Section 5. CPU and ALU

HIGHLIGHTS

This section of the manual contains the following major topics:

5.1 Introduction

The Central Processing Unit (CPU) is responsible for using the information in the program memory (instructions) to control the operation of the device. Many of these instructions operate on data memory. To operate on data memory, the Arithmetic Logical Unit (ALU) is required. In addition to performing arithmetical and logical operations, the ALU controls status bits (which are found in the STATUS register). The result of some instructions force status bits to a value depending on the state of the result.

The machine codes that the CPU recognizes are show in Table 5-1 (as well as the instruction mnemonics that the MPASM uses to generate these codes).

Table 5-1: Mid-Range MCU Instruction Set

Mnemonic, Operands		Description	Cycles	14-Bit Instruction Word				Status Bits Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lfff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lfff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff-	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

Note 1: When an I/O register is modified as a function of itself (e.g., `MOVF PORTB, 1`), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.

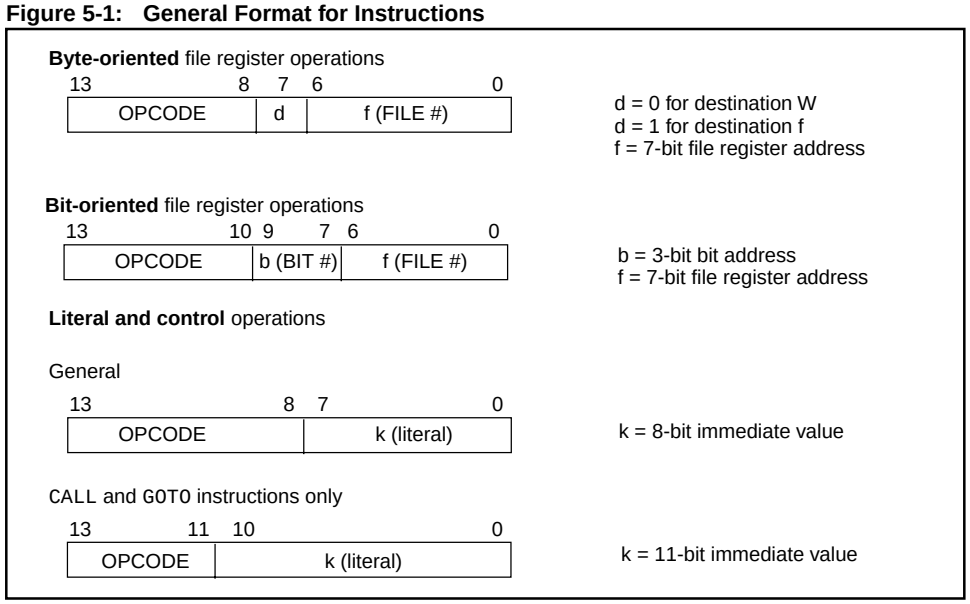
2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 Module.

3: If Program Counter (PC) is modified or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP.

5.2 General Instruction Format

The Mid-Range MCU instructions can be broken down into four general formats as shown in Figure 5-1. As can be seen the opcode for the instruction varies from 3-bits to 6-bits. This variable opcode size is what allows 35 instructions to be implemented.

Figure 5-1: General Format for Instructions



5.3 Central Processing Unit (CPU)

The CPU can be thought of as the “brains” of the device. It is responsible for fetching the correct instruction for execution, decoding that instruction, and then executing that instruction.

The CPU sometimes works in conjunction with the ALU to complete the execution of the instruction (in arithmetic and logical operations).

The CPU controls the program memory address bus, the data memory address bus, and accesses to the stack.

5.4 Instruction Clock

Each instruction cycle (TCY) is comprised of four Q cycles (Q1-Q4). The Q cycle time is the same as the device oscillator cycle time (TOSC). The Q cycles provide the timing/designation for the Decode, Read, Process Data, Write, etc., of each instruction cycle. The following diagram shows the relationship of the Q cycles to the instruction cycle.

The four Q cycles that make up an instruction cycle (TCY) can be generalized as:

Q1: Instruction Decode Cycle or forced No operation

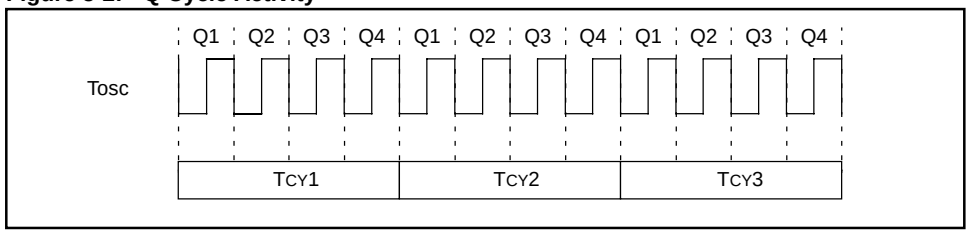
Q2: Instruction Read Data Cycle or No operation

Q3: Process the Data

Q4: Instruction Write Data Cycle or No operation

Each instruction will show a detailed Q cycle operation for the instruction.

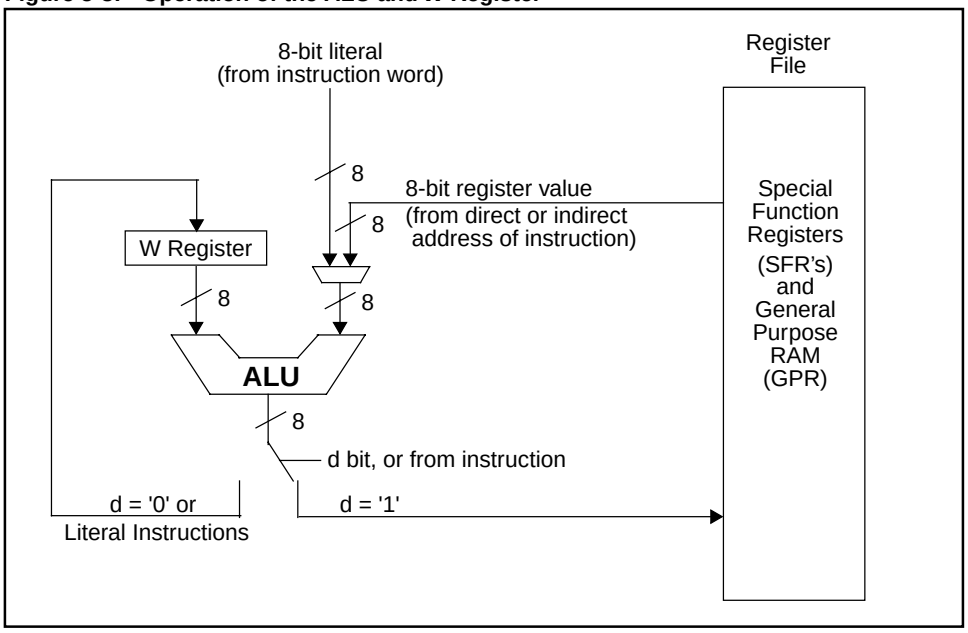
Figure 5-2: Q Cycle Activity



5.5 Arithmetic Logical Unit (ALU)

PICmicro MCUs contain an 8-bit ALU and an 8-bit working register. The ALU is a general purpose arithmetic and logical unit. It performs arithmetic and Boolean functions between the data in the working register and any register file.

Figure 5-3: Operation of the ALU and W Register



The ALU is 8-bits wide and is capable of addition, subtraction, shift and logical operations. Unless otherwise mentioned, arithmetic operations are two's complement in nature. 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.

The W register is an 8-bit working register used for ALU operations. It is not an addressable 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.

5.6 STATUS Register

The STATUS register, shown in Figure 5-1, contains the arithmetic status of the ALU, the RESET status and the bank select bits for data memory. Since the selection of the Data Memory banks is controlled by this register, it is required to be present in every bank. Also, this register is in the same relative position (offset) in each bank (see **Figure 6-5: "Register File Map"** in the **"Memory Organization"** section).

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 Table 5-1.

Note 1: Some devices do not require the IRP and RP1 (STATUS<7:6>) bits. These bits are not used by the Section 5. CPU and ALU and should be maintained clear. Use of these bits as general purpose R/W bits is NOT recommended, since this may affect upward code compatibility with future products.

Note 2: The C and DC bits operate as a $\overline{borrow}$ and $\overline{digit\ borrow}$ bit, respectively, in subtraction.

Register 5-1: STATUS Register

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit 7							bit 0

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)
1 = Bank 2, 3 (100h - 1FFh)
0 = Bank 0, 1 (00h - FFh)

For devices with only Bank0 and Bank1 the IRP bit is reserved, always maintain this bit clear.

bit 6:5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)
11 = Bank 3 (180h - 1FFh)
10 = Bank 2 (100h - 17Fh)
01 = Bank 1 (80h - FFh)
00 = Bank 0 (00h - 7Fh)

Each bank is 128 bytes. For devices with only Bank0 and Bank1 the IRP bit is reserved, always maintain this bit clear.

bit 4 **$\overline{TO}$**: Time-out bit
1 = After power-up, CLRWDT instruction, or SLEEP instruction
0 = A WDT time-out occurred

bit 3 **$\overline{PD}$**: Power-down bit
1 = After power-up or by the CLRWDT instruction
0 = By execution of the SLEEP instruction

bit2 **Z**: Zero bit
1 = The result of an arithmetic or logic operation is zero
0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions) (for $\overline{borrow}$ the polarity is reversed)
1 = A carry-out from the 4th low order bit of the result occurred
0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/$\overline{borrow}$ bit (ADDWF, ADDLW, SUBLW, SUBWF instructions)
1 = A carry-out from the most significant bit of the result occurred
0 = No carry-out from the most significant bit of the result occurred

Note: For $\overline{borrow}$ the polarity is reversed. 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.

Legend		
R = Readable bit	W = Writable bit	
U = Unimplemented bit, read as '0'		- n = Value at POR reset

5.7 OPTION_REG Register

The OPTION_REG register is a readable and writable register which contains various control bits to configure the TMR0/WDT prescaler, the external INT Interrupt, TMR0, and the weak pull-ups on PORTB.

Register 5-2: OPTION_REG Register

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
$\overline{\text{RBPU}}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7							bit 0

bit 7 **$\overline{\text{RBPU}}$:** PORTB Pull-up Enable bit
1 = PORTB pull-ups are disabled
0 = PORTB pull-ups are enabled by individual port latch values

bit 6 **INTEDG**: Interrupt Edge Select bit
1 = Interrupt on rising edge of INT pin
0 = Interrupt on falling edge of INT pin

bit 5 **T0CS**: TMR0 Clock Source Select bit
1 = Transition on T0CKI pin
0 = Internal instruction cycle clock (CLKOUT)

bit 4 **T0SE**: TMR0 Source Edge Select bit
1 = Increment on high-to-low transition on T0CKI pin
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA**: Prescaler Assignment bit
1 = Prescaler is assigned to the WDT
0 = Prescaler is assigned to the Timer0 module

bit 2-0 **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

Legend	
R = Readable bit	W = Writable bit
U = Unimplemented bit, read as '0'	- n = Value at POR reset

Note: To achieve a 1:1 prescaler assignment for the TMR0 register, assign the prescaler to the Watchdog Timer.

5.8 PCON Register

The Power Control (PCON) register contains flag bit(s), that together with the TO and PD bits, allows the user to differentiate between the device resets.

Note 1:	$\overline{BOR}$ is unknown on Power-on Reset. It must then be set by the user and checked on subsequent resets to see if $\overline{BOR}$ is clear, indicating a brown-out has occurred. The $\overline{BOR}$ status bit is a don't care and is not necessarily predictable if the brown-out circuit is disabled (by clearing the BODEN bit in the Configuration word).
Note 2:	It is recommended that the $\overline{POR}$ bit be cleared after a power-on reset has been detected, so that subsequent power-on resets may be detected.

Register 5-3: PCON Register

R-u	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
MPEEN	—	—	—	—	$\overline{PER}$	$\overline{POR}$	$\overline{BOR}$
bit 7							bit 0

bit 7 **MPEEN**: Memory Parity Error Circuitry Status bit
This bit reflects the value of the MPEEN configuration bit.

bit 6:3 **Unimplemented:** Read as '0'

bit 2 **$\overline{PER}$**: Memory Parity Error Reset Status bit
1 = No error occurred
0 = A program memory fetch parity error occurred
(must be set in software after a Power-on Reset occurs)

bit 1 **$\overline{POR}$**: Power-on Reset Status bit
1 = No Power-on Reset occurred
0 = A Power-on Reset occurred (must be set in software after a Power-on Reset occurs)

bit 0 **$\overline{BOR}$**: Brown-out Reset Status bit
1 = No Brown-out Reset occurred
0 = A Brown-out Reset occurred (must be set in software after a Brown-out Reset occurs)

Legend		
R = Readable bit	W = Writable bit	
U = Unimplemented bit, read as '0'		- n = Value at POR reset

5.9 Design Tips

Question 1: ***My program algorithm does not seem to function correctly.***

Answer 1:

1. The destination of the instruction may be specifying the W register (d = 0) instead of the file register (d = 1).
2. The register bank select bits (RP1:RP0 or IRP) may not be properly selected. Also if interrupts are used, the register bank select bits may not be properly restored when exiting the interrupt handler.

Question 2: ***I cannot seem to modify the STATUS register flags.***

Answer 2:

if the STATUS register is the destination for an instruction that affects the Z, DC, or C bits, the write to these bits is disabled. These bits are set or cleared based on device logic. Therefore, to modify bits in the STATUS register it is recommended to use the BCF and BSF instructions.

5.10 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 Mid-Range MCU family (that is they may be written for the Base-Line, or High-End families), but the concepts are pertinent, and could be used (with modification and possible limitations). The current application notes related to the CPU or the ALU are:

Title	Application Note #
Fixed Point Routines	AN617
IEEE 754 Compliant Floating Point Routines	AN575
Digital Signal Processing with the PIC16C74	AN616
Math Utility Routines	AN544
Implementing IIR Digital Filters	AN540
Implementation of Fast Fourier Transforms	AN542
Tone Generation	AN543
Servo Control of a DC Brushless Motor	AN532
Implementation of the Data Encryption Standard using the PIC17C42	AN583
PIC16C5X / PIC16CXX Utility Math Routines	AN526
Real Time Operating System for PIC16/17	AN585

5.11 Revision History

Revision A

This is the initial released revision of the CPU and ALU description.