. The cell balancing FETs are disabled in the Power-off mode. For typical current through the Cell Balancing FETs, see ”Electrical Characteristics”.

The Cell Balancing FETs are controlled by the CBCR. Neighbouring FETs cannot be simultaneouslyenabled. If trying to enable two neighbouring FETs, both will be disabled.

Figure 29-1. Cell Balancing



29.2 Register Description

29.2.1 CBCR – Cell Balancing Control Register

Bit	7	6	5	4	3	2	1	0	
(0XE73)	–	–	–	–	CBE4	CBE3	CBE2	CBE1	CBCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• Bit 7:4 – Res: Reserved Bits

These bits are reserved bits in the SE1100 and will always read as zero.

• Bit 3 – CBE4: Cell Balancing Enable 4

When this bit is set, the integrated Cell Balancing FET between terminals PV4 and PV3 will be enabled. When the bit is cleared, the Cell Balancing FET will be disabled. The Cell Balancing FETs are always disabled in Power-off mode. CBE4 cannot be set if CBE3 is set.

• Bit 2 – CBE3: Cell Balancing Enable 3

When this bit is set, the integrated Cell Balancing FET between terminals PV3 and PV2 will be enabled. When the bit is cleared, the Cell Balancing FET will be disabled. The Cell Balancing FETs are always disabled in Power-off mode. CBE3 cannot be set if CBE2 or CBE4 is set.

• Bit 1 – CBE2: Cell Balancing Enable 2

When this bit is set, the integrated Cell Balancing FET between terminals PV2 and PV1 will be enabled. When the bit is cleared, the Cell Balancing FET will be disabled. The Cell Balancing FETs are always disabled in Power-off mode. CBE2 cannot be set if CBE1 or CBE3 is set.

• Bit 0 – CBE1: Cell Balancing Enable 1

When this bit is set (one), the integrated Cell Balancing FET between terminals PV1 and NV will be enabled. When the bit is cleared (zero), the Cell Balancing FET will be disabled. The Cell Balancing FETs are always disabled in Power-off mode. CBE1 cannot be set if CBE2 is set.

30. SPECIAL FEATURES OF THE CPU

SE1100 have a Watchdog Timer which is permanently enabled via the configuration bits or software controlled. It runs off its own RC oscillator for added reliability. There are two timers that offer necessary delays on power-up. One is the Oscillator Start-up Timer (OST), intended to keep the chip in Reset until the crystal oscillator is stable. The other is the Power-up Timer (PWRT) which provides a fixed delay on power-up only, designed to keep the part in Reset while the power supply stabilizes. With these two timers on-chip, most applications need no external Reset circuitry.

30.1 Configuration Bits

The configuration bits can be programmed (read as '0') or left unprogrammed (read as '1'), to select various device configurations. These bits are mapped starting at program memory location 300000h. The user will note that address 300000h is beyond the user program memory space. In fact, it belongs to the configuration memory space (300000h-3FFFFFh) which can only be accessed using table reads. Programming the Configuration registers is done in a manner similar to programming the Flash memory.

TABLE 30-1: CONFIGURATION BITS

Config Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
Memory Lock Bits	300000	-	-	BLB12	BLB11	BLB02	BLB01	LB2	LB1	1111 1111
FUSE HIGH	300001	-	-	BODEN	OSTEN	PWRTEN	JTAGEN	STVREN	BOOTRST	1111 1111
FUSE LOW	300002	-	-	-	-	SPIEN	SUT2EN	SUT1EN	SUT0EN	1111 1111
WDT Prescaler	300003	-	-	-	-	WDTPS2	WDTPS1	WDTPS0	WDTON	1111 1111
SUT0	300004						SUT0S2	SUT0S1	SUT0S0	1111 1111
SUT1	300005						SUT1S2	SUT1S1	SUT1S0	1111 1111
SUT2	300006						SUT2S2	SUT2S1	SUT2S0	1111 1111

30.2 Program And Data Memory Lock Bits

The SE1100 provides six Lock bits which can be left unprogrammed ("1") or can be programmed ("0") to obtain the additional features listed in Table 30-3. The Lock bits can only be erased to "1" with the Chip Erase command.

Table 30-2. Lock Bit Byte

Lock Bit Byte	Bit No	Description	Default Value
	7	–	1 (unprogrammed)
	6	–	1 (unprogrammed)
BLB12	5	Boot Lock bit	1 (unprogrammed)
BLB11	4	Boot Lock bit	1 (unprogrammed)
BLB02	3	Boot Lock bit	1 (unprogrammed)
BLB01	2	Boot Lock bit	1 (unprogrammed)
LB2	1	Lock bit	1 (unprogrammed)
LB1	0	Lock bit	1 (unprogrammed)

Table 30-3. Lock Bit Protection Modes

Memory Lock Bits			Protection Type
LB Mode	LB2	LB1	
1	1	1	No memory lock features enabled
2	1	0	Further programming of the Flash and EEPROM is disabled in JTAG Serial Programming mode. The Fuse bits are locked in JTAG Serial Programming mode.
3	0	0	Further programming and verification of the Flash and EEPROM is disabled in JTAG Serial Programming mode. The Boot Lock bits and Fuse bits are locked in JTAG Serial Programming mode
BLB0 Mode	BLB02	BLB01	
1	1	1	No restrictions for EECON1 accessing the Application section.
2	1	0	EECON1 WR function is not allowed to write to the Application section.
3	0	0	EECON1 WR function is not allowed to write to the Application section, and TBLRD executing from the Boot Loader section is not allowed to read from the Application section.
4	0	1	TBLRD executing from the Boot Loader section is not allowed to read from the Application section.
BLB1 Mode	BLB12	BLB11	
1	1	1	No restrictions for EECON1 and TBLRD accessing the Boot Loader section.
2	1	0	EECON1 WR function is not allowed to write to the Boot Loader section.
3	0	0	EECON1 WR function is not allowed to write to the Boot Loader section, and TBLRD executing from the Application section is not allowed to read from the Boot Loader section.
4	0	1	TBLRD executing from the Application section is not allowed to read from the Boot Loader section.

30.3 Fuse Bits

The SE1100 has two Fuse bytes. Table 30-5 and Table 30-4 describe briefly the functionality of all the fuses and how they are mapped into the Fuse byte. Note that the fuses are read as logical zero, "0", if they are programmed.

30.3.1 High Byte

Bit No	Fuse High Byte	Description	Default Value
7-6	-		
5	BODEN	Enable Brown Out Reset	1 (unprogrammed)
4	OSTEN	Enable Oscillator Start-up Timer	1 (unprogrammed)
3	PWRTEN	Enable Power-up Timer	1 (unprogrammed)
2	JTAGEN	Enable Jtag Control	1 (unprogrammed)
1	STVREN	Enable Stack Full/Underflow Reset	1 (unprogrammed)
0	BOOTRST	Select Reset Vector	1 (unprogrammed)

30.3.2 Low Byte

Bit No	Fuse High Byte	Description	Default Value
7-4	-		
3	SPIEN	Enable jtag cpu control	1 (unprogrammed)
2	SUT2EN	Select start-up time	1 (unprogrammed)
1	SUT1EN	Select start-up time	1 (unprogrammed)
0	SUT0EN	Select start-up time	1 (unprogrammed)

30.4 WDT Bits

Config Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
WDT Prescaler	300003	-	-	-	-	WDTPS2	WDTPS1	WDTPS0	WDTON	1111 1111

Bit 7-4 Unimplemented: Read as '0'

bit 3-1 WDTPS2:WDTPS0: Watchdog Timer Postscale Select bits

111 = 1:128

110 = 1:64

101 = 1:32

100 = 1:16

011 = 1:8

010 = 1:4

001 = 1:2

000 = 1:1

Note: The Watchdog Timer postscale select bits configuration used in the SE1100 .

bit 0 WDTEN: Watchdog Timer Enable bit

1 = WDT enabled

0 = WDT disabled (control is placed on the SWDTEN bit)

30.5 Watchdog Timer (WDT)

The Watchdog Timer is a free running, on-chip RC oscillator which does not require any external components. This RC oscillator is ULP RC. During normal operation, a WDT time-out generates a device Reset (Watchdog Timer Reset). If the device is in Sleep mode, a WDT time-out causes the device to wake-up and continue with normal operation (Watchdog Timer wake-up). The TO bit in the RCON register will be cleared upon a WDT time-out.The Watchdog Timer is enabled/disabled by a device configuration bit. If the WDT is enabled, software execution may not disable this function. When the WDTEN configuration bit is cleared, the SWDTEN bit enables/disables the operation of the WDT.

The WDT time-out period values may be found in **"Electrical Characteristics" .** Values for the WDT postscaler may be assigned using the configuration bits.

Note: The CLRWDT and SLEEP instructions clear the WDT and the postscaler if assigned to the WDT and prevent it from timing out and generating a device Reset condition.

Note: When a CLRWDT instruction is executed and the postscaler is assigned to the WDT, the postscaler count will be cleared but the postscaler assignment is not changed.

30.5.1 CONTROL REGISTER

Register 24-1 shows the WDTCON register. This is a readable and writable register which contains a control bit that allows software to override the WDT enable configuration bit only when the configuration bit has disabled the WDT.

REGISTER 24-1: WDTCON: WATCHDOG TIMER CONTROL REGISTER

Read/Wite	U-0	U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	
(0XFD1)	—	—	—	—	—	—	—	SWDTEN	**WDTCON**
Bit	bit 7							bit 0	

bit 7-1 Unimplemented: Read as '0'

bit 0 SWDTEN: Software Controlled Watchdog Timer Enable bit

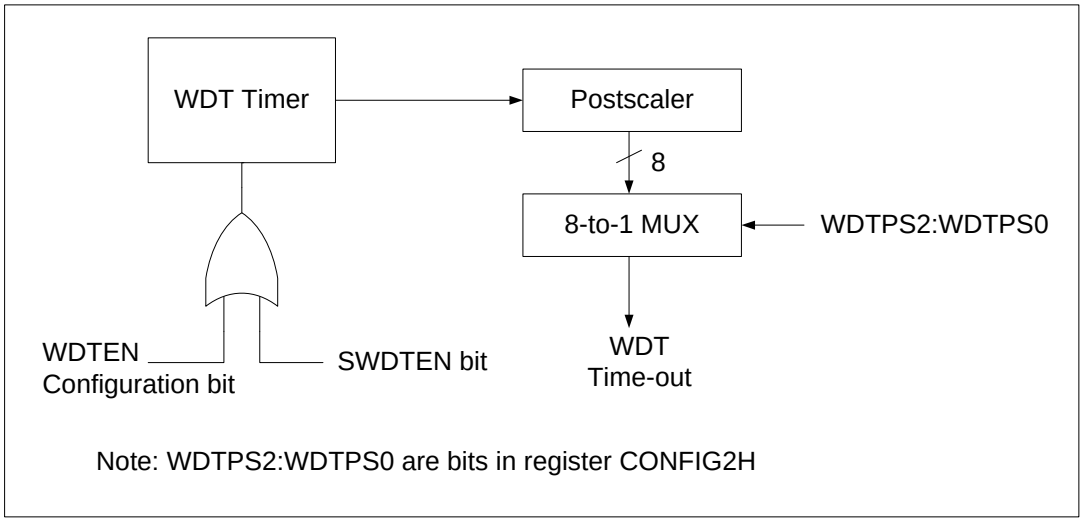
1 = Watchdog Timer is on

0 = Watchdog Timer is turned off if the WDTEN configuration bit in the Configuration register = 0

30.5.2 WDT POSTSCALER

The WDT has a postscaler that can extend the WDT Reset period. The postscaler is selected at the time of device programming by the value written to the WDT Configuration register.

FIGURE 30-1: WATCHDOG TIMER BLOCK DIAGRAM



30.5 START TIMER

Start Timer Prescaler to set the time of the reset delay .

Config Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Default
SUT0	300004						SUT0S2	SUT0S1	SUT0S0	1111 1111
SUT1	300005						SUT1S2	SUT1S1	SUT1S0	1111 1111
SUT2	300006						SUT2S2	SUT2S1	SUT2S0	1111 1111

30. 6 Signature Byte

Signature Byte Description	ADDRESS
Device ID 0, Manufacture ID	0x00
Device ID 1, Flash Size	0x02
Device ID 2, Device	0x04
FOSCCAL	0x06
SLOW RC Period L	0x07
SLOW RC Period H	0x08
SLOW RC Temp Prediction L	0x09
SLOW RC Temp Prediction H	0x0a
BGCCR Calibration Byte	0x0b
VPTAT CAL L	0x0c
VPTAT CAL H	0x0d
V-ADC Cell1 Gain Calibration Word L	0x0e
V-ADC Cell1 Gain Calibration Word H	0x0f
V-ADC Cell2 Gain Calibration Word L	0x10

V-ADC Cell2 Gain Calibration Word H	0x11
V-ADC Cell3 Gain Calibration Word L	0x12
V-ADC Cell3 Gain Calibration Word H	0x13
V-ADC Cell4 Gain Calibration Word L	0x14
V-ADC Cell4 Gain Calibration Word H	0x15
V-ADC Cell1 Offset	0x16
V-ADC Cell2 Offset	0x17
V-ADC Cell3 Offset	0x18
V-ADC Cell4 Offset	0x19
V-ADC0 Gain Calibration Word L	0x1a
V-ADC0 Gain Calibration Word H	0x1b
V-ADC1 Gain Calibration Word L	0x1c
V-ADC1 Gain Calibration Word H	0x1d
V-ADC2 Gain Calibration Word L	0x1e
V-ADC2 Gain Calibration Word H	0x1f
V-ADC3 Gain Calibration Word L	0x20
V-ADC3 Gain Calibration Word H	0x21
V-ADC4 Gain Calibration Word L	0x22
V-ADC4 Gain Calibration Word H	0x23
V-ADC ADC0 Offset	0x24
V-ADC ADC1 Offset	0x25
V-ADC ADC2 Offset	0x26
V-ADC ADC3 Offset	0x27
V-ADC ADC4 Offset	0x28
CC-ADC Instantaneous Mode 110mV Gain Calibration Word L	0x29
CC-ADC Instantaneous Mode 110mV Gain Calibration Word M	0x2a
CC-ADC Instantaneous Mode 110mV Gain Calibration Word H	0x2b
CC-ADC Instantaneous Mode 220mV Gain Calibration Word L	0x2c
CC-ADC Instantaneous Mode 220mV Gain Calibration Word M	0x2d
CC-ADC Instantaneous Mode 220mV Gain Calibration Word H	0x2e
CC-ADC Accumulate Mode 110mV Gain Calibration Word L	0x2f
CC-ADC Accumulate Mode 110mV Gain Calibration Word M	0x30
CC-ADC Accumulate Mode 110mV Gain Calibration Word H	0x31
CC-ADC Accumulate Mode 220mV Gain Calibration Word L	0x32
CC-ADC Accumulate Mode 220mV Gain Calibration Word M	0x33
CC-ADC Accumulate Mode 220mV Gain Calibration Word H	0x34
CC-ADC Instantaneous Mode 110mV Offset L	0x35
CC-ADC Instantaneous Mode 110mV Offset H	0x36
CC-ADC Instantaneous Mode 220mV Offset L	0x37
CC-ADC Instantaneous Mode 220mV Offset H	0x38
CC-ADC Accumulate Mode 110mV Offset L	0x39
CC-ADC Accumulate Mode 110mV Offset H	0x3a
CC-ADC Accumulate Mode 220mV Offset L	0x3b

CC-ADC Accumulate Mode 220mV Offset H	0x3c
THOT	0x3d
Reserves	0x3e
Reserves	0x3f
BGCCR	0x40
BGCRR	0x41
CHPOSCA	0x42
OCVGSS	0x43
ODVGSS	0x44
OPCVGSS	0x45
FOSCCAL	0x46
ULPCS	0x47
SIGNATURE PROTECTION	0x48
OSCSAD	0x49

31. Operating Circuit

Figure 31-1. Operating Circuit Diagram

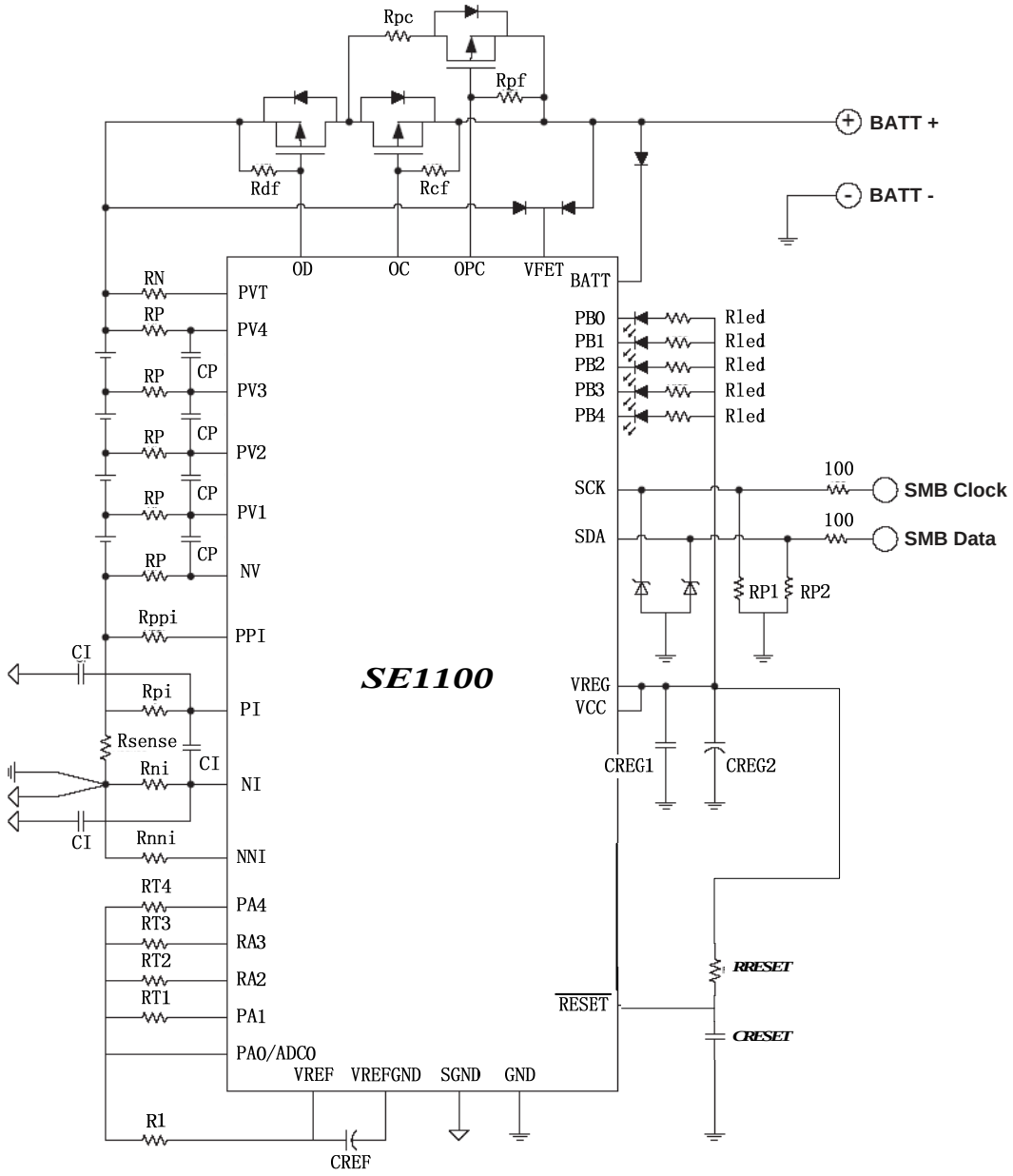


Table 31-1. Recommended values for external devices

Symbol	Use	Parameter	Min	Typ	Max	unit
R1	Pull-up resistor for thermistors	R		10		kΩ
RT1 RT2 RT3 RT4	NTC Thermistors	R@25℃		10		kΩ
		B-constant	3000		4000	K
R_S	Source Impedance when using PA0...4 as V-ADC inputs	R	0	3	7	kΩ
		Worst Gain-error due to R_S	0	1	2	%
R_{nni} R_{ppi}	Current protection LP-filter resistor	R		1		kΩ
R_{ni} R_{pi}	Current sense LP-filter resistor	R	10	100	500	Ω
C_i	Current sense LP-filter capacitor	C	0.01	0.1	0.4	uF
$(R_{ni}-R_{pi})*C_i$	Current sense LP-filter time constant	t		10	20	us
R_{sense} (1)	Current sense resistor	R		5		mΩ
RP	Cell input LP-filter resistor	R	10	500	1000	Ω
CP	Cell input LP-filter capacitor	C	0.01	0.1	0.5	uF
RP*CP	Cell input LP-filter time constant	t	6.5	25	100	us
RN	Pull-up resistor	R	0	10	TBD	Ω
R_{df} R_{cf} R_{Pf}	Pull-up resistors	R		1		mΩ
R_{pc} (2)	Pre-charge resistor	R		1		kΩ
CREF	VREF decoupling	C	1	1	22	uF
CREG1	VREG charge-storage	C		0.1		uF
CREG2				1		
RP1 RP2	TWI Pull-up resistors	C		2.2		MΩ
CRESET		C		0.1		uF
RRESET		R		1		kΩ

Note:

1. The sense resistor should be adjusted to the current flow for the application.
2. The pre-charger resistor should be adjusted to the pre-charger curret flow for the application.

32. Electrical Characteristics

32.1 Absolute Maximum Ratings*

Operating Temperature	-30°C to +85°C
Storage Temperature	-65°C to +150°C
Voltage on PA0 - PA7, PB0 - PB7, PD0 - PD1, VCC, PI, PPI, NI, NNI, XTAL1, and XTAL2 with respect to Ground	-0.5V to V_{REG} +0.5V
Voltage on SCL, SDA, NV, PV1 and $\overline{RESET}$ with respect to Ground	-0.5V to + 6.0V
Voltage on PVT and VFET with respect to Ground	-0.5V to + 35V
Voltage on PC0, OPC, OC and BATT with respect to Ground	-0.5V to VFET + 0.5V
Voltage on OD, PV2 - PV4 with respect to Ground	-0.5V to PVT + 0.5V
Maximum Operating Voltage	25V

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

32.2 DC Characteristics

DC Characteristics, TA = -30°C to 85°C, VCC = 3.3V

	Parameter	Condition	Min	Typ	Max	Unit
Supply Current	Power Supply Current	Active		2.2		mA
		Idle		540		µA
		ADC Noise Reduction		200		µA
		Power-save		200		µA
		Power-off		1		µA
Voltage Regulator(1)	Regulated Output Voltage(2) IOUT	IOUT = 5 mA	3.25	3.3	3.35	V
	Temperature Stability(2) IOUT = mA TA	IOUT = 5 mA TA = 0 - 60 °C		± 5	± 15	mV
		IOUT = 5 mA TA = -30 – 85 °C		± 20	± 70	mV
	Load Regulation	0.1 mA < IOUT < mA		± 20	± 60	mV
	Line Regulation	4V < VFET < 25V, IOUT = 1 mA		± 2	± 10	mV
VREF	Reference voltage			1.1		V
	Ref. Voltage Accuracy	After calibration, at calibration temperature		±0.1	±0.2	%

DC Characteristics, TA = -30°C to 85°C, VCC = 3.3V (Continued)

	Parameter	Condition	Min	Typ	Max	Unit
V-ADC	Reference Voltage			1.100		V
	Conversion Time			519		µs
	Effective Resolution			12		Bits
	1 LSB Un-scaled Inputs			269		µV
	1 LSB Scaled Inputs (x 0.2)			1.34		mV
	INL			± 1	± 2	LSB
	Input Voltage Range	ADC0, ADC1, ADC2, ADC3, VTEMP	0		1	V
		ADC4	0		5	V
		CELL1	2		5	V
		CELL2, PV1 ≥ 2V	0		5	V
		CELL3, PV1 ≥ 2V	0		5	V
		CELL4, PV1 ≥ 2V	0		5	V
	Offset			1.6		mV
	Gain Error Cell Inputs(6)			± 1	± 0.5	%
CC-ADC	Reference Voltage			±220		mV
	Conversion Time and Resolution	53.7 µV Resolution		3.9		ms
		1.68 µV Resolution	125		1000	ms
	INL	1000 ms conversion time			± 4	%
	CC-ADC Offset(7)	Uncompensated		± 50	± 100	%
	CC-ADC Offset Drift(5)	TA = 0 - 60°C		± 1	± 15	µV
	CC-ADC Gain Error				± 1	%
Temperature Sensor	VPTAT, Voltage Proportional to Absolute Temperature			0.6		mV/K
	Absolute Accuracy(3)				± 4	K
FET Driver	|VGS_ON|		11		15	V
	OC/OD Rise time (10 - 90%) (Switching OFF)	CL = 10 nF		10	15	µs
	OC/OD Fall time (VGS = 0 - VGS = -5V) (Switching ON)	CL = 10 nF			100	µs
	OPC Rise time (10 - 90%)(Switching OFF)	CL = 1 nF		100	500	µs
	OPC Fall time (VGS = 0 - VGS = -5V) Switching ON)	CL = 1 nF		100	500	µs
Ultra Low Power RC Oscillator	Frequency	TA = - 30 – 85°C		128		kHz
	Temperature Drift			± 2		%

DC Characteristics, TA = -30°C to 85°C, VCC = 3.3V (Continued)

	Parameter	Condition	Min	Typ	Max	Unit
Calibrated Fast RC Oscillator	Frequency	After factory calibration at TA=25°C	7.92	8	8.08	MHz
	Temperature Drift	With run-time calibration with OSI interface			3	%
Balancing Current	Battery cell Voltage VCELL = 4.2V,V-ADCfilter resistance =470Ω			4		mA
VSCD=VNNI-VPPI	Short Circuit Detection level accuracy	VSCD=20mV (min level)	14	20	26	mV
		VSCD=150mV	130	150	170	mV
		VSCD=310mV(max)	280	310	340	mV
VCOCD=VPPI-VNNI, VDOCD=VNNI-VPPI	Charge/Discharge Over Current Detection level accuracy	VCOCD,DOCD=20mV (min level)	15	20	25	mV
		VCOCD,DOCD=150mV	130	150	170	mV
		VCOCD,DOCD=310mV(max)	280	310	340	mV
VCHCD=VPPI-VNNI, VDHCD=VNNI-VPPI	Charge/Discharge High Current Detection level accuracy	VCHCD,DHCD=20mV (min level)	15	20	25	mV
		VCHCD,DHOCD=150mV	130	150	170	mV
		VCHCD,DHCD=310mV(max)	280	310	340	mV

Notes:

1. Voltage Regulator performance is based on 1 µF smooth capacitor.
2. After VREF calibration at second temperature. By default the first calibration is performed at 85 °C in Atmel factory test. The second calibration step can easily be implemented in a standard test flow at room temperature.
3. The measured VPTAT voltage must be scaled with the calibration value stored in the VPTAT Calibration Register to get the absolute temperature. The design target accuracy for this parameter assumes an exact calibration temperature. Actual accuracy of this parameter after calibration in Atmel factory test remains to be determined.
4. This value is not tested in production.
5. After system offset compensation in software.
6. After software gain error compensation.
7. This value should be measured at system level and stored in EEPROM for software offset compensation.

32.3 General I/O Lines characteristics

32.3.1 Low voltage ports

Figure 32-1. TA = -30°C to 85°C, VCC = 3.3V (unless otherwise noted) (1)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{IL}	Input Low Voltage	V_{CC} = 3.3V	-0.5		$0.3V_{CC}$(2)	V
V_{IH}	Input High Voltage	V_{CC} = 3.3V	$0.6V_{CC}$(3)		V_{CC} + 0.5	V
V_{OL}	Output Low Voltage(4)	I_{OL} = 5mA, V_{CC} = 3.3V			0.5	V
V_{OH}	Output High Voltage(5)	I_{OH} = 2 mA, V_{CC} = 3.3V	2.3			V
I_{IL}	Input Leakage Current I/O Pin	V_{CC} = 3.3V, pin low (absolute value)			1	µA
I_{IH}	Input Leakage Current I/O Pin	V_{CC} = 3.3V, pin high (absolute value)			1	µA

Notes:

1. Applicable for all except PC0.
2. “Max” means the highest value where the pin is guaranteed to be read as low.
3. “Min” means the lowest value where the pin is guaranteed to be read as high.
4. Although each I/O port can sink more than the test conditions (5 mA at VCC = 3.3V) under steady state conditions (non-transient, the following must be observed:
   - The sum of all IOL should not exceed 20 mA.
   If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test condition.
5. Although each I/O port can source more than the test conditions (2 mA at VCC = 3.3V) under steady state conditions (nontransient,the following must be observed:
   - The sum of all IOH should not exceed 2 mA.

32.3.2 High voltage ports

Figure 32-2. TA = -30°C to 85°C, VCC = 3.3V (unless otherwise noted)

Symbol	Parameter	Condition	Min.	Typ.	Max.	Units
V_{OL}(1)	Output Low Voltage	V_{CC} = 3.3V			0.5	V
t_r(1)	Rise Time	V_{CC} = 3.3V			300	ns
t_{of}(1)	Output Fall Time from V_{IHmin} to V_{ILmax}	Cb < 400 pF(2)			200	ns

Notes:

1. Parameter characterized and not tested.
2. Cb = capacitance of one bus line in pF

32.4 2-wire Serial Interface Characteristics

Table 32-1 describes the requirements for devices connected to the Two-wire Serial Bus. The SE1100 Two-wire Serial Interface meets or exceeds these requirements under the noted conditions.

Timing symbols refer to Figure f-3.

Table 32-1. Two-wire Serial Bus Requirements

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage		-0.5	0.8	V
V_{IH}	Input High-voltage		2.1	5.5	V
V_{OL} (1)	Output Low-voltage	350 µA sink current	0	0.4	V
t_r (1)	Rise Time for both SDA and SCL			300	ns
t_{of} (1)	Output Fall Time from V_{IHmin} to V_{ILmax}	C_b < 400 pF (2)		250	ns
t_{SP} (1)	Spikes Suppressed by Input Filter		0	50	ns
I_i	Input Current each I/O Pin	$0.1V_{BUS} < V_i < 0.9V_{BUS}$	-5	5	µA
C_i (1)	Capacitance for each I/O Pin		–	10	pF
f_{SCL}	SCL Clock Frequency	f_{CK} (3) > max(16f_{SCL}, 450 kHz) (4)	0	100	kHz
Rp	Value of Pull-up resistor	$f_{SCL} \le 100$ kHz	$\frac{V_{BUS}-0,4V}{350\mu A}$	$\frac{V_{BUS}-0,4V}{100\mu A}$	Ω
$t_{HD;STA}$	Hold Time (repeated) START Condition	$f_{SCL} \le 100$ kHz	4.0	–	µs
t_{LOW}	Low Period of the SCL Clock	$f_{SCL} \le 100$ kHz	4.7	–	µs
t_{HIGH}	High period of the SCL clock	$f_{SCL} \le 100$ kHz	4.0	–	µs
$t_{SU;STA}$	Set-up time for a repeated START condition	$f_{SCL} \le 100$ kHz	4.7	–	µs
$t_{HD;DAT}$	Data hold time	$f_{SCL} \le 100$ kHz	0.3	3.45	µs
$t_{SU;DAT}$	Data setup time	$f_{SCL} \le 100$ kHz	250	–	ns
$t_{SU;STO}$	Setup time for STOP condition	$f_{SCL} \le 100$ kHz	4.0	–	µs
t_{BUF}	Bus free time between a STOP and START condition	$f_{SCL} \le 100$ kHz	4.7	–	µs

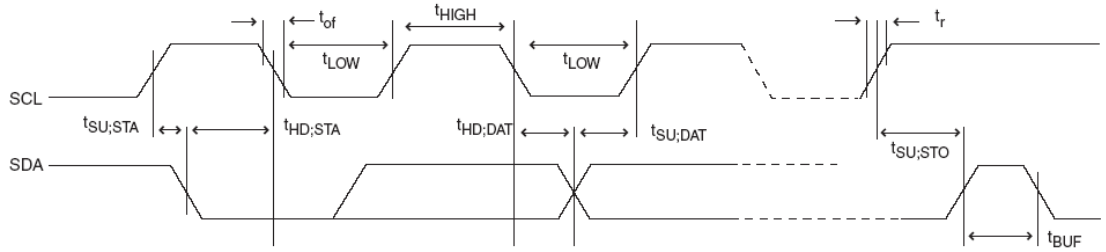
Notes: 1. In ATmega406, this parameter is characterized and not tested.

2. Cb = capacitance of one bus line in pF.

3. fCK = CPU clock frequency

4. This requirement applies to all SE1100 Two-wire Serial Interface operation. Other devices connected to the Two-wire Serial Bus need only obey the general fSCL requirement.

Figure 32-3. Two-wire Serial Bus Timing



32.5 Reset Characteristics

Table 32-4. TA= -40 to 85°C unless otherwise noted.

Symbol	Parameter	Condition	Min	Typ	Max	Unit
VPOT	Power-on Threshold Voltage (rising)(1)		4.5		7	V
	Power-on Threshold Voltage (falling)(1)(2)		4.5		6.3	V
tRST	Minimum pulse width on RESET Pin			900		ns
VBOT	Brown-Out Detection (BOD) Trigger Level	TA= 25°C		2.6		V
VHYST	BOD Level Hysteresis	TA= 25°C		50		mV

Notes:

1. The voltage at the Pack + terminal will be slightly higher than VPOT when the chip is enabled. This is because of an internal Pull-down current on the BATT pin in the range 50 - 150 uA and the RBATT resistor connected between the Pack + terminal and the BATT pin. RBATT = 1k gives a voltage drop 0.05 - 0.15V.
2. The power-on reset will not work unless the voltage has been below VPOT (falling) after a power-off condition.

Table 32-5. Characteristics for Powering-up the LDO(1)

Symbol(2)	Parameter	Min	Typ	Max	Unit
Charger Present					
VROT	Regulator Power-on Threshold	3.0		4.0	V
VCHT	Charge Voltage Threshold		1.0		V
No Charger Present					
VROT	Regulator Power-on Threshold	3.0		4.0	V
VPVIT	Voltage Threshold on Battery Cell 1		2.0		V

Notes:

1. Power-on Reset is issued when a charger is connected and the regulator has stable work conditions.
2. Values based on characterization.

Table 32-6. Power-on Reset Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Unit
VCOT	Charger-on Thresholt Voltage	Regulator must operate	6	7	8	V

Note: Internal Voltage Regulator must be on.

32.6 Voltage Regulator Characteristics

Table 32-4. TA= -40 to 85°C unless otherwise noted.

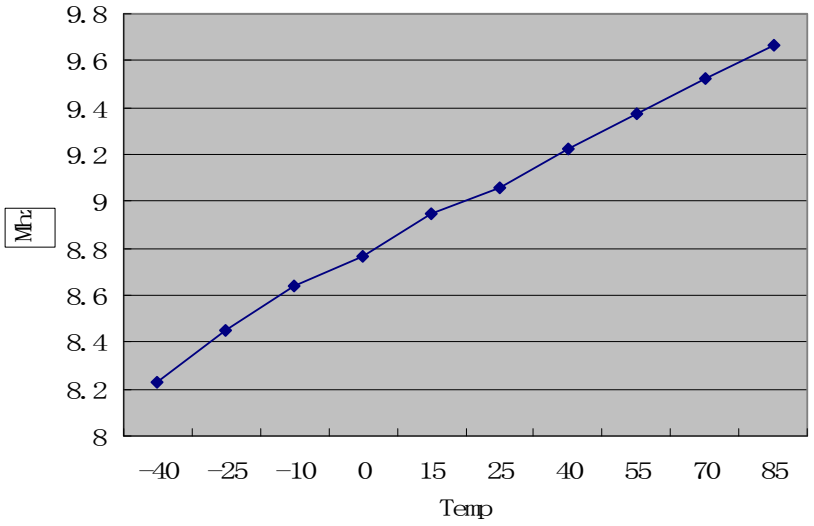
Symbol	Parameter	Condition	Min	Typ	Max	Unit
VVREG	Regulator Output Voltage	VFET=16.8V, IOUT=20mA	3.1			V
		VFET=6V, IOUT=20mA	3.1			
		VFET=4V, IOUT=7mA	3.1			
VRSCL	Voltage Regulator Short-circuit Level at VFET pin		3.3		3.7	
VBLOD	Voltage Regulator Black-out Detection Level at VREG pin	TA= 25°C		2.35		

33. Electrical Characteristics

33.1 Clock Characteristics

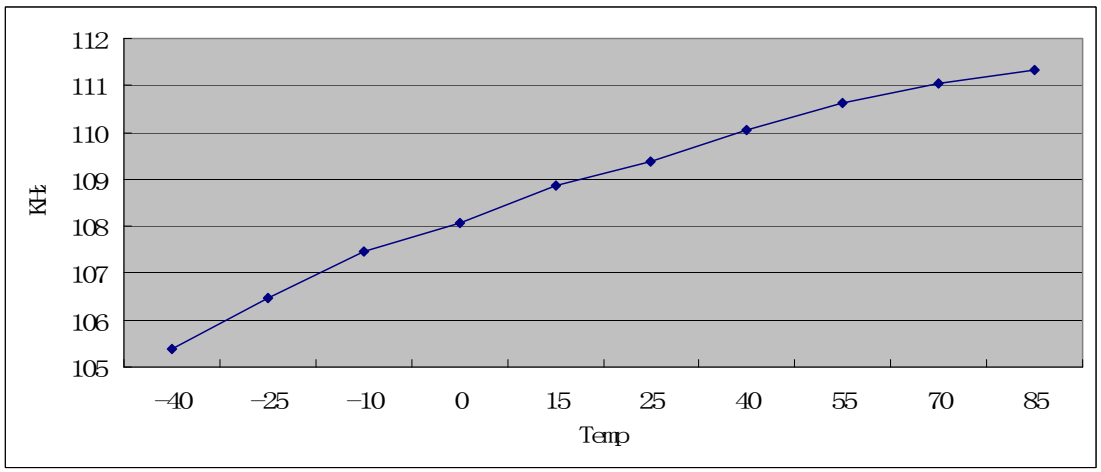
33.1.1 Fast RC Oscillator characteristics

Figure 33-1. Fast RC Oscillator Frequency vs. Temperature. (Simulation result, FOSCCAL=0x5e, WS corner)



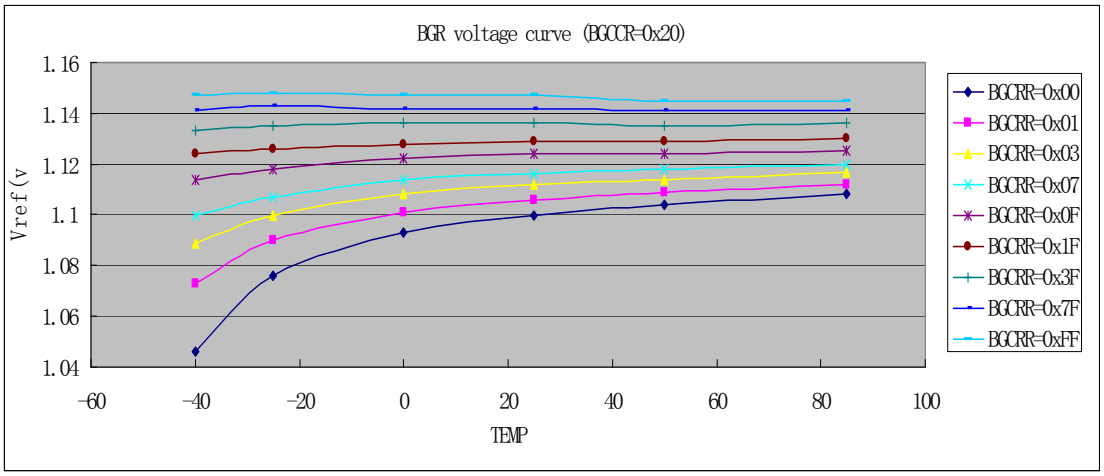
33.1.2 Slow RC Oscillator characteristics

Figure 33-2. Slow RC Oscillator Frequency vs. Temperature. (Simulation result, ULPCS=0x04,OSCCAD=0x06, WS corner)



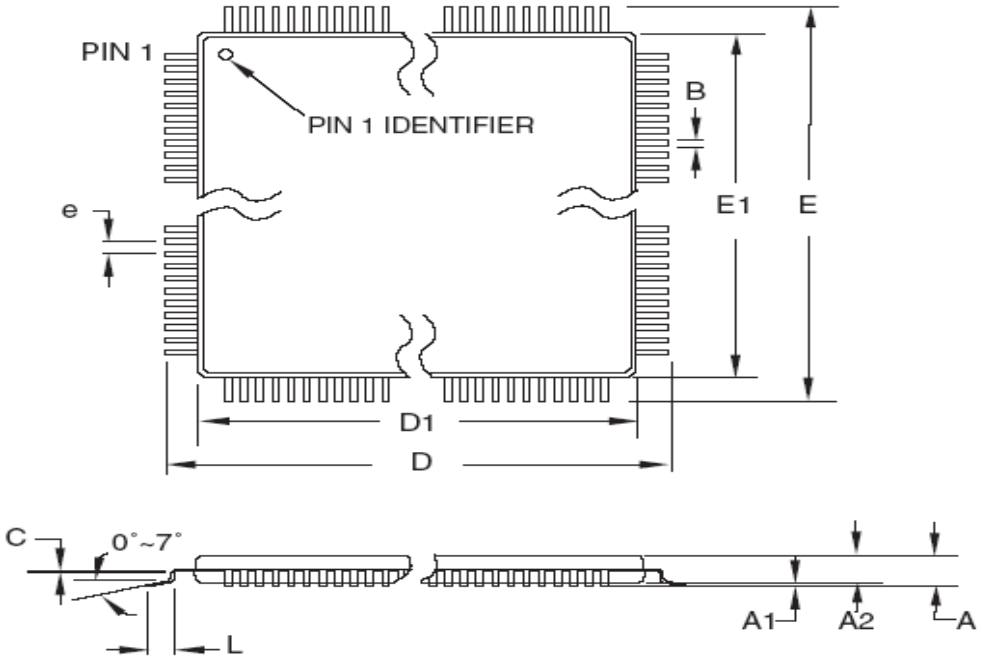
33.2 Voltage Reference characteristics

Figure 33-3. Typical VREF curve with factory calibration at 25℃ and 85℃



34. Packaging Information

34.1 48AA



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.60	
A1	0.05	–	0.15	
A2	1.35	1.40	1.45	
D	8.75	9.00	9.25	
D1	6.90	7.00	7.10	Note 2
E	8.75	9.00	9.25	
E1	6.90	7.00	7.10	Note 2
B	0.17	–	0.27	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.50 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation BBC.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.08 mm maximum.