PIC14000

28-Pin Programmable Mixed Signal Controller

High-Performance RISC CPU:

- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
- 4096 x 14 on-chip EPROM program memory
- 192 x 8 general purpose registers (SRAM)
- 6 internal and 5 external interrupt sources
- 38 special function hardware registers
- Eight-level hardware stack

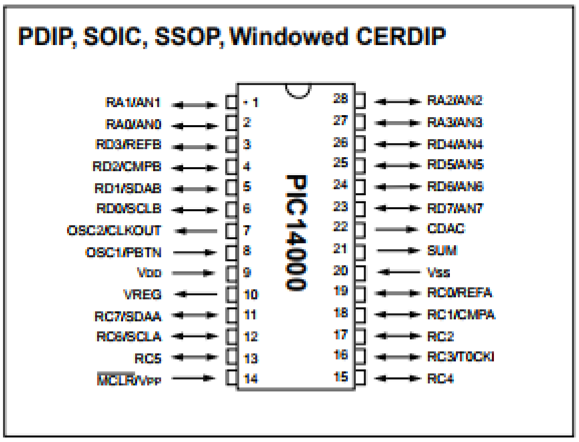
Analog Peripherals Features:

- Slope Analog-to-Digital (A/D) converter
  - Eight external input channels including two channels with selectable level shift inputs
  - Six internal input channels
  - 16-bit programmable timer with capture register
  - 16 ms maximum conversion time at maximum (16-bit) resolution and 4 MHz clock
  - 4-bit programmable current source
- Internal bandgap voltage reference
- Factory calibrated with calibration constants stored in EPROM
- On-chip temperature sensor
- Voltage regulator control output
- Two comparators with programmable references
- On-chip low voltage detector

Special Microcontroller Features:

- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Multi-segment programmable code-protection
- Selectable oscillator options
  - Internal 4 MHz oscillator
  - External crystal oscillator
- Serial in-system programming (via two pins)

Pin Diagram



Digital Peripherals Features:

- 22 I/O pins with individual direction control
- High current sink/source for direct LED drive
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler
- 16-bit A/D timer: can be used as a general purpose timer
- I^{2}C™ serial port compatible with System Management Bus

CMOS Technology:

- Low-power, high-speed CMOS EPROM technology
- Fully static design
- Wide-operating voltage range (2.7V to 6.0V)
- Commercial and Industrial Temperature Range
- Low power dissipation (typical)
  - < 3 mA @5V, 4 MHz operating mode
  - < 300 µA @3V (Sleep mode: clocks stopped with analog circuits active)
  - < 5 µA @3V (Hibernate mode: clocks stopped, analog inactive, and WDT disabled)

Applications:

- Battery Chargers
- Battery Capacity Monitoring
- Uninterruptable Power Supply Controllers
- Power Management Controllers
- HVAC Controllers
- Sensing and Data Acquisition

1.0 GENERAL DESCRIPTION

The PIC14000 features include medium to high resolution A/D conversion (10 to 16 bits), temperature sensing, closed loop charge control, serial communication, and low power operation.

The PIC14000 uses a RISC Harvard architecture CPU with separate 14-bit instruction and 8-bit data buses. A two-stage instruction pipeline allows all instructions to execute in a single cycle, except for program branches, which require two cycles. A total of 35 instructions are available. Additionally, a large register set is included.

PIC16/17 microcontrollers typically achieve a 2:1 code compression and a 4:1 speed improvement over other 8-bit microcontrollers.

Features:

The PIC14000 is a 28-pin device with these features:

- 4K of EPROM
- 192 bytes of RAM
- 22 I/O pins

The analog peripherals include:

- 8 external analog input channels, two with level shift inputs
- 6 internal analog input channels
- 2 comparators with programmable references
- A bandgap reference
- An internal temperature sensor
- A programmable current source

In addition, the I^{2}C serial port through a multiplexer supports two separate I^{2}C channels.

A special oscillator option allows either an internal 4 MHz oscillator or an external crystal oscillator. Using the internal 4 MHz oscillator requires no external components.

The PIC14000 contains three timers, the Watchdog Timer (WDT), Timer0 (TMR0), and A/D Timer (ADTMR). The Watchdog Timer includes its own on-chip RC oscillator providing protection against software lock-up. TMR0 is a general purpose 8-bit timer/counter with an 8-bit prescaler. It may be clocked externally using the RC3/T0CKI pin. The ADTMR is intended for use with the slope A/D converter, but can also be used as a general purpose timer. It has an associated capture register which can be used to measure the time between events.

An internal low-voltage detect circuit allows for tracking of voltage levels. Upon detecting the low voltage condition, the PIC14000 can be instructed to save its operating state then enter an idle state.

The internal band-gap reference is used for calibrating the measurements of the analog peripherals. The calibration factors are stored in EPROM and can be used to achieve high measurement accuracy.

Power savings modes are available for portable applications. The SLEEP and HIBERNATE modes offer different levels of power savings. The PIC14000 can wake up from these modes through interrupts or reset.

A UV erasable CERDIP packaged version is ideal for code development, while the cost-effective One-Time Programmable (OTP) version is suitable for production in any volume.

The PIC14000 fits perfectly in applications for battery charging, capacity monitoring, and data logging. The EPROM technology makes customization of application programs (battery characteristics, feature sets, etc.) extremely fast and convenient. The small footprint packages make this microcontroller based mixed signal device perfect for all applications with space limitations. Low-cost, low-power, high performance, ease of use and I/O flexibility make the PIC14000 very versatile in other applications such as temperature monitors/controllers.

1.1 Family and Upward Compatibility

Code written for PIC16C6X/7X can be easily ported to the PIC14000 (see Appendix A).

1.2 Development Support

The PIC14000 is supported by a full-featured macro assembler, a software simulator, an in-circuit emulator, a low-cost development programmer and a full-featured programmer. A “C” compiler and fuzzy logic support tools are also available.

2.0 DEVICE VARIETIES

A variety of frequency ranges and packaging options are available. The PIC14000 Product Selection System section at the end of this data sheet provides the devices options to be selected for your specific application and production requirements. When placing orders, please use the "PIC14000 Product Identification System" at the back of this data sheet to specify the correct part number.

2.1 UV Erasable Devices

The UV erasable version, offered in CERDIP package, is optimal for prototype development and pilot programs.

The UV erasable version can be erased and reprogrammed to any of the configuration modes.

> **Note:** Please note that erasing the device will also erase the pre-programmed calibration factors. Please refer to AN621 for more information.

Microchip's PICSTART®, PICSTART-PLUS and PRO MATE™ programmers all support programming of the PIC14000. Third party programmers also are available; refer to the *Microchip Third Party Guide* for a list of sources.

2.2 One-Time-Programmable (OTP) Devices

The availability of OTP devices is especially useful for customers who need the flexibility for frequent code updates or small volume applications.

The OTP devices, packaged in plastic packages permit the user to program them once. In addition to the program memory, the configuration bits must also be programmed.

2.3 Quick-Turnaround-Production (QTP) Devices

Microchip offers a QTP Programming Service for factory production orders. This service is made available for users who choose not to program a medium to high quantity of units and whose code patterns have stabilized. The devices are identical to the OTP devices but with all EPROM locations and fuse options already programmed by the factory. Certain code and prototype verification procedures do apply before production shipments are available. Please contact your local Microchip Technology sales office for more details.

2.4 Serialized Quick-Turnaround Production (SQTP^{SM}) Devices

Microchip offers a unique programming service where a few user-defined locations in each device are programmed with different serial numbers. The serial numbers may be random, pseudo-random or sequential.

Serial programming allows each device to have a unique number which can serve as an entry-code, password or ID number.

3.0 ARCHITECTURAL OVERVIEW

The PIC14000 addresses 4K x 14 program memory. All program memory is internal. The PIC14000 can directly or indirectly address its register files or data memory. All special function registers including the program counter are mapped in the data memory. The PIC14000 has an orthogonal instruction set that makes it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of 'special optimal situations' make programming with the PIC14000 simple yet efficient. In addition, the learning curve is reduced significantly.

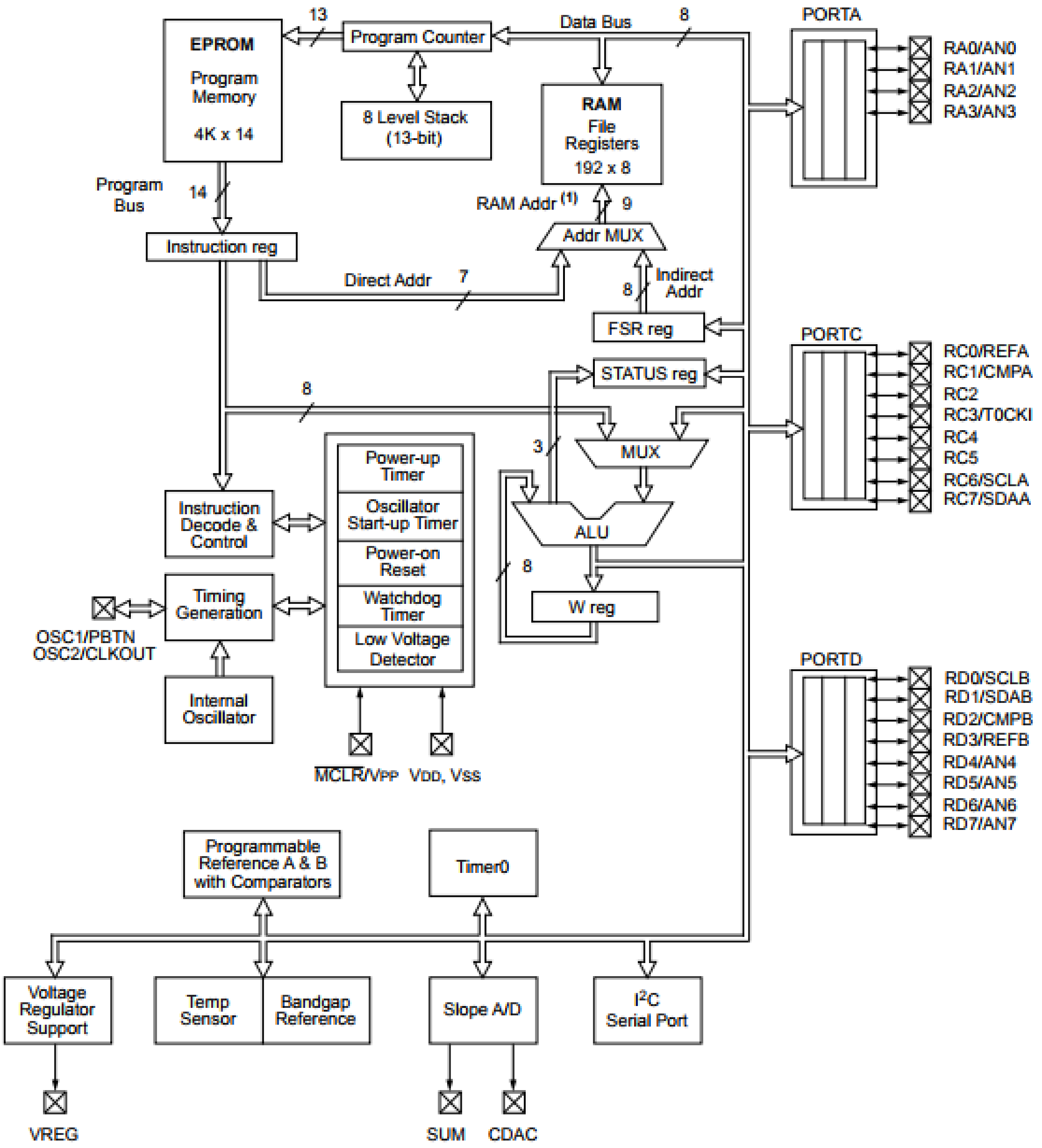
The PIC14000 contains an 8-bit ALU and working register. The ALU performs arithmetic and Boolean functions between data in the working register and any register file.

The ALU is capable of addition, subtraction, shift, and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement. In two-operand instructions, typically one operand is the working register (W register). The other operand is a file register or an immediate constant. In single operand instructions, the operand is either the W register or a file register.

Depending on the instruction executed, the ALU may affect the values of the Carry (C), Digit Carry (DC), and Zero (Z) bits in the STATUS register. The C and DC bits operate as a $\overline{\text{borrow}}$ bit and a $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

A simplified block diagram for the PIC14000 is shown in Figure 3-1, its corresponding pin description is shown in Table 3-1.

FIGURE 3-1: PIC14000 BLOCK DIAGRAM



Note 1: Higher order bits are from the STATUS register.

TABLE 3-1: PIN DESCRIPTIONS

Pin Name	Pin No.	I/O	Pin Type		Description
			Input	Output	
CDAC	22	O	—	AN	A/D ramp current source output. Normally connected to external capacitor to generate a linear voltage ramp.
RA0/AN0	2	I/O	AN/ST	CMOS	Analog input channel 0. This pin can also serve as a general-purpose I/O.
RA1/AN1	1	I/O	AN/ST	CMOS	Analog input channel 1. This pin can connect to a level shift network. If enabled, a +0.5V offset is added to the input voltage. This pin can also serve as a general-purpose I/O.
RA2/AN2	28	I/O	AN/ST	CMOS	Analog input channel 2. This pin can also serve as a general purpose digital I/O.
RA3/AN3	27	I/O	AN/ST	CMOS	Analog input channel 3. This pin can also serve as a general purpose digital I/O.
SUM	21	O	—	AN	AN1 summing junction output. This pin can be connected to an external capacitor for averaging small duration pulses.
RC0/REFA	19	I/O-PU	ST	CMOS	LED direct-drive output or programmable reference A output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to VDD.
RC1/CMPA	18	I/O-PU	ST	CMOS	LED direct-drive output or comparator A output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to VDD.
RC2	17	I/O-PU	ST	CMOS	LED direct-drive output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to VDD
RC3/T0CKI	16	I/O-PU	ST	CMOS	LED direct-drive output. This pin can also serve as a GPIO, or an external clock input for Timer0. If enabled, this pin has a weak internal pull-up to VDD.
RC4	15	I/O-PU	ST	CMOS	LED direct-drive output. This pin can also serve as a GPIO. If enabled, a change on this pin can cause a CPU interrupt. If enabled, this pin has a weak internal pull-up to VDD.
RC5	13	I/O-PU	ST	CMOS	LED direct-drive output. This pin can also serve as a GPIO. If enabled, a change on this pin can cause a CPU interrupt. If enabled, this pin has a weak internal pull-up to VDD.
RC6/SCLA	12	I/O	ST/SM	NPU/OD (No P-diode)	General purpose I/O. If enabled, is multiplexed as synchronous serial clock for I^2C interface. Also is the serial programming clock. If enabled, a change on this pin can cause a CPU interrupt. This pin has an N-channel pull-up device which is disabled in I^2C mode.
RC7/SDAA	11	I/O	ST/SM	NPU/OD (No P-diode)	General purpose I/O. If enabled, is multiplexed as synchronous serial data I/O for I^2C interface. Also is the serial programming data line. If enabled, a change on this pin can cause a CPU interrupt. This pin has an N-channel pull-up device which is disabled in I^2C mode.
RD0/SCLB	6	I/O	ST/SM	NPU/OD (No P-diode)	General purpose I/O. If enabled, is multiplexed as synchronous serial clock for I^2C interface. This pin has an N-channel pull-up device which is disabled in I^2C mode.
RD1/SDAB	5	I/O	ST/SM	NPU/OD (No P-diode)	General purpose I/O. If enabled, is multiplexed as synchronous serial data I/O for I^2C interface. This pin has an N-channel pull-up device which is disabled in I^2C mode.
RD2/CMPB	4	I/O-PU	AN/ST	CMOS	General purpose I/O or comparator B output.

TABLE 3-1: PIN DESCRIPTIONS (CONTINUED)

Pin Name	Pin No.	I/O	Pin Type		Description
			Input	Output	
RD3/REFB	3	I/O-PU	AN/ST	CMOS	General purpose I/O or programmable reference B output.
RD4/AN4	26	I/O	AN/ST	CMOS	Analog input channel 4. This pin can also serve as a GPIO.
RD5/AN5	25	I/O	AN/ST	CMOS	Analog input channel 5. This pin can connect to a level shift network. If enabled, a +0.5V offset is added to the input voltage. This pin can also serve as a GPIO.
RD6/AN6	24	I/O	AN/ST	CMOS	Analog input channel 6. This pin can also serve as a GPIO.
RD7/AN7	23	I/O	AN/ST	CMOS	Analog input channel 7. This pin can also serve as a GPIO.
VREG	10	O	—	AN	This pin is an output to control the gate of an external N-FET for voltage regulation.
OSC1/PBTN	8	I-PU	ST	—	IN Mode: Input with weak pull-up resistor, can be used to generate an interrupt. HS Mode: External oscillator input.
OSC2/ CLKOUT	7	O	—	CMOS	IN Mode: General purpose output. HS Mode: External oscillator/clock output.
$\overline{\text{MCLR}}$/VPP	14	I/PWR	ST		Master clear (reset) input / programming voltage input. This pin is an active low reset to the device.
VDD	9	PWR			Positive supply connection
VSS	20	GND			Return supply connection

Legend:

Type:	**Definition:**
TTL	TTL-compatible input
CMOS	CMOS-compatible input or output
ST	Schmitt Trigger input, with CMOS levels
SM	SMBus compatible input
OD	Open-drain output. An external pull-up resistor is required if this pin is used as an output.
NPU	N-channel pull-up. This pin will pull-up to approximately VDD - 1.0V when outputting a logical '1'.
PU	Weak internal pull-up (10K-50K ohms)
No-P diode	No P-diode to VDD. This pin may be pulled above the supply rail (to 6.0V maximum).
AN	Analog input or output

3.1 Clocking Scheme/Instruction Cycle

The clock input (from OSC1 or the internal oscillator) is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. 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 3-2.

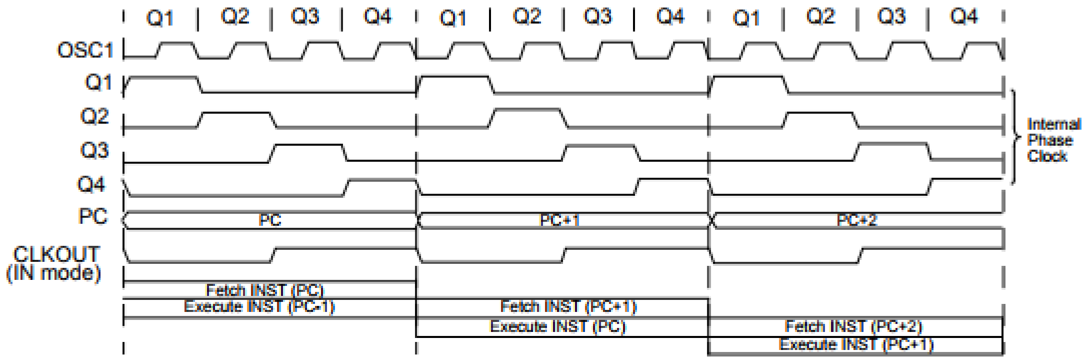
3.2 Instruction Flow/Pipelining

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 takes 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) then two cycles are required to complete the instruction (Example 3-1).

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).

FIGURE 3-2: CLOCK/INSTRUCTION CYCLE



EXAMPLE 3-1: INSTRUCTION PIPELINE FLOW

1. MOVLW 55h	Fetch 1	Execute 1			
2. MOVWF PORTB		Fetch 2	Execute 2		
3. CALL SUB_1			Fetch 3	Execute 3	
4. BSF PORTA, BIT3				Fetch 4	Flush
					Fetch SUB_1

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

4.0 MEMORY ORGANIZATION

4.1 Program Memory Organization

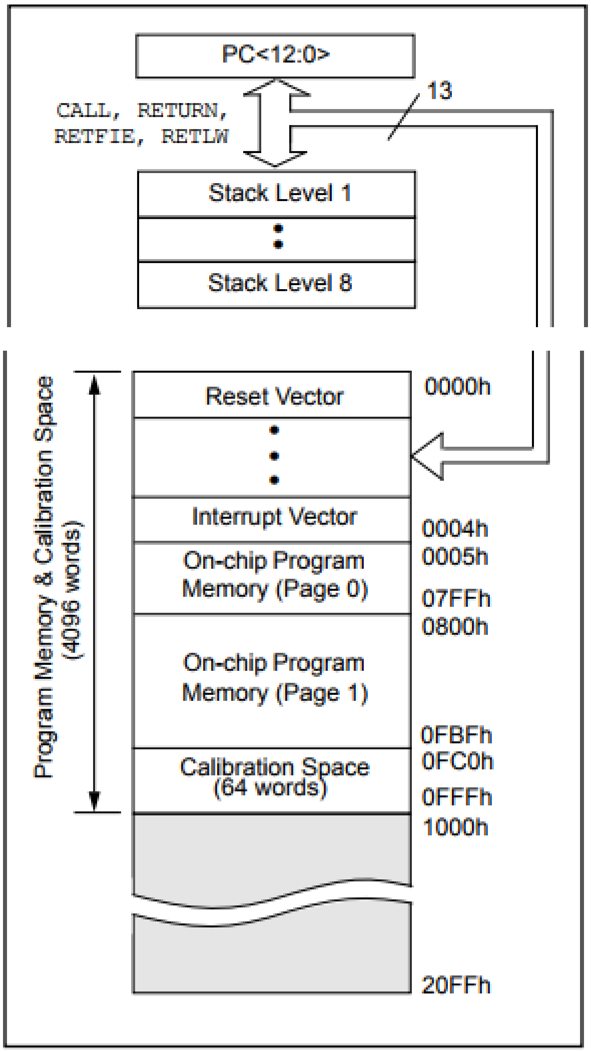
The PIC14000 has a 13-bit program counter capable of addressing an 8K x 14 program memory space. Only the first 4K x 14 (0000-0FFFh) are physically implemented. Accessing a location above the physically implemented address will cause a wraparound. The reset vector is at 0000h and the interrupt vector is at 0004h (Figure 4-1).

The 4096 words of Program Memory space are divided into:

- Address Vectors (addr 0000h-0004h)
- Program Memory Page 0 (addr 0005h-07FFH)
- Program Memory Page 1 (addr 0800h-0FBFh)
- Calibration Space (64 words, addr 0FC0h-0FFFh)

Program code may reside in Page 0 and Page 1.

FIGURE 4-1: PIC14000 PROGRAM MEMORY MAP AND STACK



4.1.1 CALIBRATION SPACE

The calibration space is not used for instructions. This section stores constants and factors for the arithmetic calculations to calibrate the analog measurements.

TABLE 4-1: CALIBRATION DATA OVERVIEW*

Address	Parameter	Symbol	Units	Format
0FC0h-0FC3h	Slope reference ratio	K_{REF}	N/A	32-bit floating point**
0FC4h-0FC7h	Bandgap reference voltage	K_{BG}	Volts	32-bit floating point
0FC8h-0FCBh	Temperature sensor voltage	V_{THERM}	Volts	32-bit floating point
0FCCh-0FCFh	Temperature sensor coefficient	K_{TC}	Volts/ degree Celsius	32-bit floating point
0FD0h	Internal oscillator frequency multiplier	F_{OSC}	N/A	byte
0FD2h	WDT time-out	T_{WDT}	ms	byte

* Refer to AN621 for details.

** Microchip modified IEEE754 32-bit floating point format. Refer to application note AN575 for details.

TABLE 4-2: CALIBRATION CONSTANT ADDRESSES

Address	Data
0FC0h	K_{REF} , exponent
0FC1h	K_{REF} , mantissa high byte
0FC2h	K_{REF} , mantissa middle byte
0FC3h	K_{REF} , mantissa low byte
0FC4h	K_{BG} , exponent
0FC5h	K_{BG} , mantissa high byte
0FC6h	K_{BG} , mantissa middle byte
0FC7h	K_{BG} , mantissa low byte
0FC8h	V_{THERM} , exponent
0FC9h	V_{THERM} , mantissa high byte
0FCAh	V_{THERM} , mantissa middle byte
0FCBh	V_{THERM} , mantissa low byte
0FCCh	K_{TC} , exponent
0FCDh	K_{TC} , mantissa high byte
0FCEh	K_{TC} , mantissa middle byte
0FCFh	K_{TC} , mantissa low byte
0FD0h	F_{OSC}, unsigned byte
0FD1h	reserved
0FD2h	T_{WDT}, unsigned byte
0FD3h - 0FF8h	reserved
0FF9h-Fh	calibration space checksums

4.2 Data Memory Organization

The data memory (Figure 4-2) is partitioned into two banks which contain the general purpose registers and the special function registers. Bank 0 is selected when the RP0 bit in the STATUS register is cleared. Bank 1 is selected when the RP0 bit in the STATUS register is set. Each bank extends up to 7Fh (128 bytes). The first 32 locations of each bank are reserved for the Special Function Registers. Several Special Function Registers are mapped in both Bank 0 and Bank 1. The general purpose registers, implemented as static RAM, are located from address 20h through 7Fh, and A0 through FF.

4.2.1 GENERAL PURPOSE REGISTER FILE

The register file is accessed either directly, or indirectly through the file select register FSR (Section 4.4).

FIGURE 4-2: REGISTER FILE MAP

File Address

00h	Indirect add.(*)	Indirect addr.(*)	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	RESERVED	RESERVED	86h
07h	PORTC	TRISC	87h
08h	PORTD	TRISD	88h
09h			89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	PIR1	PIE1	8Ch
0Dh			8Dh
0Eh	ADTMRL	PCON	8Eh
0Fh	ADTMRH	SLPCON	8Fh
10h			90h
11h			91h
12h			92h
13h	I^{2}CBUF	I^{2}CADD	93h
14h	I^{2}CCON	I^{2}CSTAT	94h
15h	ADCAPL		95h
16h	ADCAPH		96h
17h			97h
18h			98h
19h			99h
1Ah			9Ah
1Bh		PREFA	9Bh
1Ch		PREFB	9Ch
1Dh		CMCON	9Dh
1Eh		MISC	9Eh
1Fh	ADCON0	ADCON1	9Fh
20h 7F	General Purpose Register (96 Bytes)	General Purpose Register (96 Bytes)	A0h FF

* Not a physical register.
Shaded areas are unimplemented memory locations, read as '0's.

4.2.2 SPECIAL FUNCTION REGISTERS

The special function registers are registers used by the CPU and peripheral functions for controlling the desired operation of the device (Table 4-3). These registers are static RAM.

The special registers are classified into two sets. Special registers associated with the "core" functions are described in this section. Those registers related to the operation of the peripheral features are described in the section specific to that peripheral.

TABLE 4-3: SPECIAL FUNCTION REGISTERS FOR THE PIC14000

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank0									
00h*	INDF (Indirect Address)	Addressing this location uses contents of the FSR to address data memory (not a physical register).							
01h	TMR0	Timer0 data							
02h*	PCL	Program Counter's (PC's) least significant byte							
03h*	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
04h*	FSR	Indirect data memory address pointer							
05h	PORTA	PORTA data latch.							
06h	Reserved	Reserved for emulation.							
07h	PORTC	PORTC data latch							
08h	PORTD	PORTD data latch							
09h	Reserved								
0Ah*	PCLATH	Buffered register for the upper 5 bits of the Program Counter (PC)							
0Bh*	INTCON	GIE	PEIE	T0IE	r	r	T0IF	r	r
0Ch	PIR1	CMIF	—	—	PBIF	I^{2}CIF	RCIF	ADCIF	OVFIF
0Dh	Reserved								
0Eh	ADTMRL	A/D capture timer data least significant byte							
0Fh	ADTMRH	A/D capture timer data most significant byte							
10h	Reserved								
11h	Reserved								
12h	Reserved								
13h	I^{2}CBUF	I^{2}C Serial Port Receive Buffer/Transmit Register							
14h	I^{2}CCON	WCOL	I^{2}COV	I^{2}CEN	CKP	I^{2}CM3	I^{2}CM2	I^{2}CM1	I^{2}CM0
15h	ADCAPL	A/D capture latch least significant byte							
16h	ADCAPH	A/D capture latch most significant byte							
17h	Reserved								
18h	Reserved								
19h	Reserved								
1Ah	Reserved								
1Bh	Reserved								
1Ch	Reserved								
1Dh	Reserved								
1Eh	Reserved								
1Fh	ADCON0	ADCS3	ADCS2	ADCS1	ADCS0	—	AMUXOE	ADRST	ADZERO

Legend
— = unimplemented bits, read as '0' but cannot be overwritten
r = reserved bits, default is POR value and **should not be overwritten with any value**
Reserved indicates reserved register and **should not be overwritten with any value**
* indicates registers that can be addressed from either bank

TABLE 4-3: SPECIAL FUNCTION REGISTERS FOR THE PIC14000 (CONTINUED)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Bank1									
80h*	INDF (Indirect Address)	Addressing this location uses contents of FSR to address data memory (not a physical register).							
81h	OPTION	$\overline{RCPU}$	r	TOCS	TOSE	PSA	PS2	PS1	PS0
82h*	PCL	Program Counter's (PC's) least significant byte							
83h*	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
84h*	FSR	Indirect data memory address pointer							
85h	TRISA	PORTA Data Direction Register							
86h	Reserved	Reserved for emulation							
87h	TRISC	PORTC Data Direction Register							
88h	TRISD	PORTD Data Direction Register							
89h	Reserved								
8Ah*	PCLATH	Buffered register for the upper 5 bits of the Program Counter (PC)							
8Bh*	INTCON	GIE	PEIE	T0IE	r	r	T0IF	r	r
8Ch	PIE1	CMIE	—	—	PBIE	I^2CIE	RCIE	ADCIE	OVFIE
8Dh	Reserved								
8Eh	PCON	r	—	—	—	—	—	$\overline{POR}$	$\overline{LVD}$
8Fh	SLPCON	HIBEN	—	REFOFF	LSOFF	OSCOFF	CMOFF	TEMPOFF	ADOFF
90h	Reserved								
91h	Reserved								
92h	Reserved								
93h	I^2CADD	I^2C Synchronous Serial Port Address Register							
94h	I^2CSTAT	—	—	D/$\overline{A}$	P	S	R/$\overline{W}$	UA	BF
95h	Reserved								
96h	Reserved								
97h	Reserved								
98h	Reserved								
99h	Reserved								
9Ah	Reserved								
9Bh	PREFA	PRA7	PRA6	PRA5	PRA4	PRA3	PRA2	PRA1	PRA0
9Ch	PREFB	PRB7	PRB6	PRB5	PRB4	PRB3	PRB2	PRB1	PRB0
9Dh	CMCON	—	CMBOUT	CMBOE	CPOLB	—	CMAOUT	CMAOE	CPOLA
9Eh	MISC	SMHOG	SPGNDB	SPGNDA	I^2CSEL	SMBUS	INCLKEN	OSC2	OSC1
9Fh	ADCON1	ADDAC3	ADDAC2	ADDAC1	ADDAC0	PCFG3	PCFG2	PCFG1	PCFG0

Legend
— = unimplemented bits, read as '0' but cannot be overwritten
r = reserved bits, default is POR value and **should not be overwritten with any value**
Reserved indicates reserved register and **should not be overwritten with any value**
* indicates registers that can be addressed from either bank

4.2.2.1 STATUS REGISTER

The STATUS register, shown in Figure 4-3, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory.

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 or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the $\overline{TO}$ and $\overline{PD}$ bits are not writable. 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` and `MOVWF` instructions are used to alter the STATUS register because these instructions do not affect the Z, C or DC bits from the STATUS register. For other instructions, not affecting any status bits, see the "Instruction Set Summary."

> **Note 1:** The IRP and RP1 bits (STATUS<7:6>) are not used by the PIC14000 and should be programmed as cleared. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward compatibility with future products.
>
> **Note 2:** The C and DC bits operate as a $\overline{\text{borrow}}$ and $\overline{\text{digit borrow}}$ out bit, respectively, in subtraction. See the `SUBLW` and `SUBWF` instructions for examples.

FIGURE 4-3: STATUS REGISTER

83h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
STATUS	IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
Read/Write	R/W	R/W	R/W	R	R	R/W	R/W	R/W
POR value FFh	0	0	0	1	1	X	X	X

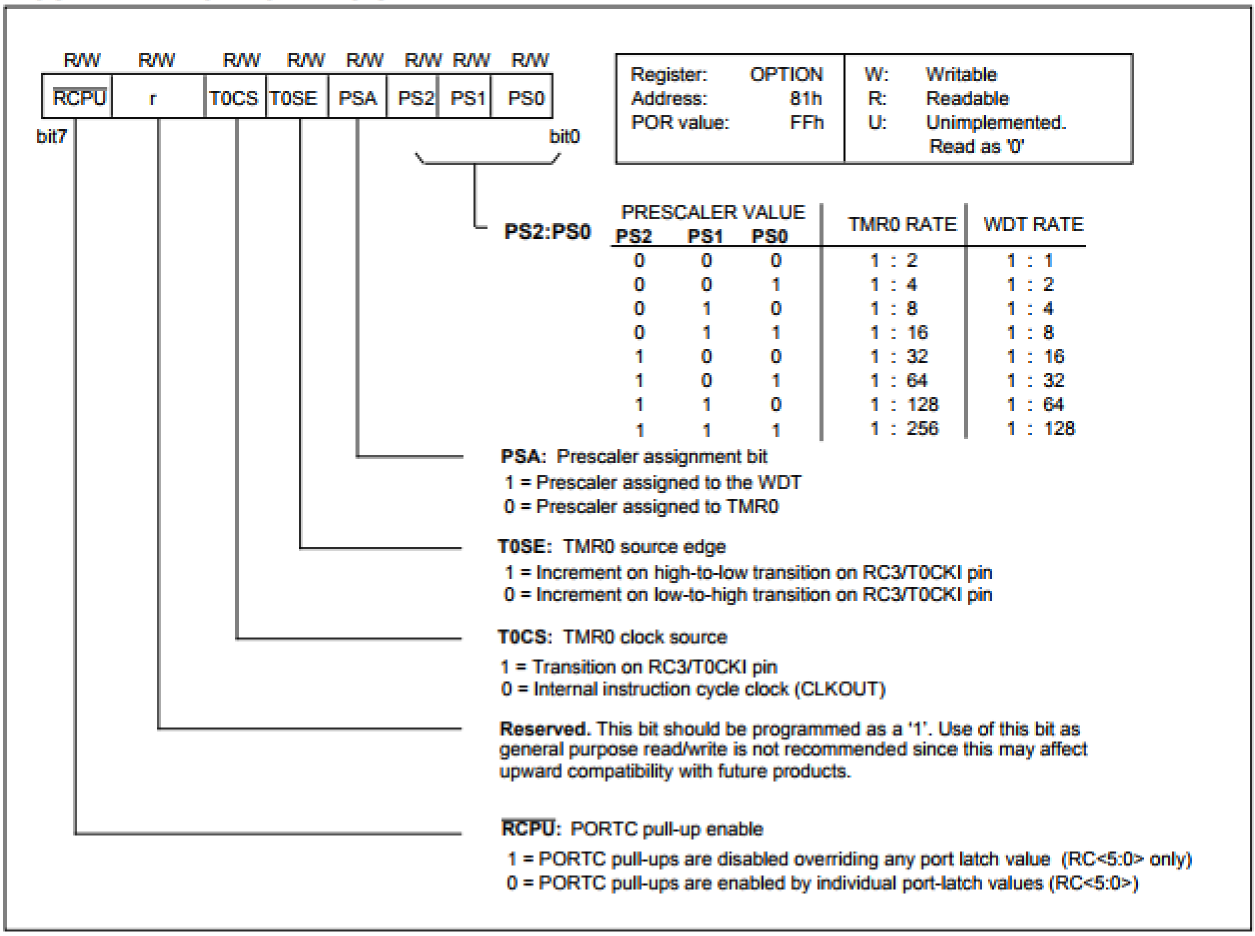
Bit	Name	Function
B7	IRP	Not used. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.
B6	RP1	Not used. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.
B5	RP0	Register page select for direct addressing. 1 = Bank1 (80h - FFh) 0 = Bank0 (00h - 7Fh) Each page is 128 bytes. Only the RP0 bit is used.
B4	$\overline{\text{TO}}$	Time-out bit. 1 = After power-up and by the `CLRWDT` and `SLEEP` instruction. 0 = A watchdog timer time-out has occurred.
B3	$\overline{\text{PD}}$	Power down bit. 1 = After power-up or by a `CLRWDT` instruction. 0 = By execution of the `SLEEP` instruction.
B2	Z	Zero bit. 1 = The result of an arithmetic or logic operation is zero. 0 = The result of an arithmetic or logical operation is not zero.
B1	DC	Digit carry / $\overline{\text{borrow}}$ bit. For `ADDWF` and `ADDLW` instructions. 1 = A carry-out from the 4th low order bit of the result. 0 = No carry-out from the 4th low order bit of the result. Note: For Borrow, the polarity is reversed.
B0	C	Carry / $\overline{\text{borrow}}$ bit. For `ADDWF` and `ADDLW` instructions. 1 = A carry-out from the most significant bit of the result occurred. Note that a subtraction is executed by adding the two'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. 0 = No carry-out from the most significant bit of the result. Note: For Borrow the polarity is reversed.

4.2.2.2 OPTION REGISTER

The OPTION register (Address 81h) is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, TMR0, and the weak pull-ups on PORTC<5:0>. Bit 6 is reserved.

> **Note:** To achieve a 1:1 prescaler assignment, assign the prescaler to the WDT (PSA=1)

FIGURE 4-4: OPTION REGISTER



R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
$\overline{\text{RCPU}}$	r	T0CS	T0SE	PSA	PS2	PS1	PS0
bit7							bit0

Register:	OPTION	W:	Writable
Address:	81h	R:	Readable
POR value:	FFh	U:	Unimplemented. Read as '0'

PS2:PS0

PRESCALER VALUE			TMR0 RATE	WDT RATE
PS2	**PS1**	**PS0**		
0	0	0	1 : 2	1 : 1
0	0	1	1 : 4	1 : 2
0	1	0	1 : 8	1 : 4
0	1	1	1 : 16	1 : 8
1	0	0	1 : 32	1 : 16
1	0	1	1 : 64	1 : 32
1	1	0	1 : 128	1 : 64
1	1	1	1 : 256	1 : 128

PSA: Prescaler assignment bit
1 = Prescaler assigned to the WDT
0 = Prescaler assigned to TMR0

T0SE: TMR0 source edge
1 = Increment on high-to-low transition on RC3/T0CKI pin
0 = Increment on low-to-high transition on RC3/T0CKI pin

T0CS: TMR0 clock source
1 = Transition on RC3/T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

Reserved. This bit should be programmed as a '1'. Use of this bit as general purpose read/write is not recommended since this may affect upward compatibility with future products.

$\overline{\text{RCPU}}$: PORTC pull-up enable
1 = PORTC pull-ups are disabled overriding any port latch value (RC<5:0> only)
0 = PORTC pull-ups are enabled by individual port-latch values (RC<5:0>)

4.2.2.3 INTCON REGISTER

The INTCON Register is a readable and writable register which contains the various enable and flag bits for the Timer0 overflow and peripheral interrupts. Figure 4-5 shows the bits for the INTCON register.

> **Note:** The T0IF will be set by the specified condition even if the corresponding Interrupt Enable Bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled). Before enabling interrupt, clear the interrupt flag, to ensure that the program does not immediately branch to the peripheral interrupt service routine

FIGURE 4-5: INTCON REGISTER

	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	GIE	PEIE	T0IE	r	r	T0IF	r	r
	bit7							bit0

Register:	INTCON	W:	Writable
Address:	0Bh or 8Bh	R:	Readable
POR value:	0000 000xb	U:	Unimplemented, read as '0'

Reserved. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

Reserved. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

T0IF: TMR0 overflow interrupt flag

1 = The TMR0 has overflowed
    Must be cleared by software
0 = TMR0 did not overflow

Reserved. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

Reserved. This bit should be programmed as '0'. Use of this bit as a general purpose read/write bit is not recommended, since this may affect upward compatibility with future products.

T0IE: TMR0 interrupt enable bit

1 = Enables T0IF interrupt
0 = Disables T0IF interrupt

PEIE: Peripheral interrupt enable bit

1 = Enables all un-masked peripheral interrupts
0 = Disables all peripheral interrupts

GIE: Global interrupt enable

1 = Enables all un-masked interrupts
0 = Disables all interrupts

4.2.2.4 PIE1 REGISTER

This register contains the individual enable bits for the Peripheral interrupts including A/D capture event, I^2C serial port, PORTC change and A/D capture timer overflow, and external push button.

Note: INTCON<6> must be enabled to enable any interrupt in PIE1.

FIGURE 4-6: PIE1 REGISTER

R/W	R	R	R/W	R/W	R/W	R/W	R/W
CMIE	—	—	PBIE	I^2CIE	RCIE	ADCIE	OVFIE
bit7							bit0

Register:	PIE1	W:	Writable
Address:	8Ch	R:	Readable
POR value:	00h	U:	Unimplemented, read as '0'

OVFIE: A/D Counter Overflow Interrupt Enable
1 = Enables A/D counter overflow interrupt
0 = Disables A/D counter overflow interrupt

ADCIE: A/D Capture Interrupt Enable
1 = A/D capture interrupt is enabled
0 = A/D capture interrupt is disabled

RCIE: PORTC Interrupt on change Enable
1 = Enables RCIF interrupt on pins, RC<7:4>
0 = Disables RCIF interrupt

I^2CIE: I^2C Port Interrupt Enable
1 = Enables I^2CIF interrupt
0 = Disables I^2CIF interrupt

PBIE: External Pushbutton Interrupt Enable
1 = Enable PBTN (pushbutton) interrupt on OSC1/PBTN.
(Note this interrupt not available in HS mode).
0 = Disable PBTN interrupt on OSC1/PBTN

Unimplemented. Read as '0'

Unimplemented. Read as '0'

CMIE: Programmable Reference Comparator Interrupt Enable
1 = Enable programmable reference comparator trip
0 = Disable programmable reference comparator trip

4.2.2.5 PIR1 REGISTER

This register contains the individual flag bits for the Peripheral interrupts (Figure 4-7).

Note: These bits will be set by the specified condition, even if the corresponding Interrupt Enable bit is cleared (interrupt disabled) or the GIE bit is cleared (all interrupts disabled). Before enabling an interrupt, the user may wish to clear the corresponding interrupt flag, to ensure that the program does not immediately branch to the Peripheral Interrupt service routine.

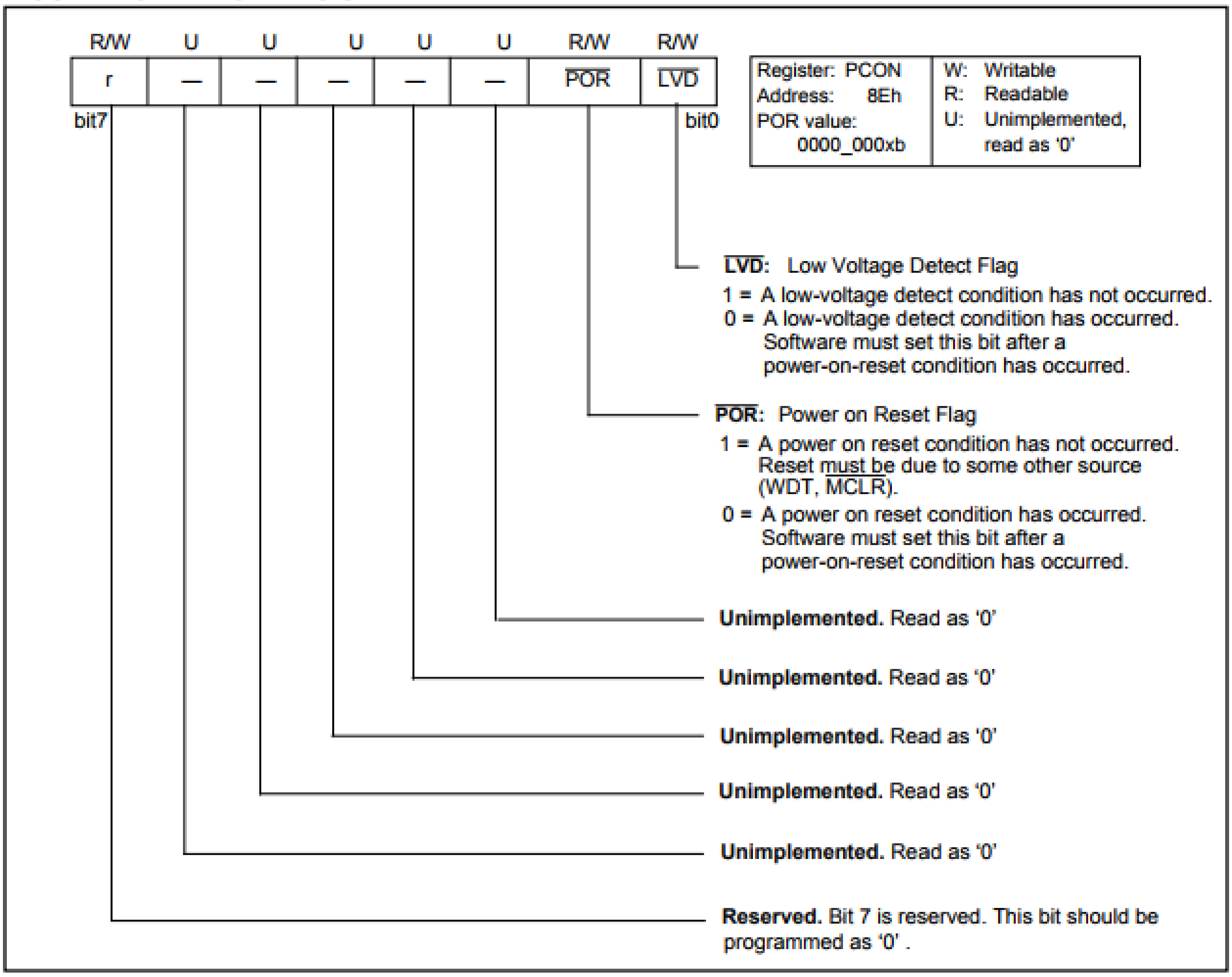
4.2.2.6 PCON REGISTER

The Power Control (PCON) register status contains 2 flag bits to allow differentiation between a Power-on Reset, an external $\overline{\text{MCLR}}$ reset, WDT reset, or low-voltage condition (Figure 4-8).

These bits are cleared on POR. The user must set these bits following POR. On a subsequent reset if POR is cleared, this is an indication that the reset was due to a power-on reset condition.

> **Note:** $\overline{\text{LVD}}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent resets to see if $\overline{\text{LVD}}$ is cleared, indicating a low voltage condition has occurred.

FIGURE 4-8: PCON REGISTER



R/W	U	U	U	U	U	R/W	R/W
r	—	—	—	—	—	$\overline{\text{POR}}$	$\overline{\text{LVD}}$
bit7							bit0

Register: PCON Address: 8Eh POR value: 0000_000xb	W: Writable R: Readable U: Unimplemented, read as '0'

$\overline{\text{LVD}}$: Low Voltage Detect Flag
1 = A low-voltage detect condition has not occurred.
0 = A low-voltage detect condition has occurred. Software must set this bit after a power-on-reset condition has occurred.

$\overline{\text{POR}}$: Power on Reset Flag
1 = A power on reset condition has not occurred. Reset must be due to some other source (WDT, $\overline{\text{MCLR}}$).
0 = A power on reset condition has occurred. Software must set this bit after a power-on-reset condition has occurred.

Unimplemented. Read as '0'

Unimplemented. Read as '0'

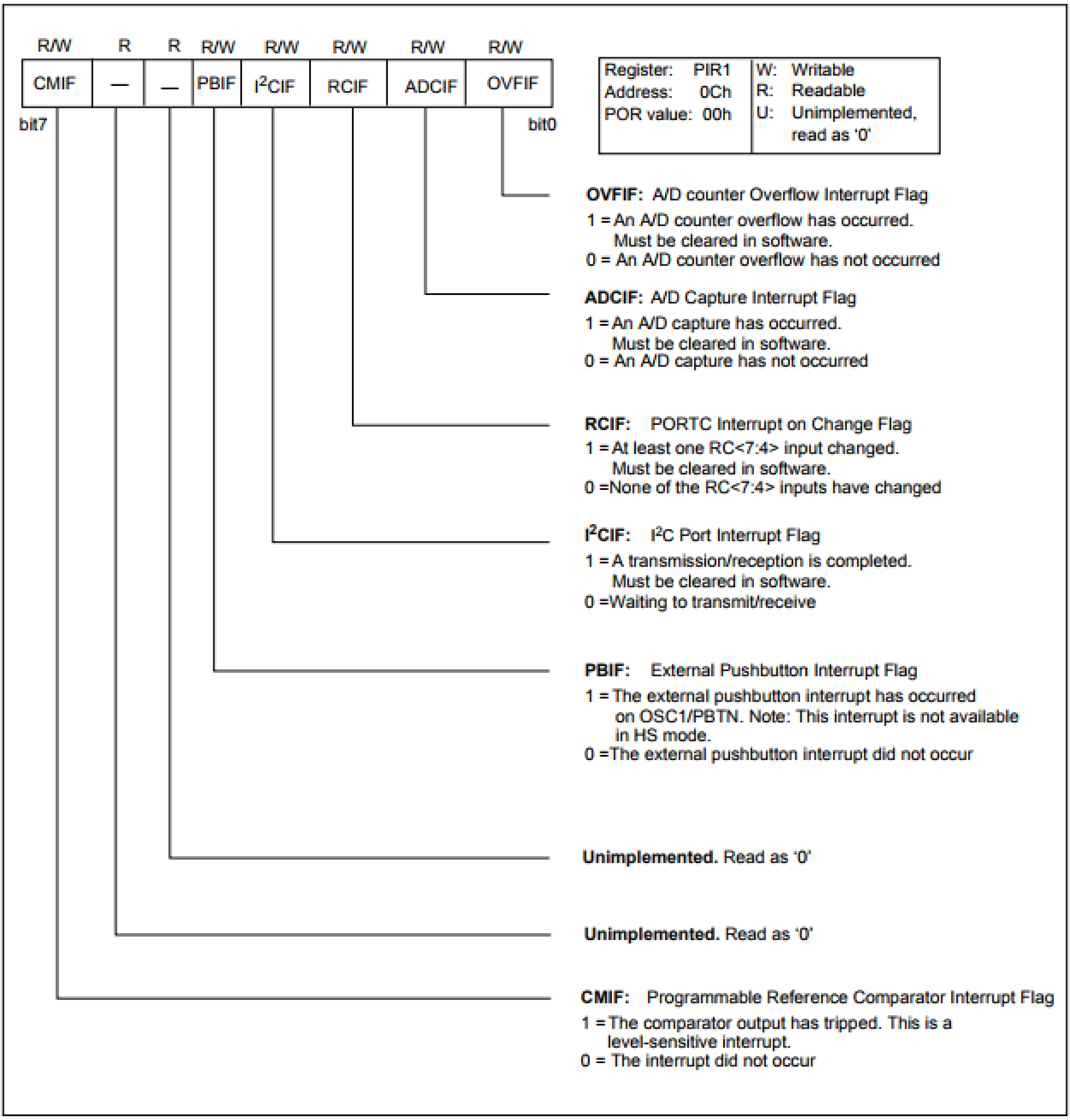
Unimplemented. Read as '0'

Unimplemented. Read as '0'

Unimplemented. Read as '0'

Reserved. Bit 7 is reserved. This bit should be programmed as '0' .

FIGURE 4-7: PIR1 REGISTER



R/W	R	R	R/W	R/W	R/W	R/W	R/W
CMIF	—	—	PBIF	I^{2}CIF	RCIF	ADCIF	OVFIF
bit7							bit0

Register: PIR1
Address: 0Ch
POR value: 00h

W: Writable
R: Readable
U: Unimplemented, read as '0'

OVFIF: A/D counter Overflow Interrupt Flag
1 = An A/D counter overflow has occurred. Must be cleared in software.
0 = An A/D counter overflow has not occurred

ADCIF: A/D Capture Interrupt Flag
1 = An A/D capture has occurred. Must be cleared in software.
0 = An A/D capture has not occurred

RCIF: PORTC Interrupt on Change Flag
1 = At least one RC<7:4> input changed. Must be cleared in software.
0 =None of the RC<7:4> inputs have changed

I^{2}CIF: I^{2}C Port Interrupt Flag
1 = A transmission/reception is completed. Must be cleared in software.
0 =Waiting to transmit/receive

PBIF: External Pushbutton Interrupt Flag
1 = The external pushbutton interrupt has occurred on OSC1/PBTN. Note: This interrupt is not available in HS mode.
0 =The external pushbutton interrupt did not occur

Unimplemented. Read as '0'

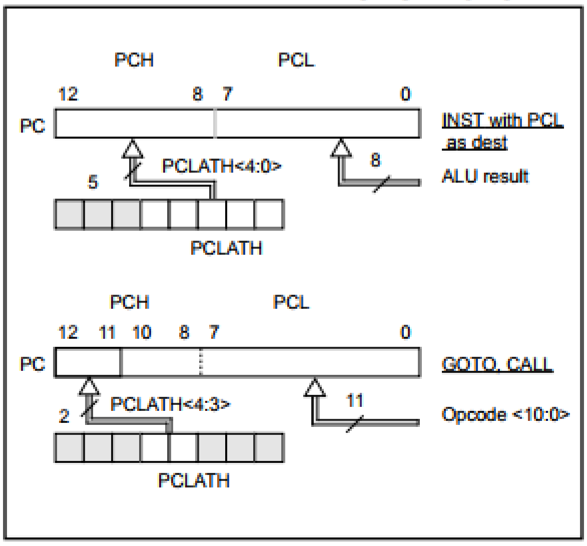
Unimplemented. Read as '0'

CMIF: Programmable Reference Comparator Interrupt Flag
1 =The comparator output has tripped. This is a level-sensitive interrupt.
0 = The interrupt did not occur

4.3 PCL and PCLATH

The program counter (PC) is 13-bits wide. The low byte, PCL, is a readable and writable register. The high byte of the PC (PCH) is not directly readable or writable. PCLATH is a holding register for PC<12:8> where contents are transferred to the upper byte of the program counter. When PC is loaded with a new value during a `CALL`, `GOTO` or a write to PCL, the high bits of PC are loaded from PCLATH as shown in Figure 4-9.

FIGURE 4-9: LOADING OF PC IN DIFFERENT SITUATIONS



Note: On POR, the contents of the PCLATH register are unknown. The PCLATH should be initialized before a CALL, GOTO, or any instruction that modifies the PCL register is executed.

4.3.1 COMPUTED GOTO

When doing a table read using a computed GOTO method, care should be exercised if the table location crosses a PCL memory boundary (each 256 byte block). Refer to the application note “Table Read Using the PIC16CXX”(AN556).

4.3.2 STACK

The PIC14000 has an 8 deep x 13-bit wide hardware stack (Figure 4-1). The stack space is not part of either program or data space and the stack pointer is not readable or writable. The PC is PUSHed in the stack when a `CALL` instruction is executed or an interrupt is acknowledged. The stack is POPed in the event of a `RETURN`, `RETLW` or a `RETFIE` instruction execution. PCLATH is not affected by a “PUSH” or a “POP” operation.

The stack operates as a circular buffer. This means that after the stack has been “PUSHed” eight times, the ninth push overwrites the value that was stored from the first push. The tenth push overwrites the second push (and so on).

Note 1: There are no STATUS bits to indicate stack overflow or stack underflow conditions.

Note 2: There are no instruction mnemonics called PUSH nor POP. These are actions that occur from the execution of the `CALL`, `RETURN`, `RETLW`, or `RETFIE` instructions, or the vectoring to an interrupt address

4.3.3 PROGRAM MEMORY PAGING

The PIC14000 has 4K of program memory, but the `CALL` and `GOTO` instructions only have a 11-bit address range. This 11-bit address range allows a branch within a 2K program memory page size. To allow `CALL` and `GOTO` instructions to address the entire 4K program memory address range, there must be another bit to specify the program memory page. This paging bit comes from the PCLATH<3> bit (Figure 4-9). When doing a `CALL` or `GOTO` instruction, the user must ensure that this page bit (PCLATH<3>) is programmed to the desired program memory page. If a `CALL` instruction (or interrupt) is executed, the entire 13-bit PC is pushed onto the stack. Therefore, manipulation of the PCLATH<3> is not required for the return instructions (which pops the PC from the stack).

Note: The PIC14000 ignores the PCLATH<4> bit, which is used for program memory pages 2 and 3 (1000h-1FFFh). The use of PCLATH<4> as a general purpose read/write bit is not recommended since this may affect upward compatibility with future products.

Example 4-1 shows the calling of a subroutine in page 1 of the program memory. This example assumes that the PCLATH is saved and restored by the interrupt service routine (if interrupts are used).

EXAMPLE 4-1: CALL OF A SUBROUTINE IN PAGE 1 FROM PAGE 0

```
ORG 0x500
BSF     PCLATH, 3 ; Select page 1 (800h-FFFh)
CALL    SUB1_P1   ; Call subroutine in
        :         ; page 1 (800h-FFFh)
        :
        :
ORG 0x900
SUB1 P1 :         ; called subroutine
        :         ; page 1 (800h-FFFh)
        :
RETURN            ; return to page 0
                  ; (000h-7FFh)
```

4.4 Indirect Addressing, INDF and FSR Registers

The INDF register is not a physical register. Addressing the INDF register will cause indirect addressing.

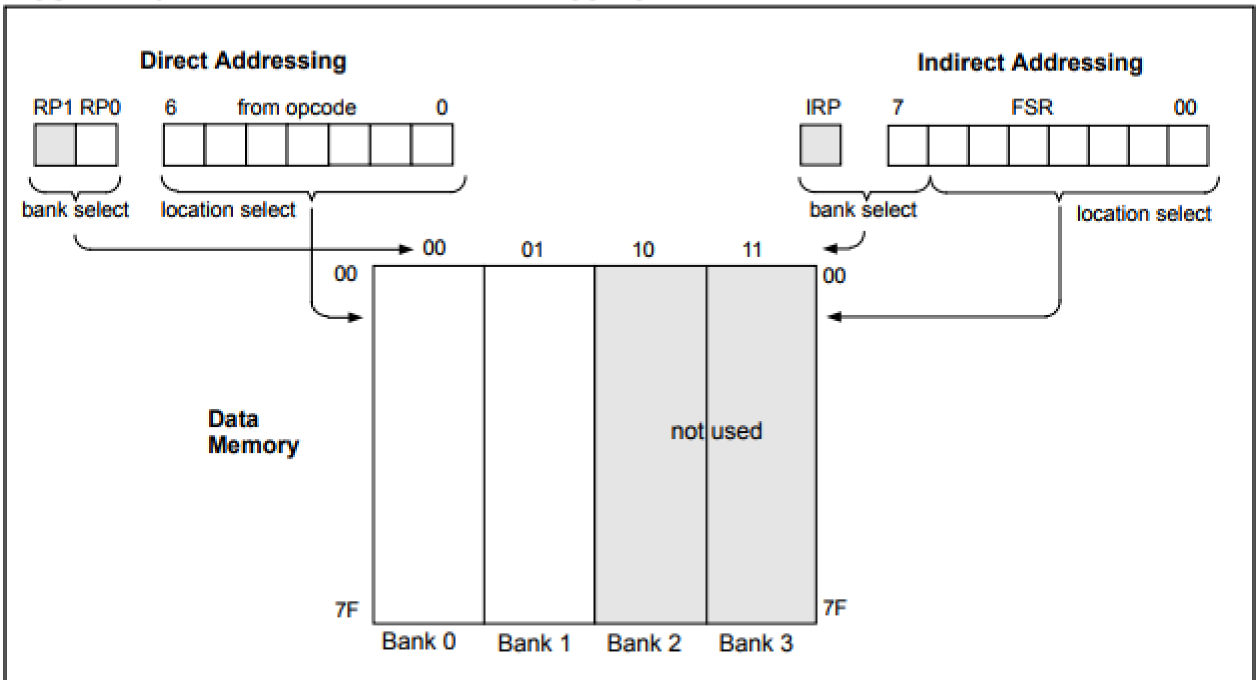
Indirect addressing is possible by using the INDF register. Any instruction using the INDF register actually accesses data pointed to by the file select register (FSR). Reading INDF itself indirectly will produce 00h. Writing to the INDF register indirectly results in a no-operation (although status bits may be affected). An effective 9-bit address is obtained by concatenating the 8-bit FSR register and the IRP bit (STATUS<7>), as shown in Figure 4-10. However, IRP is not used in the PIC14000.

A simple program to clear RAM location 20h-2Fh using indirect addressing is shown in Example 4-2.

EXAMPLE 4-2: INDIRECT ADDRESSING

```
          movlw   0x20     ;initialize pointer
          movf    FSR      ;to RAM
NEXT      clrf    INDF     ;clear INDF register
          incf    FSR      ;inc pointer
          btfss   FSR,4    ;all done?
          goto    NEXT     ;no clear next
                           ;yes continue
CONTINUE:
```

FIGURE 4-10: INDIRECT/INDIRECT ADDRESSING



Note: For memory map detail see Figure 4-1.

5.0 I/O PORTS

The PIC14000 has three ports, PORTA, PORTC and PORTD, described in the following paragraphs. Generally, PORTA is used as the analog input port. PORTC is used for general purpose I/O and for host communication. PORTD provides additional I/O lines. Four lines of PORTD may function as analog inputs.

5.1 PORTA and TRISA

PORTA is a 4-bit wide port with data register located at location 05h and corresponding data direction register (TRISA) at 85h. PORTA can operate as either analog inputs for the internal A/D converter or as general purpose digital I/O ports. These inputs are Schmitt Triggers when used as digital inputs, and have CMOS drivers as outputs.

PORTA pins are multiplexed with analog inputs. ADCON1<1:0> bits control whether these pins are analog or digital as shown in Section 8.7. When configured to the digital mode, reading the PORTA register reads the status of the pins whereas writing to it will write to the port latch. When selected as an analog input, these pins will read as '0's.

Note: On Reset, PORTA is configured as analog inputs

The TRISA register controls the direction of the PORTA pins, even when they are being used as analog inputs. The user must make sure to keep the pins configured as inputs when using them as analog inputs. A '1' in each location configures the corresponding port pin as an input. This register resets to all '1's, meaning all PORTA pins are initially inputs. The data register should be initialized prior to configuring the port as outputs. See Figure 5-2 and Figure 5-3.

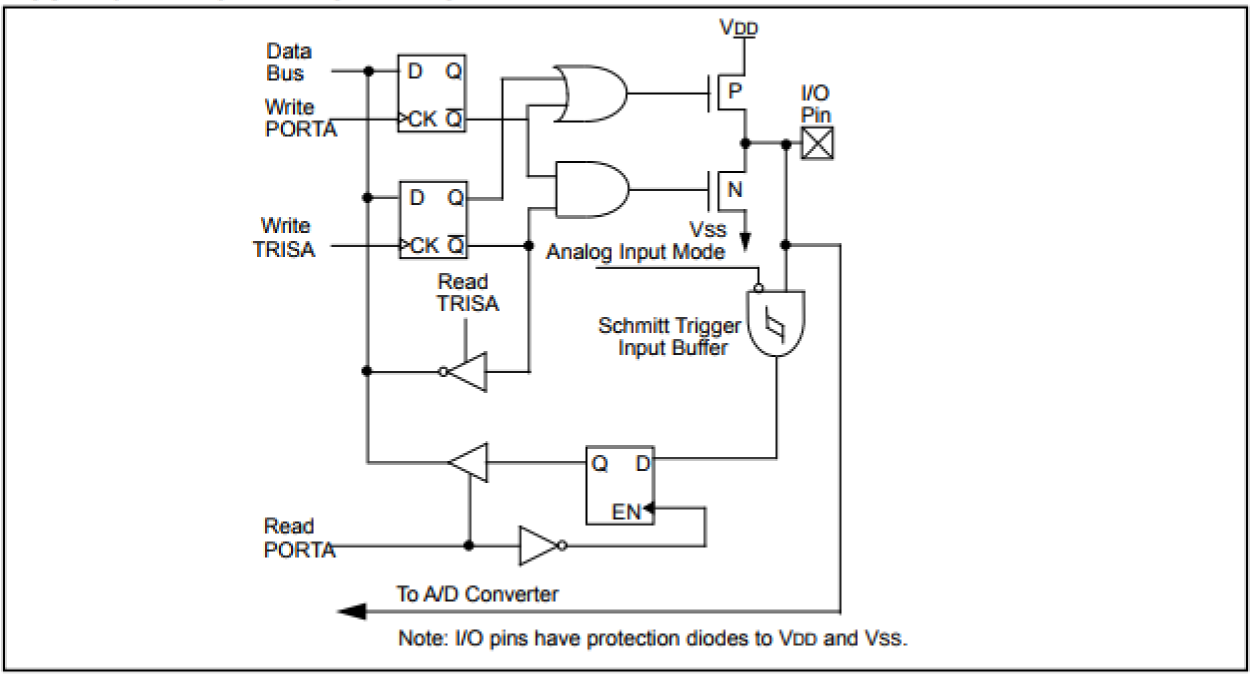
PORTA inputs go through a Schmitt Trigger AND gate that is disabled when the input is in analog mode. Refer to Figure 5-1.

Note that bits RA<7:4> are unimplemented and always read as '0'. Unused inputs should not be left floating to avoid leakage currents. All pins have input protection diodes to VDD and VSS.

EXAMPLE 5-1: INITIALIZING PORTA

```
CLRF   PORTA          ;Initialize PORTA by setting
                      ;output data latches
BSF    STATUS, RP0    ;Select Bank1
MOVLW  0x0F           ;Value used to initialize
                      ;data direction
MOVWF  TRISA          ;Set RA<3:0> as inputs
```

FIGURE 5-1: PORTA BLOCK DIAGRAM



Note: I/O pins have protection diodes to VDD and VSS.

FIGURE 5-2: PORTA DATA REGISTER

05h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTA	—	—	—	—	RA3/AN3	RA2/AN2	RA1/AN1	RA0/AN0
Read/Write	U	U	U	U	R/W	R/W	R/W	R/W
POR value 0xh	0	0	0	0	X	X	X	X

Bit	Name	Function
B7-B4	—	Unimplemented. Reads as‘0’.
B3	RA3/AN3	GPIO or analog input. Returns value on pin RA3/AN3 when used as a digital input. When configured as an analog input, reads as ‘0’.
B2	RA2/AN2	GPIO or analog input. Returns value on pin RA2/AN2 when used as a digital input. When configured as an analog input, reads as ‘0’.
B1	RA1/AN1	GPIO or analog input. Returns value on RA1/AN1 when used as a digital input. This pin can connect to a level shift network. If enabled, a +0.5V offset is added to the input voltage. When configured as an analog input, reads as ‘0’.
B0	RA0/AN0	GPIO or analog input. Returns value on pin RA0/AN0 when used as a digital input. When configured as an analog input, reads as ‘0’.

5.2 PORTC and TRISC

PORTC is a 8-bit wide bidirectional port, with Schmitt Trigger inputs, that serves the following functions depending on programming:

- Direct LED drive (PORTC<7:0>).
- I^{2}C communication lines (PORTC<7:6>), refer to Section 7.0 I^{2}C Serial Port.
- Interrupt on change function (PORTC<7:4>), discussed below and in Section 10.3 Interrupts.
- Programmable reference and comparator outputs.
- Timer0 clock source on RC3

The PORTC data register is located at location 07h and its data direction register (TRISC) is at 87h.

PORTC<5:0> have weak internal pull-ups (~100 uA typical). A single control bit can turn on all the pull-ups. This is done by clearing bit $\overline{\text{RCPU}}$ (OPTION<7>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on power-on reset and in hibernate mode.

When using PORTC<0> as an analog output (CMCON<1> bit is set), the TRISC<0> bit should be cleared to disable the weak pull-up on this pin. Refer to Table 5-1.

Four of the PORTC pins, RC<7:4> have an interrupt on change feature. Only pins configured as inputs can cause this interrupt to occur. In other words, any pin RC<7:4> configured as an output is excluded from the interrupt on change comparison. The input pins of RC<7:4> are compared with the old value latched on the last read of PORTC. The “mismatch" outputs of RC<7:4> are OR'ed together to assert the RCIF flag (PIR1 register<2>) and cause a CPU interrupt, if enabled.

> **Note:** If the I^{2}C function is enabled, (I^{2}CCON<5>, address 14h), RC<7:6> are automatically excluded from the interrupt-on-change comparison.

This interrupt can wake the device up from SLEEP. The user, in the interrupt service routine, can clear the interrupt in one of two ways:

- Disable the interrupt by clearing the RCIE (PIE1<2>) bit
- Read PORTC. This will end mismatch condition. Then, clear the RCIF (PIR1<2>) bit.

A mismatch condition will continue to set the RCIF bit. Reading PORTC will end the mismatch condition, and allow the RCIF bit to be cleared.

If bit CMAOE (CMCON<1>) is set, the RC0/REFA pin becomes the programmable reference A and analog output. Pin RC1/CMPA becomes the comparator A output.

> **Note:** Setting CMAOE changes the definition of RC0/REFA and RC1/CMPA, bypassing the PORTC data and TRISC register settings.

PORTC<7:6> also serves multiple functions. These pins act as the I^2C data and clock lines when the I^2C module is enabled. They also serve as the serial programming interface data and clock line for in-circuit programming of the EPROM.

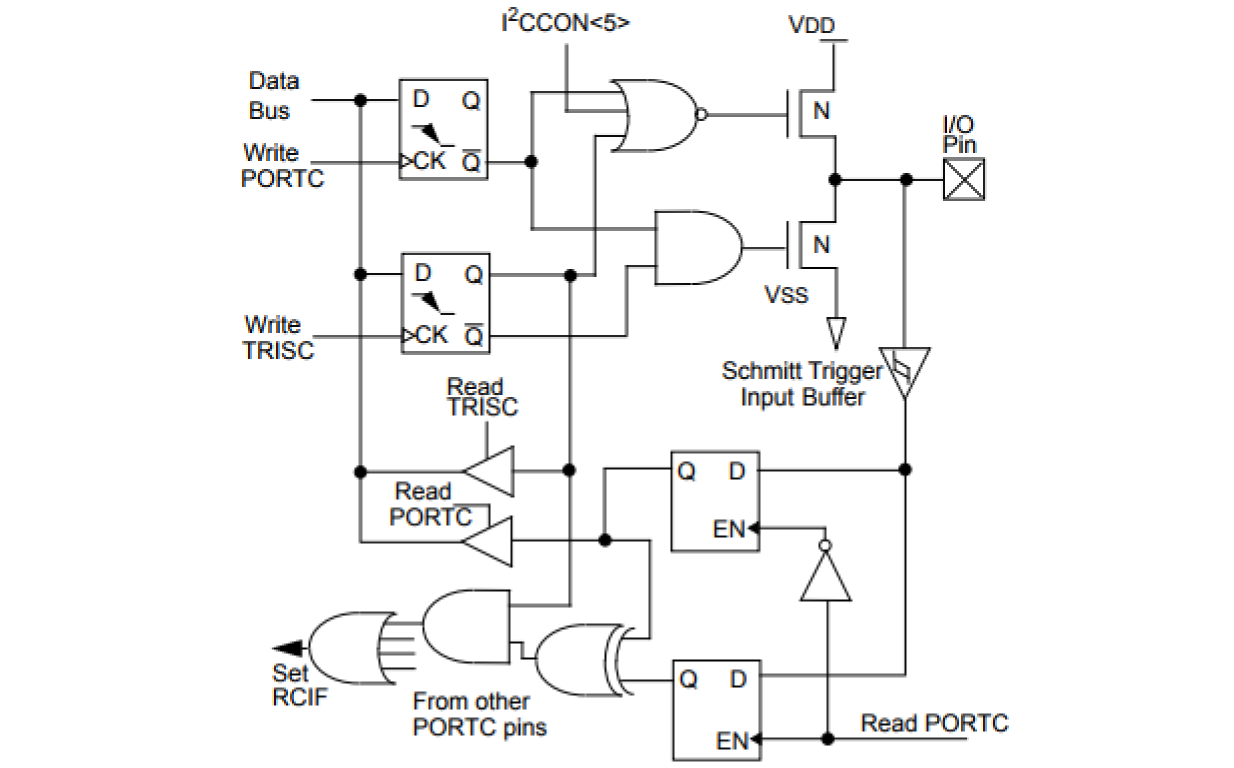
The TRISC register controls the direction of the PORTC pin. A '1' in each location configures the corresponding port pin as an input. Upon reset, this register sets to FFh, meaning all PORTC pins are initially inputs. The data register should be initialized prior to configuring the port as outputs.

Unused inputs should not be left floating to avoid leakage currents. All pins have input protection diodes to VDD and VSS.

EXAMPLE 5-2: INITIALIZING PORTC

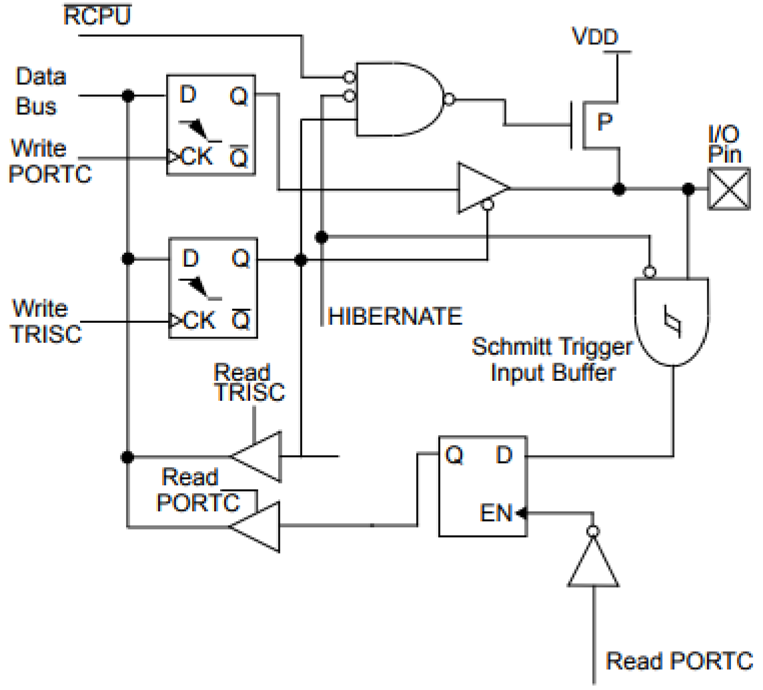
```
CLRF   PORTC        ; Initialize PORTC data
                    ;    latches before setting
                    ;    the data direction
                    ;    register
BSF    STATUS, RPO  ; Select Bank1
MOVLW  0xCF         ; Value used to initialize
                    ; data direction
MOVWF  TRISC        ; Set RC<3:0> as inputs
                    ;    RC<5:4> as outputs
                    ;    RC<7:6> as inputs
```

FIGURE 5-3: BLOCK DIAGRAM OF PORTC<7:6> PINS



Note: I/O pins have protection diodes to VDD and VSS. These pins do not have a P-channel pull-up.

FIGURE 5-5: BLOCK DIAGRAM OF PORTC<3:0> PINS

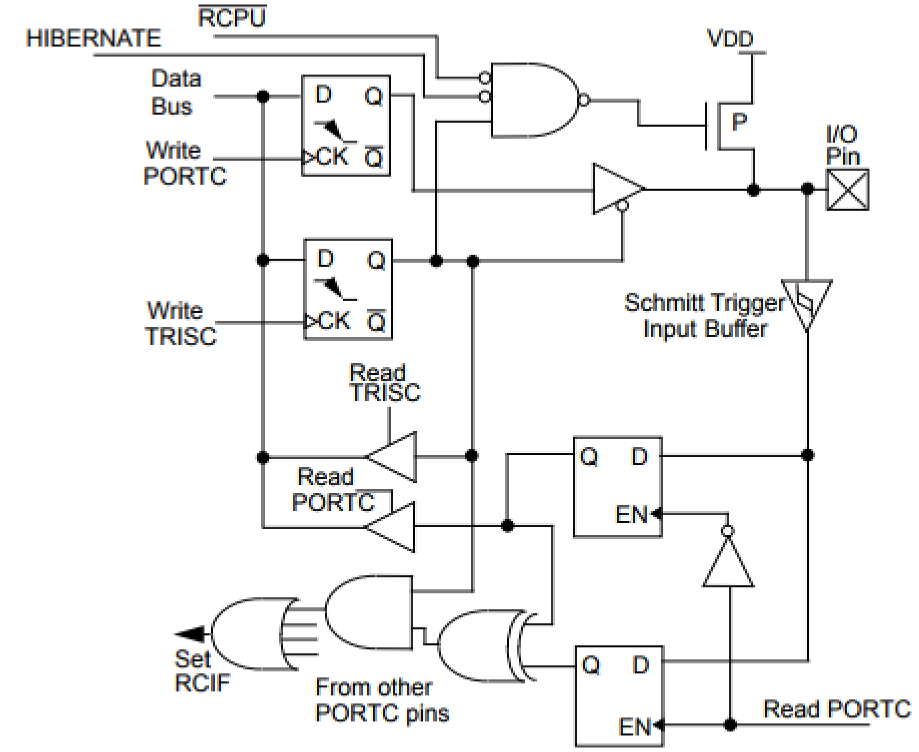


1. I/O pins have protection diodes to VDD and VSS.
2. Port Latch ='1' and TRISC ='1' enables weak pull-up if $\overline{\text{RCPU}}$ ='0' in OPTION register.
3. If the CMAOE bit (CMCON<1>) is set to'1', RC0 becomes REFA, RC1 becomes CMPA, ignoring the PORTC<1:0> data and TRISC<1:0> register settings.

TABLE 5-1: PORT RC0 PIN CONFIGURATION SUMMARY

RC0 Pin Configuration	TRISC<0>	$\overline{\text{RCPU}}$ OPTION<7>	CMAOE CMCON<1>	Comment
Digital Input (weak pull-up)	1	0	0	
Digital Input (no pull-up)	1	1	0	
Digital Output	0	X	0	
Analog Output	0	X	1	Must clear TRISC<0> to disable pull-up when used as an analog output.

FIGURE 5-4: BLOCK DIAGRAM OF PORTC<5:4> PINS



1. I/O pins have protection diodes to VDD and VSS.

2. Port Latch = ‘1’ and TRISC = ‘1’ enables weak pull-up if $\overline{\text{RCPU}}$ = ‘0’ in OPTION register.

FIGURE 5-6: PORTC DATA REGISTER

07h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTC	RC7/SDAA	RC6/SCLA	RC5	RC4	RC3/T0CKI	RC2	RC1/CMPA	RC0/REFA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value xxh	x	x	x	x	x	x	x	x

Bit	Name	Function
B7	RC7/SDAA	Synchronous serial data I/O for I^{2}C interface. Also is the serial programming data line. This pin can also serve as a general purpose I/O. If enabled, a change on this pin can cause a CPU interrupt. This pin has an N-channel pull-up to VDD which is disabled in I^{2}C mode.
B6	RC6/SCLA	Synchronous serial clock for I^{2}C interface. Also is the serial programming clock. This pin can also serve as a general purpose I/O. If enabled, a change on this pin can cause a CPU interrupt. This pin has an N-channel pull-up to VDD which is disabled in I^{2}C mode.
B5	RC5	LED direct-drive output. This pin can also serve as a GPIO. If enabled, a change on this pin can cause a CPU interrupt. If enabled, this pin has a weak internal pull-up to VDD.
B4	RC4	LED direct-drive output. This pin can also serve as a GPIO. If enabled, a change on this pin can cause a CPU interrupt. If enabled, this pin has a weak internal pull-up to VDD.
B3	RC3/T0CKI	LED direct-drive output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to VDD. T0CKI is enabled as TMR0 clock via the OPTION register.
B2	RC2	LED direct-drive output. This pin can also serve as a GPIO. If enabled, this pin has a weak internal pull-up to VDD.
B1	RC1/CMPA	LED direct-drive output. This pin can also serve as a GPIO, or comparator A output. If enabled, this pin has a weak internal pull-up to VDD.
B0	RC0/REFA	LED direct-drive output. This pin can also serve as a GPIO, or programmable reference A output. If enabled, this pin has a weak internal pull-up to VDD.

U= unimplemented, X = unknown.

5.2.1 TRISC PORTC DATA DIRECTION REGISTER

This register defines each pin of PORTC as either an input or output under software control. A ‘1’ in each location configures the corresponding port pin as an input. This register resets to all ‘1’s, meaning all PORTC pins are initially inputs. The data register should be initialized prior to configuring the port as outputs.

FIGURE 5-7: TRISC REGISTER

87h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value FFh	1	1	1	1	1	1	1	1

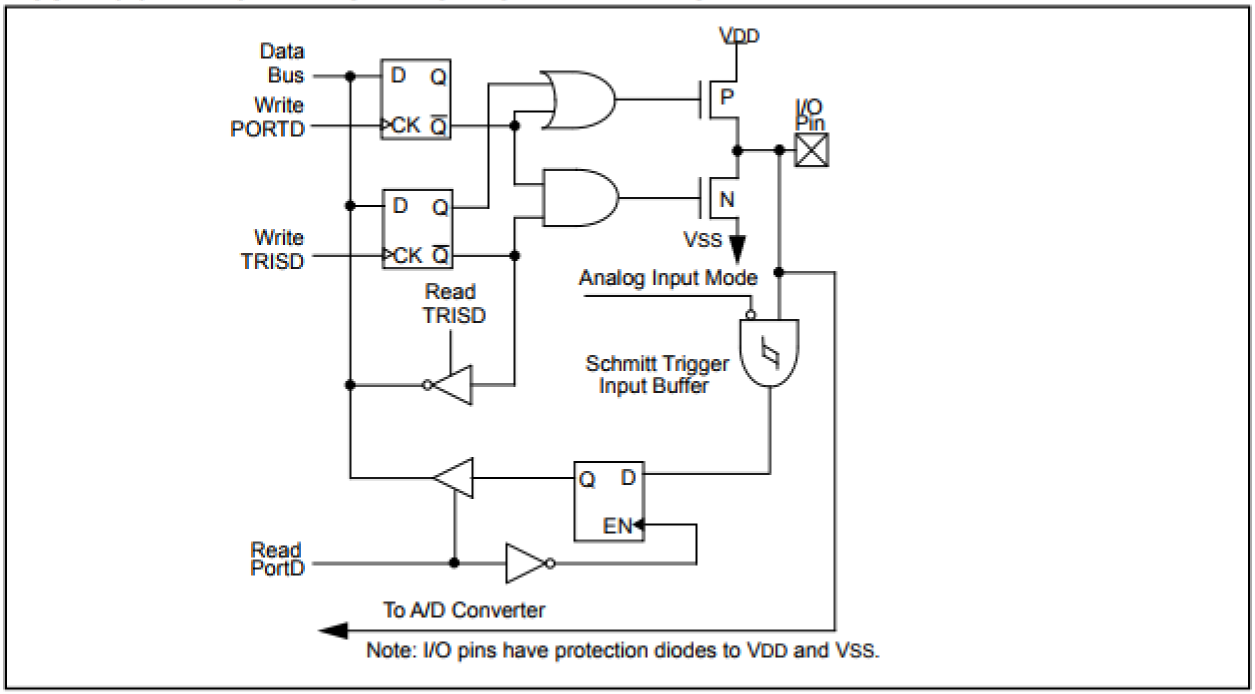
Bit	Name	Function
B7	TRISC7	Control direction on pin RC7/SDAA (has no effect if I^2C is enabled): 0 = pin is an output 1 = pin is an input
B6	TRISC6	Control direction on pin RC6/SCLA (has no effect if I^2C is enabled): 0 = pin is an output 1 = pin is an input
B5	TRISC5	Control direction on pin RC5: 0 = pin is an output 1 = pin is an input
B4	TRISC4	Control direction on pin RC4: 0 = pin is an output 1 = pin is an input
B3	TRISC3	Control direction on pin RC3: 0 = pin is an output 1 = pin is an input
B2	TRISC2	Control direction on pin RC2: 0 = pin is an output 1 = pin is an input
B1	TRISC1	Control direction on pin RC1/CMPA (has no effect if the CMAOE bit is set): 0 = pin is an output 1 = pin is an input
B0	TRISC0	Control direction on pin RC0/REFA (has no effect if the CMAOE bit is set): 0 = pin is an output 1 = pin is an input

U= unimplemented, X = unknown.

5.3 PORTD and TRISD

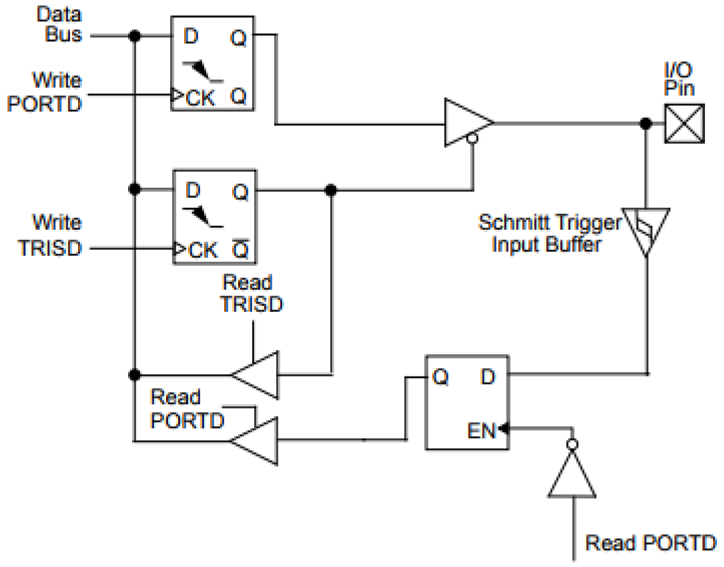
PORTD is an 8-bit port that may be used for general purpose I/O. Four pins can be configured as analog inputs.

FIGURE 5-8: BLOCK DIAGRAM OF PORTD<7:4> PINS



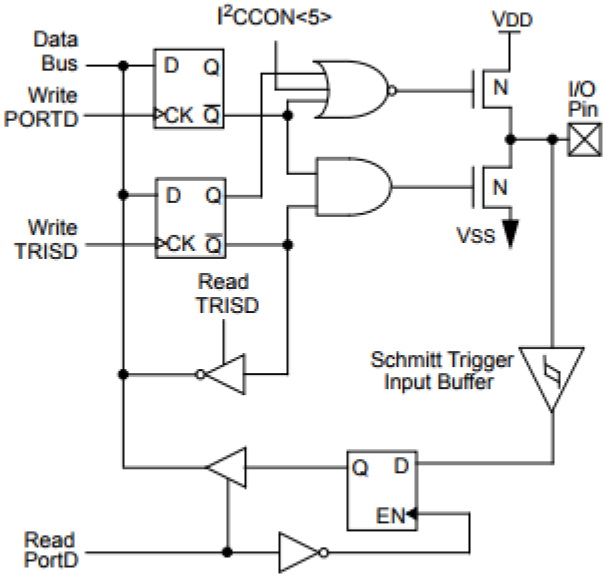
Note: I/O pins have protection diodes to VDD and VSS.

FIGURE 5-9: BLOCK DIAGRAM OF PORTD<3:2> PINS



1. I/O pins have protection diodes to VDD and VSS.
2. If CMBOE (CMCON<5>) is set to '1', RD2 becomes CMPB, RD3 becomes REFB, ignoring the PORTD<3:2> data and TRISD<3:2> register settings.

FIGURE 5-10: BLOCK DIAGRAM OF PORTD<1:0> PINS



Note: I/O pins have protection diodes to VDD and VSS. These pins do not have a P-channel pull-up.

FIGURE 5-11: PORTD DATA REGISTER

08h	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PORTD	RD7/AN7	RD6/AN6	RD5/AN5	RD4/AN4	RD3/REFB	RD2/CMPB	RD1/SDAB	RD0/SCLB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POR value xxh	X	X	X	X	X	X	X	X

Bit	Name	Function
B7	RD7/AN7	GPIO or analog input. Returns value on pin RD7/AN7 when used as a digital input. When configured as an analog input, reads as ‘0’.
B6	RD6/AN6	GPIO or analog input. Returns value on pin RD6/AN6 when used as a digital input. When configured as an analog input, reads as ‘0’.
B5	RD5/AN5	GPIO or analog input. This pin can connect to a level shift network. If enabled, a +0.5V offset is added to the input voltage. When configured as an analog input, reads as ‘0’.
B4	RD4/AN4	GPIO or analog input. Returns value on pin RD4/AN4 when used as a digital input. When configured as an analog input, reads as ‘0’.
B3	RD3/REFB	This pin can serve as a GPIO, or programmable reference B output.
B2	RD2/CMPB	This pin can serve as a GPIO, or comparator B output.
B1	RD1/SDAB	Alternate synchronous serial data I/O for I^{2}C interface enabled by setting the I^{2}CSEL bit in the MISC register. This pin can also serve as a general purpose I/O. This pin has an N-channel pull-up to VDD which is disabled in I^{2}C mode.
B0	RD0/SCLB	Alternate synchronous serial clock for I^{2}C interface, enabled by setting the I^{2}CSEL bit in the MISC register. This pin can also serve as a general purpose I/O. This pin has an N-Channel pull-up to VDD which is disabled in I^{2}C mode.

Legend: U = unimplemented, read as ‘0’, x = unknown.