PIC18F4550- A General Overview

Deepak Bhat Debapratim Ghosh

Virtual Labs
Indian Institute of Technology Bombay

19 September 2011

Abstract

With the world so heavily dependent on embedded systems, microcontrollers have become indispensable devices in embedded design. Put simply, a microcontroller (MCU) is a device that combines the core of a microprocessor (MPU) with features such as I/O ports, timers, some dedicated communication controllers etc. Whereas an MPU needs additional *glue logic* to interface to the outside world, an MCU is a self-sufficient device as far as peripheral handling is concerned, due to the availability of rich on-chip resources. Depending on a specific application, the choice of an appropriate MCU can indeed lead to a very efficient design. This document is a brief introductory giving an insight into the PIC18F4550 MCU-architecture and programming, and is aimed to give a headstart to a first-time user.

1 The PIC Family Overview

The PIC (Peripheral Interface Controller) is a family of MCUs manufactured by Microchip Technology. Originally developed as a supplementary to General Instruments' CP1600 CPU in 1975, the first 8-bit PIC MCU was a success due to its dedicated I/O operational architecture. Till about the early 1990s, PIC devices were primarily used as supplementary controllers to offload I/O handling from the main CPU. PIC MCUs began to be used as stand-alone programmable units after the first PIC with on-chip EEPROM, the PIC16C84 was developed in 1993.

PIC MCUs are available as 8-bit (PIC18 and earlier families), 16-bit (PIC24 and dsPIC families) and 32-bit devices (PIC32 family). PIC18F4550 is an 8-bit microcontroller, on which the board Aurum is centered.

2 The PIC18F4550 Architecture

The PIC18F4550 (hereafter referred to as 4550) is an 8-bit MCU with Harvard architecture, i.e. it has separate code and data spaces and pathways. It is commonly available in 40-pin DIP, 44-pin TQFP and QFN packages. Some of its features are enumerated as shown.

- Powered by 2.5-5.5V DC supplies, hence suitable for fairly low-power applications.
- Contains an on-chip USB 2.0 controller with voltage regulator. USB communication is used for programmability.
- Flexible oscillator structure- has an on-chip clock generator, and can also use an external crystal oscillator.
- Three external interrupts, four timers, dual analog comparators with multiplexed inputs, master synchronous serial port (MSSP) supporting SPI and I2C communications (all modes).

- On-chip 10-bit, 13-channel A/D converter, capture/compare PWM modules.
- On-chip 32KB program memory (instructions), 16KB program memory (bytes), with 2KB data memory. Supports in-system programmability (ISP), with facility for in-circuit debugging.

An overview of some of the important architectural parts are being described in the following subsections. The oscillator configurations, general I/O port structure, A/D converters and some of the communication buses such as USB, I2C and SPI are covered.

2.1 Oscillator and Clocking

The 4550 supports twelve distinct oscillator modes. Correlating with the conventional Intel 8051, which supports only one clock mode using an external crystal, the 4550 seems more complex. It should be kept in mind that the PIC family of microcontrollers are targeted specifically for a large class of I/O control. Thus, the different clock modes are used to make interfacing of certain peripherals easier.

Broadly, the twelve different modes can be classified as follows.

- Using an external crystal- The 4550 can work with an external crystal in four modes. External crystal (XT), crystal with PLL enabled (XTPLL), high-speed crystal (HS) and high-speed crystal with PLL enabled (HSPLL). The XT and XTPLL modes are used with crystals with frequencies below 4 MHz, and the high-speed modes are for applications above 4 MHz frequency. A PLL is provided to take reduce drift in frequncies, and is especially important for USB applications.
- Using an external clock source- The 4550 can support four modes when it is clocked by an external clock generator. During these modes, one of the pins dedicated for an external crystal (OSC2) can be used as an I/O port pin (of port A) or as a clock output, and can be used as a reference for external devices. The EC and ECPLL modes support the clock-out facility, dividing the input frequency by 4 to generate the clock-out signal, whereas the ECIO and ECPIO modes make the OSC2 pin work as an I/O pin, in this case, RA6 of port A. This is useful in asynchronous peripheral applications, where a reference clock for those peripherals is not required.
- Mixed clock modes-The 4550 supports four mixed modes of clocking. The INTXT and INTHS modes use the internal oscillator as clock to the microcontroller core, and the XT and HS clocks respectively, for the USB. The INTIO and INTCKO modes also use the internal oscillator as the core clock, but uses an external clock (EC) for the usb controller. The two modes differ only in the fact that INTIO uses the RA6 port pin as an I/O pin, while INTCKO gives a clockout signal on that pin, as discusses in the external clock modes.

For information on configuraing the clock/ oscillator modes, refer the datasheet regarding the various SFRs used for this purpose.

2.2 Memory Organization

To use the 4550 judiciously for various applications, it is essential to understand a memory map of the device. There are three types of memory in PIC18 enhanced microcontroller devices.

- Program memory
- Data memory
- Data EEPROM

As discussed, the 4550 is a Harvard architecture device. Hence concurrent access to program and data memory is enabled. The data EEPROM, for practical purposes, can be regarded as a peripheral device, since it is addressed and accessed through a set of control registers.

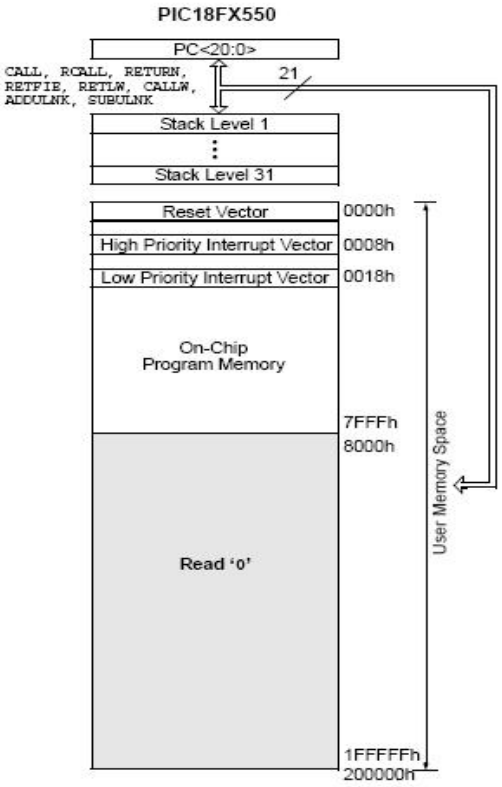

Figure 1: Program Memory Organization in PIC18F4550

2.2.1 Program Memory Organization

The 4550 implements a 21-bit program counter which is capable of addressing a 2MB program memory space. It has 32KB of Flash memory and can store up to 16,384 single-word instructions. Figure 1 shows the program memory organization.

2.2.2 Data Memory Organization

The data memory in the 4550 is implemented as static RAM. Each register in the data memory has a 12-bit address, thus allowing up to 4096 bytes of data memory. The memory space in the PIC18 family is divided into as many as 16 banks that contain 256 bytes each. The 4550 in particular implements eight complete banks, for a total of 2048 bytes. Figure 2 shows the data memory organization for this microcontroller.

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. A result of a read operation of an unimplemented location will be read as '0's.

The instruction set and architecture allow operations across all banks. The entire data memory may be accessed by direct, indirect or indexed addressing modes. To ensure that commonly used registers (SFRs and select GPRs) can be accessed in a single cycle, PIC18 devices implement an Access Bank. This is a 256-byte memory space that provides fast access to SFRs and the lower portion of GPR Bank 0 without using the Bank Select Register (BSR).

2.2.3 Data EEPROM

The data EEPROM in 4550 is a nonvolatile memory array, separate from the data RAM and program memory, that is used for long-term storage of program data. It is not directly mapped

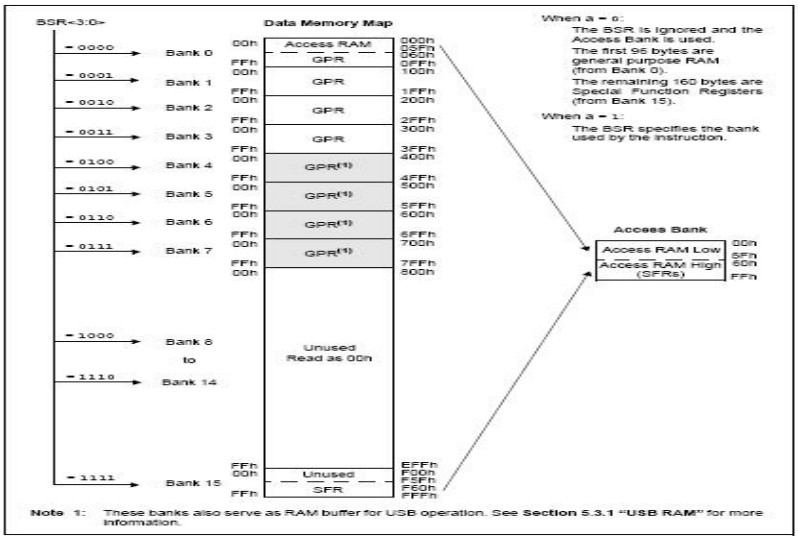

Figure 2: Data Memory Organization in PIC18F4550

in either the register file or program memory space, but is indirectly addressed through SFRs. The EEPROM can be read and written during normal operation, working over the entire supply voltage range. Four SFRs are used to read and write to the data EEPROM as well as the program memory. These are the EECON1, EECON2, EEADR and EEDATA.

The data EEPROM allows byte read and write. When interfacing to the data memory block, EEDATA holds the 8-bit data for read/write and the EEADR register holds the address of the EEPROM location being accessed. The EEPROM data memory is rated for high erase/write cycle endurance. 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; it will vary with voltage and temperature as well as from chip to chip.

2.3 Interrupt Structure

The 4550 has 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 location 000008H and the low priority interrupt vector is at 000018H. There are ten registers which are used for interrupt handling. These registers are:

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

Refer the datasheet for information on how to use these registers.

Each interrupt source has three bits to control its operation. The functions of these bits are:

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

The interrupt priority feature is enabled by setting the IPEN bit (RCON bit 7). When interrupt priority is enabled, there are two bits which enable interrupts globally. Setting the GIEH bit (INTCON bit 7) enables all interrupts that have the priority bit set (high priority). Setting the GIEL bit (INTCON bit 6) enables all interrupts that have the priority bit cleared (low priority). 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 bit setting. Individual interrupts can be disabled through their corresponding enable bits.

When the IPEN bit is cleared (default state), the interrupt priority feature is disabled and interrupts are compatible with PIC mid-range devices. In Compatibility mode, the interrupt priority bits for each source have no effect. INTCON bit 6 is the PEIE bit which enables/disables all peripheral interrupt sources. INTCON bit 7 is the GIE bit which 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. Low priority interrupts are not processed while a high priority interrupt service is in progress.

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. Individual interrupt flag bits are set regardless of the status of their corresponding enable bit or the GIE bit. The block diagram of the interrupt logic is shown in Figure 3.

2.4 General Purpose I/O Ports

The 4550 has five I/O ports depending on their configuration. Some pins of the I/O ports are multiplexed with an alternate function from the peripheral features of the device. When the particular peripheral is enabled the pin will not be used as a general-purpose I/O. Each port has three registers for its operation. These registers are:

- TRISx register (data direction register of Port x)
- PORTx register (indicates the levels on the pins of Port x)
- LATx register (output latch of Port x)

For information on the functions of these registers, refer the datasheet. Along with the above registers, there are other registers associated with Port A to configure the pin as the one belonging to peripheral or an I/O pin in general. Figure 4 shows an I/O port structure of the 4550.

2.5 On-chip Timers

The 4550 contains four timer modules- timer 0,1,2 and 3. A brief description of each timer is provided in this section.

2.5.1 Timer0 Module

The Timer0 module incorporates the following features:

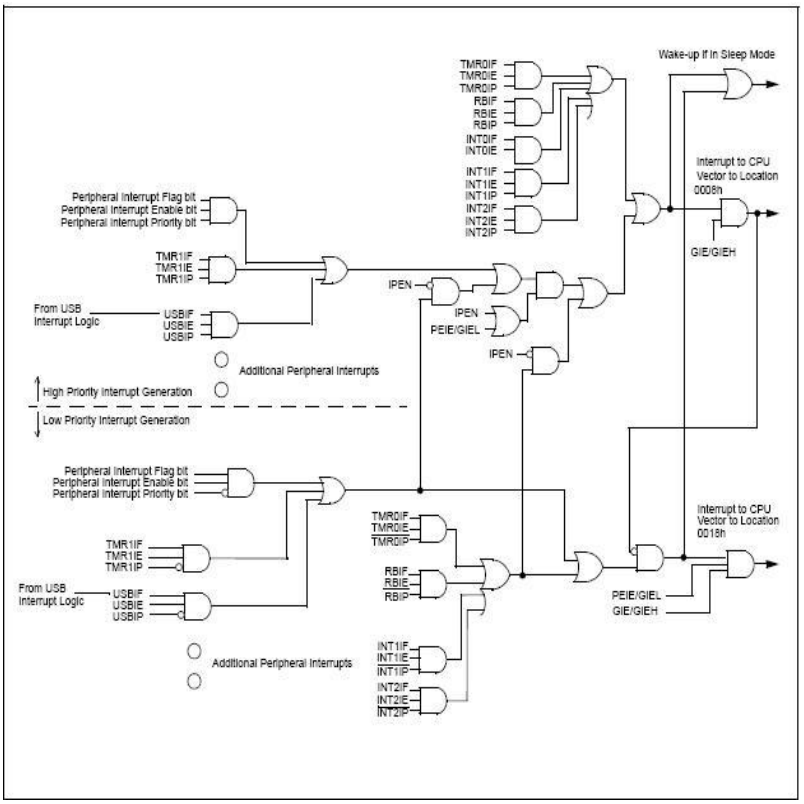

Figure 3: Interrupt Structure of PIC18F4550

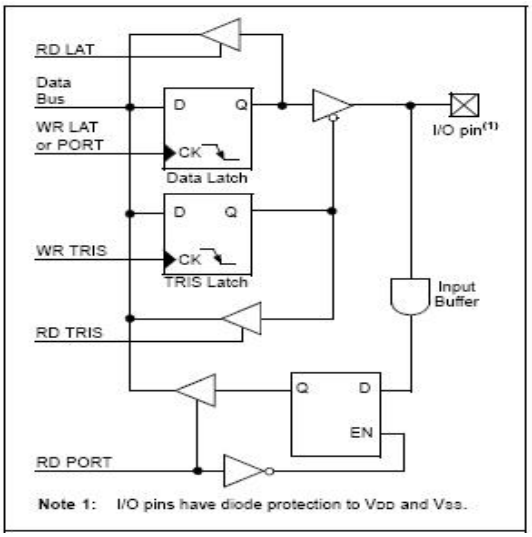

Figure 4: I/O Port logic of PIC18F4550

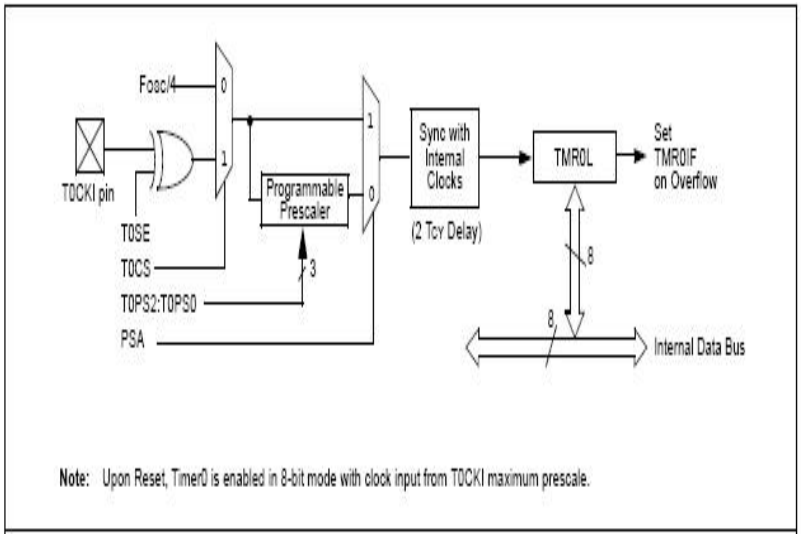

Figure 5: Timer0 operation in 8-bit mode

- Software selectable operation as a timer or counter in both 8-bit or 16-bit modes.
- Readable and writable registers.
- Dedicated 8-bit, software programmable prescaler.
- Selectable clock source (internal or external).
- Edge select for external clock.
- Interrupt on overflow.

The T0CON register controls all aspects of the module's operation, including the prescale selection. It is both readable and writable. Figure 5 shows the Timer0 block diagram in 8-bit mode.

2.5.2 Timer1 Module

In addition to the features provided by the Timer0 module, the Timer1 module incorporates:

- Module reset on CCP special event trigger
- Device clock status flag (T1RUN)

Figure 6 shows a simplified block diagram of the Timer1 module. It incorporates its own low-power oscillator to provide an additional clocking option. The Timer1 oscillator can also be used as a low-power clock source for the microcontroller in power-managed operation. It can also be used to provide Real-Time Clock (RTC) functionality to applications with only a minimal addition of external components and code overhead. Timer1 is controlled through the T1CON Control register. Timer1 can operate in one of these modes:

- Timer
- Synchronous Counter
- Asynchronous Counter

The operating mode is determined by the clock select bit, TMR1CS (T1CON bit 1). When TMR1CS is cleared, Timer1 increments on every internal instruction cycle (FOSC/4). When set, Timer1 increments on every rising edge of the Timer1 external clock input or the Timer1 oscillator, if enabled.

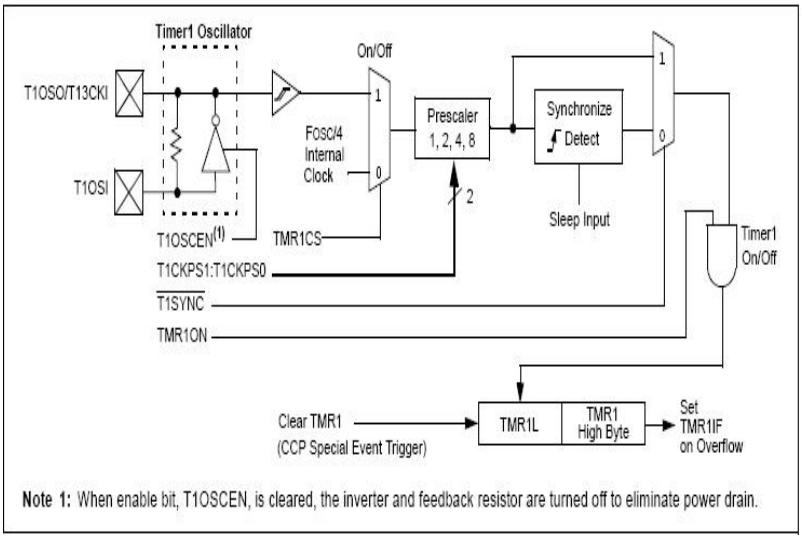

Figure 6: Simplified Timer1 module

2.5.3 Timer2 Module

The Timer2 module timer incorporates the following features:

- 8-bit timer and period registers (TMR2 and PR2 respectively)
- Readable and writable (both registers)
- Software programmable prescaler (1:1, 1:4 and 1:16)
- Software programmable postscaler (1:1 through 1:16)
- Interrupt on TMR2 to PR2 match

Timer2 is controlled through the T2CON register. In normal operation, TMR2 is incremented from 00h on each clock (FOSC/4). A 2-bit counter/prescaler on the clock input gives direct input, divide-by-4 and divide-by-16 prescale options. These are selected by the prescaler control bits, T2CKPS1:T2CKPS0 (T2CON[1:0]). The value of TMR2 is compared to that of the period register, PR2, on each clock cycle. When the two values match, the comparator generates a match signal as the timer output. This signal also resets the value of TMR2 to 00h on the next cycle and drives the output counter/postscaler. Figure 7 shows the block diagram of Timer2.

The Timer3 module operates similar to that of the Timer1 module. Refer the datasheet (Section 14.0) for more details.

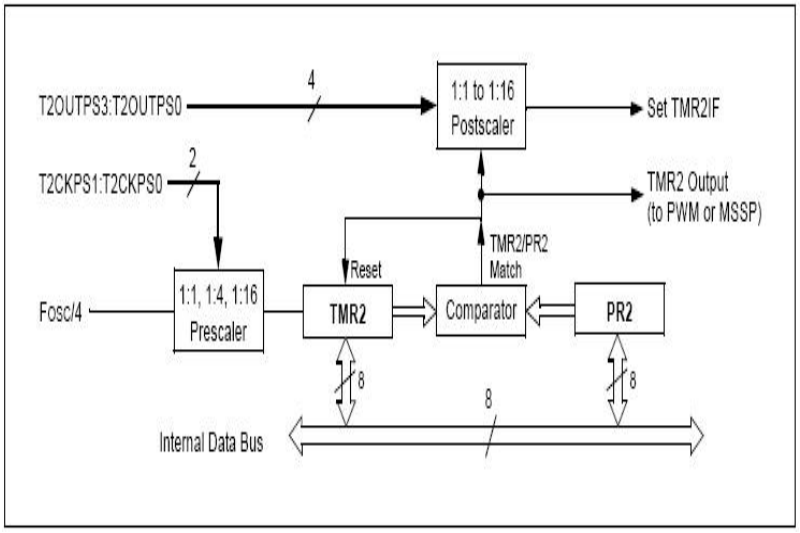

Figure 7: Timer2 Block Diagram

2.5.4 Analog-to-Digital Converters

The Analog-to-Digital (A/D) converter of the 4550 has 13 multiplexed inputs for the 40/44-pin devices. This module allows conversion of an analog input signal to a corresponding 10-bit digital code.The module has five registers:

- A/D Result High Register (ADRESH)
- A/D Result Low Register (ADRESL)
- A/D Control Register 0 (ADCON0)
- A/D Control Register 1 (ADCON1)
- A/D Control Register 2 (ADCON2)

The ADCON0 register controls the operation of the A/D module, while the ADCON1 register configures the functions of the port pins. The ADCON2 register configures the A/D clock source, programmed acquisition time and justification. Figure 8 shows the block diagram of the A/D converter.

2.6 Universal Serial Bus-USB 2.0

The PIC18FX455/X550 device family contains a full-speed and low-speed compatible USB Serial Interface Engine (SIE) that allows fast communication between any USB host and the PIC microcontroller. The SIE can be interfaced directly to the USB, utilizing the internal transceiver, or it can be connected through an external transceiver. An internal 3.3V regulator is also available to power the internal transceiver in 5V applications.

Some special hardware features have been included to improve performance. Dual port memory (refer Section 17.3 in datasheet) in the device's data memory space (USB RAM) has been supplied to share direct memory access between the microcontroller core and the SIE. Buffer descriptors (refer Section 17.4) are also provided, allowing users to freely program endpoint memory usage within the USB RAM space. A Streaming Parallel Port (refer Section 17.7) has been provided to support the uninterrupted transfer of large volumes of data, such as isochronous data, to external memory buffers. Figure 9 indicates the architecture of the USB module.

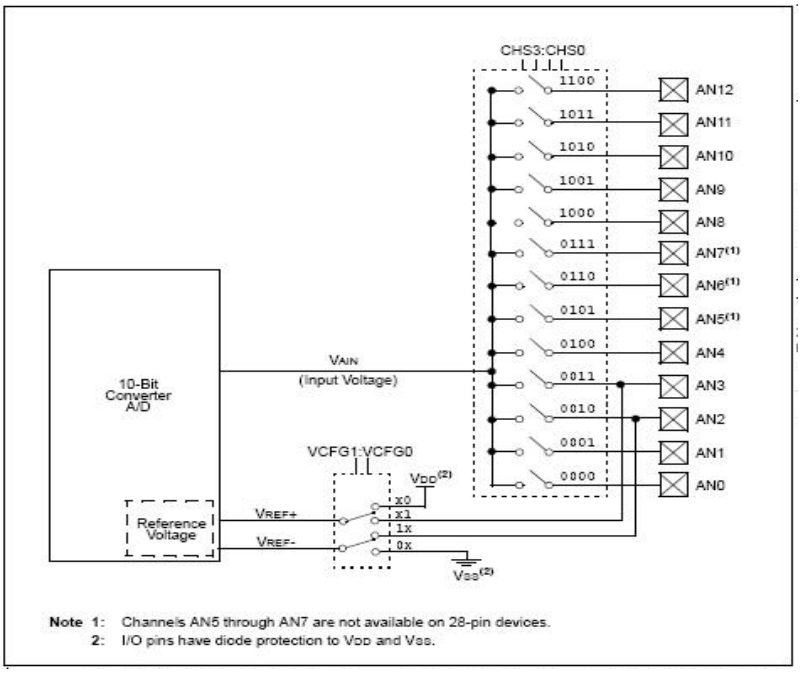

Figure 8: A/D Converter in PIC18F4550

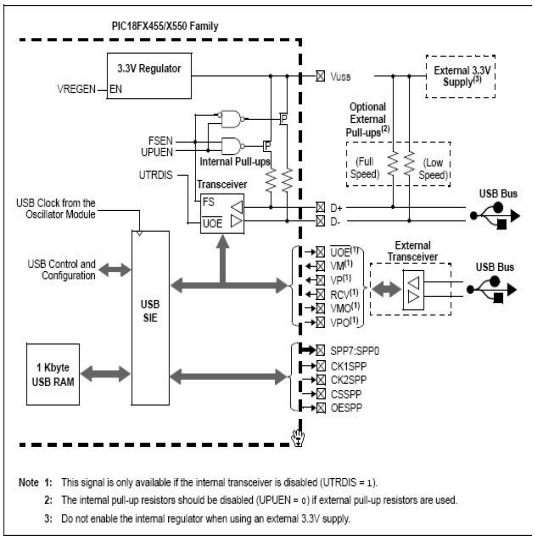

Figure 9: USB Module in PIC18F4550

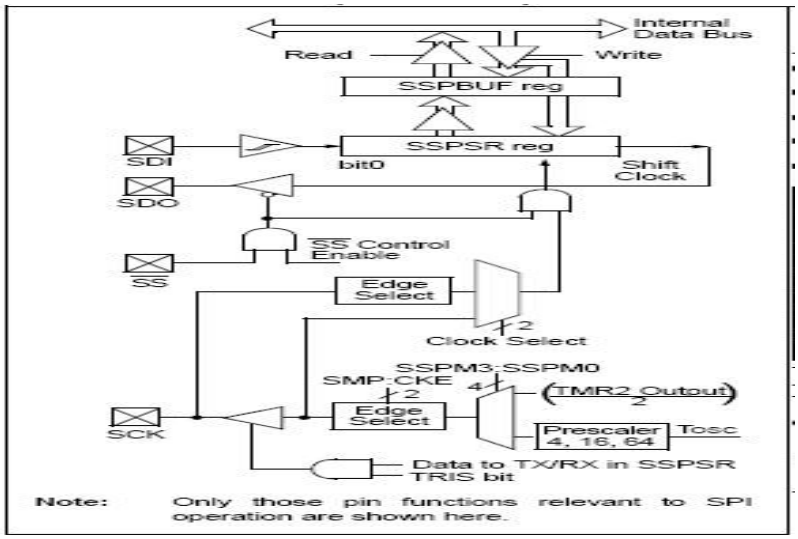

Figure 10: Block diagram of the SPI Communication module

2.7 Master Synchronous Serial Port (MSSP)

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, to name a few.

The MSSP module can operate in one of two modes:

- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I2C)

in control modes

- Full Master mode
- Slave mode (with general address call)

The following subsections discuss in brief, the working of the SPI and I2C bus modes.

2.7.1 SPI Mode

The SPI mode allows 8 bits of data to be synchronously transmitted and received simultaneously. All four modes of SPI are supported. To accomplish communication, typically three pins are used:

- Serial Data Out (SDO) – RC7/RX/DT/SDO
- Serial Data In (SDI) – RB0/AN12/INT0/FLT0/SDI/SDA
- Serial Clock (SCK) – RB1/AN10/INT1/SCK/SCL

Additionally, a fourth pin may be used in a slave mode of operation, the Slave Select (SS) – RA5/AN4/SS/HLVDIN/C2OUT pin. Figure 10 shows the block diagram of the SPI communication module.

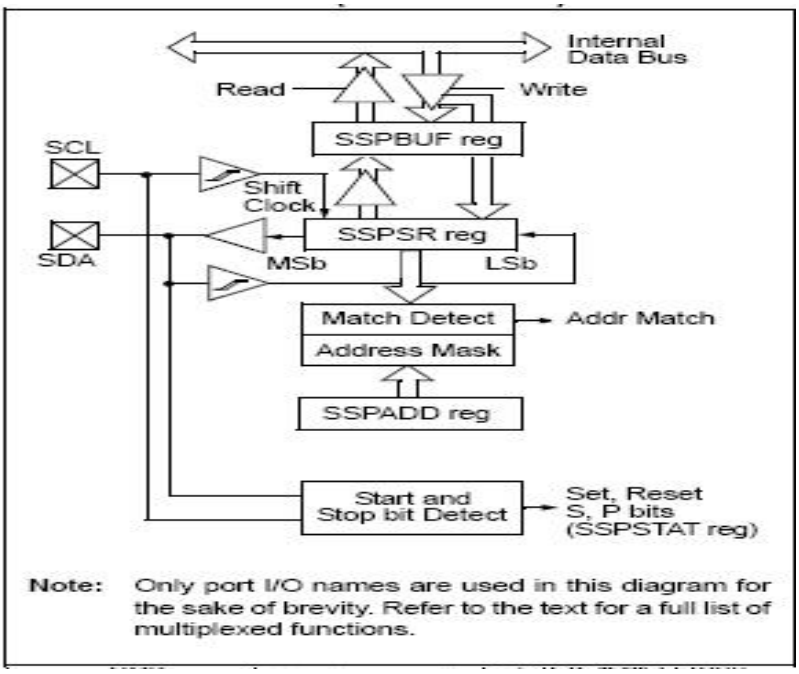

Figure 11: Block diagram of the I2C communication module

2.7.2 I2C Mode

The MSSP module in I2C 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.The I2C interface supports the following modes in hardware:

- Master mode
- Multi-Master mode
- Slave mode

Two pins are used for data transfer:

- Serial clock (SCL) – RB1/AN10/INT1/SCK/SCL
- Serial data (SDA) – RB0/AN12/INT0/FLT0/SDI/SDA

The user must configure these pins as inputs by setting the associated TRISx bits (Where x denotes the port name). Figure 11 shows the block diagram of the I2C communication module.

3 The PIC18F4550 Instruction Set

The 4550 has instructions which are 16-bit wide, which can be partitioned into an opcode, with one or more operands. The instruction set is highly orthogonal, which means that instructions with similar functions are not implemented. This is the methodology of RISC architecture, to focus on the prospective applications of the MCU and thereby doing away with unwanted instructions. The instruction set may be categorized into the following types based on their operations.

- Byte-oriented operations
- Bit-oriented operations
- Literal operations
- Control operations

For **Byte-oriented instructions**, the field `f` represents a file register designator and `d` represents a destination designator. The file register designator specifies which file register is to be used by the instruction. For example, the instruction `ADDWF f,d` adds the contents of registers `f` and `W`, while `d` indicates which of `f` or `W` will be the desitination. If `d` is zero, the result is placed in the `W` register. If `d` is one, the result is placed in the file register specified in the instruction, i.e. `f`.

For **Bit-oriented instructions**, `b` represents a bit field designator which selects the number of the bit affected by the operation, while `f` represents the number of the file in which the bit is located.

For **Literal and Control operations**, `k` represents an eight or eleven bit constant or literal value. Read the datasheet (Section 26 onwards) for a comprehensive overview of the instruction set.

References

[1] *PIC18F2455/2550/4455/4550 Data Sheet*, Microchip Technology Inc., 2007

[2] *PICmicro Mid-Range MCU Family*, Microchip Technology Inc., 1997