f commonts begin with semicolon the instruction mov w3, w5 ; copy value in w3 to ws is an example of register direct addressing. File register addressing is used to move a value from a register to data memory or from data memory to a register. This is clone by speifying a location in memory in the mor instruction. Example: Copy the value in Latte momory at location 0x0802 into WY MOV 0x0802,W4 before after little endian! 0 x31 04801 0×801 0x31 0 x802 0 ×4A () 802 0x4A 0 2803 0×1234 0x67 0x674A 0x67 01804 0x804 0xFF WH OXFF 8-bits 8-bits

24

()

Section 5. Instruction Descriptions

MOV

Move f to Wnd

Implemented in:	PIC24F	PIC24H	PIC24E	dsPIC30F	dsPIC33F	dsPIC33E
	X	Х	Х	Х	Х	X

Syntax:

{label:}

MOV

Wnd

Operands:

f ∈ [0 ... 65534]

Wnd ∈ [W0 ... W15]

Operation:

(f) \rightarrow Wnd

Status Affected:

None

Encoding: Description:
 1000
 0fff
 ffff
 ffff
 ffff
 dddd

 Move the word contents of the specified file register to Wnd. The file

register may reside anywhere in the 32K words of data memory, but must be word-aligned. Register direct addressing must be used for Wnd.

The 'f' bits select the address of the file register. The 'd' bits select the destination register.

Note 1: This instruction operates on word operands only.

- 2: Since the file register address must be word-aligned, only the upper 15 bits of the file register address are encoded (bit 0 is assumed to be '0').
- 3: To move a byte of data from file register memory, the "MOV f to Destination" instruction (page 279) may be used.

Words:

1

Cycles:

1(1)

Note 1: In dsPIC33E and PIC24E devices, the listed cycle count does not apply to read and read-modify-write operations on non-CPU Special Function Registers. For more details, see Note 3 in Section 3.2.1 "Multi-Cycle Instructions".

Example 1:

MOV

CORCON, W12

; move CORCON to W12

	Deloie					
	Instruction					
W12	78FA					
CORCON	00F0					
SR	0000					

After
Instruction
W12 00F0

W12 00F0 CORCON 00F0 SR 0000

Example 2:

MOV

0x27FE, W3

; move (0x27FE) to W3

| Before | Instruction | W3 | 0035 | Data 27FE | ABCD | SR | 0000 |

After Instruction
W3 ABCD
Data 27FE ABCD
SR 0000

Ę

Instruction Descriptions

File register addressing can also be used to copy

mov W4,0x0804

	befoze	<u>g</u>		after	
	0x67	0x803			0x803
	OXFF	0×804		•	0x804
0x674A	0x01	0x805	0x674A	<u> </u>	0 x 805
WH	OxB2	0×806	wy	OxB2	0x806
	l	1			

There are some restrictions, moving data between a file register and a working register can only be done with 16-bit words. Both the source and the destination connot be file registers

Note that Microchip refers to Jata memory locations as file registery. Most other microcontroller rendors would use the term memory direct cubbressing to inducate the memory andress is directly specified. The addresses specified with this mor metrotion are absolute addresses

MOV		Move Wns	to f					
Implemented in:	PIC24F	PIC24H	PIC24E	dsPIC30F	dsPIC33F	dsPIC33E		
	Х	Х	Х	X	Х	X		
Syntax:	{label:}	MOV	Wns,	f				
Operands:	f∈ [0 65 Wns∈ [W							
Operation:	(Wns) →f							
Status Affected:	None							
Encoding:	1000	1fff	ffff	ffff	ffff	ssss		
Description:	register. TI memory, b	Move the word contents of the working register Wns to the specified file register. The file register may reside anywhere in the 32K words of data memory, but must be word-aligned. Register direct addressing must be used for Wn.						
		select the ad select the sc						
	Note 1: 2:							
	3:	·						
Words:	1							
Cycles:	1							
Example 1:	OV W4, XM	IDOSRT ;	move W4 to	XMODSRT		•		
XMODSF	Before Instruction V4 1200 RT 1340 BR 0000	XMODS	After Instruction W4 1200 SRT 1200 SR 0000					
Example 2:	O W8, 0	x1222 ;	move W8 to	o data addr	ess 0x1222			
	Before		After					

Before Instruction F200 Data 1222

FD88 SR 0000

Instruction W8 F200 Data 1222 F200 0000

Literal Addrewing

The move 16-bit literal to register instruction use literal addressing, i.e. the Nebue, instead of the address of the Nature of the Nature.

Example: Load 0x8765 into register 113

mou & 0x8765, W3

hefore ofter

W3 0×3678 W3 0×8765

W4 0×9ABC W4 0×9ABC

This would be encoded at , w3 0x8765 0010 1000 0111 0110 0101 0011

0×287653

Note that the literal can be signed or unsigned

MOU X-30875, W3

MOV

Move 16-bit Literal to Wnd

MOV	Move 16-bit Literal to Wnd					
Implemented in:	PIC24F	PIC24H	PIC24E	dsPIC30F	dsPIC33F	dsPIC33E
	Х	Х	Х	Х	Х	X
Syntax:	{label:}	MOV	#lit16,	Wnd		
Operands:	•	2768 65535 /0 W15]	5]			
Operation:	lit16 →Wn	nd				
Status Affected:	None					
Encoding:	0010	kkkk	kkkk	kkkk	kkkk	dddd
Description:	The 16-bit be used for	t literal 'k' is lo or Wnd.	aded into Wr	nd. Register o	direct addres	sing must
		s specify the value select the a			ster.	
	Note 1: 2:	This instructi The literal m or unsigned	ay be specific	ed as a signe		768:32767],
Words:	1					
Cycles:	1					
Example 1:	10V #0×423	31, W13	; load W13	with #0x42	31	
W	Before Instruction 13 091B SR 0000	٧	After Instruction V13 4231			
Example 2:	10V #0×4,	W2	; load W2 v	with #0×4		

Before

Instruction W2 B004 After Instruction W2 0004 SR 0000

SR 0000

MOV

Example 3:

#-1000, W8

; load W8 with #-1000

Before Instruction W8 23FF

0000

Instruction
W8 FC18
SR 0000

After

Another important foun of and drewing is indirect culdressing. An indirect culdress in some square breakets arount the register to indicate the source or destination and breas where the article relief is found.

The following instruction uses indirect

MOU IW67, IW75

Here the value at the address in W6 is copied to the address in W7.

Example: Copy the walve at data memory address 0x80 & to date memory address 0x806 using indirect addressing

peloce			after		
	OXHA	0x802		1	0x80Z
0×0802	Ox 67	0 v803	0x0802	0x67	Ox 303
W6	0x01	04809	WG	0x01	0 x 804
0×0806	0xBZ	0 × 80 S	0x0806	OxB2	0 x80S
W7	Ox20	0x806	ω7	Ox 9A	0x806
	OV3C	0 x 807		0x67	0x807
The second secon				1	127

A useful foun of indirect addressing uses an offset from a base address in a working register to determine the effective address

Example: Copy the word in register wo to the address two words after the address in W6

mou wo, [we+0x2]

before.		after	1
A THE STATE OF THE	0×4A		Ox4A
0x4321	0×67	Ox4321	0x67
WO	0x01	WO	0x01
Processor Action Co.	OxB2		Ox B2
0×0802	0×20	0.0802	0x21
W6	0x5C	W6	0×43

The offset can be in the range - 1024 to 1022

(For the byte version of this instruction the range is -512 to 511.)

MOV

Move [Ws with offset] to Wnd

Implemented in:	PIC24F	PIC24H	PIC24E	dsPIC30F	dsPIC33F	dsPIC33E
	Х	X	Х	X	X	Х

Syntax:

{label:}

MOV{.B}

[Ws + Slit10], Wnd

Operands:

Ws ∈ [W0 ... W15]

Slit $10 \in [-512 \dots 511]$ for byte operation

Slit10 ∈ [-1024 ... 1022] (even only) for word operation

Wnd ∈ [W0 ... W15]

Operation:

[Ws + Slit10] →Wnd

Status Affected:

None

Encoding: Description:

1001 0kkk kBkk kddd dkkk The contents of [Ws + Slit10] are loaded into Wnd. In Word mode, the

range of Slit10 is increased to [-1024 ... 1022] and Slit10 must be even to maintain word address alignment. Register indirect addressing must be used for the source, and direct addressing must be used for Wnd.

The 'k' bits specify the value of the literal.

The 'B' bit selects byte or word operation ('0' for word, '1' for byte).

The 'd' bits select the destination register. The 's' bits select the source register.

Note 1: The extension .B in the instruction denotes a byte move rather than a word move. You may use a . w extension to denote a word move, but it is not required.

In Byte mode, the range of Slit10 is not reduced as specified in Section 4.6 "Using 10-bit Literal Operands", since the literal represents an address offset from Ws.

Words:

Cycles:

1(1)

In dsPIC33E and PIC24E devices, the listed cycle count does not apply to read and read-modify-write operations on non-CPU Special Function Registers. For more details, see Note 3 in Section 3.2.1 "Multi-Cycle Instructions".

Example 1:

MOV.B

[W8+0x13], W10; load W10 with [W8+0x13]

; (Byte mode)

I	Before Instruction			
W8	1008			
W10	4009			
Data 101A	3312			
SR	0000			

		After			
	Instruction				
	W8 1008				
V	V10	4033			
Data 10)1A	3312			
	SR	0000			

Example 2:

MOV

 $[W4+0\times3E8]$, W2; load W2 with $[W4+0\times3E8]$

; (Word mode)

Before Instruction 9088 W2 W4 0800 Data 0BE8 5634

SR

After Instruction W2 5634 W4 0800 5634 Data 0BE8 0000 SR

MOV

Move Wns to [Wd with offset]

Implemented in:	PIC24F	PIC24H	PIC24E	dsPIC30F	dsPIC33F	dsPIC33E
	Х	Х	Х	Х	Х	X

Syntax:

{label:}

0000

MOV{.B}

Wns,

[Wd + Slit10]

Operands:

Wns ∈ [W0 ... W15]

Slit10 ∈ [-512 ... 511] in Byte mode

Slit10 ∈ [-1024 ... 1022] (even only) in Word mode

Wd ∈ [W0 ... W15]

Operation:

 $(Wns) \rightarrow [Wd + Slit10]$

Status Affected:

None

Encoding:

1001

Description:

1kkk kBkk kddd dkkk SSSS The contents of Wns are stored to [Wd + Slit10]. In Word mode, the range of Slit10 is increased to [-1024 ... 1022] and Slit10 must be even to

maintain word address alignment. Register direct addressing must be used for Wns, and indirect addressing must be used for the destination.

The 'k' bits specify the value of the literal.

The 'B' bit selects byte or word operation ('0' for word, '1' for byte).

The 'd' bits select the destination register. The 's' bits select the source register.

Note 1: The extension .B in the instruction denotes a byte move rather than a word move. You may use a . w extension to denote a word move, but it is not required.

2: In Byte mode, the range of Slit10 is not reduced as specified in Section 4.6 "Using 10-bit Literal Operands", since the literal represents an address offset from Wd.

Words:

1

Cycles:

1

Example 1:

MOV.B W0, [W1+0x7] ; store W0 to [W1+0x7]

; (Byte mode)

Before Instruction W0 9015 1800 W1 Data 1806 2345 0000

After Instruction W0 9015 W1 1800 1545 Data 1806 SR 0000

Anthmetic

The most common arithmetic operation in any microcontroller is addition. Addition is not only used in processing data but is also used for operations such as computing offsets in addressing.

The PIC24 implements both three and Two operand vorsions of audition Cand subtraviour). The three operand version allows separate registers for each of the Two numbers to be added and the result. As with the mon instruction various addressing modes are supported with variants of these instructions that will operate on bytes

Example: Add the contents of wo to wi and place the result in w3.

add wo, wi, wz

This instruction uses register direct addressing with the Two source operands preceeding the destination operand,

						all agricon blacks for a last ways replaced processing and stated becomes account.
	The a	stude opera	Two is Co	NO)+(WI)	→ W3	774-34-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4-4
		The same of the sa	markkin-misk, kantike van herriet fankaren, miskantik gre vaka maar fe die 1807 verkin			
	mak karawar angi kalingian kapata kadaman da Badama njija dipadansi	a contract of the contract of		10.024	n an an Andrewskin Fig. (1949) 1967 197 197 197 197 197 197 197 197 197 19	derference table trade and any partie are consequently
	WO	Ox1234	WO	0x1234		
	WI	0x5678	WI	0x5678	gyppopula manami i dali (*). And i dag Milabul di un daput aput di propunsion I dali (*). And i dag Milabul di un daput aput di un daput aput di un daput aput di un daput di un daput aput	
		0×9ABC	<i>W3</i>	Ox 68 AC	and the state of t	
	<i>W3</i>	[OX MOC]		1023110		
	Lig to a second only deficiency of the second on the secon	ann mill million in del coloni e ferminale del million la del film ferni en fejan (1 a fejan million en del mil	r ankarana narrangaratiga (p. 10) a sada wan biwat da ara da Babbah (kan sa sada sa sa sa		and producers and all the State of the State of the State of the Assessing Principles of State of the State o	i di Bi di ambi Shinki Ving Shink Andrew Wilderstan, andrew sakan dan sakan
	an top y and a second real real of a second sec		websak bekert, any calculational distribution for General Association (Association) and Learning with the Contract of Contract	والمرافقة		
	angen garan ang kabupatèn kabupatèn dan dan dan dan dan dan dan dan dan da				and Allen and a survey of the party of the survey of the s	order of the second state of the second seco
The second state of the se				ميدود والمراجعة المراجعة المر		
	The state of the s					
The second secon	gamagandagad agiya ya agana yesan dan acarara dana dan bada di didikisini P kilotet			raphorate and a single property and appropriate specifications, the last, april 1972, and the state of the st	and the second second provide provide provide the second s	
	magain na magaine magaine agus falbhful didh laffic le "Philipse		emande aller am den julijanski far blike serane Block for en benink U.Vinci unde setter (EFF)	of contrast of the frequency of contrast of the contrast of th	and the state of t	
	angunum aka masalir sangka kabanahari di Pril mandali 14 (2)		damind holosophilasis in the Country's speed provided as being the constraint with the country of the country o	anno de archego a de histórico e estad descripción de archido hace de archido de la compansión de la compans		
	and security in the structure of the security		rika kawa kawa 13 minjalawa ka wasa ya mara manaka kali 2 mata ka kalimba Marakaba Carl da d	action was proceed on wind the shadely black a state of the soul or class of a distribution of		ENGINEER AND WAS ASSESSED ASSESSED FOR STRUCKEN SHEET
		managamang man dispunstration and and dispuns y designation of galaxies and the state of property of the particular designation of the state of the				Section of the second section of the second section of the section of the second section of the
			norman sama makina dake e die y Njelyen kumi untu dan ride ministrativa dan dan dan sama dan sama dan sama dan	*************************************	Marie de August entes vann i Jaryon, Spain min Cast (1981 allent) na	and any or the track of the second of the se
anadografia godina, mad haddinkogo ori viving ad dalikina o po negli arabinak (APP) da				analysis since we manufacture hand the water analysis of the or the contract of the contract o		.30
				antigative generalization of general professional performance (CP 2012) and a second color color color		50

and the second s

Example 2:

ADD

W3, #0×6, [--W4]

; Add W3 and 6 (Word mode) ; Store the result in [--W4]

Before Instruction 6006 W3 1000 W4 DDEE Data 0FFE DDEE Data 1000 0000 SR

After Instruction W3 6006 W4 **OFFE** Data 0FFE 600C Data 1000 DDEE SR 0000

ADD

Add Wb to Ws

Implemented in:	PIC24F	PIC24H	PIC24E	dsPIC30F	dsPIC33F	dsPIC33E
	Х	Х	Х	Х	Х	X

Syntax:

{label:}

ADD{.B}

Wb,

Wd

[Ws],

[Wd]

[Ws++],

Ws,

[Wd++]

[Ws---],

[Wd--]

[++Ws], [--Ws],

[++Wd] [--Wd]

Operands:

Wb ∈ [W0 ... W15]

Ws ∈ [W0 ... W15]

Wd ∈ [W0 ... W15]

Operation:

(Wb) + (Ws) →Wd

Status Affected:

DC, N, OV, Z, C

Encoding: Description:

0100 0www ···wBqq qddd dppp Add the contents of the source register Ws and the contents of the base

register Wb, and place the result in the destination register Wd. Register direct addressing must be used for Wb. Either register direct or indirect

addressing may be used for Ws and Wd.

The 'w' bits select the address of the base register.

The 'B' bit selects byte or word operation ('0' for word, '1' for byte).

The 'q' bits select the destination Address mode.

The 'd' bits select the destination register.

The 'p' bits select the source Address mode.

The 's' bits select the source register.

Note: The extension .B in the instruction denotes a byte operation rathjer than a word operation. You may use a . w extension to

denote a word operation, but it is not required.

Words:

1

Cycles:

1(1)

Note 1:

In dsPIC33E and PIC24E devices, the listed cycle count does not apply to read and read-modify-write operations on non-CPU Special Function Registers. For more

details, see Note 3 in Section 3.2.1 "Multi-Cycle Instructions".

Section 5. Instruction Descriptions

Example 1:

ADD.B

W5, W6, W7

; Add W5 to W6, store result in W7

; (Byte mode)

Before Instruction

AB00 W5 0030 W6

Instruction AB00 W5 0030 W6 FF30 W7

After

FFFF W7 0000 SR

0000 SR

Example 2:

ADD

W5, W6, W7

; Add W5 to W6, store result in W7

; (Word mode)

Before Instruction AB00 W5 W6 0030 W7 FFFF SR 0000

After Instruction AB00 W5 W6 0030 W7 **AB30** 0008 (N = 1) SR

ADD

Add Accumulators

Implemented in:	PIC24F	PIC24H	PIC24E	dsPIC30F	dsPIC33F	dsPIC33E
				Х	Х	Х

Syntax:

{label:}

ADD

Acc

Operands:

 $Acc \in [A,B]$

Operation:

If (Acc = A):

(ACCA) + (ACCB) →ACCA

(ACCA) + (ACCB) →ACCB

Status Affected:

Encoding:

OA, OB, OAB, SA, SB, SAB 1100 1011 A000

Description:

Add the contents of Accumulator A to the contents of Accumulator B and

0000

0000

0000

place the result in the selected accumulator. This instruction performs a

40-bit addition.

The 'A' bit specifies the destination accumulator.

Words:

Cycles:

1

ADD

Example 1:

; Add ACCB to ACCA

Before

Instruction

00 1833 4558 0000

After Instruction

ACCA 00 1855 7858 **ACCB** 00 1833 4558

ACCA 00 0022 3300 **ACCB** SR

0000 SR

>xc16-objdump -d newmainXC16.o

Disassembly of section .text:

00000000	<_mai:	n>:		
0:	06 00) fa	lnk	#0x6
2:	50 f	3 2f	mov.w	#0xff85, w0
4:	00 0:	£ 78	mov.w	w0, [w14]
6 :	e0 0:	2 20	mov.W	#0x2e, $w0$
8:	10 0	7 98	mov.w	w0, [w14+2]
a:	1e 0	90	mov.w	[w14+2], w0
c:	1e 0	0 40	add.w	w0, [w14], w0
e:	20 0	7 98	mov.W	w0, [w14+4]
10:	2e 0	0 90	mov.W	[w14+4], w0
12:	00 0	88 0	mov.W	w0, 0x0
14:	00 0	0 eb	clr.w	w0
16:	00 8	0 fa	ulnk	
18:	00 0	0 06	return	