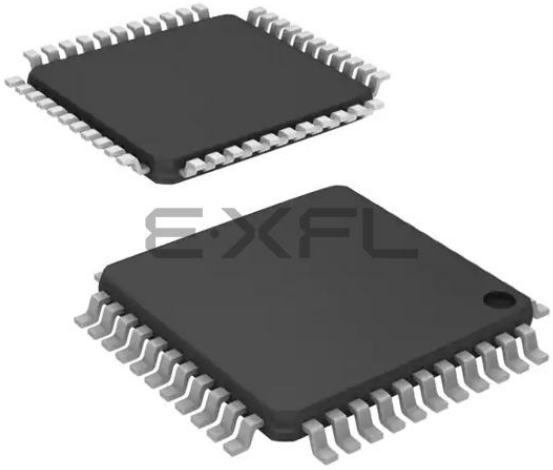


**Welcome to E-XFL.COM**

## 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 "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Active |
| Core Processor | PIC |
| Core Size | 8-Bit |
| Speed | 64MHz |
| Connectivity | I²C, LINbus, SPI, UART/USART |
| Peripherals | Brown-out Detect/Reset, DMA, HLVD, POR, PWM, WDT |
| Number of I/O | 36 |
| Program Memory Size | 128KB (64K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | 1K x 8 |
| RAM Size | 8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.3V ~ 5.5V |
| Data Converters | A/D 35x12b; D/A 1x5b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 44-TQFP |
| Supplier Device Package | 44-TQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/pic18f47k42-e-pt |

### 4.4.2 INSTRUCTIONS IN PROGRAM MEMORY

The program memory is addressed in bytes. Instructions are stored as either two bytes or four bytes in program memory. The Least Significant Byte of an instruction word is always stored in a program memory location with an even address (LSb = 0). To maintain alignment with instruction boundaries, the PC increments in steps of two and the LSb will always read '0' (see **Section 4.2.4 "Program Counter"**).

Figure 4-2 shows an example of how instruction words are stored in the program memory.

The CALL and GOTO instructions have the absolute program memory address embedded into the instruction. Since instructions are always stored on word boundaries, the data contained in the instruction is a word address. The word address is written to PC<20:1>, which accesses the desired byte address in program memory. Instruction #2 in Figure 4-2 shows how the instruction GOTO 0006h is encoded in the program memory. Program branch instructions, which encode a relative address offset, operate in the same manner. The offset value stored in a branch instruction represents the number of single-word instructions that the PC will be offset by. **Section 41.0 "Instruction Set Summary"** provides further details of the instruction set.

### 4.4.3 MULTI-WORD INSTRUCTIONS

The standard PIC18 instruction set has four two-word instructions: CALL, MOVFF, GOTO and LFSR and two three-word instructions: MOVFFL and MOVSFL. In all cases, the second and the third word of the instruction always has '1111' as its four Most Significant bits; the other 12 bits are literal data, usually a data memory address.

The use of '1111' in the four MSbs of an instruction specifies a special form of NOP. If the instruction is executed in proper sequence – immediately after the first word – the data in the second word is accessed and used by the instruction sequence. If the first word is skipped for some reason and the second or third word is executed by itself, a NOP is executed instead. This is necessary for cases when the multi-word instruction is preceded by a conditional instruction that changes the PC. Example 4-4 shows how this works.

**FIGURE 4-2: INSTRUCTIONS IN PROGRAM MEMORY**

Program Memory Byte Locations →

Instruction 1: MOVLW 055h
Instruction 2: GOTO 0006h
Instruction 3: MOVFF 123h, 456h
Instruction 4: MOVFFL 123h, 456h

| LSB = 1 | LSB = 0 | Word Address ↓ |
|---|---|---|
| | | 000000h |
| | | 000002h |
| | | 000004h |
| | | 000006h |
| 0Fh | 55h | 000008h |
| EFh | 03h | 00000Ah |
| F0h | 00h | 00000Ch |
| C1h | 23h | 00000Eh |
| F4h | 56h | 000010h |
| 00h | 60h | 000012h |
| F4h | 8Ch | 000014h |
| F4h | 56h | 000016h |
| | | 000018h |
| | | 00001Ah |

## 5.0 DEVICE CONFIGURATION

Device configuration consists of the Configuration Words, User ID, Device ID, Rev ID, Device Information Area (DIA), (see **Section 5.7 "Device Information Area"**), and the Device Configuration Information (DCI) regions, (see **Section 5.8 "Device Configuration Information"**).

### 5.1 Configuration Words

There are six Configuration Word bits that allow the user to setup the device with several choices of oscillators, Resets and memory protection options. These are implemented as Configuration Word 1 through Configuration Word 6 at 300000h through 30000Bh.

**REGISTER 5-5: CONFIGURATION WORD 3L (30 0004h)**

| U-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 | R/W-1 |
|---|---|---|---|---|---|---|---|
| — | WDTE<1:0> | | WDTCPS<4:0> | | | | |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '1' | |
| -n = Value for blank device | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

bit 7 **Unimplemented:** Read as '1'

bit 6-5 **WDTE<1:0>:** WDT Operating Mode bits
00 = WDT is disabled, SWDTEN is ignored
01 = WDT is enabled/disabled by the SWDTEN bit in WDTCON0
10 = WDT is enabled while Sleep = 0, suspended when Sleep = 1; SWDTEN is ignored
11 = WDT is enabled regardless of Sleep; SWDTEN is ignored

bit 4-0 **WDTCPS<4:0>:** WDT Period Select bits

| WDTCPS<4:0> | WDTPS at POR | | | | Software Control of WDTPS? |
|---|---|---|---|---|---|
| | Value | Divider Ratio | | Typical Time-out (FIN = 31 kHz) | |
| 00000 | 00000 | 1:32 | $2^5$ | 1 ms | No |
| 00001 | 00001 | 1:64 | $2^6$ | 2 ms | |
| 00010 | 00010 | 1:128 | $2^7$ | 4 ms | |
| 00011 | 00011 | 1:256 | $2^8$ | 8 ms | |
| 00100 | 00100 | 1:512 | $2^9$ | 16 ms | |
| 00101 | 00101 | 1:1024 | $2^{10}$ | 32 ms | |
| 00110 | 00110 | 1:2048 | $2^{11}$ | 64 ms | |
| 00111 | 00111 | 1:4096 | $2^{12}$ | 128 ms | |
| 01000 | 01000 | 1:8192 | $2^{13}$ | 256 ms | |
| 01001 | 01001 | 1:16384 | $2^{14}$ | 512 ms | |
| 01010 | 01010 | 1:32768 | $2^{15}$ | 1s | |
| 01011 | 01011 | 1:65536 | $2^{16}$ | 2s | |
| 01100 | 01100 | 1:131072 | $2^{17}$ | 4s | |
| 01101 | 01101 | 1:262144 | $2^{18}$ | 8s | |
| 01110 | 01110 | 1:524299 | $2^{19}$ | 16s | |
| 01111 | 01111 | 1:1048576 | $2^{20}$ | 32s | |
| 10000 | 10000 | 1:2097152 | $2^{21}$ | 64s | |
| 10001 | 10001 | 1:4194304 | $2^{22}$ | 128s | |
| 10010 | 10010 | 1:8388608 | $2^{23}$ | 256s | |
| 10011 ... 11110 | 10011 ... 11110 | 1:32 | $2^5$ | 1 ms | No |
| 11111 | 01011 | 1:65536 | $2^{16}$ | 2s | Yes |

## 9.6 Returning from Interrupt Service Routine (ISR)

The “Return from Interrupt” instruction (`RETFIE`) is used to mark the end of an ISR.

When `RETFIE 1` instruction is executed, the PC is loaded with the saved PC value from the top of the PC stack. Saved context is also restored with the execution of this instruction. Thus, execution returns to the previous state of operation that existed before the interrupt occurred.

When `RETFIE 0` instruction is executed, the saved context is not restored back to the registers.

## 9.7 Interrupt Latency

By assigning each interrupt with a vector address/ number (MVECEN = `1`), scanning of all interrupts is not necessary to determine the source of the interrupt.

When MVECEN = `1`, Vectored interrupt controller requires three clock cycles to vector to the ISR from main routine, thereby removing dependency of interrupt timing on compiled code.

There is a fixed latency of three instruction cycles between the completion of the instruction active when the interrupt occurred and the first instruction of the Interrupt Service Routine. Figure 9-7, Figure 9-8 and Figure 9-9 illustrate the sequence of events when a peripheral interrupt is asserted when the last executed instruction is one-cycle, two-cycle and three-cycle respectively, when MVECEN = `1`.

After the Interrupt Flag Status bit is set, the current instruction completes executing. In the first latency cycle, the contents of the PC, STATUS, WREG, BSR, FSR0/1/2, PRODL/H and PCLATH/U registers are context saved and the IVTBASE+ Vector number is calculated. In the second latency cycle, the PC is loaded with the calculated vector table address for the interrupt source and the starting address of the ISR is fetched. In the third latency cycle, the PC is loaded with the ISR address. All the latency cycles are executed as a `FNOP` instruction.

When MVECEN = `0`, Vectored interrupt controller requires two clock cycles to vector to the ISR from main routine. There is a latency of two instruction cycles plus the software latency between the completion of the instruction active when the interrupt occurred and the first instruction of the Interrupt Service Routine.

**REGISTER 9-26: IPR1: PERIPHERAL INTERRUPT PRIORITY REGISTER 1**

| R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 | R/W-1/1 |
|---|---|---|---|---|---|---|---|
| SMT1PWAIP | SMT1PRAIP | SMT1IP | C1IP | ADTIP | ADIP | ZCDIP | INT0IP |
| bit 7 | | | | | | | bit 0 |

| **Legend:** | | |
|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' |
| u = Bit is unchanged | x = Bit is unknown | -n/n = Value at POR and BOR/Value at all other Resets |
| '1' = Bit is set | '0' = Bit is cleared | |

bit 7 **SMT1PWAIP:** SMT1 Pulse Width Acquisition Interrupt Priority bit
1 = High priority
0 = Low priority

bit 6 **SMT1PRAIP:** SMT1 Period Acquisition Interrupt Priority bit
1 = High priority
0 = Low priority

bit 5 **SMT1IP:** SMT1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 4 **C1IP:** C1 Interrupt Priority bit
1 = High priority
0 = Low priority

bit 3 **ADTIP:** ADC Threshold Interrupt Priority bit
1 = High priority
0 = Low priority

bit 2 **ADIP:** ADC Interrupt Priority bit
1 = High priority
0 = Low priority

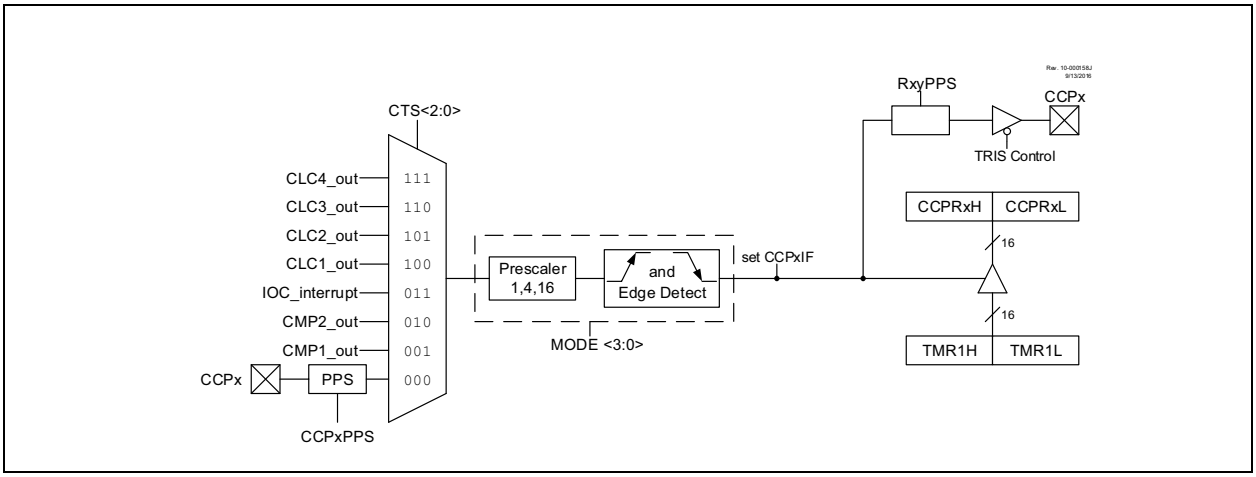
bit 1 **ZCDIP:** ZCD Interrupt Priority bit
1 = High priority
0 = Low priority

bit 0 **INT0IP:** External Interrupt 0 Interrupt Priority bit
1 = High priority
0 = Low priority

**TABLE 17-1: PPS INPUT REGISTER DETAILS**

| Peripheral | PPS Input Register | Default Pin Selection at POR | Register Reset Value at POR | Input Available from Selected PORTx | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | PIC18(L)F26/27K42 | | | PIC18(L)F45/46/47K42 | | | | | PIC18(L)F55/56/57K42 | | | | | | |
| ADC Conversion Trigger | ADACTPPS | RB4 | 0b0 1100 | — | B | C | — | B | — | D | — | — | B | — | D | — | — |
| SPI1 Clock | SPI1SCKPPS | RC3 | 0b1 0011 | — | B | C | — | B | C | — | — | — | B | C | — | — | — |
| SPI1 Data | SPI1SDIPPS | RC4 | 0b1 0100 | — | B | C | — | B | C | — | — | — | B | C | — | — | — |
| SPI1 Slave Select | SPI1SSPPS | RA5 | 0b0 0101 | A | — | C | A | — | — | D | — | A | — | — | D | — | — |
| $I^2C1$ Clock | I2C1SCLPPS | RC3 | 0b1 0011 | — | B | C | — | B | C | — | — | — | B | C | — | — | — |
| $I^2C1$ Data | I2C1SDAPPS | RC4 | 0b1 0100 | — | B | C | — | B | C | — | — | — | B | C | — | — | — |
| $I^2C2$ Clock | I2C2SCLPPS | RB1 | 0b0 1001 | — | B | C | — | B | — | D | — | — | B | — | D | — | — |
| $I^2C2$ Data | I2C2SDAPPS | RB2 | 0b0 1010 | — | B | C | — | B | — | D | — | — | B | — | D | — | — |
| UART1 Receive | U1RXPPS | RC7 | 0b1 0111 | — | B | C | — | B | C | — | — | — | — | C | — | — | F |
| UART1 Clear To Send | U1CTSPPS | RC6 | 0b1 0110 | — | B | C | — | B | C | — | — | — | — | C | — | — | F |
| UART2 Receive | U2RXPPS | RB7 | 0b0 1111 | — | B | C | — | B | — | D | — | — | B | — | D | — | — |
| UART2 Clear To Send | U2CTSPPS | RB6 | 0b0 1110 | — | B | C | — | B | — | D | — | — | B | — | D | — | — |

**FIGURE 23-1: CAPTURE MODE OPERATION BLOCK DIAGRAM**

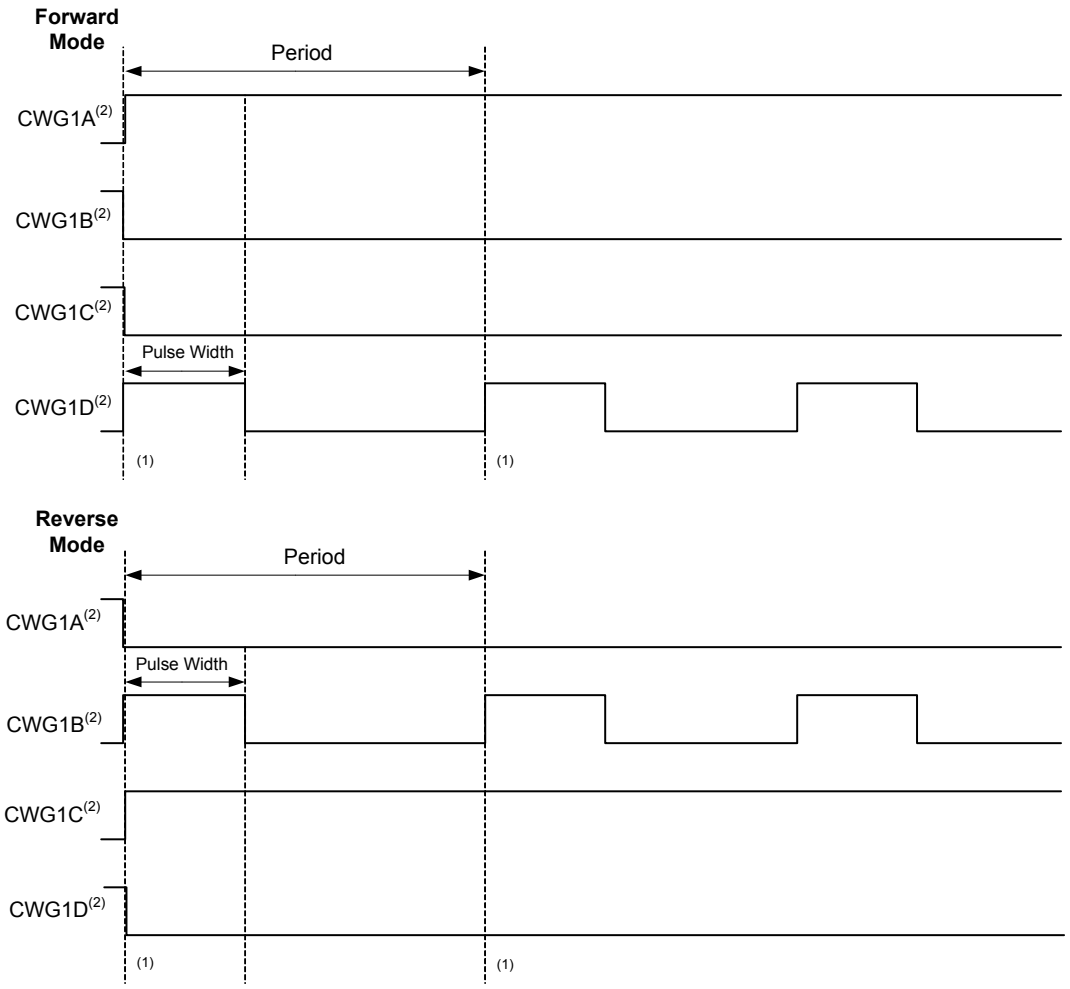


In Forward Full-Bridge mode (MODE<2:0> = 010), CWGxA is driven to its active state, CWGxB and CWGxC are driven to their inactive state, and CWGxD is modulated by the input signal, as shown in Figure 26-7.

In Reverse Full-Bridge mode (MODE<2:0> = 011), CWGxC is driven to its active state, CWGxA and CWGxD are driven to their inactive states, and CWGxB is modulated by the input signal, as shown in Figure 26-7.

In Full-Bridge mode, the dead-band period is used when there is a switch from forward to reverse or vice-versa. This dead-band control is described in **Section 26.6 "Dead-Band Control"**, with additional details in **Section 26.7 "Rising Edge and Reverse Dead Band"** and **Section 26.8 "Falling Edge and Forward Dead Band"**. Steering modes are not used with either of the Full-Bridge modes. The mode selection may be toggled between forward and reverse toggling the MODE<0> bit of the CWGxCON0 while keeping MODE<2:1> static, without disabling the CWG module.

**FIGURE 26-7: EXAMPLE OF FULL-BRIDGE OUTPUT**




**Note 1:** A rising CWG data input creates a rising event on the modulated output.
**2:** Output signals shown as active-high; all POLy bits are clear.

## 26.12 Operation During Sleep

The CWG module operates independently from the system clock and will continue to run during Sleep, provided that the clock and input sources selected remain active.

The HFINTOSC remains active during Sleep when all the following conditions are met:

- CWG module is enabled
- Input source is active
- HFINTOSC is selected as the clock source, regardless of the system clock source selected.

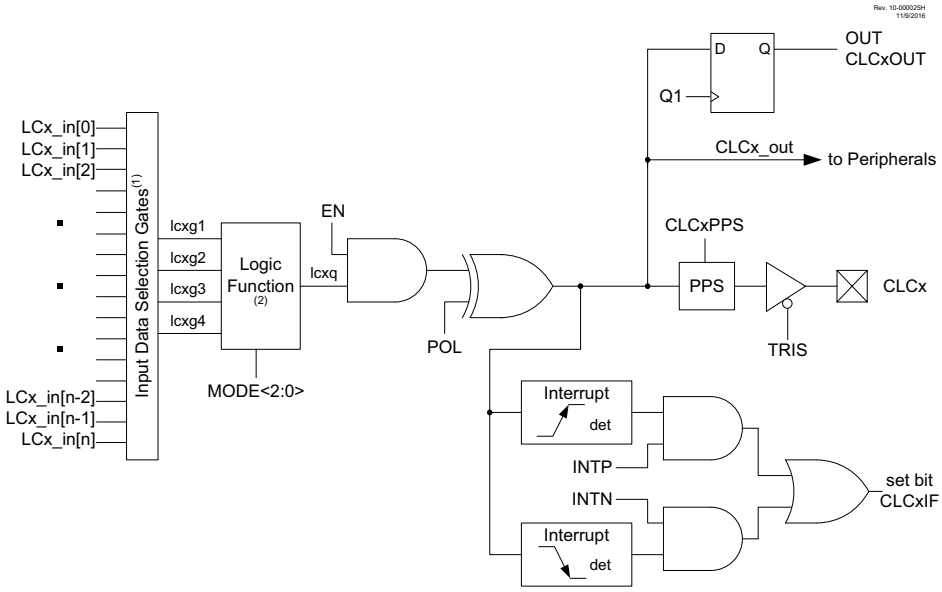
In other words, if the HFINTOSC is simultaneously selected as system clock and CWG clock, when the CWG is enabled and the input source is active, then the CPU will go idle during Sleep, but the HFINTOSC will remain active and the CWG will continue to operate. This will have a direct effect on the Sleep mode current.

## 26.13 Configuring the CWG

1. Ensure that the TRIS control bits corresponding to CWG outputs are set so that all are configured as inputs, ensuring that the outputs are inactive during setup. External hardware should ensure that pin levels are held to safe levels.
2. Clear the EN bit, if not already cleared.
3. Configure the MODE<2:0> bits of the CWGxCON0 register to set the output operating mode.
4. Configure the POLy bits of the CWGxCON1 register to set the output polarities.
5. Configure the ISM<4:0> bits of the CWGxISM register to select the data input source.
6. If a steering mode is selected, configure the STRx bits to select the desired output on the CWG outputs.
7. Configure the LSBD<1:0> and LSAC<1:0> bits of the CWGxASD0 register to select the auto-shutdown output override states (this is necessary even if not using auto-shutdown because start-up will be from a shutdown state).
8. If auto-restart is desired, set the REN bit of CWGxAS0.
9. If auto-shutdown is desired, configure the ASxE bits of the CWGxAS1 register to select the shutdown source.
10. Set the desired rising and falling dead-band times with the CWGxDBR and CWGxDBF registers.
11. Select the clock source in the CWGxCLKCON register.
12. Set the EN bit to enable the module.
13. Clear the TRIS bits that correspond to the CWG outputs to set them as outputs.

If auto-restart is to be used, set the REN bit and the SHUTDOWN bit will be cleared automatically. Otherwise, clear the SHUTDOWN bit in software to start the CWG.

**FIGURE 27-1: CLCx SIMPLIFIED BLOCK DIAGRAM**




**Note 1:** See Figure 27-2: Input Data Selection and Gating

**2:** See Figure 27-3: Programmable Logic Functions.

## 27.1 CLCx Setup

Programming the CLCx module is performed by configuring the four stages in the logic signal flow. The four stages are:

- Data selection
- Data gating
- Logic function selection
- Output polarity

Each stage is setup at run time by writing to the corresponding CLCx Special Function Registers. This has the added advantage of permitting logic reconfiguration on-the-fly during program execution.

### 27.1.1 DATA SELECTION

There are 32 signals available as inputs to the configurable logic. Four 32-input multiplexers are used to select the inputs to pass on to the next stage.

Data selection is through four multiplexers as indicated on the left side of Figure 27-2. Data inputs in the figure are identified by a generic numbered input name.

Table 27-1 correlates the generic input name to the actual signal for each CLC module. The column labeled 'DyS<4:0> Value' indicates the MUX selection code for the selected data input. DyS is an abbreviation for the MUX select input codes: D1S<4:0> through D4S<4:0>.

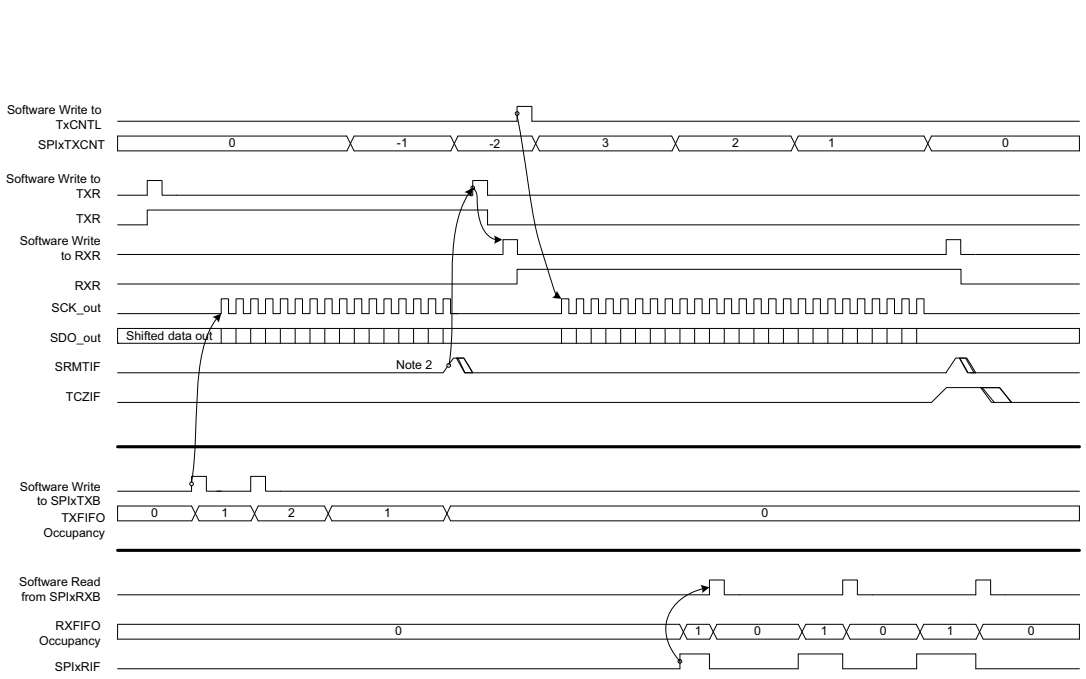
Data inputs are selected with CLCxSEL0 through CLCxSEL3 registers (Register 27-3 through Register 27-6).

**Note:** Data selections are undefined at power-up.

### 32.5.3 RECEIVE ONLY MODE

When RXR is set and TXR is clear, the SPI master is in Receive Only mode. In this mode, data transfers when the RXFIFO is not full and the Transfer Counter is non-zero. In this mode, writing a value to SPIxTCNTL will start the clocks for transfer. The clocks will suspend while the RXFIFO is full and cease when the SPIxTCNT reaches zero (see **Section 32.4 "Transfer Counter"**). If there is any data in the TXFIFO, the first data written to the TXFIFO will be transmitted on each data exchange, although the TXFIFO occupancy will not change, meaning that the same message will be sent on each transmission. If there is no data in the TXFIFO, the most recently received data will instead be transmitted. Figure 32-5 shows an example of sending a command using **Section 32.5.2 "Transmit Only Mode"** and then receiving a byte of data using this mode.

**FIGURE 32-5: SPI MASTER OPERATION, COMMAND+READ DATA, TXR/RXR=0/1**

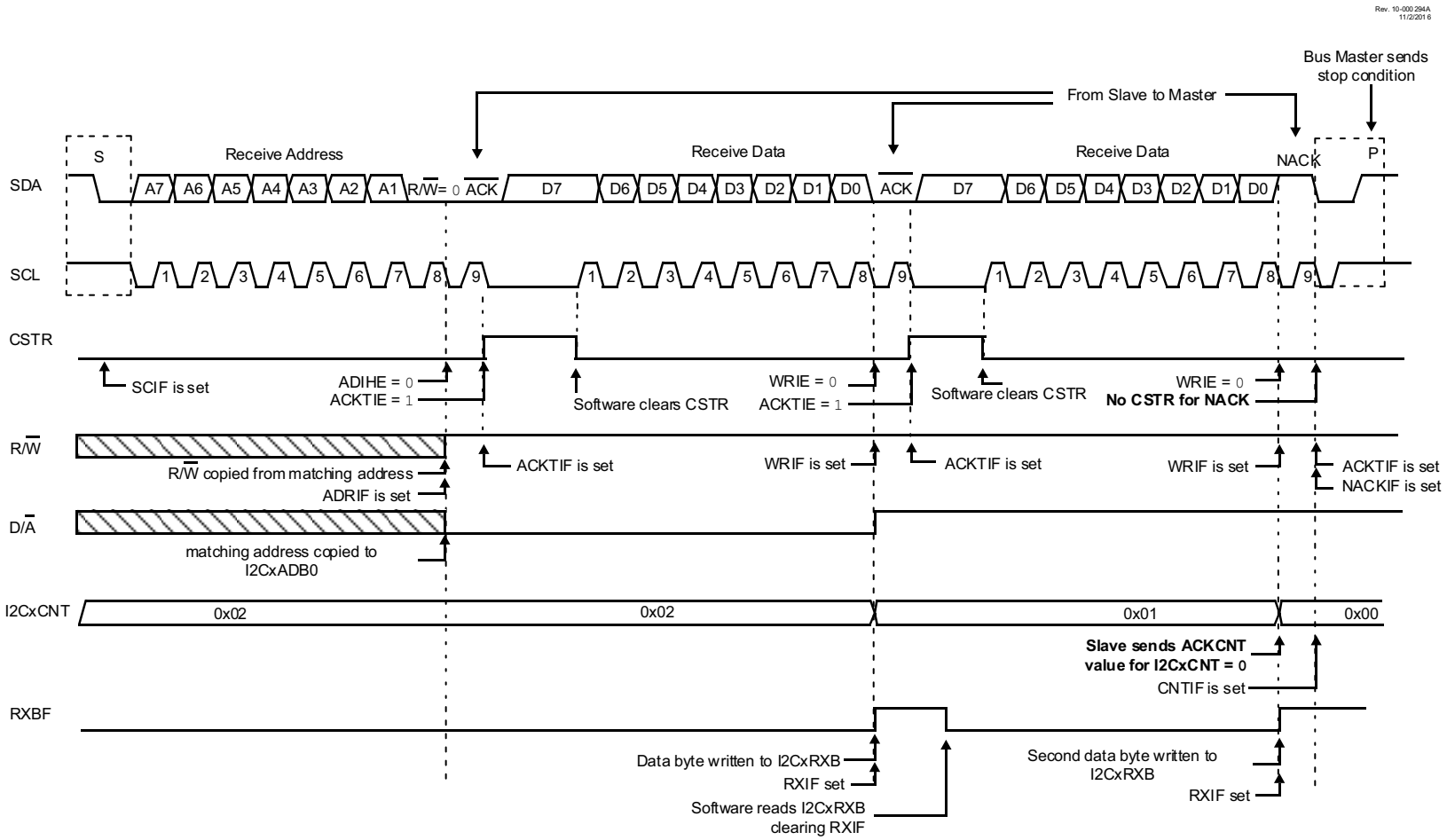



Note: 1. SS(out) is not shown
2. Software must wait for shift-register empty (SRMTIF) before changing TXR, RXR, BYTES and BITS controls. This is not considered an imposition in this case, because the slave probably needs time to load output data (see also Figure 4-14).

### 32.5.4 TRANSFER OFF MODE

When both TXR and RXR are cleared, the SPI master is in Transfer Off mode. In this mode, SCK will not toggle and no data is exchanged. However, writes to SPIxTXB will be transferred to the TXFIFO which will be transmitted if the TXR bit is set.

**FIGURE 33-7: I²C SLAVE, 7-BIT ADDRESS, RECEPTION WITH I2CxCNT (ACKTIE = 1, ADRIE = 0, WRIE = 0)**



#### 33.4.3.2 Slave Transmission (7-bit Addressing Mode)

This section describes the sequence of events for the $I^2C$ module configured as an $I^2C$ slave in 7-bit Addressing mode and is transmitting data. Figure 33-9 and Figure 33-10 are used as a visual reference for this description.

1. Master asserts Start condition (can also be a restart) on the bus. Start condition Interrupt Flag (SCIF) in I2CxPIR register is set.
2. If Start condition interrupt is enabled (SCIE bit is set), generic interrupt I2CxIF is set.
3. Master transmits eight bits – 7-bit address and R/W = 1.
4. Received address is compared with the values in I2CxADR0/I2CxADR1/I2CxADR2/I2CxADR3 registers. Refer to **Section 33.4.1 "Slave Addressing Modes"** for Slave Addressing modes
5. If address matches; SMA in I2CxSTAT0 register is set, R/W is copied to R/W bit, D/A bit is cleared. If the address does not match; module becomes idle.
6. The matched address data is loaded into I2CxADB0 and ADRIF in I2CxPIR register is set.
7. If Address hold interrupt is enabled (ADRIE = 1), CSTR is set. I2CxIF is set. Slave software can read address from I2CxADB0 and set/clear ACKDT before releasing SCL. SCL line can be released by clearing CSTR.
8. If the transmit buffer is empty from the previous transaction, i.e. TXBE = 1 and I2CxCNT!= 0 (I2CxTXIF = 1), CSTR is set. Slave software must load data into I2CxTXB to release SCL. I2CxCNT decrements after the byte is loaded into the shift register.
9. Slave hardware waits for 9th SCL pulse with ACK data from Master.
10. If I2CxCNT = 0, CNTIF is set.
11. If the Acknowledge interrupt and hold is enabled (ACKTIE = 1), CSTR is set, I2CxIF is set.
12. Slave software can change the value of ACKDT before releasing SCL by clearing CSTR.
13. Master sends eight SCL pulses to clock out data or asserts a Stop condition to end the transaction.
14. Go to step 8.

**FIGURE 33-9: I²C SLAVE, 7-BIT ADDRESS, TRANSMISSION**

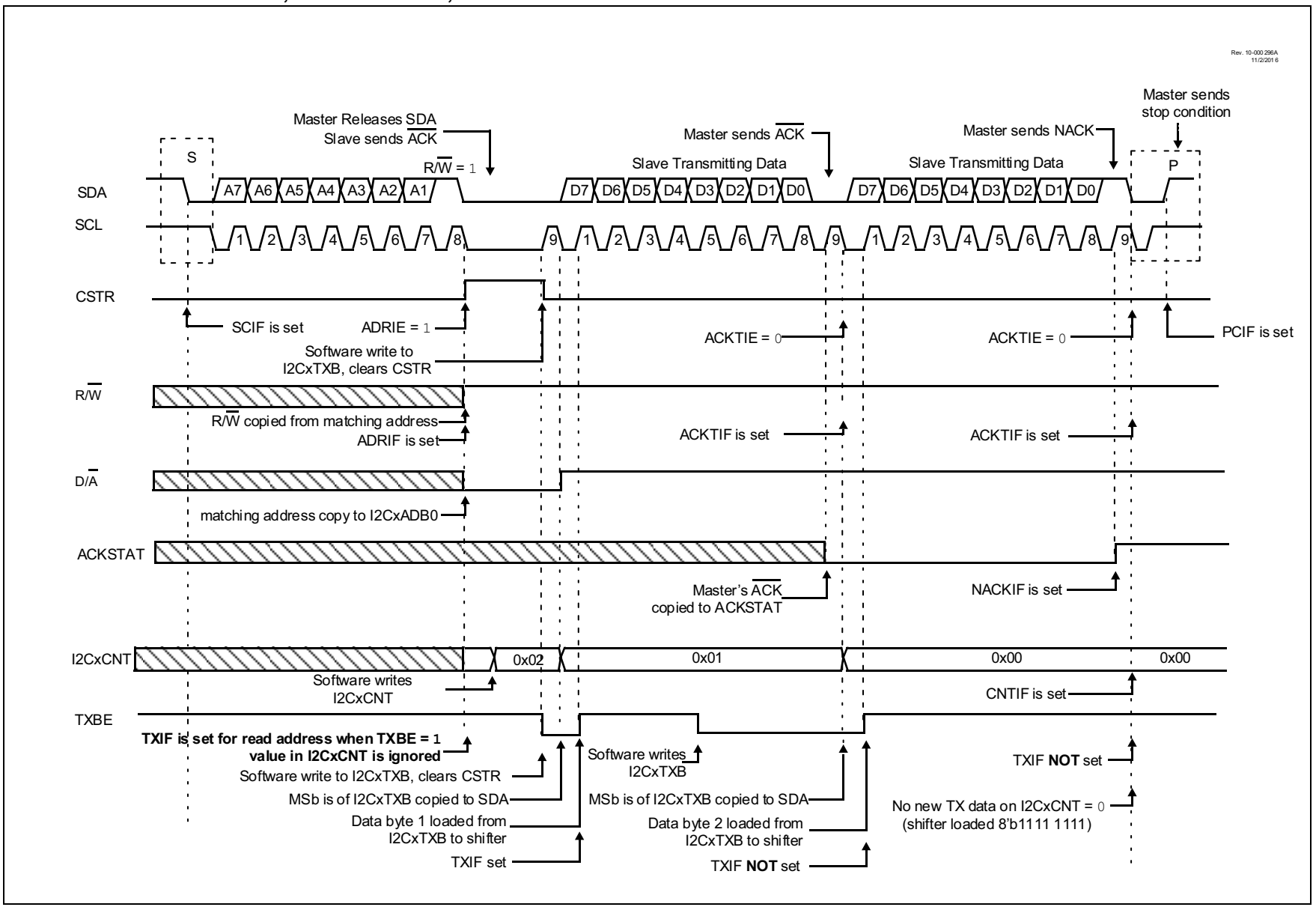
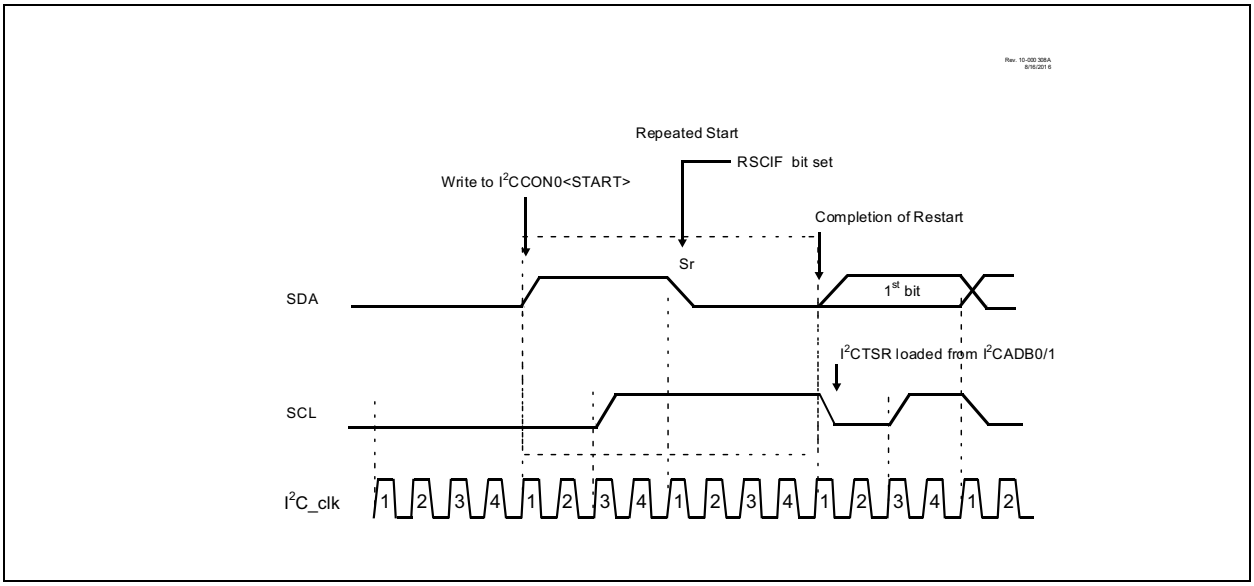


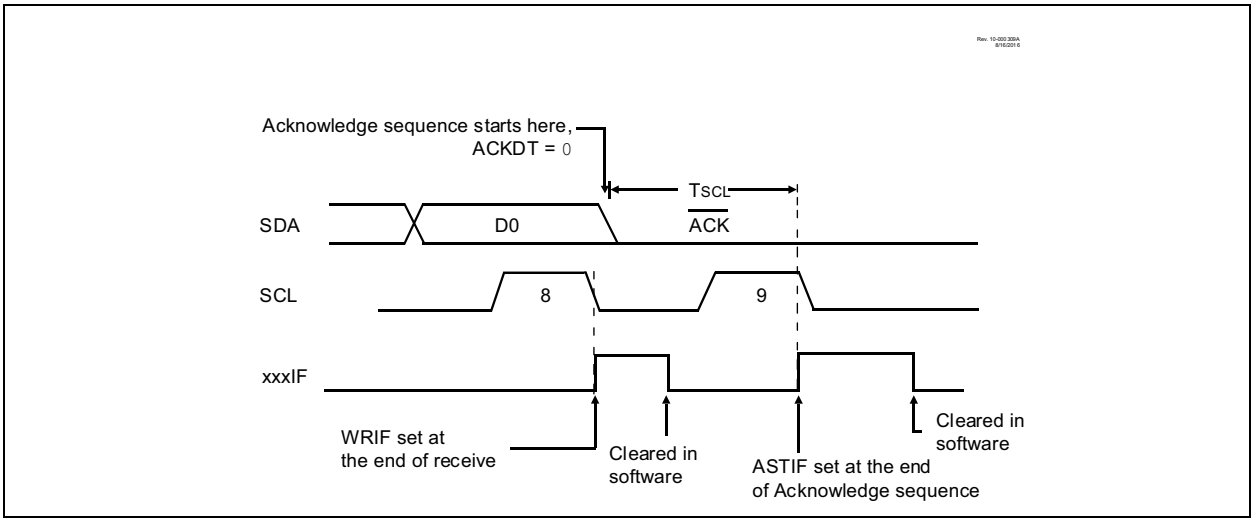
**FIGURE 33-16: REPEATED START CONDITION TIMING**




### 33.5.7 ACKNOWLEDGE SEQUENCE TIMING

An Acknowledge sequence is enabled automatically following an address/data byte transmission. The SCL pin is pulled low and the contents of the Acknowledge Data bits (ACKDT/ACKCNT) are presented on the SDA pin. If the user wishes to generate an Acknowledge, then the ACKDT bit should be cleared. If not, the user should set the ACKDT bit before starting an Acknowledge sequence. The master then waits one clock period (TSCL) and the SCL pin is released high. When the SCL pin is sampled high (clock arbitration), the master counts another TSCL. The SCL pin is then pulled low. Figure 33-17 shows the timings for Acknowledge sequence.

**FIGURE 33-17: ACKNOWLEDGE SEQUENCE TIMING**




### 33.5.8 STOP CONDITION TIMING

A Stop bit is asserted on the SDA pin at the end of receive/transmit when I2CxCNT = 0. After the last byte of a receive/transmit sequence, the SCL line is held low. The master asserts the SDA line low. The SCL pin is then released high TSCL/2 later and is detected high. The SDA pin is then released. When the SDA pin transitions high while SCL is high, the PCIF bit of the I2CxIF register is set. Figure 33-18 shows the timings for a Stop condition.

**MOVSF** **Move Indexed to f**

Syntax: MOVSF [$z_s$], $f_d$

Operands: $0 \le z_s \le 127$
$0 \le f_d \le 4095$

Operation: ((FSR2) + $z_s$) → $f_d$

Status Affected: None

Encoding:

| | | | | |
|---|---|---|---|---|
| 1st word (source) | 1110 | 1011 | 0zzz | $zzzz_s$ |
| 2nd word (destin.) | 1111 | ffff | ffff | $ffff_d$ |

Description: The contents of the source register are moved to destination register '$f_d$'. The actual address of the source register is determined by adding the 7-bit literal offset '$z_s$' in the first word to the value of FSR2. The address of the destination register is specified by the 12-bit literal '$f_d$' in the second word. Both addresses can be anywhere in the 4096-byte data space (000h to FFFh).
MOVSF has curtailed the destination range to the lower 4 Kbyte space in memory (Banks 1 through 15). For everything else, use MOVSFL.

Words: 2

Cycles: 2

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | Determine source addr | Determine source addr | Read source reg |
| Decode | No operation<br>No dummy read | No operation | Write register 'f' (dest) |

Example: MOVSF [05h], REG2

Before Instruction
FSR2 = 80h
Contents of 85h = 33h
REG2 = 11h
After Instruction
FSR2 = 80h
Contents of 85h = 33h
REG2 = 33h

**MOVSFL** **Move Indexed to f (Long Range)**

Syntax: MOVSFL [$z_s$], $f_d$

Operands: $0 \le z_s \le 127$
$0 \le f_d \le 16383$

Operation: ((FSR2) + $z_s$) → $f_d$

Status Affected: None

Encoding:

| | | | | |
|---|---|---|---|---|
| 1st word (opcode) | 0000 | 0000 | 0110 | 0010 |
| 2nd word (source) | 1111 | xxxz | zzzz | $zz_s$ff |
| 3rd word (full destin.) | 1111 | ffff | ffff | $ffff_d$ |

Description: The contents of the source register are moved to destination register '$f_d$'. The actual address of the source register is determined by adding the 7-bit literal offset '$z_s$' in the first word to the value of FSR2 (14 bits). The address of the destination register is specified by the 14-bit literal '$f_d$' in the second word. Both addresses can be anywhere in the 16 Kbyte data space (0000h to 3FFFh). The MOVSFL instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register. If the resultant source address points to an indirect addressing register, the value returned will be 00h.

Words: 3

Cycles: 3

Q Cycle Activity:

| Q1 | Q2 | Q3 | Q4 |
|---|---|---|---|
| Decode | No operation | No operation | No operation |
| Decode | Read register "z" (src.) | Process data | No operation |
| Decode | No operation<br>No dummy read | No operation | Write register "f" (dest.) |

Example: MOVSFL [05h], REG2

Before Instruction
FSR2 = 80h
Contents of 85h = 33h
REG2 = 11h
After Instruction
FSR2 = 80h
Contents of 85h = 33h

**TABLE 44-5: POWER-DOWN CURRENT (IPD)(1,2)**

| PIC18LF27/47/57K42 | | | | Standard Operating Conditions (unless otherwise stated) | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| PIC18F27/47/57K42 | | | | Standard Operating Conditions (unless otherwise stated)<br>VREGPM = 1 | | | | | |
| Param. No. | Symbol | Device Characteristics | Min. | Typ.† | Max. +85°C | Max. +125°C | Units | Conditions | |
| | | | | | | | | VDD | Note |
| D200 | IPD | IPD Base | — | 0.07 | 2 | 10.5 | µA | 3.0V | |
| D200<br>D200A | IPD | IPD Base | — | 0.4 | 4 | 12 | µA | 3.0V | |
| | | | — | 20 | 38 | 42 | µA | 3.0V | VREGPM = 0 |
| D201 | IPD_WDT | Low-Frequency Internal Oscillator/ WDT | — | 0.9 | 3.2 | 11.2 | µA | 3.0V | |
| D201 | IPD_WDT | Low-Frequency Internal Oscillator/ WDT | — | 1.1 | 3.2 | 13 | µA | 3.0V | |
| D202 | IPD_SOSC | Secondary Oscillator (SOSC) | — | 0.75 | 6 | 14 | µA | 3.0V | LP mode |
| D202 | IPD_SOSC | Secondary Oscillator (SOSC) | — | 1.0 | 7 | 15 | µA | 3.0V | LP mode |
| D203 | IPD_FVR | FVR | — | 45 | 74 | 75 | µA | 3.0V | FVRCON = 0x81 or 0x84 |
| D203 | IPD_FVR | FVR | — | 40 | 70 | 76 | µA | 3.0V | FVRCON = 0x81 or 0x84 |
| D204 | IPD_BOR | Brown-out Reset (BOR) | — | 9.4 | 14 | 18 | µA | 3.0V | |
| D204 | IPD_BOR | Brown-out Reset (BOR) | — | 9.4 | 15 | 18 | µA | 3.0V | |
| D205 | IPD_LPBOR | Low-Power Brown-out Reset (LPBOR) | — | 0.2 | 3 | 11 | µA | 3.0V | |
| D206 | IPD_HLVD | High/Low Voltage Detect (HLVD) | — | 9.5 | 14.8 | 18 | µA | 3.0V | |
| D206 | IPD_HLVD | High/Low Voltage Detect (HLVD) | — | 9.7 | 14.2 | 17 | µA | 3.0V | |
| D207 | IPD_ADCA | ADC - Converting | — | .01 | 2 | 10.5 | µA | 3.0V | |
| D207 | IPD_ADCA | ADC - Converting | — | 0.1 | 4 | 12 | µA | 3.0V | ADC not converting (4) |
| D208 | IPD_CMP | Comparator | — | 33 | 49 | 50 | µA | 3.0V | |
| D208 | IPD_CMP | Comparator | — | 30 | 49 | 50 | µA | 3.0V | |

† Data in "Typ." column is at 3.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

**Note 1:** The peripheral current is the sum of the base IDD and the additional current consumed when this peripheral is enabled. The peripheral Δ current can be determined by subtracting the base IDD or IPD current from this limit. Max. values should be used when calculating total current consumption.

**2:** The power-down current in Sleep mode does not depend on the oscillator type. Power-down current is measured with the part in Sleep mode with all I/O pins in high-impedance state and tied to VSS.

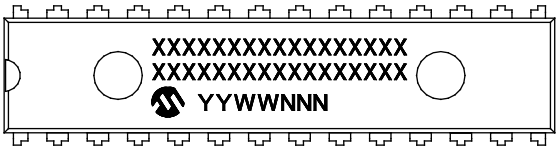
**3:** All peripheral currents listed are on a per-peripheral basis if more than one instance of a peripheral is available.

**4:** ADC clock source is FRC.

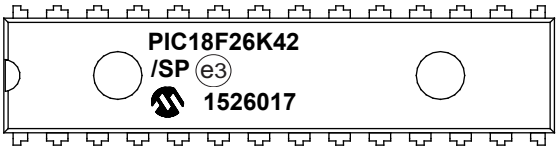
## 46.0 PACKAGING INFORMATION

### Package Marking Information

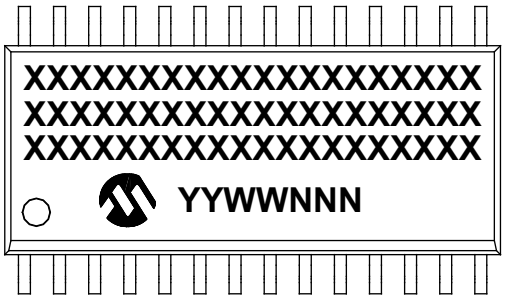
28-Lead SPDIP (.300”)



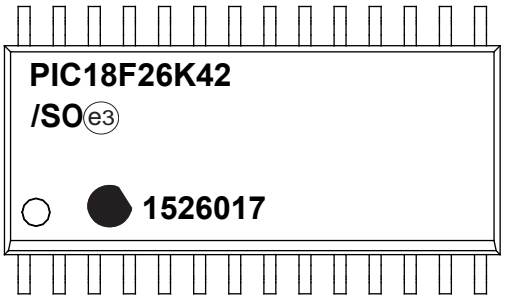

Example



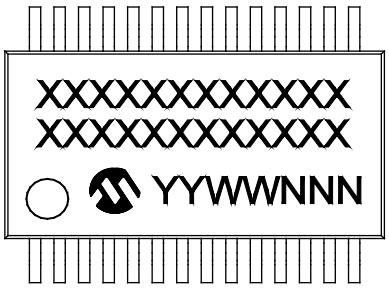

28-Lead SOIC (7.50 mm)



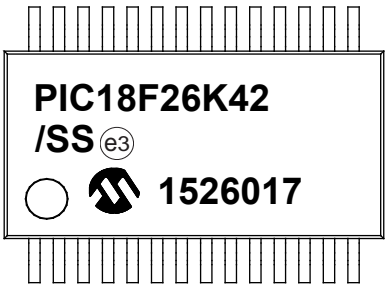

Example




28-Lead SSOP (5.30 mm)




Example

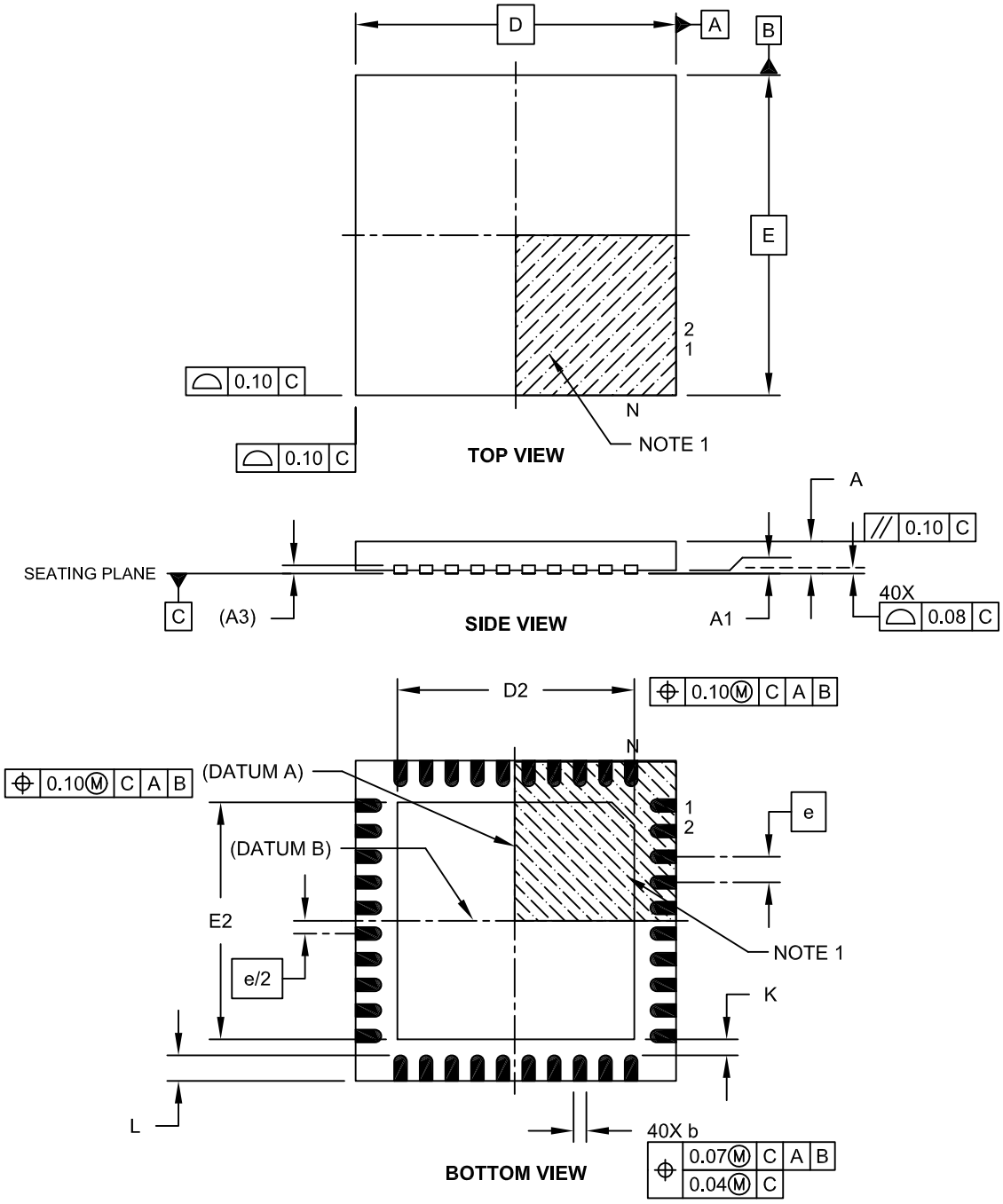



| | | |
|---|---|---|
| **Legend:** | XX...X | Customer-specific information or Microchip part number |
| | Y | Year code (last digit of calendar year) |
| | YY | Year code (last 2 digits of calendar year) |
| | WW | Week code (week of January 1 is week ‘01’) |
| | NNN | Alphanumeric traceability code |
| | (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 event the full Microchip part number cannot be marked on one line, it will be carried over to the next line, thus limiting the number of available characters for customer-specific information.

**40-Lead Ultra Thin Plastic Quad Flat, No Lead Package (MV) – 5x5x0.5 mm Body [UQFN]**

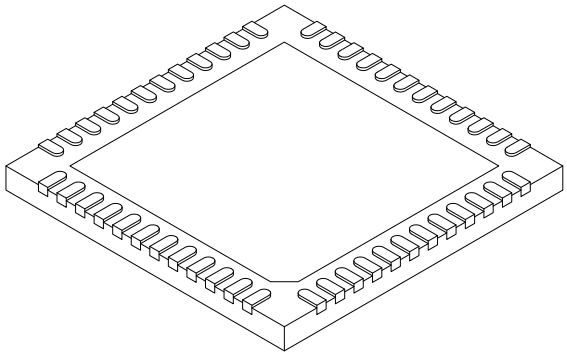
| **Note:** | For the most current package drawings, please see the Microchip Packaging Specification located at http://www.microchip.com/packaging |
|---|---|




Microchip Technology Drawing C04-156A Sheet 1 of 2

### 48-Lead Plastic Ultra Thin Quad Flat, No Lead Package (MV) – 6x6x0.5 mm Body [UQFN]

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



| Units | | MILLIMETERS | | |
|---|---|---|---|---|
| Dimension Limits | | MIN | NOM | MAX |
| Number of Pins | N | | 48 | |
| Pitch | e | | 0.40 BSC | |
| Overall Height | A | 0.45 | 0.50 | 0.55 |
| Standoff | A1 | 0.00 | 0.02 | 0.05 |
| Contact Thickness | A3 | | 0.127 REF | |
| Overall Width | E | | 6.00 BSC | |
| Exposed Pad Width | E2 | 4.45 | 4.60 | 4.75 |
| Overall Length | D | | 6.00 BSC | |
| Exposed Pad Length | D2 | 4.45 | 4.60 | 4.75 |
| Contact Width | b | 0.15 | 0.20 | 0.25 |
| 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 C04-153A Sheet 2 of 2