MICROCHIP

Section 4. Architecture

HIGHLIGHTS

This section of the manual contains the following major topics:

4.1 Introduction

The high performance of the PIC18CXXX devices can be attributed to a number of architectural features commonly found in RISC microprocessors. These include:

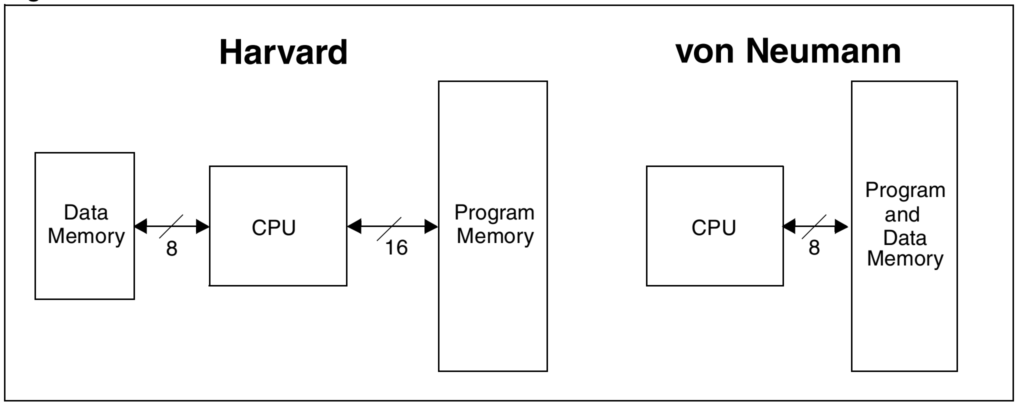
- Harvard architecture
- Long Word Instructions
- Single Word Instructions
- Single Cycle Instructions
- Instruction Pipelining
- Reduced Instruction Set
- Register File Architecture
- Orthogonal (Symmetric) Instructions

Figure 4-2 shows a general block diagram for PIC18CXXX devices.

Harvard Architecture:

Harvard architecture has the program memory and data memory as separate memories which are accessed from separate buses. This improves bandwidth over traditional von Neumann architecture in which program and data are fetched from the same memory using the same bus. To execute an instruction, a von Neumann machine must make one or more (generally more) accesses across the 8-bit bus to fetch the instruction. Then data may need to be fetched, operated on and possibly written. As can be seen from this description, the bus can become extremely congested. With a Harvard architecture, the instruction is fetched in a single instruction cycle (all 16 bits). While the program memory is being accessed, the data memory is on an independent bus and can be read and written. These separated busses allow one instruction to execute, while the next instruction is fetched. A comparison of Harvard and von Neumann architectures is shown in Figure 4-1.

Figure 4-1: Harvard vs. von Neumann Block Architectures



Long Word Instructions:

Long word instructions have a wider (more bits) instruction bus than the 8-bit data memory bus. This is possible because the two buses are separate. This allows instructions to be sized differently than the 8-bit wide data word and allows a more efficient use of the program memory, since the program memory width is optimized to the architectural requirements.

Single Word Instructions:

Single word instruction opcodes are 16-bits wide making it possible to have all but a few instructions be single word instructions. A 16-bit wide program memory access bus fetches a 16-bit instruction in a single cycle. With single word instructions, the number of words of program memory locations equals the number of instructions for the device. This means that all locations are valid instructions.

Typically in the von Neumann architecture, most instructions are multi-byte. In general, a device with 4 Kbytes of program memory would allow approximately 2K of instructions. This 2:1 ratio is generalized and dependent on the application code. Since each instruction may take multiple bytes, there is no assurance that each location is a valid instruction.

Double Word Instructions:

Some operations require more information then can be stored in the 16 bits of a program memory location. These operations require a double word instruction, and are therefore 32-bits wide. Instructions that require this second instruction word are:

- Memory to memory move instruction (12 bits for each RAM address)
  - `MOVFF   SourceReg, DestReg`
- Literal value to FSR move instruction (12 bits for data and 2 bits for FSR to load)
  - `LFSR    FSR#, Address`
- Call and goto operations (20 bits for address)
  - `CALL    Address`
  - `GOTO    Address`

The first word indicates to the CPU that the next program memory location is the additional information for this instruction and not an instruction. If the CPU tries to execute the second word of an instruction (due to a software modified PC pointing to that location as an instruction), the fetched data is executed as a `NOP`.

Double word instruction execution is not split between the two TCY cycles by an interrupt request. That is, when an interrupt request occurs during the execution of a double word instruction, the execution of the instruction is completed before the processor vectors to the interrupt address. The interrupt latency is preserved.

Instruction Pipeline:

The instruction pipeline is a two-stage pipeline that overlaps the fetch and execution of instructions. The fetch of the instruction takes one TCY, while the execution takes another TCY. However, due to the overlap of the fetch of current instruction and execution of previous instruction, an instruction is fetched and another instruction is executed every TCY.

Single Cycle Instructions:

With the program memory bus being 16-bits wide, the entire instruction is fetched in a single machine cycle (TCY), except for double word instructions. The instruction contains all the information required and is executed in a single cycle. There may be a one cycle delay in execution if the result of the instruction modified the contents of the program counter. This requires the pipeline to be flushed and a new instruction to be fetched.

Two Cycle Instructions:

Double word instructions require two cycles to execute, since all the required information is in the 32 bits.

Reduced Instruction Set:

When an instruction set is well designed and highly orthogonal (symmetric), fewer instructions are required to perform all needed tasks. With fewer instructions, the whole set can be more rapidly learned.

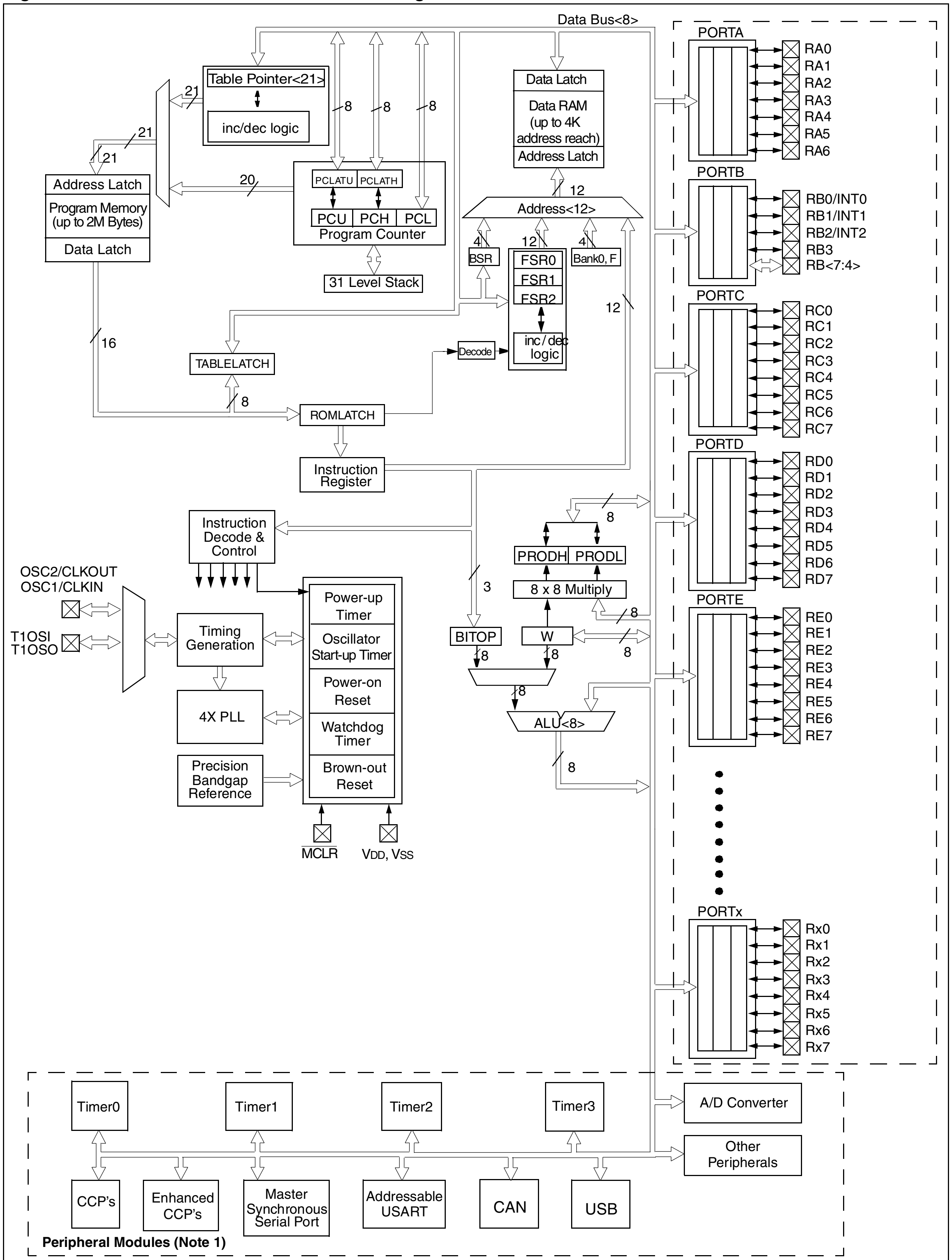
Register File Architecture:

The register files/data memory can be directly or indirectly addressed. All special function registers, including the program counter, are mapped in the data memory.

Orthogonal (Symmetric) Instructions:

Orthogonal instructions make it possible to carry out any operation on any register using any addressing mode. This symmetrical nature and lack of “special instructions” make programming simple yet efficient. In addition, the learning curve is reduced significantly. The Enhanced MCU instruction set uses only three non-register oriented instructions, which are used for two of the cores features. One is the `SLEEP` instruction, which places the device into the lowest power use mode. The second is the `CLRWDT` instruction, which verifies the chip is operating properly by preventing the on-chip Watchdog Timer (WDT) from overflowing and resetting the device. The third is the `RESET` instruction, which resets the device.

Figure 4-2: General Enhanced MCU Block Diagram

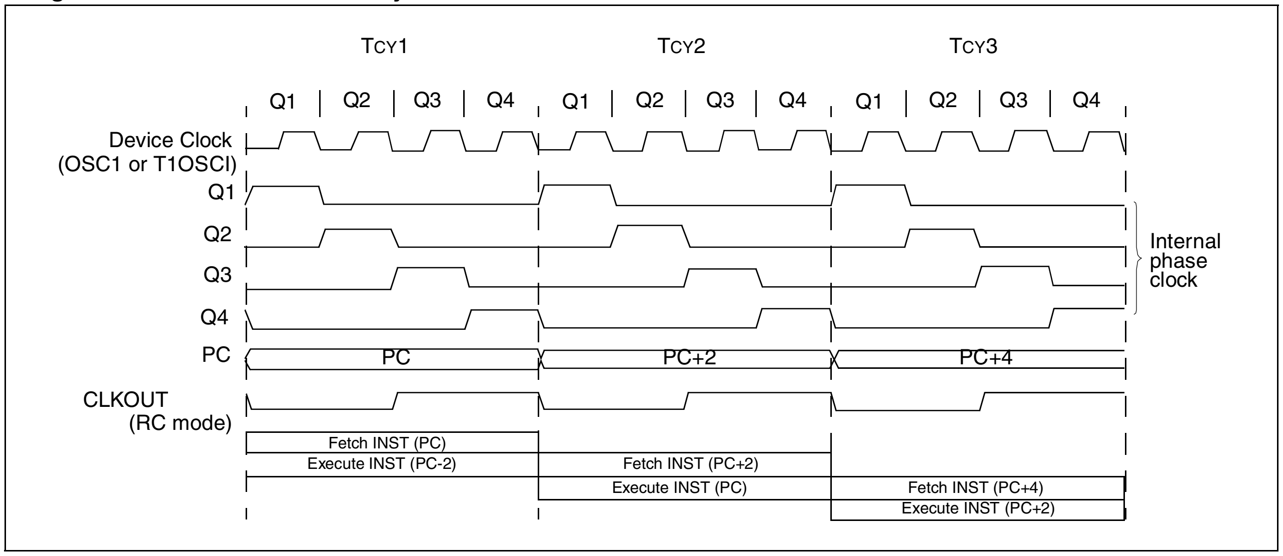


Note 1: Many of the general purpose I/O pins are multiplexed with one or more peripheral module functions. The multiplexing combinations are device dependent.

4.2 Clocking Scheme/Instruction Cycle

The clock input is internally divided by four to generate four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4. Internally, the program counter is incremented every Q1, and 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 illustrated in Figure 4-3 and Example 4-1.

Figure 4-3: Clock/Instruction Cycle

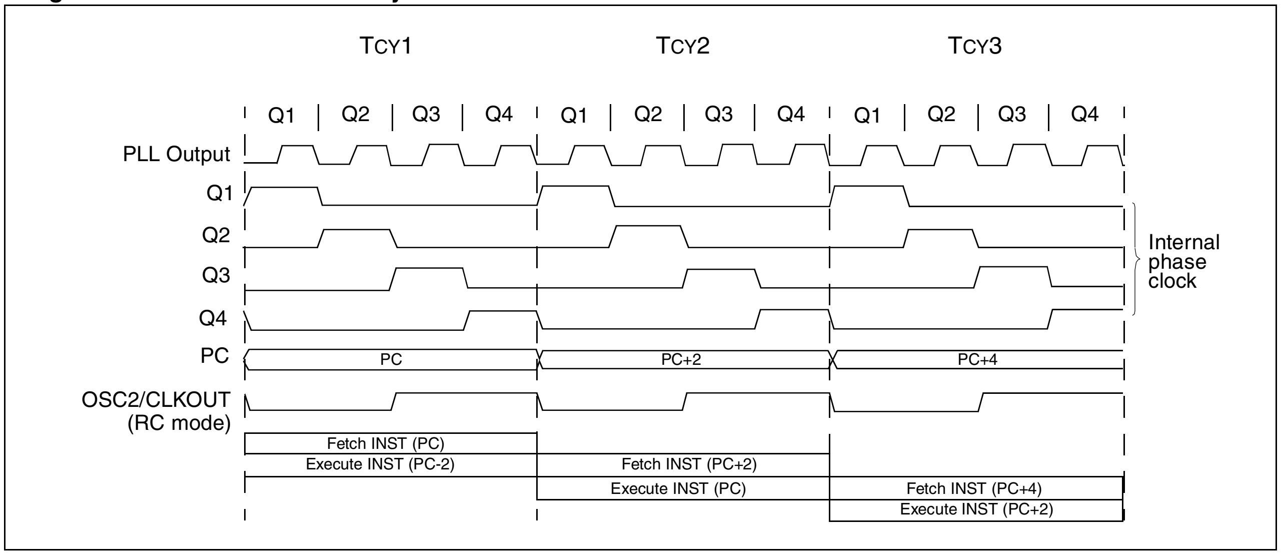


4.2.1 Phase Lock Loop (PLL)

The clock input is multiplied by four by the PLL. Therefore, when it is internally divided by four, it provides an instruction cycle that is the same frequency as the external clock frequency. Four non-overlapping quadrature clocks, namely Q1, Q2, Q3 and Q4 are still generated internally.

Internally, the program counter (PC) is incremented every Q1, and 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 illustrated in Figure 4-4 and Example 4-1.

Figure 4-4: Clock/Instruction Cycle with PLL



4.3 Instruction Flow/Pipelining

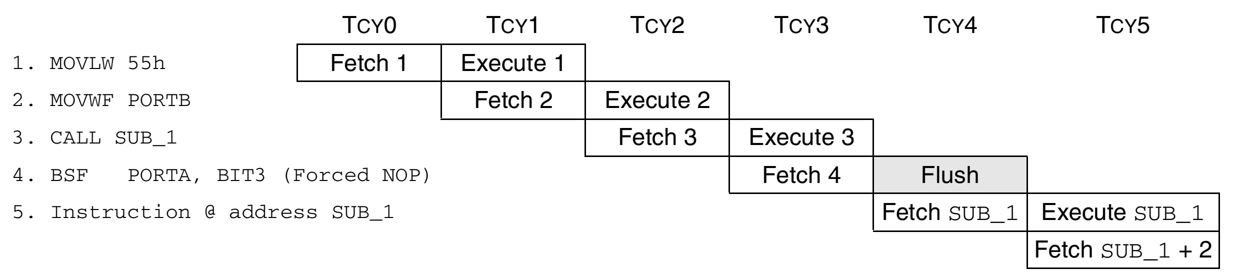
An “Instruction Cycle” consists of four Q cycles (Q1, Q2, Q3 and Q4). Fetch takes one instruction cycle, while decode and execute takes another instruction cycle. However, due to pipelining, each instruction effectively executes in one cycle. If an instruction causes the program counter to change (e.g. `GOTO` instruction), then an extra cycle is required to complete the instruction (See Example 4-1).

The instruction **fetch** begins with the program counter 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).

Example 4-1 shows the operation of the two stage pipeline for the instruction sequence shown. At time TCY0, the first instruction is fetched from program memory. During TCY1, the first instruction executes, while the second instruction is fetched. During TCY2, the second instruction executes, while the third instruction is fetched. During TCY3, the fourth instruction is fetched, while the third instruction (`CALL SUB_1`) is executed. When the third instruction completes execution, the CPU forces the address of instruction four onto the Stack and then changes the Program Counter (PC) to the address of `SUB_1`. This means that the instruction that was fetched during TCY3 needs to be “flushed” from the pipeline. During TCY4, instruction four is flushed (executed as a `NOP`) and the instruction at address `SUB_1` is fetched. Finally during TCY5, instruction five is executed and the instruction at address `SUB_1` + 2 is fetched.

Example 4-1: Instruction Pipeline Flow



Most instructions are single cycle. Program branches take two cycles, since the fetch instruction is “flushed” from the pipeline while the new instruction is being fetched and then executed.

4.4 I/O Descriptions

Table 4-1 gives a brief description of device pins and the functions that may be multiplexed to a port pin. Multiple functions may exist on one port pin. When multiplexing occurs, the peripheral module’s functional requirements may force an override of the data direction (TRIS bit) of the port pin (such as in the A/D and Comparator modules).

Table 4-1: I/O Descriptions

Pin Name	Pin Type	Buffer Type	Description
A19	O	—	System bus address line 19
A18	O	—	System bus address line 18
A17	O	—	System bus address line 17
A16	O	—	System bus address line 16
AD15	I/O	TTL	System bus address/data line 15
AD14	I/O	TTL	System bus address/data line 14
AD13	I/O	TTL	System bus address/data line 13
AD12	I/O	TTL	System bus address/data line 12
AD11	I/O	TTL	System bus address/data line 11
AD10	I/O	TTL	System bus address/data line 10
AD9	I/O	TTL	System bus address/data line 9
AD8	I/O	TTL	System bus address/data line 8
AD7	I/O	TTL	System bus address/data line 7
AD6	I/O	TTL	System bus address/data line 6
AD5	I/O	TTL	System bus address/data line 5
AD4	I/O	TTL	System bus address/data line 4
AD3	I/O	TTL	System bus address/data line 3
AD2	I/O	TTL	System bus address/data line 2
AD1	I/O	TTL	System bus address/data line 1
AD0	I/O	TTL	System bus address/data line 0
ALE	O	—	System bus address latch enable strobe
			Analog Input Channels
AN0	I	Analog	
AN1	I	Analog	
AN2	I	Analog	
AN3	I	Analog	
AN4	I	Analog	
AN5	I	Analog	
AN6	I	Analog	
AN7	I	Analog	
AN8	I	Analog	
AN9	I	Analog	
AN10	I	Analog	
AN11	I	Analog	
AN12	I	Analog	
AN13	I	Analog	
AN14	I	Analog	
AN15	I	Analog	
AVDD	P	P	Analog Power

Legend: TTL = TTL-compatible input
ST = Schmitt Trigger input with CMOS levels
PU = Weak internal pull-up
Analog = Analog input or output
CMOS = CMOS compatible input or output
O = output
I = input
P = Power

Table 4-1: I/O Descriptions (Continued)

Pin Name	Pin Type	Buffer Type	Description
AVSS	P	P	Analog Ground
BA0	O	—	System bus byte address 0
CANRX	I	ST	CAN bus receive pin
CANTX0	O	—	CAN bus transmit
CANTX1	O	—	CAN bus complimentary transmit or CAN bus bit time clock
CCP1	I/O	ST	Capture1 input/Compare1 output/PWM1 output
CCP2	I/O	ST	Capture2 input/Compare2 output/PWM2 output.
CK	I/O	ST	USART Synchronous Clock, always associated with TX pin function (See related TX, RX, DT)
CLKI	I	ST/CMOS	External clock source input. Always associated with pin function OSC1. (See related OSC1/CLKIN, OSC2/CLKOUT pins)
CLKO	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate. Always associated with OSC2 pin function. (See related OSC2, OSC1)
CMPA	O	—	Comparator A output
CMPB	O	—	Comparator B output
$\overline{\text{CS}}$	I	TTL	Chip select control for parallel slave port (See related $\overline{\text{RD}}$ and $\overline{\text{WR}}$)
CVREF	O	Analog	Comparator voltage reference output
DT	I/O	ST	USART Synchronous Data. Always associated RX pin function. (See related RX, TX, CK)

Legend: TTL = TTL-compatible input	CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels	O = output
PU = Weak internal pull-up	I = input
Analog = Analog input or output	P = Power

Table 4-1: I/O Descriptions (Continued)

Pin Name	Pin Type	Buffer Type	Description
INT0	I	ST	External Interrupt0
INT1	I	ST	External Interrupt1
INT2	I	ST	External Interrupt2
$\overline{\text{LB}}$	O	—	System bus low byte strobe
LVDIN	I	Analog	Low voltage detect input
$\overline{\text{MCLR}}$	I/P	ST	Master clear (reset) input or programming voltage input. This pin is an active low reset to the device.
NC	—	—	These pins should be left unconnected.
$\overline{\text{OE}}$	O	—	System bus output enable strobe
OSC1	I	ST/CMOS	Oscillator crystal input or external clock source input. ST buffer when configured in RC mode. CMOS otherwise.
OSC2	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
PSP0	I/O	TTL	Parallel Slave Port for interfacing to a microprocessor port. These pins have TTL input buffers when PSP module is enabled.
PSP1	I/O	TTL	
PSP2	I/O	TTL	
PSP3	I/O	TTL	
PSP4	I/O	TTL	
PSP5	I/O	TTL	
PSP6	I/O	TTL	
PSP7	I/O	TTL	
			PORTA is a bi-directional I/O port.
RA0	I/O	TTL	
RA1	I/O	TTL	
RA2	I/O	TTL	
RA3	I/O	TTL	
RA4	I/O	ST	RA4 is an open drain when configured as output.
RA5	I/O	TTL	
RA6	I/O	TTL	
			PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0	I/O	TTL	
RB1	I/O	TTL	
RB2	I/O	TTL	
RB3	I/O	TTL	
RB4	I/O	TTL	Interrupt on change pin.
RB5	I/O	TTL	Interrupt on change pin.
RB6	I/O	TTL/ST	Interrupt on change pin. Serial programming clock. TTL input buffer as general purpose I/O, Schmitt Trigger input buffer when used as the serial programming clock.
RB7	I/O	TTL/ST	Interrupt on change pin. Serial programming data. TTL input buffer as general purpose I/O, Schmitt Trigger input buffer when used as the serial programming data.

Legend: TTL = TTL-compatible input — CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels — O = output
PU = Weak internal pull-up — I = input
Analog = Analog input or output — P = Power

Table 4-1: I/O Descriptions (Continued)

Pin Name	Pin Type	Buffer Type	Description
			PORTC is a bi-directional I/O port.
RC0	I/O	ST	
RC1	I/O	ST	
RC2	I/O	ST	
RC3	I/O	ST	
RC4	I/O	ST	
RC5	I/O	ST	
RC6	I/O	ST	
RC7	I/O	ST	
$\overline{\text{RD}}$	I	TTL	Read control for parallel slave port. (See also $\overline{\text{WR}}$ and $\overline{\text{CS}}$ pins.)
			PORTD is a bi-directional I/O port.
RD0	I/O	ST	
RD1	I/O	ST	
RD2	I/O	ST	
RD3	I/O	ST	
RD4	I/O	ST	
RD5	I/O	ST	
RD6	I/O	ST	
RD7	I/O	ST	
			PORTE is a bi-directional I/O port.
RE0	I/O	ST	
RE1	I/O	ST	
RE2	I/O	ST	
RE3	I/O	ST	
RE4	I/O	ST	
RE5	I/O	ST	
RE6	I/O	ST	
RE7	I/O	ST	
			PORTF is a digital input
RF0	I/O	ST	
RF1	I/O	ST	
RF2	I/O	ST	
RF3	I/O	ST	
RF4	I/O	ST	
RF5	I/O	ST	
RF6	I/O	ST	
RF7	I/O	ST	

Legend: TTL = TTL-compatible input	CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels	O = output
PU = Weak internal pull-up	I = input
Analog = Analog input or output	P = Power

Table 4-1: I/O Descriptions (Continued)

Pin Name	Pin Type	Buffer Type	Description
			PORTG is a digital input
RG0	I/O	ST	
RG1	I/O	ST	
RG2	I/O	ST	
RG3	I/O	ST	
RG4	I/O	ST	
RG5	I/O	ST	
RG6	I/O	ST	
RG7	I/O	ST	
			PORTH is a digital input
RH0	I/O	ST	
RH1	I/O	ST	
RH2	I/O	ST	
RH3	I/O	ST	
RH4	I/O	ST	
RH5	I/O	ST	
RH6	I/O	ST	
RH7	I/O	ST	
			PORTJ is a digital input
RJ0	I/O	ST	
RJ1	I/O	ST	
RJ2	I/O	ST	
RJ3	I/O	ST	
RJ4	I/O	ST	
RJ5	I/O	ST	
RJ6	I/O	ST	
RJ7	I/O	ST	
			PORTK is a digital input
RK0	I/O	ST	
RK1	I/O	ST	
RK2	I/O	ST	
RK3	I/O	ST	
RK4	I/O	ST	
RK5	I/O	ST	
RK6	I/O	ST	
RK7	I/O	ST	

Legend: TTL = TTL-compatible input — CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels — O = output
PU = Weak internal pull-up — I = input
Analog = Analog input or output — P = Power

Table 4-1: I/O Descriptions (Continued)

Pin Name	Pin Type	Buffer Type	Description
			PORTL is a digital input
RL0	I/O	ST	
RL1	I/O	ST	
RL2	I/O	ST	
RL3	I/O	ST	
RL4	I/O	ST	
RL5	I/O	ST	
RL6	I/O	ST	
RL7	I/O	ST	
RX	I	ST	USART Asynchronous Receive
SCL	I/O	ST	Synchronous serial clock input/output for I^2C mode.
SCLA	I/O	ST	Synchronous serial clock for I^2C interface.
SCLB	I/O	ST	Synchronous serial clock for I^2C interface.
SDA	I/O	ST	$I^2C^{™}$ Data I/O
SDAA	I/O	ST	Synchronous serial data I/O for I^2C interface
SDAB	I/O	ST	Synchronous serial data I/O for I^2C interface
SCK	I/O	ST	Synchronous serial clock input/output for SPI mode.
SDI	I	ST	SPI Data In
SDO	O	—	SPI Data Out (SPI mode)
$\overline{\text{SS}}$	I	ST	SPI Slave Select input
T0CKI	I	ST	Timer0 external clock input
T1CKI	I	ST	Timer1 external clock input
T1OSO	O	CMOS	Timer1 oscillator output
T1OSI	I	CMOS	Timer1 oscillator input
TX	O	—	USART Asynchronous Transmit (See related RX)
$\overline{\text{UB}}$	O	—	System bus upper byte strobe

Legend: TTL = TTL-compatible input	CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels	O = output
PU = Weak internal pull-up	I = input
Analog = Analog input or output	P = Power

Table 4-1: I/O Descriptions (Continued)

Pin Name	Pin Type	Buffer Type	Description
VREF	I	Analog	Analog High Voltage Reference input. DR reference voltage output on devices with comparators.
VREF+	I	Analog	Analog High Voltage Reference input. Usually multiplexed onto an analog pin.
VREF-	I	Analog	Analog Low Voltage Reference input. Usually multiplexed onto an analog pin.
VSS	P	—	Ground reference for logic and I/O pins.
VDD	P	—	Positive supply for logic and I/O pins.
VPP	P	—	Programming voltage input
$\overline{\text{WR}}$	I	TTL	Write control for parallel slave port (See $\overline{\text{CS}}$ and $\overline{\text{RD}}$ pins also).
$\overline{\text{WRL}}$	O	—	System bus write low byte strobe
$\overline{\text{WRH}}$	O	—	System bus write high byte strobe

Legend: TTL = TTL-compatible input CMOS = CMOS compatible input or output
ST = Schmitt Trigger input with CMOS levels O = output
PU = Weak internal pull-up I = input
Analog = Analog input or output P = Power

4.5 Design Tips

No related design tips at this time.

4.6 Related Application Notes

This section lists application notes that are related to this section of the manual. These application notes may not be written specifically for the Enhanced family (that is, they may be written for the Base-Line, the Mid-Range, or High-End families), but the concepts are pertinent and could be used (with modification and possible limitations). The current application notes related to Architecture are:

Title	**Application Note #**
No related application notes at this time.	

Note: Please visit the Microchip Web site for additional software code examples. These code examples are stand alone examples to assist in the understanding of the PIC18CXXX. The web address for these examples is:

http://www.microchip.com/10/faqs/codeex/

4.7 Revision History

Revision A

This is the initial released revision of the Enhanced MCU Architecture description.