

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

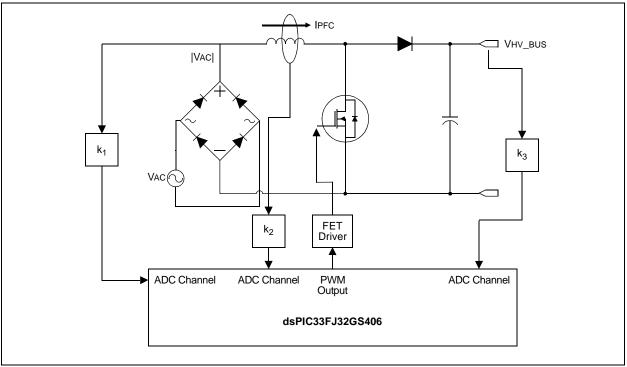
E·XFI

| Details | |
|----------------------------|---|
| Product Status | Active |
| Core Processor | dsPIC |
| Core Size | 16-Bit |
| Speed | 40 MIPs |
| Connectivity | I ² C, IrDA, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, QEI, POR, PWM, WDT |
| Number of I/O | 58 |
| Program Memory Size | 32KB (32K x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 4K x 8 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | A/D 16x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-VFQFN Exposed Pad |
| Supplier Device Package | 64-VQFN (9x9) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/dspic33fj32gs406-e-mr |
| | |

Email: info@E-XFL.COM

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







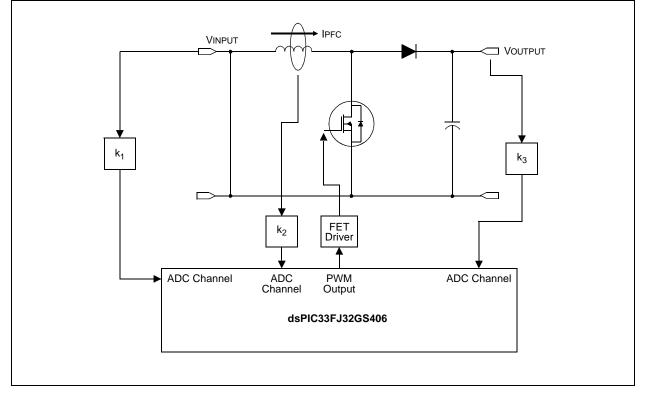


TABLE 4-18: HIGH-SPEED PWM GENERATOR 2 REGISTER MAP

| File Name | SFR Addr | Bit 15 | Bit 14 | Bit 13 | Bit 12 | Bit 11 | Bit 10 | Bit 9 | Bit 8 | Bit 7 | Bit 6 | Bit 5 | Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | All Resets |
|--------------|-------------|---------|-------------------|---------|---------|-----------|-----------|-----------|-----------|----------|---------|----------|----------|----------|----------|----------|----------|---------------|
| PWMCON2 | 0440 | FLTSTAT | CLSTAT | TRGSTAT | FLTIEN | CLIEN | TRGIEN | ITB | MDCS | DTC1 | DTC0 | DTCP | | MTBS | CAM | XPRES | IUE | 0000 |
| IOCON2 | 0442 | PENH | PENL | POLH | POLL | PMOD1 | PMOD0 | OVRENH | OVRENL | OVRDAT1 | OVRDAT0 | FLTDAT1 | FLTDAT0 | CLDAT1 | CLDAT0 | SWAP | OSYNC | 0000 |
| FCLCON2 | 0444 | IFLTMOD | CLSRC4 | CLSRC3 | CLSRC2 | CLSRC1 | CLSRC0 | CLPOL | CLMOD | FLTSRC4 | FLTSRC3 | FLTSRC2 | FLTSRC1 | FLTSRC0 | FLTPOL | FLTMOD1 | FLTMOD0 | 0000 |
| PDC2 | 0446 | | | | | | | | PDC2 | 2<15:0> | | | | | | | | 0000 |
| PHASE2 | 0448 | | PHASE2<15:0> 00 | | | | | | | | | | 0000 | | | | | |
| DTR2 | 044A | _ | — — DTR2<13:0> 0 | | | | | | | | | 0000 | | | | | | |
| ALTDTR2 | 044C | _ | — — ALTDTR2<13:0> | | | | | | | | 0000 | | | | | | | |
| SDC2 | 044E | | | | | | | | SDC2 | 2<15:0> | | | | | | | | 0000 |
| SPHASE2 | 0450 | | | | | | | | SPHAS | E2<15:0> | | | | | | | | 0000 |
| TRIG2 | 0452 | | | | | | | TRGCMP<12 | 2:0> | | | | | | _ | _ | — | 0000 |
| TRGCON2 | 0454 | TRGDIV3 | TRGDIV2 | TRGDIV1 | TRGDIV0 | _ | _ | — | _ | DTM | _ | TRGSTRT5 | TRGSTRT4 | TRGSTRT3 | TRGSTRT2 | TRGSTRT1 | TRGSTRT0 | 0000 |
| STRIG2 | 0456 | | | | | | | STRGCMP<1 | 2:0> | | | | | | _ | _ | — | 0000 |
| PWMCAP2 | 0458 | | | | | | | PWMCAP<1 | 2:0> | | | | | | _ | _ | — | 0000 |
| LEBCON2 | 045A | PHR | PHF | PLR | PLF | FLTLEBEN | CLLEBEN | _ | _ | _ | _ | BCH | BCL | BPHH | BPHL | BPLH | BPLL | 0000 |
| LEBDLY2 | 045C | _ | _ | _ | — | • | | | L | EB<8:0> | • | | | - | | | | 0000 |
| AUXCON2 | 045E | HRPDIS | HRDDIS | _ | _ | BLANKSEL3 | BLANKSEL2 | BLANKSEL1 | BLANKSEL0 | _ | — | CHOPSEL3 | CHOPSEL2 | CHOPSEL1 | CHOPSEL0 | CHOPHEN | CHOPLEN | 0000 |

Legend: x = unknown value on Reset, — = unimplemented, read as '0'. Reset values are shown in hexadecimal.

| REGISTER 7 | 7-6: IFS1: I | NTERRUPT | FLAG STAT | | ER 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 |
| U2TXIF | U2RXIF | INT2IF | T5IF | T4IF | OC4IF | OC3IF | DMA2IF |
| bit 15 | | | | | | | bit 8 |
| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
| _ | _ | _ | INT1IF | CNIF | AC1IF | MI2C1IF | SI2C1IF |
| bit 7 | | | | | | | bit (|
| Legend: | | | | | | | |
| R = Readable | e bit | W = Writable | bit | U = Unimpler | mented bit, rea | d as '0' | |
| -n = Value at | POR | '1' = Bit is set | t | '0' = Bit is cle | ared | x = Bit is unkr | iown |
| bit 12 | U2TXIF: UAR | RT2 Transmitte | r Interrupt Fla | g Status bit | | | |
| | 1 = Interrupt r | request has oc | curred | - | | | |
| | • | request has no | | | | | |
| bit 11 | | RT2 Receiver I | | Status bit | | | |
| | | request has oc | | | | | |
| bit 13 | - | request has no | | :. | | | |
| DIL 13 | | nal Interrupt 2 request has oc | 0 | п | | | |
| | • | request has no | | | | | |
| bit 12 | - | Interrupt Flag | | | | | |
| | 1 = Interrupt r | request has oc request has no | curred | | | | |
| bit 11 | - | Interrupt Flag | | | | | |
| | | request has oc | | | | | |
| | | request has no | | | | | |
| bit 10 | OC4IF: Outpu | ut Compare Ch | nannel 4 Interr | upt Flag Status | s bit | | |
| | • | request has oc request has no | | | | | |
| bit 9 | - | - | | upt Flag Status | bit | | |
| | - | request has oc | | optillag etailet | 2 | | |
| | • | equest has no | | | | | |
| bit 8 | DMA2IF: DM | A Channel 2 D | ata Transfer (| Complete Interr | upt Flag Status | s bit | |
| | | request has oc request has no | | | | | |
| bit 7-5 | Unimplemen | ted: Read as ' | 0' | | | | |
| bit 4 | INT1IF: Exter | nal Interrupt 1 | Flag Status b | it | | | |
| | | request has oc request has no | | | | | |
| bit 3 | - | - | | Flag Status bit | | | |
| | 1 = Interrupt r | request has oc request has no | curred | - | | | |
| bit 2 | • | g Comparator | | ag Status bit | | | |
| - | | request has oc | - | J | | | |
| | 0 = Interrupt r | - | | | | | |

REGISTER 7-6: IFS1: INTERRUPT FLAG STATUS REGISTER 1

| | -1: DMAX | | | | | | |
|--------------------|---|--------------------------------------|---|------------------------------------|-----------------|------------------|------------|
| 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 |
| | | AMODE1 | AMODE0 | — | — | MODE1 | MODE0 |
| bit 7 | | | | | | | bit C |
| Legend: | | | | | | | |
| R = Readable | bit | W = Writable | hit | U = Unimpler | nented bit, rea | d as '0' | |
| -n = Value at F | | '1' = Bit is set | | '0' = Bit is cle | | x = Bit is unkn | own |
| | ÖN | 1 – Dit 13 36t | | | aleu | | lowin |
| bit 15 | CHEN: DMA | Channel Enabl | e bit | | | | |
| | 1 = Channel i | s enabled | | | | | |
| | 0 = Channel i | s disabled | | | | | |
| bit 14 | SIZE: Data Tr | ansfer Size bit | | | | | |
| | 1 = Byte 0 = Word | | | | | | |
| bit 13 | DIR: Transfer | Direction bit (s | source/destina | tion bus select | t) | | |
| | | m DMA RAM a m peripheral a | • | | | | |
| bit 12 | | Block Transfer | | | | | |
| 5.1.12 | 1 = Initiates b | lock transfer co lock transfer co | omplete interru | upt when half c | of the data has | | |
| bit 11 | | Data Periphera | - | - | | | |
| | | write to periphe | | | write (DIR bit | must also be cle | ar) |
| bit 10-6 | - | ted: Read as ' | 0' | | | | |
| bit 5-4 | - | -: DMA Chann | | lode Select bit | S | | |
| | 11 = Reserve | | 5 | | | | |
| | | al Indirect Add | | | | | |
| | - | Indirect withou Indirect with F | | | | | |
| | - | | | mode | | | |
| hit 3-2 | Unimplemen | ted: Read as ' | | | | | |
| bit 3-2 bit 1-0 | Unimplemen | | | de Select hits | | | |
| bit 3-2 bit 1-0 | MODE<1:0>: | DMA Channel | Operating Mo | | k transfer from | n/to each DMA R | AM buffer) |
| | MODE<1:0>: 11 = One-Sho | DMA Channel | Operating Mo nodes are ena | abled (one bloc | k transfer from | n/to each DMA R | AM buffer) |
| | MODE<1:0>: 11 = One-Sho 10 = Continuo 01 = One-Sho | DMA Channel ot, Ping-Pong r | Operating Mo nodes are ena modes are en nodes are disa | abled (one bloc nabled abled | k transfer from | n/to each DMA R | AM buffer) |

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

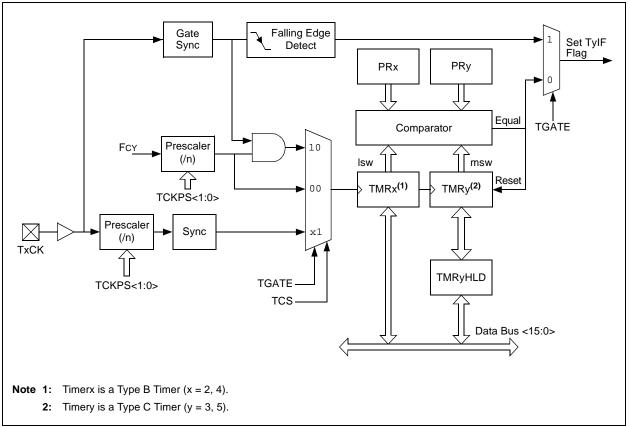
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | U-0 | U-0 |
|-----------------------------------|--------------|-------------------------------------|----------------|----------------------|------------------|--------------------|-------|
| — | — | — | — | — | CMPMD | — | — |
| bit 15 | | | | | | | bit 8 |
| | | | | | | | |
| U-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 | R/W-0 | U-0 |
| — | — | QEI2MD | | — | — | I2C2MD | — |
| bit 7 | | | | | | | bit 0 |
| | | | | | | | |
| Legend: | | | | | | | |
| R = Readable bit W = Writable bit | | | | U = Unimplen | nented bit, read | d as '0' | |
| -n = Value a | t POR | '1' = Bit is set | | '0' = Bit is cleared | | x = Bit is unknown | |
| | | | | | | | |
| bit 15-11 | Unimplemen | ted: Read as ' | כ' | | | | |
| bit 10 | CMPMD: Ana | alog Comparato | or Module Disa | ble bit | | | |
| | | omparator mode | | | | | |
| bit 9-6 | Unimplemen | ted: Read as ' |)' | | | | |
| bit 5 | QEI2MD: QE | I2 Module Disa | ble bit | | | | |
| | | dule is disabled dule is enabled | | | | | |
| bit 4-2 | Unimplemen | ted: Read as ' |)' | | | | |
| bit 1 | - | 2 Module Disab | | | | | |
| | 1 = I2C2 mod | lule is disabled | | | | | |
| | 0 = I2C2 mod | lule is enabled | | | | | |
| bit 0 | Unimplemen | ted: Read as ' |)' | | | | |
| | | | | | | | |

REGISTER 10-3: PMD3: PERIPHERAL MODULE DISABLE CONTROL REGISTER 3

REGISTER 10-4: PMD4: PERIPHERAL MODULE DISABLE CONTROL REGISTER 4

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|------------|---|--|--|---|--|--|
| — | — | — | — | — | — | — |
| | | | | | | bit 8 |
| | | | | | | |
| U-0 | U-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 |
| — | — | _ | REFOMD | _ | — | |
| · | • | | | | | bit 0 |
| | | | | | | |
| | | | | | | |
| e bit | W = Writable I | oit | U = Unimplem | nented bit, read | l as '0' | |
| POR | '1' = Bit is set | | 0' = Bit is cleared $x = Bit is unl$ | | | nown |
| | | | | | | |
| Unimplemen | ted: Read as 'o |)' | | | | |
| REFOMD: Re | eference Clock | Generator Mod | dule Disable bit | | | |
| | - | | | | | |
| | | | | | | |
| | U-0 U-0 e bit POR Unimplemen REFOMD: Re 1 = Reference | U-0 U-0 U-0 U-0 — — — e bit W = Writable I POR '1' = Bit is set Unimplemented: Read as '0 REFOMD: Reference Clock 1 = Reference clock generat | U-0 U-0 U-0 U-0 U-0 U-0 Hermitian Hermitian Hermitian Bit W = Writable bit Hermitian POR '1' = Bit is set Hermitian Unimplemented: Read as '0' REFOMD: Reference Clock Generator Moor 1 = Reference clock generator module is dited in the set | — — — — U-0 U-0 U-0 R/W-0 — — — REFOMD e bit W = Writable bit U = Unimplem POR '1' = Bit is set '0' = Bit is clear Unimplemented: Read as '0' U U | - - - - U-0 U-0 U-0 R/W-0 U-0 - - - REFOMD - e bit W = Writable bit U = Unimplemented bit, read POR '1' = Bit is set '0' = Bit is cleared Unimplemented: Read as '0' REFOMD: Reference Clock Generator Module Disable bit 1 = Reference clock generator module is disabled | Image: definition of the second state of the second st |





| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | | | | | |
|--------------|---------------|---|----------------|------------------|----------------------------|-------------------------|-------------------------|--|--|--|--|--|
| _ | _ | — | — | — | — | — | — | | | | | |
| bit 15 | · | | | | | · | bit 8 | | | | | |
| | | | | | | | | | | | | |
| U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | | | | | |
| _ | — | — | — | — | PCLKDIV2 ⁽¹⁾ | PCLKDIV1 ⁽¹⁾ | PCLKDIV0 ⁽¹⁾ | | | | | |
| bit 7 | · | | | | | • | bit 0 | | | | | |
| | | | | | | | | | | | | |
| Legend: | | | | | | | | | | | | |
| R = Readab | ole bit | W = Writable | bit | U = Unimpler | mented bit, read | l as '0' | | | | | | |
| -n = Value a | at POR | '1' = Bit is set | | '0' = Bit is cle | eared | x = Bit is unknown | | | | | | |
| | | | | | | | | | | | | |
| bit 15-3 | Unimplemen | ted: Read as ' | 0' | | | | | | | | | |
| bit 2-0 | PCLKDIV<2: | 0>: PWM Input | t Clock Presca | aler (Divider) S | Select bits ⁽¹⁾ | | | | | | | |
| | 111 = Reserv | red | | | | | | | | | | |
| | 110 = Divide- | by-64, maximu | ım PWM timin | g resolution | | | | | | | | |
| | | 101 = Divide-by-32, maximum PWM timing resolution | | | | | | | | | | |
| | | hy-16 maxim | | • | | | | | | | | |

100 = Divide-by-16, maximum PWM timing resolution

011 = Divide-by-8, maximum PWM timing resolution

010 = Divide-by-4, maximum PWM timing resolution

001 = Divide-by-2, maximum PWM timing resolution

000 = Divide-by-1, maximum PWM timing resolution (power-on default)

Note 1: These bits should be changed only when PTEN = 0. Changing the clock selection during operation will yield unpredictable results.

REGISTER 16-7: STPER: PWM SECONDARY MASTER TIME BASE PERIOD REGISTER

| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | |
|--|------------------|---------------------------------|---|---|---|---|--|
| | - | STPE | R<15:8> | | | | |
| | | | | | | bit 8 | |
| | | | | | | | |
| R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-0 | R/W-0 | R/W-0 | |
| | | STPE | R<7:0> | | | | |
| | | | | | | bit 0 | |
| | | | | | | | |
| Legend: R = Readable bit W = Writable bit | | | | nented bit, read | d as '0' | | |
| OR | '1' = Bit is set | | '0' = Bit is cle | ared | x = Bit is unknown | | |
| | R/W-1 | R/W-1 R/W-1 bit W = Writable | STPE R/W-1 R/W-1 R/W-1 STPE bit W = Writable bit | STPER<15:8> R/W-1 R/W-1 R/W-1 STPER<7:0> STPER<7:0> | STPER<15:8> R/W-1 R/W-1 R/W-0 STPER<7:0> bit W = Writable bit U = Unimplemented bit, read | $STPER < 15:8 >$ $R/W-1 \qquad R/W-1 \qquad R/W-0 \qquad R/W-0$ $STPER < 7:0 >$ bit W = Writable bit U = Unimplemented bit, read as '0' | |

bit 15-0 STPER<15:0>: Secondary Master Time Base (SMTMR) Period Value bits

REGISTER 16-12: PDCx: PWM GENERATOR DUTY CYCLE x REGISTER^(1,2,3)

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|-----------------|-------|------------------|-------|------------------|-----------------|-----------------|-------|
| | | | PDC | x<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 |
| | | | PDC | x<7:0> | | | |
| bit 7 | | | | | | | bit 0 |
| Legend: | | | | | | | |
| • | h:+ | W/ Writchlah | :4 | | nantad hit raa | d aa '0' | |
| R = Readable | | W = Writable b | It | • | nented bit, rea | | |
| -n = Value at P | POR | '1' = Bit is set | | '0' = Bit is cle | ared | x = Bit is unkr | nown |

bit 15-0 PDCx<15:0>: PWM Generator # Duty Cycle Value bits

- **Note 1:** In Independent PWM mode, the PDCx register controls the PWMxH duty cycle only. In the Complementary, Redundant and Push-Pull PWM modes, the PDCx register controls the duty cycle of both the PWMxH and PWMxL.
 - **2:** The smallest pulse width that can be generated on the PWM output corresponds to a value of 0x0009, while the maximum pulse width generated corresponds to a value of Period 0x0009.
 - **3:** As the duty cycle gets closer to 0% or 100% of the PWM period (0 to 40 ns, depending on the mode of operation), PWM duty cycle resolution will increase from 1 to 3 LSBs.

REGISTER 16-13: SDCx: PWM SECONDARY DUTY CYCLE x REGISTER^(1,2,3)

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|-----------------|-------|------------------|--|-------------------|-------|--------------------|-------|
| | | | SDC | x<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 |
| | | | | 5x<7:0> | | | |
| bit 7 | | | | | | | bit 0 |
| Legend: | | | | | | | |
| R = Readable | bit | W = Writable b | bit U = Unimplemented bit, read as '0' | | | | |
| -n = Value at P | OR | '1' = Bit is set | | '0' = Bit is clea | ared | x = Bit is unknown | |

bit 15-0 **SDCx<15:0>:** Secondary Duty Cycle bits for PWMxL Output Pin

- **Note 1:** The SDCx register is used in Independent PWM mode only. When used in Independent PWM mode, the SDCx register controls the PWMxL duty cycle.
 - **2:** The smallest pulse width that can be generated on the PWM output corresponds to a value of 0x0009, while the maximum pulse width generated corresponds to a value of Period 0x0009.
 - **3:** As the duty cycle gets closer to 0% or 100% of the PWM period (0 to 40 ns, depending on the mode of operation), PWM duty cycle resolution will increase from 1 to 3 LSBs.

| U-0 | U-0 | U-0 | U-0 | U-0 | U-0 | R/W-0 | R/W-0 | |
|-----------------|-------|------------------|-------|-------------------|------------------|--------------------|-------|--|
| _ | — | _ | — | — | _ | AMSK | <9:8> | |
| 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 | |
| | | | AMS | K<7:0> | | | | |
| bit 7 | | | | | | | bit 0 | |
| | | | | | | | | |
| Legend: | | | | | | | | |
| R = Readable I | bit | W = Writable | bit | U = Unimplen | nented bit, read | l as '0' | | |
| -n = Value at P | OR | '1' = Bit is set | | '0' = Bit is clea | ared | x = Bit is unknown | | |

REGISTER 19-3: I2CxMSK: I2Cx SLAVE MODE ADDRESS MASK REGISTER

bit 15-10 Unimplemented: Read as '0'

bit 9-0

AMSK<9:0>: Mask for Address bit x Select bits

1 = Enables masking for bit x of incoming message address; bit match is not required in this position 0 = Disables masking for bit x; bit match is required in this position

| U-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | | | |
|--------------------|--|---|-------|-----------------------------|-------|--------------------|-------|--|--|--|
| _ | | FBP5 | FBP4 | FBP3 | FBP2 | FBP1 | FBP0 | | | |
| bit 15 | · | | | | | | bit 8 | | | |
| | | | | | | | | | | |
| U-0 | U-0 | R-0 | R-0 | R-0 | R-0 | R-0 | R-0 | | | |
| — | — | FNRB5 | FNRB4 | FNRB3 | FNRB2 | FNRB1 | FNRB0 | | | |
| bit 7 | | | | | | | bit (| | | |
| | | | | | | | | | | |
| Legend: | | | | | | | | | | |
| R = Readable bit | | W = Writable bit | | U = Unimplemented bit, read | | d as '0' | | | | |
| -n = Value at POR | | '1' = Bit is set | | '0' = Bit is cleared | | x = Bit is unknown | | | | |
| 6:4 <i>6</i> 44 | | atadı Daadaa (| o' | | | | | | | |
| bit 15-14 | - | nted: Read as ' | | | | | | | | |
| bit 13-8 | FBP<5:0>: FIFO Buffer Pointer bits | | | | | | | | | |
| | 011111 = RB31 buffer 011110 = RB30 buffer | | | | | | | | | |
| | 011110 = Ri | B30 buller | | | | | | | | |
| | • | | | | | | | | | |
| | • | | | | | | | | | |
| | 000001 = TRB1 buffer | | | | | | | | | |
| | 000000 = TF | | | | | | | | | |
| | Unimplemented: Read as '0' | | | | | | | | | |
| | - | | | | | | | | | |
| | FNRB<5:0>: | FIFO Next Rea | | ter bits | | | | | | |
| | FNRB<5:0>: 011111 = RI | FIFO Next Rea B31 buffer | | ter bits | | | | | | |
| | FNRB<5:0>: | FIFO Next Rea B31 buffer | | ter bits | | | | | | |
| | FNRB<5:0>: 011111 = RI | FIFO Next Rea B31 buffer | | ter bits | | | | | | |
| | FNRB<5:0>: 011111 = RI | FIFO Next Rea B31 buffer | | ter bits | | | | | | |
| bit 7-6 bit 5-0 | FNRB<5:0>: 011111 = RI | FIFO Next Rea B31 buffer B30 buffer | | ter bits | | | | | | |

REGISTER 21-5: CxFIFO: ECANx FIFO STATUS REGISTER

BUFFER 21-5: ECANx MESSAGE BUFFER WORD 4

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|-------------------------------------|-------|-------|------------------|------------------|-----------------|-------|-------|
| _ | | | Ву | te 3 | | | |
| 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 |
| _ | | | Ву | te 2 | | | |
| bit 7 | | | | | | | bit 0 |
| Legend: | | | | | | | |
| R = Readable bit $W = Writable bit$ | | bit | U = Unimpler | mented bit, read | d as '0' | | |
| -n = Value at POR '1' = Bit is set | | | '0' = Bit is cle | ared | x = Bit is unkr | nown | |

| bit 15-8 | Byte 3<15:8>: ECANx Message Byte 3 |
|----------|------------------------------------|
| bit 7-0 | Byte 2<7:0>: ECANx Message Byte 2 |

BUFFER 21-6: ECANx MESSAGE BUFFER WORD 5

| R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x | R/W-x |
|-----------------------------------|-------|------------------|------------------------------------|-------------------|-------|-----------------|-------|
| | | | B | yte 5 | | | |
| 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 |
| | | | B | yte 4 | | | |
| bit 7 | | | | | | bit 0 | |
| Legend: | | | | | | | |
| R = Readable bit W = Writable bit | | oit | U = Unimplemented bit, read as '0' | | | | |
| -n = Value at POR '1' = | | '1' = Bit is set | | '0' = Bit is clea | ared | x = Bit is unkr | nown |

 bit 15-8
 Byte 5<15:8>: ECANx Message Byte 5

 bit 7-0
 Byte 4<7:0>: ECANx Message Byte 4

© 2009-2014 Microchip Technology Inc.

22.3 Module Functionality

The High-Speed, 10-Bit ADC is designed to support power conversion applications when used with the High-Speed PWM module. The ADC may have one or two SAR modules, depending on the device variant. If two SARs are present on a device, two conversions can be processed at a time, yielding 4 Msps conversion rate. If only one SAR is present on a device, only one conversion can be processed at a time, yielding 2 Msps conversion rate. The High-Speed, 10-Bit ADC produces two 10-bit conversion results in a 0.5 microsecond.

The ADC module supports up to 24 external analog inputs and two internal analog inputs. To monitor reference voltage, two internal inputs, AN24 and AN25, are connected to EXTREF and INTREF, respectively.

The analog reference voltage is defined as the device supply voltage (AVDD/AVSS).

The ADC module uses the following control and status registers:

- ADCON: ADC Control Register
- ADSTAT: ADC Status Register
- ADBASE: ADC Base Register^(1,2)
- ADPCFG: ADC Port Configuration Register
- ADPCFG2: ADC Port Configuration Register 2
- ADCPC0: ADC Convert Pair Control Register 0
- ADCPC1: ADC Convert Pair Control Register 1
- ADCPC2: ADC Convert Pair Control Register 2
- ADCPC3: ADC Convert Pair Control Register 3
- ADCPC4: ADC Convert Pair Control Register 4
- ADCPC5: ADC Convert Pair Control Register 5
- ADCPC6: ADC Convert Pair Control Register 6(2)

The ADCON register controls the operation of the ADC module. The ADSTAT register displays the status of the conversion processes. The ADPCFG registers configure the port pins as analog inputs or as digital I/O. The ADCPCx registers control the triggering of the ADC conversions. See Register 22-1 through Register 22-12 for detailed bit configurations.

Note: A unique feature of the ADC module is its ability to sample inputs in an asynchronous manner. Individual Sample-and-Hold circuits can be triggered independently of each other.

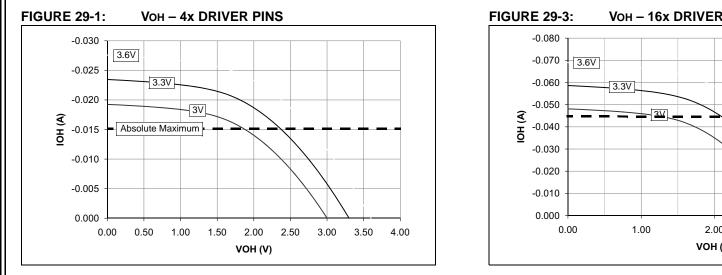
| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | | | |
|-------------------|--|---------------------------------------|----------------|------------------------------------|---------------------------------|--------------------|-----------|--|--|--|
| IRQEN9 | IRQEN9 PEND9 | | TRGSRC94 | TRGSRC93 | TRGSRC92 | TRGSRC91 | TRGSRC90 | | | |
| 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 | | | |
| IRQEN8 | PEND8 | SWTRG8 | TRGSRC84 | TRGSRC83 | TRGSRC82 | TRGSRC81 | TRGSRC80 | | | |
| 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 | IRQEN9: Inte | RQEN9: Interrupt Request Enable 9 bit | | | | | | | | |
| | 1 = Enable IF 0 = IRQ is no | | when requeste | d conversion o | of channels AN | 19 and AN18 is | completed | | | |
| bit 14 | PEND9: Pen | ding Conversio | n Status 9 bit | | | | | | | |
| | 1 = Conversion of channels AN19 and AN18 is pending; set when selected trigger is asserted 0 = Conversion is complete | | | | | | | | | |
| bit 13 | SWTRG9: So | SWTRG9: Software Trigger 9 bit | | | | | | | | |
| | | | | | the TRGSRCx< he PEND9 bit is | | | | | |
| | 0 = Conversion is not started | | | | | | | | | |
| | | | | | | | | | | |

REGISTER 22-10: ADCPC4: ADC CONVERT PAIR CONTROL REGISTER 4

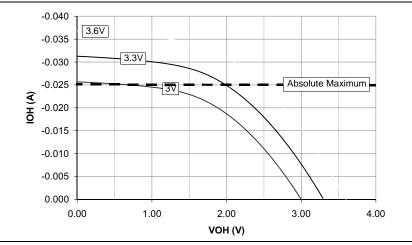
Note 1: The trigger source must be set as an individual software trigger prior to setting this bit to '1'. If other conversions are in progress, the conversion is performed when the conversion resources are available.

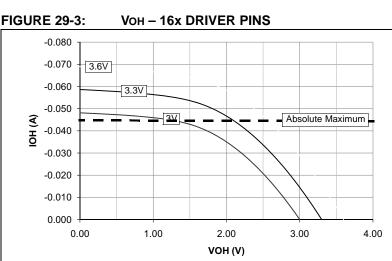
DC AND AC DEVICE CHARACTERISTICS GRAPHS 29.0

The graphs provided following this note are a statistical summary based on a limited number of samples and are provided for design guidance purposes Note: only. The performance characteristics listed herein are not tested or guaranteed. In some graphs, the data presented may be outside the specified operating range (e.g., outside specified power supply range) and therefore, outside the warranted range.



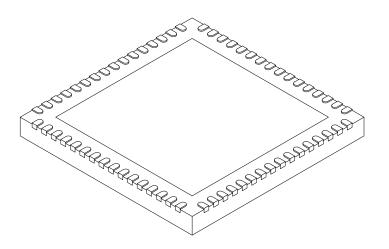
VOH – 8x DRIVER PINS FIGURE 29-2:





64-Lead Plastic Quad Flat, No Lead Package (MR) – 9x9x0.9 mm Body [QFN] With 7.15 x 7.15 Exposed Pad [QFN]

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



| | Units | | | S | |
|------------------------|------------------|------|----------|------|--|
| Dimensior | Dimension Limits | | NOM | MAX | |
| Number of Pins | | 64 | | | |
| Pitch | е | | 0.50 BSC | | |
| Overall Height | Α | 0.80 | 0.90 | 1.00 | |
| Standoff | A1 | 0.00 | 0.02 | 0.05 | |
| Contact Thickness | A3 | | 0.20 REF | | |
| Overall Width | E | | 9.00 BSC | | |
| Exposed Pad Width | E2 | 7.05 | 7.15 | 7.50 | |
| Overall Length | D | | 9.00 BSC | | |
| Exposed Pad Length | D2 | 7.05 | 7.15 | 7.50 | |
| Contact Width | b | 0.18 | 0.25 | 0.30 | |
| Contact Length | L | 0.30 | 0.40 | 0.50 | |
| Contact-to-Exposed Pad | K | 0.20 | - | - | |

Notes:

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

2. Package is saw singulated.

3. 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 C04-149C Sheet 2 of 2

| High-Speed 10-Bit ADC Module (dsPIC33FJ32GS610 |
|---|
| and dsPIC33FJ64GS610 Devices)83 |
| High-Speed 10-Bit ADC Module (for dsPIC33FJ32GS406 and dsPIC33FJ64GS406 |
| Devices) |
| High-Speed 10-Bit ADC Module (for |
| dsPIC33FJ32GS606 and dsPIC33FJ64GS606 |
| Devices) 86 |
| High-Speed PWM71 |
| High-Speed PWM Generator 171 |
| High-Speed PWM Generator 2 |
| High-Speed PWM Generator 3 |
| High-Speed PWM Generator 4 |
| High-Speed PWM Generator 5 |
| High-Speed PWM Generator 6 |
| High-Speed PWM Generator 7 (All Devices |
| except dsPIC33FJ32GS406 and |
| dsPIC33FJ64GS406) |
| High-Speed PWM Generator 8 (All Devices |
| except dsPIC33FJ32GS406 and |
| dsPIC33FJ64GS406) |
| High-Speed PWM Generator 9 (dsPIC33FJ32GS610 |
| and dsPIC33FJ64GS610 Devices)79 |
| I2C1 |
| 12C2 |
| Input Capture |
| Interrupt Controller (dsPIC33FJ32GS406 and |
| dsPIC33FJ64GS406 Devices)61 |
| Interrupt Controller (dsPIC33FJ32GS606 |
| Devices) |
| Interrupt Controller (dsPIC33FJ32GS608 |
| Devices) |
| Interrupt Controller (dsPIC33FJ32GS610 |
| Devices) |
| Interrupt Controller (dsPIC33FJ64GS606 |
| Devices) |
| Interrupt Controller (dsPIC33EJb4GSb08 |
| |
| Devices) |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ34GS406 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ34GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS610 Devices) 97 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ34GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS610 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS610 Devices) 97 PMD (dsPIC33FJ32GS610 Devices) 97 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS610 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 96 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS610 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 96 PORTA (dsPIC33FJ32GS608 and 96 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 96 PORTA (dsPIC33FJ64GS608 Devices) 97 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS610 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 96 PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 96 PORTA (dsPIC33FJ32GS608 and dsPIC33FJ32GS608 and 92 PORTA (dsPIC33FJ32GS610 and 92 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 96 PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 92 PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS610 Devices) 92 PORTA (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 92 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 96 PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 92 PORTA (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 92 PORTA (dsPIC33FJ64GS610 Devices) 92 PORTB 92 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 96 PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 92 PORTA (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 92 PORTB 92 PORTC (dsPIC33FJ32GS406/606 and 92 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 96 PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 92 PORTA (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 92 PORTB 92 PORTC (dsPIC33FJ32GS406/606 and dsPIC33FJ64GS406/606 Devices) 93 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 92 PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS610 Devices) 92 PORTB 92 PORTC (dsPIC33FJ32GS406/606 and dsPIC33FJ64GS406/606 Devices) 93 PORTC (dsPIC33FJ32GS608 and 93 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS406 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 96 PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS610 Devices) 92 PORTA (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 92 PORTB 92 PORTC (dsPIC33FJ32GS406/606 and dsPIC33FJ64GS406/606 Devices) 93 PORTC (dsPIC33FJ32GS608 and dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 93 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 92 PORTA (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 92 PORTB 92 PORTC (dsPIC33FJ32GS406/606 and dsPIC33FJ64GS406/606 Devices) 93 PORTC (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 93 PORTC (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 93 PORTC (dsPIC33FJ32GS610 and 93 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 97 PMD (dsPIC33FJ32GS610 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 92 PORTA (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 92 PORTB 92 PORTC (dsPIC33FJ32GS406/606 and dsPIC33FJ64GS608 Devices) 93 PORTC (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 93 PORTC (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 93 PORTC (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 93 |
| Devices)57Interrupt Controller (dsPIC33FJ64GS61055NVM96Output Compare70PMD (dsIPC33FJ64GS606 Devices)98PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS606 Devices)98PMD (dsPIC33FJ32GS606 Devices)98PMD (dsPIC33FJ32GS606 Devices)98PMD (dsPIC33FJ32GS606 Devices)97PMD (dsPIC33FJ32GS608 Devices)97PMD (dsPIC33FJ32GS608 Devices)97PMD (dsPIC33FJ64GS608 Devices)97PMD (dsPIC33FJ64GS608 Devices)97PMD (dsPIC33FJ64GS608 Devices)96PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS610 Devices)92PORTA (dsPIC33FJ32GS608 and dsPIC33FJ64GS610 Devices)92PORTE92PORTC (dsPIC33FJ32GS406/606 and dsPIC33FJ64GS608 Devices)93PORTC (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices)93PORTC (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices)93PORTC (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices)93PORTC (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices)93PORTC (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices)93PORTC (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices)92PORTD (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices)92PORTD (dsPIC33FJ32GS406/606 and92 |
| Devices) 57 Interrupt Controller (dsPIC33FJ64GS610 55 Devices) 55 NVM 96 Output Compare 70 PMD (dsIPC33FJ64GS606 Devices) 98 PMD (dsPIC33FJ32GS406 and dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 98 PMD (dsPIC33FJ32GS606 Devices) 97 PMD (dsPIC33FJ32GS608 Devices) 97 PMD (dsPIC33FJ64GS608 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 97 PMD (dsPIC33FJ32GS610 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 97 PMD (dsPIC33FJ64GS610 Devices) 92 PORTA (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 92 PORTB 92 PORTC (dsPIC33FJ32GS406/606 and dsPIC33FJ64GS608 Devices) 93 PORTC (dsPIC33FJ32GS608 and dsPIC33FJ64GS608 Devices) 93 PORTC (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 93 PORTC (dsPIC33FJ32GS610 and dsPIC33FJ64GS610 Devices) 93 |

| PORTE (dsPIC33FJ32GS406/606 and | |
|---|---|
| dsPIC33FJ64GS406/606 Devices) | 4 |
| PORTE (dsPIC33FJ32GS608/610 and | |
| dsPIC33FJ64GS608/610 Devices) | 1 |
| PORTF (dsPIC33FJ32GS406/606 and | |
| dsPIC33FJ64GS406/606 Devices) | - |
| , | 2 |
| PORTF (dsPIC33FJ32GS608 and | |
| dsPIC33FJ64GS608 Devices)94 | 1 |
| PORTF (dsPIC33FJ32GS610 and | |
| dsPIC33FJ64GS610 Devices) | 4 |
| PORTG (dsPIC33FJ32GS406/606 and | |
| dsPIC33FJ64GS406/606 Devices) | 5 |
| | , |
| PORTG (dsPIC33FJ32GS608 and | _ |
| dsPIC33FJ64GS608 Devices)95 | 5 |
| PORTG (dsPIC33FJ32GS610 and | |
| dsPIC33FJ64GS610 Devices) | 5 |
| Quadrature Encoder Interface 1 (QEI1) |) |
| Quadrature Encoder Interface 2 (QEI2) | |
| SPI1 | |
| | |
| SPI2 | |
| System Control | |
| Timers | 9 |
| UART1 | 1 |
| UART2 | 1 |
| Registers | |
| ACLKCON (Auxiliary Clock Divisor Control) | a |
| | |
| ADBASE (ADC Base) | |
| ADC Base Register (ADBASE) | |
| ADCON (ADC Control) 319 | |
| ADCPC0 (ADC Convert Pair Control 0) 324 | 1 |
| ADCPC1 (ADC Convert Pair Control 1) | 7 |
| ADCPC2 (ADC Convert Pair Control 2) | |
| ADCPC3 (ADC Convert Pair Control 3) | |
| ADCPC4 (ADC Convert Pair Control 4) | |
| | |
| ADCPC5 (ADC Convert Pair Control 5) | |
| ADCPC6 (ADC Convert Pair Control 6) 342 | |
| ADPCFG (ADC Port Configuration) | 3 |
| ADPCFG2 (ADC Port Configuration 2) | 3 |
| ADSTAT (ADC Status) | |
| ALTDTRx (PWM Alternate Dead-Time x) | |
| AUXCONx (PWM Auxiliary Control x) | 2 |
| | |
| CHOP (PWM Chop Clock Generator) 241 | |
| CLKDIV (Clock Divisor) 196 | |
| CMPCONx (Comparator Control x) | 7 |
| CMPDACx (Comparator DAC Control x) | 3 |
| CORCON (Core Control) | |
| CxBUFPNT1 (ECANx Filter 0-3 | |
| | 5 |
| Buffer Pointer 1) | 2 |
| CxBUFPNT2 (ECANx Filter 4-7 | |
| Buffer Pointer 2) 299 | Э |
| CxBUFPNT3 (ECANx Filter 8-11 | |
| Buffer Pointer 3) 300 |) |
| CxBUFPNT4 (ECANx Filter 12-15 | |
| Buffer Pointer 4) | 1 |
| CxCFG1 (ECANx Baud Rate Configuration 1) | |
| | |
| CxCFG2 (ECANx Baud Rate Configuration 2) | |
| CxCTRL1 (ECANx Control 1) 288 | |
| CxCTRL2 (ECANx Control 2) 289 | |
| CxEC (ECANx Transmit/Receive Error Count) 296 | 3 |
| CxFCTRL (ECANx FIFO Control) | |
| CxFEN1 (ECANx Acceptance Filter Enable 1) | |
| CxFIFO (ECANx FIFO Status) | |
| | - |
| CxFMSKSEL1 (ECANx Filter 7-0 Mask | _ |
| Selection 1) 303 | 3 |