

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

E·XFI

Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	40 MIPs
Connectivity	CANbus, I ² C, IrDA, LINbus, SPI, UART/USART
Peripherals	AC'97, Brown-out Detect/Reset, DMA, I2S, POR, PWM, WDT
Number of I/O	85
Program Memory Size	256КВ (256К х 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	30K x 8
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 32x10b/12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TQFP
Supplier Device Package	100-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj256gp710a-e-pf

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

Pin Diagrams (Continued)



3.5 Arithmetic Logic Unit (ALU)

The dsPIC33FJXXXGPX06A/X08A/X10A ALU is 16 bits wide and is capable of addition, subtraction, bit shifts and logic operations. Unless otherwise mentioned, arithmetic operations are 2's complement in nature. Depending on the operation, the ALU may affect the values of the Carry (C), Zero (Z), Negative (N), Overflow (OV) and Digit Carry (DC) Status bits in the SR register. The C and DC Status bits operate as Borrow and Digit Borrow bits, respectively, for subtraction operations.

The ALU can perform 8-bit or 16-bit operations, depending on the mode of the instruction that is used. Data for the ALU operation can come from the W register array, or data memory, depending on the addressing mode of the instruction. Likewise, output data from the ALU can be written to the W register array or a data memory location.

Refer to the *"16-bit MCU and DSC Programmer's Reference Manual"* (DS70157) for information on the SR bits affected by each instruction.

The dsPIC33FJXXXGPX06A/X08A/X10A CPU incorporates hardware support for both multiplication and division. This includes a dedicated hardware multiplier and support hardware for 16-bit-divisor division.

3.5.1 MULTIPLIER

Using the high-speed 17-bit x 17-bit multiplier of the DSP engine, the ALU supports unsigned, signed or mixed-sign operation in several MCU multiplication modes:

- 16-bit x 16-bit signed
- 16-bit x 16-bit unsigned
- 16-bit signed x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit unsigned
- 16-bit unsigned x 5-bit (literal) unsigned
- 16-bit unsigned x 16-bit signed
- 8-bit unsigned x 8-bit unsigned

3.5.2 DIVIDER

The divide block supports 32-bit/16-bit and 16-bit/16-bit signed and unsigned integer divide operations with the following data sizes:

- 32-bit signed/16-bit signed divide
- 32-bit unsigned/16-bit unsigned divide
- 16-bit signed/16-bit signed divide
- 16-bit unsigned/16-bit unsigned divide

The quotient for all divide instructions ends up in W0 and the remainder in W1. 16-bit signed and unsigned DIV instructions can specify any W register for both the 16-bit divisor (Wn) and any W register (aligned) pair (W(m + 1):Wm) for the 32-bit dividend. The divide algorithm takes one cycle per bit of divisor, so both 32-bit/16-bit and 16-bit/16-bit instructions take the same number of cycles to execute.

3.6 DSP Engine

The DSP engine consists of a high-speed, 17-bit x 17-bit multiplier, a barrel shifter and a 40-bit adder/subtracter (with two target accumulators, round and saturation logic).

The dsPIC33FJXXXGPX06A/X08A/X10A is a single-cycle, instruction flow architecture; therefore, concurrent operation of the DSP engine with MCU instruction flow is not possible. However, some MCU ALU and DSP engine resources may be used concurrently by the same instruction (e.g., ED, EDAC).

The DSP engine also has the capability to perform inherent accumulator-to-accumulator operations which require no additional data. These instructions are ADD, SUB and NEG.

The DSP engine has various options selected through various bits in the CPU Core Control register (CORCON), as listed below:

- Fractional or integer DSP multiply (IF)
- Signed or unsigned DSP multiply (US)
- · Conventional or convergent rounding (RND)
- Automatic saturation on/off for AccA (SATA)
- Automatic saturation on/off for AccB (SATB)
- Automatic saturation on/off for writes to data memory (SATDW)
- Accumulator Saturation mode selection (ACCSAT)

Table 3-1 provides a summary of DSP instructions. A block diagram of the DSP engine is shown in Figure 3-3.

TABLE 3-1:	DSP INSTRUCTIONS
	SUMMARY

Instruction	Algebraic Operation	ACC Write Back
CLR	A = 0	Yes
ED	$A = (x - y)^2$	No
EDAC	$A = A + (x - y)^2$	No
MAC	$A = A + (x \bullet y)$	Yes
MAC	A = A + x2	No
MOVSAC	No change in A	Yes
MPY	$A = x \bullet y$	No
MPY	$A = x^2$	No
MPY.N	$A = -x \bullet y$	No
MSC	$A = A - x \bullet y$	Yes

TRAPR IOPUWR — — — VF bit 15 </th <th></th>	
bit 15	1200
	bit 8
R/W-0 R/W-0 R/W-0 R/W-0 R/W-0 R/W-1 F	R/W-1
EXTR SWR SWDTEN ⁽²⁾ WDTO SLEEP IDLE BOR	POR
bit 7	bit 0
Legend:	
R = Readable bit $W = Writable bit II = Unimplemented bit read as '0'$	
-n = Value at POR (1' = Bit is set (0' = Bit is cleared x = Bit is unknown)	
bit 15 TRAPR: Trap Reset Flag bit	
1 = A Trap Conflict Reset has occurred	
bit 14 IOPLINE: Illegel Opende er Uninitialized W Access Repet Flag bit	
1 = An illegal opcode detection, an illegal address mode or uninitialized W register use	ed as an
Address Pointer caused a Reset	
0 = An illegal opcode or uninitialized W Reset has not occurred	
bit 13-9 Unimplemented: Read as '0'	
bit 8 VREGS: Voltage Regulator Standby During Sleep bit ⁽³⁾	
1 = Voltage Regulator goes into standby mode during Sleep	
bit 7 EXTR: External Reset (MCLR) Pin bit	
1 = A Master Clear (pin) Reset has occurred	
0 = A Master Clear (pin) Reset has not occurred	
bit 6 SWR: Software Reset (Instruction) Flag bit	
1 = A RESET INSTRUCTION has been executed 0 = A RESET instruction has not been executed	
bit 5 SWDTEN: Software Enable/Disable of WDT bit ⁽²⁾	
1 = WDT is enabled	
0 = WDT is disabled	
bit 4 WDTO: Watchdog Timer Time-out Flag bit	
1 = WDT time-out has occurred	
bit 3 SI FEP: Wake-up from Sleep Flag bit	
1 = Device has been in Sleep mode	
0 = Device has not been in Sleep mode	
bit 2 IDLE: Wake-up from Idle Flag bit	
1 = Device was in Idle mode	
bit 1 BOB: Brown out Depet Fleg bit	
1 = A Brown-out Reset has occurred	
0 = A Brown-out Reset has not occurred	
Note 1: All of the Reset status bits may be set or cleared in software. Setting one of these bits in software of a device Reset	does not
2: If the FWDTEN Configuration bit is '1' (unprogrammed) the WDT is always enabled regardless	of the

REGISTER 6-1: RCON: RESET CONTROL REGISTER⁽¹⁾

SWDTEN bit setting.
3: For dsPIC33FJ256GPX06A/X08A/X10A devices, this bit is unimplemented and reads back programmed value.

SR: CPU STATUS REGISTER⁽¹⁾

R-0	R-0	R/C-0	R/C-0	R-0	R/C-0	R -0	R/W-0
OA	OB	SA	SB	OAB	SAB	DA	DC
bit 15							bit 8
R/W-0 ⁽³⁾	R/W-0 ⁽³⁾	R/W-0 ⁽³⁾	R-0	R/W-0	R/W-0	R/W-0	R/W-0
IPL2 ⁽²⁾	IPL1 ⁽²⁾	IPL0 ⁽²⁾	RA	N	OV	Z	С
bit 7							bit 0
Legend:							
C = Clear only bit R = Readable bit U = L			U = Unimpler	mented bit, read	1 as '0'		
S = Set only bi	t	W = Writable I	bit	-n = Value at	POR		

x = Bit is unknown

bit 7-5

1' = Bit is set

REGISTER 7-1:

IPL<2:0>: CPU Interrupt Priority Level Status bits⁽²⁾

'0' = Bit is cleared

- 111 = CPU Interrupt Priority Level is 7 (15), user interrupts disabled
- 110 = CPU Interrupt Priority Level is 6 (14)
- 101 = CPU Interrupt Priority Level is 5 (13)
- 100 = CPU Interrupt Priority Level is 4 (12)
- 011 = CPU Interrupt Priority Level is 3 (11)
- 010 = CPU Interrupt Priority Level is 2 (10)
- 001 = CPU Interrupt Priority Level is 1 (9) 000 = CPU Interrupt Priority Level is 0 (8)
- **Note 1:** For complete register details, see Register 3-1.
 - 2: The IPL<2:0> bits are concatenated with the IPL<3> bit (CORCON<3>) to form the CPU Interrupt Priority Level. The value in parentheses indicates the IPL if IPL<3> = 1. User interrupts are disabled when IPL<3> = 1.
 - 3: The IPL<2:0> Status bits are read-only when NSTDIS (INTCON1<15>) = 1.

REGISTER 7-2: CORCON: CORE CONTROL REGISTER⁽¹⁾

U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-0	R-0		
_	—	—	US	EDT		DL<2:0>			
bit 15							bit 8		
R/W-0	R/W-0	R/W-1	R/W-0	R/C-0	R/W-0	R/W-0	R/W-0		
SATA	SATB	SATDW	ACCSAT	IPL3 ⁽²⁾	PSV	RND	IF		
bit 7							bit 0		
Legend:		C = Clear only	/ bit						
R = Readable b	oit	W = Writable	bit	-n = Value at POR		'1' = Bit is set			
0' = Bit is cleared 'x = Bit			nown U = Unimplemented bit, read as '0'						
				(2)					
bit 3	IPL3: CPU Inf	terrupt Priority	Level Status b	bit 3 ⁽²⁾					
1 = CPU interrupt priority level is greater than 7									

0 = CPU interrupt priority level is 7 or less

Note 1: For complete register details, see Register 3-2.

2: The IPL3 bit is concatenated with the IPL<2:0> bits (SR<7:5>) to form the CPU Interrupt Priority Level.

REGISTER 7-9: IFS4: INTERRUPT FLAG STATUS REGISTER 4

U-0	U-0	U-0	U-0	U-0	U-0	U-0	U-0
_	_	—	_	_	—	—	
bit 15							bit 8
P/M/_0	P/M/ 0	P/M/_0	PM/0	11-0	P/M/ 0	P/M/_0	11.0
C2TXIE	C1TXIE		DMA6IF				
bit 7	0 T T/AI	Billinin	Billini		OZEN	01Ell	bit C
Legend:							
R = Readabl	e bit	W = Writable	bit	U = Unimpler	mented bit, read	l as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkn	own
			e.1				
DIT 15-8	Unimplemen	ited: Read as	0.				
bit /	C2TXIF: EC/	AN2 Transmit D	ata Request I	nterrupt Flag S	status bit		
	1 = Interrupt 0 = Interrupt	request has oc request has no	curred t occurred				
bit 6	C1TXIE: FCA	AN1 Transmit D	ata Request I	nterrupt Flag S	status bit		
	1 = Interrupt	request has oc	curred				
	0 = Interrupt	request has no	t occurred				
bit 5	DMA7IF: DM	IA Channel 7 D	ata Transfer C	Complete Interr	upt Flag Status	bit	
	1 = Interrupt	request has oc	curred				
	0 = Interrupt	request has no	t occurred				
bit 4	DMA6IF: DM	IA Channel 6 D	ata Transfer C	Complete Interr	upt Flag Status	bit	
	1 = Interrupt	request has oc	curred				
L:1 0		request has no	t occurred				
DIT 3	Unimplemen	ited: Read as	0.				
bit 2	U2EIF: UAR	12 Interrupt Fla	g Status bit				
	\perp = Interrupt 0 = Interrupt	request has oc request has no	currea t occurred				
bit 1	U1EIF: UAR	T1 Interrupt Fla	g Status bit				
-	1 = Interrupt	request has oc	curred				
	0 = Interrupt	request has no	t occurred				
bit 0	Unimplemer	nted: Read as '	0'				

REGISTER 7	-33: INTTR	EG: INTERRU	JPT CONT	ROL AND ST	ATUS REGIS	TER		
U-0	U-0	U-0	U-0	R-0	R-0	R-0	R-0	
	_	—			ILR<	<3:0>		
bit 15		•					bit 8	
U-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	
				VECNUM<6:0	>			
bit 7							bit 0	
Legend:								
R = Readable	bit	W = Writable b	bit	U = Unimplen	nented bit, read	l as '0'		
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is clea	ared	x = Bit is unkr	iown	
bit 15-12	Unimplemer	ted: Read as '0	3					
bit 11-8	ILR<3:0>: No	ew CPU Interrup	t Priority Lev	vel bits				
	1111 = CPU	Interrupt Priority	/ Level is 15					
	•							
	•							
	0001 = CPU	Interrupt Priority	/ Level is 1					
	0000 = CPU	Interrupt Priority	Level is 0					
bit 7	Unimplemer	ted: Read as '0	,					
bit 6-0	VECNUM<6:	0>: Vector Numl	ber of Pendi	ing Interrupt bits				
	0111111 = 	nterrupt Vector p	ending is nu	umber 135				
	•		C C					
	•							
	•	nterrunt Vector n	ondina is nu	umber 0				
	0000001 = Interrupt Vector pending is number 9							

0000000 = Interrupt Vector pending is number 8

NOTES:

REGISTER 8-1:	DMAxCON: DMA CHANNEL x CONTROL REGISTER
---------------	---

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	U-0	U-0	
CHEN	SIZE	DIR	HALF	NULLW	_	_	_	
bit 15							bit 8	
U-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	
		AMOD	E<1:0>			MODE	<1:0>	
bit 7							bit 0	
Legend:								
R = Readable	bit	W = Writable	bit	U = Unimplei	mented bit, read	as '0'		
-n = Value at P	POR	'1' = Bit is set		'0' = Bit is cle	eared	x = Bit is unkr	IOWN	
		F						
DIT 15	1 - Channel							
	0 = Channel o	disabled						
bit 14	SIZE: Data Tr	ansfer Size bit						
	1 = Byte							
	0 = Word							
bit 13	DIR: Transfer Direction bit (source/destination bus select)							
	1 = Read fron 0 = Read fron	n DMA RAM ao n peripheral ad	ddress, write t dress, write to	o peripheral ao DMA RAM ao	ddress ddress			
bit 12	HALF: Early I	Block Transfer	Complete Inte	errupt Select b	it			
	1 = Initiate blo	ock transfer col	mplete interru	pt when half of pt when all of t	f the data has be the data has bee	een moved		
bit 11	• – millate block transfer complete interrupt when all of the data has been moved NIII I W. Null Data Perinheral Write Mode Select hit							
	1 = Null data	write to periphe	eral in additior	n to DMA RAM	l write (DIR bit m	nust also be cle	ar)	
bit 10-6	Unimplemen	ted: Read as '	0'					
bit 5-4	AMODE<1:0	: DMA Chann	el Operating N	Node Select bi	ts			
	11 = Reserve	d						
	10 = Peripher	al Indirect Add	ressing mode					
	01 = Register 00 = Register	Indirect without Indirect with F	ost-Increm	t mode				
bit 3-2	Unimplemen	ted: Read as '	0'					
bit 1-0	MODE<1:0>:	DMA Channel	Operating Mo	ode Select bits				
	11 = One-Sho	ot, Ping-Pong r	nodes enable	d (one block tr	ansfer from/to e	ach DMA RAM	buffer)	
	10 = Continuo	ous, Ping-Pong	modes enab	led				
	01 = One-Sno	ous. Pina-Pong r	noues uisable I modes disab	u led				
	ee continue							

REGISTER 8-5: DMAXPAD: DMA CHANNEL x PERIPHERAL ADDRESS REGISTER⁽¹⁾

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
			PAD	<15:8>			
bit 15							bit 8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
			PAC)<7:0>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable I	bit	U = Unimpler	mented bit, rea	id as '0'	
-n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is u				x = Bit is unkr	nown		

bit 15-0 PAD<15:0>: Peripheral Address Register bits

Note 1: If the channel is enabled (i.e., active), writes to this register may result in unpredictable behavior of the DMA channel and should be avoided.

REGISTER 8-6: DMAxCNT: DMA CHANNEL x TRANSFER COUNT REGISTER⁽¹⁾

U-0	U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0
—	—	—	—	—	—	CNT<9:8> ⁽²⁾	
bit 15							bit 8

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
			CNT<	7:0> (2)			
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	l as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-10 Unimplemented: Read as '0'

bit 9-0 CNT<9:0>: DMA Transfer Count Register bits⁽²⁾

Note 1: If the channel is enabled (i.e., active), writes to this register may result in unpredictable behavior of the DMA channel and should be avoided.

2: Number of DMA transfers = CNT<9:0> + 1.

REGISTER 1	9-16: CiRX	FnSID: ECAN™	ACCEPTAN	ICE FILTER n	STANDARD II	DENTIFIER (n =	: 0, 1,, 15)	
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	
			SID	<10:3>				
bit 15							bit 8	
R/W-x	R/W-x	R/W-x	U-0	R/W-x	U-0	R/W-x	R/W-x	
	SID<2:0>		_	EXIDE	_	EID<	17:16>	
bit 7							bit 0	
Legend:								
R = Readabl	le bit	W = Writable I	bit	U = Unimpler	mented bit, rea	d as '0'		
-n = Value at	t POR	'1' = Bit is set		'0' = Bit is cleared			x = Bit is unknown	
bit 15-5	SID-10.0-	Standard Identifi	er hite					
bit 10-0	1 = Messag 0 = Messag	e address bit SIE e address bit SIE)x must be '1)x must be '0	' to match filter ' to match filter				
bit 4	Unimpleme	ented: Read as 'o)'					
bit 3	EXIDE: Ext	ended Identifier I	Enable bit					
	If MIDE = 1	then:						
	1 = Match o	nly messages wi	th extended i	identifier addre	sses			
	0 = Match o	nly messages wi	th standard i	dentifier addres	ses			
	If MIDE = 0	then:						
	Ignore EXII	DE bit.						

bit 2
 Unimplemented: Read as '0'

 bit 1-0
 EID<17:16>: Extended Identifier bits

 1 = Message address bit EIDx must be '1' to match filter

 0 = Message address bit EIDx must be '0' to match filter

REGISTER 19-17: CIRXFnEID: ECAN™ ACCEPTANCE FILTER n EXTENDED IDENTIFIER (n = 0, 1, ..., 15)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
			EID	<15:8>			
bit 15							bit 8
]
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
			EID	<7:0>			
bit 7							bit 0
Legend:							
R = Readable b	bit	W = Writable b	oit	U = Unimpler	nented bit, rea	d as '0'	
-n = Value at P	OR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown

bit 15-0 EID<15:0>: Extended Identifier bits

1 = Message address bit EIDx must be '1' to match filter 0 = Message address bit EIDx must be '0' to match filter

Note: The buffers, SID, EID, DLC, Data Field and Receive Status registers are located in DMA RAM.

REGISTER 19-27: CiTRBnSID: ECAN™ BUFFER n STANDARD IDENTIFIER (n = 0, 1, ..., 31)

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—			SID<10:6>		
bit 15							bit 8
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x

SID<5:0>	SRR	IDE
bit 7		bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read	as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 15-13	Unimplemented: Read as '0'
bit 12-2	SID<10:0>: Standard Identifier bits
bit 1	SRR: Substitute Remote Request bit
	1 = Message will request remote transmission 0 = Normal message
bit 0	IDE: Extended Identifier bit
	1 = Message will transmit extended identifier

0 = Message will transmit standard identifier

REGISTER 19-28: CITRBnEID: ECAN™ BUFFER n EXTENDED IDENTIFIER (n = 0, 1, ..., 31)

U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	—		EID<	17:14>	
bit 15							bit 8
R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
			EID<	:13:6>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplen	nented bit, rea	d as '0'	
-n = Value at P	OR	'1' = Bit is set		'0' = Bit is cleared x = Bit is unknown			

bit 15-12 Unimplemented: Read as '0'

bit 11-0 EID<17:6>: Extended Identifier bits



REGISTER 21-5: ADxCHS123: ADCx INPUT CHANNEL 1, 2, 3 SELECT REGISTER

U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
	_	—	_	—	CH123	CH123NB<1:0>	
bit 15							bit 8
U-0	U-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0
—	_		—	—	CH123	NA<1:0>	CH123SA
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable b	oit	U = Unimple	mented bit, rea	ad as '0'	
-n = Value at	POR	'1' = Bit is set		'0' = Bit is cle	eared	x = Bit is unk	nown
bit 15-11	Unimplemen	ted: Read as '0	,				
bit 10-9	CH123NB<1:	0>: Channel 1,	2, 3 Negative	e Input Select fo	or Sample B bi	ts	
	When AD12E	B = 1, CHxNB is	s: U-0, Unim	plemented, Re	ead as '0'		
	11 = CH1 neg	pative input is Al	N9, CH2 neg	ative input is A	N10, CH3 neg	ative input is Al	N11
	10 = CH1 neg 0x = CH1 CH	Jative input is Al	N6, CH2 neg	ative input is A	N7, CH3 nega	live input is AN	8
hit 8	CH123SB: CH	nannel 1 2 3 P	ositive Input :	Select for Sam	nle B hit		
bit o	When AD12F	3 = 1. CHxSB is	: U-0. Unimi	plemented. Re	ad as '0'		
	1 = CH1 posit	tive input is AN3	, CH2 positiv	ve input is AN4	, CH3 positive	input is AN5	
	0 = CH1 posit	tive input is ANC	, CH2 positiv	e input is AN1	, CH3 positive	input is AN2	
bit 7-3	Unimplemen	ted: Read as '0	,				
bit 2-1	CH123NA<1:	0>: Channel 1,	2, 3 Negative	e Input Select fo	or Sample A bi	ts	
	When AD12E	B = 1, CHxNA is	s: U-0, Unim	plemented, Re	ad as '0'		
	11 = CH1 negative input is AN9, CH2 negative input is AN10, CH3 negative input is AN11						N11
	10 = CH1 neg	gative input is Al	N6, CH2 neg	ative input is A	N7, CH3 nega	tive input is AN	8
hit 0			e input is VRI	EF- Rolant for Rom	nlo A hit		
DILU				Select for Sam	pie A bit		
	1 = CH1 positi	ive input is ANG	CH2 positiv	e input is AN4	CH3 positive	input is AN5	
	0 = CH1 posit	tive input is ANC	, CH2 positiv	e input is AN1	, CH3 positive	input is AN2	

REGISTER 2	1-6: ADxCl	HS0: ADCx IN	IPUT CHAN	INEL 0 SELE	CT REGISTE	ER			
R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
CH0NB	_	—			CH0SB<4:0>	(1)			
bit 15							bit 8		
R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0		
CH0NA		—			CH0SA<4:0>	(1)			
bit 7							bit 0		
									
Legend:									
R = Readable	bit	W = Writable b	bit	U = Unimpler	mented bit, rea	ad as '0'			
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unki	nown		
bit 15	CHONB: Cha	nnel () Negative	Input Select	for Sample R b	it				
	Same definition	on as bit 7.		ior campie B s					
bit 14-13	Unimplemen	ted: Read as '0	,						
bit 12-8	CH0SB<4:0>: Channel 0 Positive Input Select for Sample B bits ⁽¹⁾								
	11111 = Cha 11110 = Cha	nnel 0 positive i nnel 0 positive i	nput is AN31 nput is AN30	F					
	•								
	•								
	00010 = Channel 0 positive input is AN2								
	00001 = Channel 0 positive input is AN1								
	00000 = Cha	nnel 0 positive i	nput is AN0						
bit 7	CH0NA: Cha	nnel 0 Negative	Input Select	for Sample A b	it				
	1 = Channel (0 = Channel (0 negative input 0 negative input	is AN1 is Vref-						
bit 6-5	Unimplemen	ted: Read as '0	,						
bit 4-0	CH0SA<4:0>: Channel 0 Positive Input Select for Sample A bits ⁽¹⁾								
	11111 = Channel 0 positive input is AN31								
	11110 = Cha	nnel 0 positive i	nput is AN30						
	•								
	•								
	00010 = Cha	nnel 0 positive i	nput is AN2						
	00001 = Cha	nnel 0 positive i	nput is AN1						
	00000 = Cna	nnei o positive i	nput is AINU						

Note 1: ADC2 can only select AN0 through AN15 as positive input.

48 MPY μrv Mim * Nin . Acc., Xix . Rod., Wy , Wyd. Multiply Wn by Wn to Accumulator 1 1 0. A OB OAB. ASB SAB 49 MPY . Ni Mim * Win . Acc., Xix . Rod., Wy , Wyd. Square Wn to Accumulator 1 1 0. A OB OAB. ASB SAB 49 MPY . Ni Mim * Wn . Acc., Xix . Rod., Wy , Wyd. (Multiply Wm by Wn to Accumulator 1 1 1 0. A OB OAB. ASB SAB 50 MSC Rec _ Xix . Rod., Wy , Wyd. Multiply and Subtract from Accumulator 1 1 1 0. A OB OAB. ASB SAB 51 MUL _ ST _ Wn , No. Ro. Nr.A. (Wnd + 1, Wnd) = signed(Wb) * singend(Wb) 1 1 None 400101 KD , No. No. No. Ro. Nr.A. (Wnd + 1, Wnd) = signed(Wb) * singend(Wb) 1 1 None 400101 KD , No. No. No. Ro. Nr.A. (Wnd + 1, Wnd) = signed(Wb) * singend(Wb) 1 1 None 401101 KD , No. Ro. Nr.A. (Wnd + 1, Wnd) = signed(Wb) * unsigned(Wb) 1 1 None 401101 KD , No. Ro. Nr.A. (Wnd + 1, Wnd) = signed(Wb) * unsigned(Wb) 1 1 None 401110 KD , Ro. Nr.A. Acco Noge N	Base Instr #	Assembly Mnemonic		Assembly Syntax	Description	# of Words	# of Cycles	Status Flags Affected
NP NP<	48	MPY	MPY Wm*Wn,A	cc,Wx,Wxd,Wy,Wyd	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
49 MPY.N MPY.N			MPY Wm*Wm,A	cc,Wx,Wxd,Wy,Wyd	Square Wm to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
50 MSC KSC Non-AcC, NZ, Kod, Wy, Wyd, YM, YM, YM, YM, YM, YM, YM, YM, YM, YM	49	MPY.N	MPY.N Wm*Wn,A	cc,Wx,Wxd,Wy,Wyd	-(Multiply Wm by Wn) to Accumulator	1	1	None
51 MUL MUL, SU MD, SU, MD, MD, MD, MD, MD (Mnd + 1, Wnd) = signed(WD) * signed(WD) 1 1 None MUL, SU MD, SU, MD, MD, MD, MD, MD, MD, MD, MD, MD, MD	50	MSC	MSC	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd , AWB	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
MUL. SU Wo. Ms. Mcd. (Mrd + 1, Mrd) = signed(Wb) * unsigned(Wb) 1 1 None MUL. US Kb., Ma, Krad (Wrd + 1, Mrd) = unsigned(Wb) * unsigned(Wb) 1 1 None MUL. US Kb., Ma, Krad (Wrd + 1, Mrd) = unsigned(Wb) * unsigned(Wb) * unsigned(Wb) 1 1 None MUL. US Kb., Hill, E., Wrd (Wrd + 1, Mrd) = unsigned(Wb) * unsigned(Wb) * 1 1 None MUL. US Kb., Hill, E., Wrd (Wrd + 1, Mrd) = unsigned(Wb) * unsigned(Wb) * 1 1 None MUL. US Kb., Hill, E., Wrd (Wrd + 1, Mrd) = unsigned(Wb) * 1 1 None MUL f C Wild, H., Mrd) Wild, H., Wrd) = unsigned(Wb) * 1 1 None MUL, US Kb., Ma, Md Wd = Tis + 1 1 1 C.OC, NOVZ NBG Niggi E.Auge WrEG = i + 1 1 1 C.OC, NOVZ NBG NoP NoP NoOP No Operation 1 1 None So NOP POP POP POP from	51	MUL	MUL.SS	Wb,Ws,Wnd	{Wnd + 1, Wnd} = signed(Wb) * signed(Ws)	1	1	None
Mark Mark (Wnd + 1, Wnd) = unsigned(Wb)*signed(Wb) = 1 1 None MUL.UU Wb, He, Mnd (Wnd + 1, Wnd) = unsigned(Wb)* unsigned(Wb) = 1 1 1 None MUL.UU Wb, #1:15, Wnd (Wnd + 1, Wnd) = unsigned(Wb)* unsigned(Wb)* 1 1 None MUL.UU Wb, #1:15, Wnd (Wnd + 1, Wnd) = unsigned(Wb)* unsigned(Wb)* 1 1 None MUL.UU Wb, #1:15, Wnd (Wnd + 1, Wnd) = unsigned(Wb)* 1 1 None MUL.UU Wb, #1:15, Wnd (Wnd + 1, Wnd) = unsigned(Wb)* 1 1 None MUL.UU Wb, #1:15, Wnd (Wnd + 1, Wnd) = unsigned(Wb)* 1 1 None MUL.UU Wb, #1:15, Wnd (Wnd + 1, Wnd) = unsigned(Wb)* 1 1 None MUL.UU Wb, #1:15, Wnd (Wnd + 1, Wnd) = unsigned(Wb)* 1 1 None MUL.UU Wb, #1:15, Wnd Wol Wall Wol Wall None MUL.UU Wb, #1:15, Wnd Wol Wol Wol Wol None			MUL.SU	Wb,Ws,Wnd	{Wnd + 1, Wnd} = signed(Wb) * unsigned(Ws)	1	1	None
MUL.UU Wb.Ws.Ma. [Wnd+1, Wnd) = unsigned(Wb)* 1 1 None MUL.UU Wb.#lit5, Mnd [Wnd+1, Wnd] = unsigned(Wb)* 1 1 None MUL.UU Wb.#lit5, Mnd [Wnd+1, Wnd] = unsigned(Wb)* 1 1 None MUL f Ws.Miss.Wnd [Wnd+1, Wnd] = unsigned(Wb)* 1 1 None S2 MUL f Mode Masser None None MUL f Acc NegateAccumulator 1 1 1 C.DC.NOVZ NBG MSG w.Md WdEG =f+1 1 1 C.DC.NOVZ NBG NOP NOP No Operation 1 1 None S4 POP POP f Popffrom Top-of-Stack (TOS) 1 1 None S4 POP POP Wdd Popffrom Top-of-Stack (TOS) 1 1 None S54 POP POP Wdd Popffrom Top-of-Stack (TOS) 1 1 <td< td=""><td></td><td></td><td>MUL.US</td><td>Wb,Ws,Wnd</td><td>{Wnd + 1, Wnd} = unsigned(Wb) * signed(Ws)</td><td>1</td><td>1</td><td>None</td></td<>			MUL.US	Wb,Ws,Wnd	{Wnd + 1, Wnd} = unsigned(Wb) * signed(Ws)	1	1	None
NUL. SU NDL. BULLS, Wid. (Wnd + 1, Wnd) = signed(Wb) * unsigned(Wb) * unsigned(Wb) * 1 1 None MUL. UV Wb. #11L5, Wnd (Wnd + 1, Wnd) = unsigned(Wb) * unsigned(Wb) * 1 1 1 None S2 NEG Acc Negate Accumulator 1 1 1 OA GB QAB, SASB,SAB NEG Acc Negate Accumulator 1 1 1 C.DC.N.OVZ NEG f, WEBG f = f + 1 1 1 1 C.DC.N.OVZ S3 NOP NOP No Operation 1 1 1 None 54 POP POP P POP POP POP No Portion Top-of-Stack (TOS) 1 1 None F00. D Wnd Pop from Top-of-Stack (TOS) 1 1 None P09. N Pop from Top-of-Stack (TOS) 1 1 None P09. N Pop from Top-of-Stack (TOS) 1 1 None P09. N Pop Shadow Registers 1			MUL.UU	Wb,Ws,Wnd	{Wnd + 1, Wnd} = unsigned(Wb) * unsigned(Ws)	1	1	None
			MUL.SU	Wb,#lit5,Wnd	{Wnd + 1, Wnd} = signed(Wb) * unsigned(lit5)	1	1	None
MUL £ W3W2 = f * WREG 1 1 None 52 NEG NEG Acc Negate Accumulator 1 1 0A,0B,0AB, SA,8B,3AB NEG £ meg f=f+1 1 1 1 C,DC,N,OVZ NEG £,WEEG WREG=f+1 1 1 1 C,DC,N,OVZ 53 NOP NOP No Operation 1 1 None 54 POP £ Pop ff from Top-of-Stack (TOS) 1 1 None 54 POP £ Pop ff from Top-of-Stack (TOS) to Wdo 1 1 None 55 POS. Pop Stadox Registers 1 1 All 709.8.1 £ Push Wao Push Wao Registers 1 1 None 705.8.1 PUSH PUSH Wao Push Wao Registers 1 1 None 705.8.1 PUSH Wao Push Mone/Vach Registers 1 1 None 705.			MUL.UU	Wb,#lit5,Wnd	{Wnd + 1, Wnd} = unsigned(Wb) * unsigned(lit5)	1	1	None
52 NEG Acc Negate Accumulator 1 1 0A,0B,0AB,SAB NKG £ f=f+1 1 1 C,DC,NO,VZ NKG £,MKEG WREG=f+1 1 1 C,DC,NO,VZ NKG No No No 1 1 C,DC,NO,VZ S3 NOP NOP No Operation 1 1 None 54 POP £ Pop from Top-of-Stack (TOS) 1 1 None 55 POP. Wado Pop from Top-of-Stack (TOS) to Wdo 1 1 All 55 PUSH £ POP.D Wad Pop from Top-of-Stack (TOS) to Wdo 1 1 All 55 PUSH £ Push fito Top-of-Stack (TOS) 1 1 None FUSH £ PUSH fito Top-of-Stack (TOS) 1 1 None FUSH F PUSH Nso Push Wins/Wink +1) to Top-of-Stack (TOS) 1 1 None			MUL	f	W3:W2 = f * WREG	1	1	None
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$	52	NEG	NEG	Acc	Negate Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$			NEG	f	$f = \overline{f} + 1$	1	1	C,DC,N,OV,Z
NEG Ws. Wd Wd = Ws + 1 1 1 C.D.C.N.OV.Z 53 NOP NOP No Operation 1 1 None 54 POP POP F Pop from Top-of-Stack (TOS) 1 1 None 54 POP POP Mdo Pop from Top-of-Stack (TOS) 1 1 None 54 POP POP Mdo Pop from Top-of-Stack (TOS) 1 1 None 54 POP. Wdo Pop from Top-of-Stack (TOS) 1 1 None 55 PUSH FUSH f Push Mso Push Wso to Top-of-Stack (TOS) 1 1 None 56 PUSH Mso Push Wso Push Wso to Top-of-Stack (TOS) 1 1 None 57 RCALL Mso Push Madow Registers 1 1 None 56 PWRSAV #lit1 Go into Steep or Idle mode 1 1 None 57 RCALL KALL <			NEG	f,WREG	WREG = \overline{f} + 1	1	1	C,DC,N,OV,Z
53 NOP NOP No Operation 1 1 1 None 54 POP POP f Pop ff or Top-of-Stack (TOS) 1 1 None 54 POP POP f Pop for Top-of-Stack (TOS) 1 1 None 54 POP Wdo Pop form Top-of-Stack (TOS) to Wdo 1 1 None 55 POF.D Wnd Pop form Top-of-Stack (TOS) to Wdo 1 1 All 55 PUSH POSH Pop Shadow Registers 1 1 All 56 PUSH. DUSH Msso Push Wso to Top-of-Stack (TOS) 1 1 None 56 PWRSAV PUSH wso Push Wso to Top-of-Stack (TOS) 1 1 None 57 RCALL Expx Push Wso to Top-of-Stack (TOS) 1 1 None 57 RCALL Expx Relative Call 1 1 None 58 REPEAT #Lit1 Go into Ste			NEG	Ws,Wd	$Wd = \overline{Ws} + 1$	1	1	C,DC,N,OV,Z
NOPR No Operation 1 1 None 54 POP f Pop from Top-of-Stack (TOS) 1 1 None 54 POP f Pop from Top-of-Stack (TOS) to Wdo 1 1 None 50 POP. D Wind Pop from Top-of-Stack (TOS) to Wdo 1 1 None 55 POP. J. Wind Pop from Top-of-Stack (TOS) 1 1 All 55 PUSH PUSH f Push viso Push viso top-of-Stack (TOS) 1 1 None 709.8.2 PUSH Wiso Push Wiso top-of-Stack (TOS) 1 1 None 7098.9 PUSH Wiso Push Wiso top-of-Stack (TOS) 1 1 None 7098.0 Push Wiso Push Wins)*/(ms + 1) to Top-of-Stack (TOS) 1 1 None 7098.0 PRESAV PRESAV #Itit1 Go into Steep or Idle mode 1 1 None 56 PWRSAV PRESAV #Itit1 Repeat	53	NOP	NOP		No Operation	1	1	None
54POPfPop from Top-of-Stack (TOS)111NonePDPWdoPop from Top-of-Stack (TOS) to Wdo111NonePOP.DWndWoloPop from Top-of-Stack (TOS) to Wdo112NonePOP.SPop from Top-of-Stack (TOS) to12NoneNoneFUSHPUSHfPush for Top-of-Stack (TOS)111All55PUSHPUSHfPush for Top-of-Stack (TOS)111NonePUSH.JWsoPush Wso to Top-of-Stack (TOS)111NonePUSH.JWsoPush Wso to Top-of-Stack (TOS)111NoneFUSH.JWinsPush Wso to Top-of-Stack (TOS)111None56PWRSAVPWSAV#111Go into Steep or Idle mode11None57RCALLRCALLExprRelative Call12None58REPEATRELLWnComputed Call12None59RESETSoftware device Reset11None60RETFLERETURNReturn from Subroutine13 (2)None61RETURNRETURNReturn from Subroutine13 (2)None62RETURNRETURNReturn from Subroutine110.N.Z64RLCffRotale Left through Carry f11N.Z64 <td></td> <td></td> <td>NOPR</td> <td></td> <td>No Operation</td> <td>1</td> <td>1</td> <td>None</td>			NOPR		No Operation	1	1	None
POP WdoPop from Top-of-Stack (TOS) to Wdo111NonePOP.DWindPop from Top-of-Stack (TOS) to W(nd):W(nd+1)12None55PUSHFOP.SPop Shadow Registers111All55PUSHFUSHfPush fo Top-of-Stack (TOS)111None56PUSH.DWinsPush Wso to Top-of-Stack (TOS)111None57FUSH.SPush Shadow Registers111None57RCALLExprRelative Call12None58REPEATRCALLExprRelative Call12None59RESETROMComputed Call12None60RETFIERETIEReturn from interrupt13 (2)None61RETURRETUR#1it10, WnReturn from Subroutine11None62RETURRETURRETURReturn from Subroutine13 (2)None63RLCffReturn from Subroutine11C,N,Z64RLCfffReturn from Subroutine11N,Z64RLNCf, WREGWREG = Rotate Left through Carry f11N,Z65RECffRotate Left (No Carry) f11N,Z66RLNCf, WREGWREG = Rotate Left (No Carry) f11 <t< td=""><td>54</td><td>POP</td><td>POP</td><td>f</td><td>Pop f from Top-of-Stack (TOS)</td><td>1</td><td>1</td><td>None</td></t<>	54	POP	POP	f	Pop f from Top-of-Stack (TOS)	1	1	None
$ \begin{array}{ c c c c c c c c c c c c c c c c c c c$			POP	Wdo	Pop from Top-of-Stack (TOS) to Wdo	1	1	None
POP.SPop Shadow Registers11All55PUSHPUSH fPush fto Top-of-Stack (TOS)11NonePUSH.PUSH fPush WsoPush Wso to Top-of-Stack (TOS)11NonePUSH.DWnoPush Wso to Top-of-Stack (TOS)11NonePUSH.DWnoPush Wso to Top-of-Stack (TOS)12NonePUSH.DWnsPush Shadow Registers11None56PWRSAV#1it1Go into Steep or Idle mode11WDTO,Steep57RCALLExprRelative Call12None58RCALLWnComputed Call12None58REPEAT#lit14Repeat Next Instruction lit14 + 1 times11None59RESETRESETSoftware device Reset11None60RETFIERETFIEReturn from interrupt13 (2)None61RETURNRETURNReturn from Subroutine13 (2)None62RETURNRETURNReturn from Subroutine11C.N.Z64RLCffRotate Left through Carry f11N.Z.64RLNCF.N.WWREGRotate Left Noc Garry f11N.Z.65RCCRCCff=Rotate Left (No Carry) f11N.Z.66RLNCf, WREGWREG = Rotate Left Nongh Carry f11			POP.D	Wnd	Pop from Top-of-Stack (TOS) to W(nd):W(nd + 1)	1	2	None
55PUSHPUSHfPush foTop-of-Stack (TOS)111NonePUSHWsoPush Wso to Top-of-Stack (TOS)111NonePUSH.DWnsPush Wso to Top-of-Stack (TOS)12NonePUSH.SPush Shadow Registers111None56PWRSAVPWRSAV#1it1Go into Sleep or Idle mode111WDTO,Sleep57RCALLExprRelative Call12None58REPEATRCALLExprRelative Call12None59RESETREPEAT#1it14Repeat Next Instruction lit14 + 1 times11None59RESETRETFIEReturn from interrupt13 (2)None61RETUWRETTUW#1it10. WnReturn with literal in Wn13 (2)None62RETURNRETURNREturnf = Rotate Left through Carry f11C,N,Z63RLCf.f.f = Rotate Left through Carry f11N,Z64RLNCF.MREGWREG = Rotate Left (No Carry) f11N,Z65RRCRLNCf.KREGWREG = Rotate Left (No Carry) f11N,Z64RLNCf.KREGWREG = Rotate Left (No Carry) f11N,Z65RRCRRCf.KREGWREG = Rotate Left (No Carry) f11N,Z <tr< td=""><td></td><td></td><td>POP.S</td><td></td><td>Pop Shadow Registers</td><td>1</td><td>1</td><td>All</td></tr<>			POP.S		Pop Shadow Registers	1	1	All
PUSHWisoPush WisoPush Wiso to Top-of-Stack (TOS)111NonePUSH.DWinsPush Wins): W(ns + 1) to Top-of-Stack (TOS)12NoneFUSH.SPush Shadow Registers111None56PWRSAVPWRSAV#1it1Go into Sleep or Idle mode11WDTO,Sleep57RCALLExprRelative Call12None58REPEATREPEAT#1it14Repeat Next Instruction lit14 + 1 times11None59RESETREPEATWinRepeat Next Instruction lit14 + 1 times11None59RESETRESETSoftware device Reset11None60RETFIERETFIEReturn from interrupt13 (2)None61RETURRETURNReturn with literal in Wn13 (2)None62RETURNRETURNReturn with literal in Wn13 (2)None63RLCff = Rotate Left through Carry f11C,N,Z64RLNCf, WREGWREG = Rotate Left through Carry f11N,Z64RLNCf, WREGWREG = Rotate Left (No Carry) f11N,Z65RRCff = Rotate Left (No Carry) f11N,Z66RLNCf, WREGWREG = Rotate Left (No Carry) f11N,Z66RLNCf, WREGWREG = Rotate Left (No Carry) f1 <td>55</td> <td>PUSH</td> <td>PUSH</td> <td>f</td> <td>Push f to Top-of-Stack (TOS)</td> <td>1</td> <td>1</td> <td>None</td>	55	PUSH	PUSH	f	Push f to Top-of-Stack (TOS)	1	1	None
$\begin{array}{ c c c c c c c c c c c c c c c c c c c$			PUSH	Wso	Push Wso to Top-of-Stack (TOS)	1	1	None
PUSH.SPush Shadow Registers11None56PWRSAVPWRSAV#lit1Go into Sleep or Idle mode11WDTO,Sleep57RCALLExprRelative Call12None58REPEATREPEAT#lit14Repeat Next Instruction lit14 + 1 times11None58REPEAT#REPEAT#lit14Repeat Next Instruction (Wn) + 1 times11None59RESETRESETSoftware device Reset11None60RETFIERETFIEReturn from interrupt13 (2)None61RETUWRETUW#lit10, WnReturn with literal in Wn13 (2)None62RETURRETURNReturn from Subroutine13 (2)None63RLCfff e Rotate Left through Carry f11C,N,Z64RLCf, WREGWREG = Rotate Left through Carry f11N,Z64RLNCf, WREGWREG = Rotate Left (No Carry) f11N,Z65RRCRRCff e Rotate Left (No Carry) f11N,Z65RRCRRCff e Rotate Right through Carry f11N,Z66RLNCf, WREGWREG = Rotate Left (No Carry) f11N,Z67RLNCf, WREGWREG = Rotate Left (No Carry) f11N,Z68RLNCf, WREGWREG = Rotate Left (No Ca			PUSH.D	Wns	Push W(ns):W(ns + 1) to Top-of-Stack (TOS)	1	2	None
56PWRSAVPWRSAV#lit1Go into Sleep or Idle mode11WDTO,Sleep57RCALLRCALLExprRelative Call12None58REPEATREPEAT#lit14Repeat Next Instruction lit14 + 1 times112None58REPEATREPEAT#lit14Repeat Next Instruction lit14 + 1 times11None59RESETRESETWnRepeat Next Instruction (Wn) + 1 times11None60RETFIERETFIEReturn from interrupt13 (2)None61RETLWRETUW#lit10, WnReturn with literal in Wn13 (2)None62RETURNRETURNReturn from Subroutine13 (2)None63RLCff = Rotate Left through Carry f11C,N,Z64RLNCf, WREGWREG = Rotate Left through Carry f11N,Z65RRCRRCf = Rotate Left (No Carry) f11N,Z65RRCRRCf = Rotate Left (No Carry) f11N,Z66RRCRRCf = Rotate Left (No Carry) f11N,Z66RRCFf = Rotate Left (No Carry) f11N,Z67RRCff = Rotate Left (No Carry) f11N,Z68RLNCf, WREGWREG = Rotate Left (No Carry) f11N,Z69RRCRRCf =			PUSH.S		Push Shadow Registers	1	1	None
57RCALLRCALLExprRelative Call12None757RCALLWnComputed Call12None758REPEATREPEAT#1114Repeat Next Instruction lit14 + 1 times11None758REPEATREPEAT#1114Repeat Next Instruction lit14 + 1 times11None759RESETRESETWnRepeat Next Instruction (Wn) + 1 times11None759RESETRESETSoftware device Reset11None760RETFIERETFIEReturn from interrupt13 (2)None761RETLW#1110, WnReturn with literal in Wn13 (2)None762RETURNRETURNReturn from Subroutine13 (2)None763RLCffRotate Left through Carry f11C,N,Z764RLNCfgKREGWREG = Rotate Left through Carry f11N,Z764RLNCffRotate Left (No Carry) f11N,Z765RRCRRCff = Rotate Left (No Carry) f11N,Z765RRCRRCff = Rotate Right through Carry f11C,N,Z765RRCRRCff = Rotate Right through Carry f11C,N,Z765RRCRRCff = Rotate Right through Carry f11C,N,Z765<	56	PWRSAV	PWRSAV	#lit1	Go into Sleep or Idle mode	1	1	WDTO,Sleep
RCALLWnComputed Call12None58REPEATREPEAT#lit14Repeat Next Instruction lit14 + 1 times11None59RESETRESETWnRepeat Next Instruction (Wn) + 1 times11None60RETFIERESETSoftware device Reset11None61RETLWRETLW#lit10,WnReturn from interrupt13 (2)None62RETURNRETURNReturn With literal in Wn13 (2)None63RLCffRotate Left through Carry f11C,N,Z64RLCf, WREGWREG = Rotate Left through Carry f11N,Z65RRCRRCff = Rotate Left (No Carry) f11N,Z65RRCKRCff = Rotate Right through Carry f11N,Z66RRCFRotate Right through Carry f11N,Z67RLNCffRotate Left (No Carry) f11N,Z68RLNCF, WREGWREG = Rotate Left (No Carry) f11N,Z69RLNCF, WREGWREG = Rotate Left (No Carry) f11N,Z7RLNCffRotate Left (No Carry) f11N,Z7RLNCffRotate Left (No Carry) f11N,Z7RLNCffRotate Left (No Carry) f11 <td>57</td> <td>RCALL</td> <td>RCALL</td> <td>Expr</td> <td>Relative Call</td> <td>1</td> <td>2</td> <td>None</td>	57	RCALL	RCALL	Expr	Relative Call	1	2	None
58REPEATREPEAT#lit14Repeat Next Instruction lit14 + 1 times11None59RESETRESETWnRepeat Next Instruction (Wn) + 1 times11None60RETFIERESETSoftware device Reset11None61RETLWRETLW#lit10, WnReturn from interrupt13 (2)None61RETURNRETURNRETURNReturn from Subroutine13 (2)None62RETURNRETURNReturn from Subroutine13 (2)None63RLCffRotate Left through Carry f11C,N,Z64RLNCffRotate Left (No Carry) f11N,Z65RRCRRCffRotate Left (No Carry) f11N,Z65RRCRRCffRotate Left (No Carry) f11C,N,Z7RRCffRotate Left (No Carry) f11N,Z65RRCRRCffRotate Left (No Carry) f11N,Z65RRCRRCfMREGWREG = Rotate Right through Carry f11C,N,Z7RRCffRotate Left Right through Carry f11C,N,Z8RRCffRotate Left Right through Carry f11C,N,Z8RRCfgggGGGGG			RCALL	Wn	Computed Call	1	2	None
EXAMPLERepeatWnRepeat Next Instruction (Wn) + 1 times11None59RESETRESETRESETSoftware device Reset11None60RETFIERETFIEReturn from interrupt13 (2)None61RETLWRETLW#1it10,WnReturn with literal in Wn13 (2)None62RETURNRETURNReturn from Subroutine13 (2)None63RLCfff = Rotate Left through Carry f11C,N,Z64RLCf, WREGWREG = Rotate Left through Carry Ms11N,Z64RLNCf, WREGWREG = Rotate Left (No Carry) f11N,Z65RRCRRCff = Rotate Left (No Carry Ms11N,Z65RRCRRCff = Rotate Right through Carry f11C,N,Z66RRCRRCfMREGWREG = Rotate Left (No Carry) f11N,Z67RRCfRRCf = Rotate Left (No Carry) f11N,Z68RRCRRCfF = Rotate Left (No Carry) f11N,Z69RRCRRCfRRCf = Rotate Right through Carry f11C,N,Z60RRCRRCfRRCf = Rotate Right through Carry f11C,N,Z61RRCRRCRRCf = Rotate Right through Carry f11C,N	58	REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14 + 1 times	1	1	None
59RESETRESETSoftware device Reset11None60RETFIERETFIEReturn from interrupt13 (2)None61RETLWRETLW#lit10,WnReturn with literal in Wn13 (2)None62RETURNRETURNReturn from Subroutine13 (2)None63RLCfff = Rotate Left through Carry f11C,N,Z64RLCf,WREGWREG = Rotate Left through Carry fs11C,N,Z64RLNCf,WREGWREG = Rotate Left (No Carry) fs11N,Z65RRCRRCff = Rotate Left (No Carry fs11N,Z65RRCRRCff = Rotate Left (hrough Carry fs11N,Z66RRCRRCff = Rotate Left (ho Carry) fs11N,Z67RRCRRCfRotate Left (No Carry) fs11N,Z68RRCRRCfRotate Left (No Carry) fs11N,Z69RRCRRCfRotate Left (No Carry) fs11N,Z60RRCRRCfRotate Left (No Carry) fs11N,Z61RRCRRCfRotate Left (No Carry) fs11N,Z62RRCRRCfRotate Left (No Carry) fs11C,N,Z63RRCRRCfRotate Right through Carry fs <td></td> <td></td> <td>REPEAT</td> <td>Wn</td> <td>Repeat Next Instruction (Wn) + 1 times</td> <td>1</td> <td>1</td> <td>None</td>			REPEAT	Wn	Repeat Next Instruction (Wn) + 1 times	1	1	None
60RETFIERETFIEReturn from interrupt13 (2)None61RETLWRETLW#1it10,WnReturn with literal in Wn13 (2)None62RETURNRETURNReturn from Subroutine13 (2)None63RLCfff = Rotate Left through Carry f11C,N,Z64RLNCf,WREGWREG = Rotate Left through Carry f11N,Z64RLNCf,WREGWREG = Rotate Left (No Carry) f11N,Z65RRCRRCff = Rotate Left (No Carry f)11N,Z66RRCRRCff = Rotate Left (No Carry) f11N,Z67RRCRRCMREGWREG = Rotate Left (No Carry) f11N,Z68RRCRRCf,WREGWREG = Rotate Left (No Carry) f11N,Z69RRCRRCfRotate Left (No Carry) f11N,Z60RRCRRCfRotate Right through Carry f11C,N,Z61RRCF,WREGWREG = Rotate Right through Carry f11C,N,Z62RRCRRCfReturn for through Carry f11C,N,Z63RRCRRCRRCfRotate Right through Carry f11C,N,Z64RCRRCRRCRRCRRCRRCRRCRRCRRCRRC65RRC	59	RESET	RESET		Software device Reset	1	1	None
61RETLWRETLW#lit10, WnReturn with literal in Wn13 (2)None62RETURNRETURNReturn from Subroutine13 (2)None63RLCfffReturn from Subroutine11C,N,Z63RLCfffReturn from Subroutine11C,N,Z64RLCf, WREGWREGRotate Left through Carry f11C,N,Z64RLNCf, WREGWREGRotate Left (No Carry) f11N,Z64RLNCf, WREGWREG = Rotate Left (No Carry) f11N,Z65RRCRRCff = Rotate Left (No Carry f11N,Z65RRCRRCf, WREGWREG = Rotate Right through Carry f11C,N,Z7RRCf, WREGWREG = Rotate Right through Carry f11C,N,Z7RRCf, WREGWREG = Rotate Right through Carry f11C,N,Z7RRCf, WREGWREG = Rotate Right through Carry f11C,N,Z	60	RETFIE	RETFIE		Return from interrupt	1	3 (2)	None
62 RETURN RETURN Return from Subroutine 1 3 (2) None 63 RLC f f f = Rotate Left through Carry f 1 1 C,N,Z 64 RLC f, WREG WREG = Rotate Left through Carry Ws 1 1 C,N,Z 64 RLNC f, WREG WREG = Rotate Left through Carry Ms 1 1 N,Z 64 RLNC f, WREG WREG = Rotate Left (No Carry) f 1 1 N,Z 64 RLNC f, WREG WREG = Rotate Left (No Carry) f 1 1 N,Z 65 RRC RRC f Rotate Left (No Carry) Ms 1 1 C,N,Z 65 RRC RRC f Rotate Left (No Carry) Ms 1 1 N,Z 65 RRC RRC f Rotate Right through Carry f 1 1 C,N,Z REC f, WREG WREG = Rotate Right through Carry f 1 1 C,N,Z	61	RETLW	RETLW	#lit10,Wn	Return with literal in Wn	1	3 (2)	None
63RLCffRotate Left through Carry f11C,N,ZRLCf, WREGWREG = Rotate Left through Carry f11C,N,ZRLCWs, WdWd = Rotate Left through Carry Ws11C,N,Z64RLNCfff = Rotate Left (No Carry) f11N,ZRLNCf, WREGWREG = Rotate Left (No Carry) f11N,ZRLNCf, WREGWREG = Rotate Left (No Carry) f11N,Z65RRCRRCff = Rotate Right through Carry f11C,N,ZRRCf, WREGWREG = Rotate Right through Carry f11C,N,Z	62	RETURN	RETURN		Return from Subroutine	1	3 (2)	None
RLC f, WREG WREG = Rotate Left through Carry f 1 1 C,N,Z RLC Ws, Wd Wd = Rotate Left through Carry Ws 1 1 C,N,Z 64 RLNC f f R Rotate Left (No Carry) f 1 1 N,Z 64 RLNC f, WREG WREG = Rotate Left (No Carry) f 1 1 N,Z 64 RLNC f, WREG WREG = Rotate Left (No Carry) f 1 1 N,Z 65 RRC RRC f Restant f = Rotate Left (No Carry) Ws 1 1 N,Z 65 RRC RRC f Restant Restant f = Rotate Right through Carry f 1 1 C,N,Z 65 RRC RRC f, WREG WREG = Rotate Right through Carry f 1 1 C,N,Z 8 RRC f, WREG WREG = Rotate Right through Carry f 1 1 C,N,Z 9 RRC f, WREG Wd = Dotate Dight through Carry f 1 1 C,N,Z	63	RLC	RLC	f	f = Rotate Left through Carry f	1	1	C,N,Z
RLC Ws, Wd Wd = Rotate Left through Carry Ws 1 1 C,N,Z 64 RLNC f f Rotate Left (No Carry) f 1 1 N,Z 64 RLNC f mean f = Rotate Left (No Carry) f 1 1 N,Z 64 RLNC f, WREG WREG = Rotate Left (No Carry) f 1 1 N,Z 65 RRC RRC f f = Rotate Right through Carry f 1 1 N,Z 65 RRC RRC f, WREG WREG = Rotate Right through Carry f 1 1 C,N,Z 65 RRC RRC f, WREG Wd = Rotate Right through Carry f 1 1 C,N,Z 65 RRC RRC f, WREG Wd = Rotate Right through Carry f 1 1 C,N,Z 66 RRC RRC f, WREG Wd = Rotate Right through Carry f 1 1 C,N,Z			RLC	1,WREG	WREG = Rotate Left through Carry f	1	1	C,N,Z
04 RLNC I I I N,Z RLNC f, WREG WREG = Rotate Left (No Carry) f 1 1 N,Z RLNC f, WREG WREG = Rotate Left (No Carry) f 1 1 N,Z 65 RRC RRC f Ferror Ferror 1 1 N,Z 65 RRC RRC f, WREG WREG = Rotate Right through Carry f 1 1 C,N,Z RRC f, WREG WREG = Rotate Right through Carry f 1 1 C,N,Z RRC f, WREG Wd = Pototo Pictor Pictor Pictor Carry Wo 1 1 C,N,Z	64	D7.2-7	RLC	WS,Wd	wu = Rotate Left through Carry Ws	1	1	U,N,Z
RENC F, WREG WREG Rotate Left (No Carry) T T T T N,Z 65 RRC RRC f Rec f Rec f N,Z RRC f, WREG Wd = Rotate Left (No Carry) Ws 1 1 N,Z 65 RRC RRC f Rec f Rotate Right through Carry f 1 1 C,N,Z RRC f, WREG WREG = Rotate Right through Carry f 1 1 C,N,Z	04	RLNC	RLNC	I C UDDO		1	1	N,Z
REC REC f f N.Z 65 RRC f f Rec f N.Z RRC f Max Wd = Rotate Right through Carry f 1 1 C,N,Z RRC f, WREG WREG = Rotate Right through Carry f 1 1 C,N,Z RRC f, WREG Wd = Dotate Dight through Carry f 1 1 C,N,Z			RLNC	I,WREG	WREG = Rotate Left (No Carry) T	1	1	N,∠
$\frac{1}{1} = RC + RC$	6F	DDC	RLNC	ws,Wa	f = Pototo Pight through Correct	1	1	
RKC L, WKEG WKEG = Kolale Kight through Carry Ma I I C,N,Z PPC Ma Wd = Datata Diabt through Carry Ma I I C,N,Z	00	KKC	RRC	L f WDEC	I - Rotate Right through Carry f	1	1	
			NRC DAG	L, WKEG	Wite Brotate Right through Carry We	1	1	

TABLE 23-2: INSTRUCTION SET OVERVIEW (CONTINUED)

25.1 DC Characteristics

Charactoristic	VDD Range	Temp Range	Max MIPS		
Characteristic	(in Volts)	(in °C)	dsPIC33FJXXXGPX06A/X08A/X10A		
—	VBOR-3.6V ⁽¹⁾	-40°C to +85°C	40		
_	VBOR-3.6V ⁽¹⁾	-40°C to +125°C	40		

Note 1: Device is functional at VBORMIN < VDD < VDDMIN. Analog modules such as the ADC will have degraded performance. Device functionality is tested but not characterized. Refer to parameter BO10 in Table 25-11 for the minimum and maximum BOR values.

TABLE 25-2: THERMAL OPERATING CONDITIONS

Rating	Symbol	Min	Тур	Max	Unit
dsPIC33FJXXXGPX06A/X08A/X10A					
Operating Junction Temperature Range	TJ	-40	—	+125	°C
Operating Ambient Temperature Range	TA	-40	—	+85	°C
Extended Temperature Devices					
Operating Junction Temperature Range	TJ	-40	—	+150	°C
Operating Ambient Temperature Range	TA	-40	—	+125	°C
Power Dissipation: Internal chip power dissipation: $PINT = VDD x (IDD - \Sigma IOH)$ I/O Pin Power Dissipation:		Pint + Pi/o		w	
$I/O = \Sigma (\{VDD - VOH\} \times IOH) + \Sigma (VOL \times IOL)$					
Maximum Allowed Power Dissipation	PDMAX	(TJ - TA)/θJ	A	W

TABLE 25-3: THERMAL PACKAGING CHARACTERISTICS

Characteristic	Symbol	Тур	Max	Unit	Notes
Package Thermal Resistance, 100-pin TQFP (14x14x1 mm)	θja	40	—	°C/W	1
Package Thermal Resistance, 100-pin TQFP (12x12x1 mm)	θја	40	—	°C/W	1
Package Thermal Resistance, 80-pin TQFP (12x12x1 mm)	θја	40	_	°C/W	1
Package Thermal Resistance, 64-pin TQFP (10x10x1 mm)	θја	40	—	°C/W	1
Package Thermal Resistance, 64-pin QFN (9x9x0.9 mm)	θja	28	—	°C/W	1

Note 1: Junction to ambient thermal resistance, Theta-JA (θ JA) numbers are achieved by package simulations.



TABLE 25-39: DCI MODULE (AC-LINK MODE) TIMING REQUIREMENTS

AC CHARACTERISTICS			$\begin{tabular}{lllllllllllllllllllllllllllllllllll$				
Param No.	Symbol	Characteristic ^(1,2)	Min	Тур ⁽³⁾	Max	Units	Conditions
CS60	TBCLKL	BIT_CLK Low Time	36	40.7	45	ns	—
CS61	TBCLKH	BIT_CLK High Time	36	40.7	45	ns	—
CS62	TBCLK	BIT_CLK Period	—	81.4	_	ns	Bit clock is input
CS65	TSACL	Input Setup Time to Falling Edge of BIT_CLK	—	—	10	ns	_
CS66	THACL	Input Hold Time from Falling Edge of BIT_CLK	—	—	10	ns	_
CS70	TSYNCLO	SYNC Data Output Low Time	_	19.5		μs	Note 1
CS71	TSYNCHI	SYNC Data Output High Time	—	1.3	_	μs	Note 1
CS72	TSYNC	SYNC Data Output Period	—	20.8	_	μS	Note 1
CS75	TRACL	Rise Time, SYNC, SDATA_OUT	—	10	25	ns	CLOAD = 50 pF, VDD = 5V
CS76	TFACL	Fall Time, SYNC, SDATA_OUT	—	10	25	ns	CLOAD = 50 pF, VDD = 5V
CS77	TRACL	Rise Time, SYNC, SDATA_OUT	—	—	30	ns	CLOAD = 50 pF, VDD = 3V
CS78	TFACL	Fall Time, SYNC, SDATA_OUT	—	—	30	ns	CLOAD = 50 pF, VDD = 3V
CS80	TOVDACL	Output Valid Delay from Rising Edge of BIT_CLK	—	_	15	ns	_

Note 1: These parameters are characterized but not tested in manufacturing.

2: These values assume BIT_CLK frequency is 12.288 MHz.

3: Data in "Typ" column is at 3.3V, 25°C unless otherwise stated. Parameters are for design guidance only and are not tested.

AC CHARACTERISTICS		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +150^{\circ}C$ for High Temperature					
Param No.	Symbol	Characteristic ⁽¹⁾	Min	Тур	Max	Units	Conditions
HSP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge	—	_	35	ns	_
HSP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	25	_	—	ns	_
HSP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	25	—	_	ns	_
HSP51	TssH2doZ	SSx ↑ to SDOx Output High-Impedance	15	_	55	ns	See Note 2

TABLE 26-12: SPIX MODULE SLAVE MODE (CKE = 0) TIMING REQUIREMENTS

Note 1: These parameters are characterized but not tested in manufacturing.

2: Assumes 50 pF load on all SPIx pins.

TABLE 26-13: SPIX MODULE SLAVE MODE (CKE = 1) TIMING REQUIREMENTS

AC CHARACTERISTICS		Standard Operating Conditions: 3.0V to 3.6V (unless otherwise stated)Operating temperature $-40^{\circ}C \le TA \le +150^{\circ}C$ for High Temperature						
Param No.	Symbol	Characteristic ⁽¹⁾	Min	Тур	Max	Units	Conditions	
HSP35	TscH2doV, TscL2doV	SDOx Data Output Valid after SCKx Edge			35	ns	_	
HSP40	TdiV2scH, TdiV2scL	Setup Time of SDIx Data Input to SCKx Edge	25	—	—	ns	_	
HSP41	TscH2diL, TscL2diL	Hold Time of SDIx Data Input to SCKx Edge	25	_	_	ns	_	
HSP51	TssH2doZ	SSx ↑ to SDOx Output High-Impedance	15	_	55	ns	See Note 2	
HSP60	TssL2doV	SDOx Data Output Valid after SSx Edge	_	_	55	ns	—	

Note 1: These parameters are characterized but not tested in manufacturing.

2: Assumes 50 pF load on all SPIx pins.

28.0 PACKAGING INFORMATION

28.1 Package Marking Information



Legend	: XXX	Customer-specific information					
	Y	Year code (last digit of calendar year)					
	ΥY	ear code (last 2 digits of calendar year)					
	WW	/eek code (week of January 1 is week '01')					
	NNN	Alphanumeric traceability code					
	(e3)	Pb-free JEDEC designator for Matte Tin (Sn)					
	*	This package is Pb-free. The Pb-free JEDEC designator ($_{(e3)}$)					
		can be found on the outer packaging for this package.					
Note:	In the ever	nt the full Microchip part number cannot be marked on one line, it will					
	be carried	J over to the next line, thus limiting the number of available					
	characters	for customer-specific information.					

100-Lead Plastic Thin Quad Flatpack (PT) – 12x12x1 mm Body, 2.00 mm Footprint [TQFP]

Note: For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging



Notes:

1. Pin 1 visual index feature may vary, but must be located within the hatched area.

2. Chamfers at corners are optional; size may vary.

3. Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.

- 4. Dimensioning and tolerancing per ASME Y14.5M.
 - BSC: Basic Dimension. Theoretically exact value shown without tolerances.

REF: Reference Dimension, usually without tolerance, for information purposes only.

Microchip Technology Drawing C04-100B