

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·XF

Product Status	Active
Core Processor	PIC
Core Size	8-Bit
Speed	64MHz
Connectivity	I ² C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	60
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.5K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 3.6V
Data Converters	A/D 45x10b; D/A 1x5b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf66k40t-i-pt

Email: info@E-XFL.COM

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

Digital Peripherals (Continued)

- Programmable CRC with Memory Scan:
 - Reliable data/program memory monitoring for Fail-Safe operation (e.g., Class B)
 - Calculate CRC over any portion of Flash or EEPROM
 - High-speed or background operation
- Hardware Limit Timer (TMR2/4/6/8+HLT):
- Hardware monitoring and Fault detection
- Peripheral Pin Select (PPS):
- Enables pin mapping of digital I/O
- Data Signal Modulator (DSM)
- Two Signal Measurement Timer (SMT1/2):
 - 24-bit timer/counter with prescaler
 - Multiple gate and clock inputs

Analog Peripherals

- 10-Bit Analog-to-Digital Converter with Computation (ADC²):
 - 47 external channels
 - Conversion available during Sleep
 - Four internal analog channels
 - Internal and external trigger options
 - Automated math functions on input signals:
 averaging, filter calculations, oversampling and threshold comparison
 - pling and threshold company
- Zero-Cross Detect (ZCD):
 - Detect when AC signal on pin crosses ground
- 5-Bit Digital-to-Analog Converter (DAC):
 - Output available externally
 - Programmable 5-bit voltage (% of VDD)
 - Internal connections to comparators, Fixed Voltage Reference and ADC
- Three Comparators (CMP):
 - Five external inputs
 - External output via PPS
- Fixed Voltage Reference (FVR) module:
 - 1.024V, 2.048V and 4.096V output levels

Clocking Structure

- High-Precision Internal Oscillator Block (HFINTOSC):
- Selectable frequency range up to 64 MHz
 ±1% at calibration
- 32 kHz Low-Power Internal Oscillator (LFINTOSC)
- External 32 kHz Crystal Oscillator (SOSC)
- External Oscillator Block:
 - Three crystal/resonator modes
 - 4x PLL with external sources
- · Fail-Safe Clock Monitor:
 - Allows for safe shutdown if peripheral clock stops
- Oscillator Start-up Timer (OST)

Programming/Debug Features

- In-Circuit Debug Integrated On-Chip
- In-Circuit Serial Programming[™] (ICSP[™]) via Two Pins
- In-Circuit Debug (ICD) with Three Breakpoints via Two Pins

4.3.2.6 Oscillator Status and Manual Enable

The Ready status of each oscillator (including the ADCRC oscillator) is displayed in OSCSTAT (Register 4-4). The oscillators (but not the PLL) may be explicitly enabled through OSCEN (Register 4-7).

4.3.2.7 HFOR and MFOR Bits

The HFOR and MFOR bits indicate that the HFINTOSC and MFINTOSC is ready. These clocks are always valid for use at all times, but only accurate after they are ready.

When a new value is loaded into the OSCFRQ register, the HFOR and MFOR bits will clear, and set again when the oscillator is ready. During pending OSCFRQ changes the MFINTOSC clock will stall at a high or a low state, until the HFINTOSC resumes operation.

4.4 Clock Switching

The system clock source can be switched between external and internal clock sources via software using the New Oscillator Source (NOSC) bits of the OSCCON1 register. The following clock sources can be selected using the following:

- External oscillator
- Internal Oscillator Block (INTOSC)

Note:	The Clock	Switch	Enable	bit	in
	Configuration	Word 1	can be	used	to
	enable or dis	able the	clock s	witchi	ng
	capability. Whe	en cleare	d, the NC)SC a	nd
	NDIV bits car	not be	changed	by us	ser
	software. Whe	n set, wri	ting to NC	DSC a	nd
	NDIV is allow	ed and	would sw	/itch t	he
	clock frequenc	у.			

4.4.1 NEW OSCILLATOR SOURCE (NOSC) AND NEW DIVIDER SELECTION REQUEST (NDIV) BITS

The New Oscillator Source (NOSC) and New Divider Selection Request (NDIV) bits of the OSCCON1 register select the system clock source and frequency that are used for the CPU and peripherals.

When new values of NOSC and NDIV are written to OSCCON1, the current oscillator selection will continue to operate while waiting for the new clock source to indicate that it is stable and ready. In some cases, the newly requested source may already be in use, and is ready immediately. In the case of a divider-only change, the new and old sources are the same, so the old source will be ready immediately. The device may enter Sleep while waiting for the switch as described in **Section 4.4.2 "Clock Switch and Sleep"**. When the new oscillator is ready, the New Oscillator Ready (NOSCR) bit of OSCCON3 is set and also the Clock Switch Interrupt Flag (CSWIF) bit of PIR1 sets. If Clock Switch Interrupts are enabled (CSWIE = 1), an interrupt will be generated at that time. The Oscillator Ready (ORDY) bit of OSCCON3 can also be polled to determine when the oscillator is ready in lieu of an interrupt.

Note:	The CSWIF interrupt will not wake the
	system from Sleep.

If the Clock Switch Hold (CSWHOLD) bit of OSCCON3 is clear, the oscillator switch will occur when the New Oscillator is Ready bit (NOSCR) is set, and the interrupt (if enabled) will be serviced at the new oscillator setting.

If CSWHOLD is set, the oscillator switch is suspended, while execution continues using the current (old) clock source. When the NOSCR bit is set, software should:

- Set CSWHOLD = 0 so the switch can complete, or
- Copy COSC into NOSC to abandon the switch.

If DOZE is in effect, the switch occurs on the next clock cycle, whether or not the CPU is operating during that cycle.

Changing the clock post-divider without changing the clock source (i.e., changing Fosc from 1 MHz to 2 MHz) is handled in the same manner as a clock source change, as described previously. The clock source will already be active, so the switch is relatively quick. CSWHOLD must be clear (CSWHOLD = 0) for the switch to complete.

The current COSC and CDIV are indicated in the OSCCON2 register up to the moment when the switch actually occurs, at which time OSCCON2 is updated and ORDY is set. NOSCR is cleared by hardware to indicate that the switch is complete.

5.1 Clock Source

The input to the reference clock output can be selected using the CLKRCLK register.

5.1.1 CLOCK SYNCHRONIZATION

Once the reference clock enable (EN) is set, the module is ensured to be glitch-free at start-up.

When the reference clock output is disabled, the output signal will be disabled immediately.

Clock dividers and clock duty cycles can be changed while the module is enabled, but glitches may occur on the output. To avoid possible glitches, clock dividers and clock duty cycles should be changed only when the CLKREN is clear.

5.2 Programmable Clock Divider

The module takes the clock input and divides it based on the value of the DIV<2:0> bits of the CLKRCON register (Register 5-1).

The following configurations can be made based on the DIV<2:0> bits:

- · Base Fosc value
- Fosc divided by 2
- Fosc divided by 4
- Fosc divided by 8
- Fosc divided by 16
- Fosc divided by 32
- Fosc divided by 64
- Fosc divided by 128

The clock divider values can be changed while the module is enabled; however, in order to prevent glitches on the output, the DIV<2:0> bits should only be changed when the module is disabled (EN = 0).

5.3 Selectable Duty Cycle

The DC<1:0> bits of the CLKRCON register can be used to modify the duty cycle of the output clock. A duty cycle of 25%, 50%, or 75% can be selected for all clock rates, with the exception of the undivided base Fosc value.

The duty cycle can be changed while the module is enabled; however, in order to prevent glitches on the output, the DC<1:0> bits should only be changed when the module is disabled (EN = 0).

Note: The DC1 bit is reset to '1'. This makes the default duty cycle 50% and not 0%.

5.4 Operation in Sleep Mode

The reference clock output module clock is based on the system clock. When the device goes to Sleep, the module outputs will remain in their current state. This will have a direct effect on peripherals using the reference clock output as an input signal. No change should occur in the module from entering or exiting from Sleep.

11.1.2 CONTROL REGISTERS

Several control registers are used in conjunction with the TBLRD and TBLWT instructions. These include the following registers:

- NVMCON1 register
- NVMCON2 register
- TABLAT register
- TBLPTR registers

11.1.2.1 NVMCON1 and NVMCON2 Registers

The NVMCON1 register (Register 11-1) is the control register for memory accesses. The NVMCON2 register is not a physical register; it is used exclusively in the memory write and erase sequences. Reading NVMCON2 will read all '0's.

The NVMREG<1:0> control bits determine if the access will be to Data EEPROM Memory locations. PFM locations or User IDs, Configuration bits, Rev ID and Device ID. When NVMREG<1:0> = 00, any subsequent operations will operate on the Data EEPROM Memory. When NVMREG<1:0> = 10, any subsequent operations will operate on the program memory. When NVMREG<1:0> = x1, any subsequent operations will operate on the Data IDS, Rev ID and Device ID.

The FREE bit allows the program memory erase operation. When the FREE bit is set, an erase operation is initiated on the next WR command. When FREE is clear, only writes are enabled. This bit is applicable only to the PFM and not to data EEPROM.

When set, the WREN bit will allow a program/erase operation. The WREN bit is cleared on power-up.

The WRERR bit is set by hardware when the WR bit is set and cleared when the internal programming timer expires and the write operation is successfully complete.

The WR control bit initiates erase/write cycle operation when the NVMREG<1:0> bits point to the Data EEPROM Memory location, and it initiates a write operation when the NVMREG<1:0> bits point to the PFM location. The WR bit cannot be cleared by firmware; it can only be set by firmware. Then the WR bit is cleared by hardware at the completion of the write operation.

The NVMIF Interrupt Flag bit of the PIR7 register is set when the write is complete. The NVMIF flag stays set until cleared by firmware.

11.1.2.2 TABLAT – Table Latch Register

The Table Latch (TABLAT) is an 8-bit register mapped into the SFR space. The Table Latch register is used to hold 8-bit data during data transfers between program memory and data RAM.

11.1.2.3 TBLPTR – Table Pointer Register

The Table Pointer (TBLPTR) register addresses a byte within the program memory. The TBLPTR is comprised of three SFR registers: Table Pointer Upper Byte, Table Pointer High Byte and Table Pointer Low Byte (TBLPTRU:TBLPTRH:TBLPTRL). These three registers join to form a 22-bit wide pointer. The low-order 21 bits allow the device to address up to 2 Mbytes of program memory space. The 22nd bit allows access to the Device ID, the User ID and the Configuration bits.

The Table Pointer register, TBLPTR, is used by the TBLRD and TBLWT instructions. These instructions can update the TBLPTR in one of four ways based on the table operation. These operations on the TBLPTR affect only the low-order 21 bits.

11.1.2.4 Table Pointer Boundaries

TBLPTR is used in reads, writes and erases of the Program Flash Memory.

When a TBLRD is executed, all 22 bits of the TBLPTR determine which byte is read from program memory directly into the TABLAT register.

When a TBLWT is executed the byte in the TABLAT register is written, not to Flash memory but, to a holding register in preparation for a program memory write. The holding registers constitute a write block which varies depending on the device (see Table 11-3). The 3, 4, or 5 LSbs of the TBLPTRL register determine which specific address within the holding register block is written to. The MSBs of the Table Pointer have no effect during TBLWT operations.

When a program memory write is executed the entire holding register block is written to the Flash memory at the address determined by the MSbs of the TBLPTR. The 3, 4, or 5 LSBs are ignored during Flash memory writes. For more detail, see **Section 11.1.6 "Writing to Program Flash Memory"**.

Figure 11-3 describes the relevant boundaries of TBLPTR based on Program Flash Memory operations.

U-0	U-0	U-0	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1	R/W-1/1		
	_		CCP5IP	CCP4IP	CCP3IP	CCP2IP	CCP1IP		
bit 7					•		bit 0		
Legend:									
R = Readable	bit	W = Writable	bit	U = Unimpler	mented bit, read	l as '0'			
-n = Value at F	POR	'1' = Bit is set		'0' = Bit is cle	ared	x = Bit is unkr	nown		
bit 7-5	Unimplemen	ted: Read as '	כ'						
bit 4	CCP5IP: ECCP5 Interrupt Priority bit 1 = High priority 0 = Low priority								
bit 3	CCP4IP: ECC 1 = High prio 0 = Low prior	CP4 Interrupt P rity ity	riority bit						
bit 2	CCP3IP: ECC 1 = High prio 0 = Low prior	CP3 Interrupt P rity ity	riority bit						
bit 1	CCP2IP: ECC 1 = High prio 0 = Low prior	CP2 Interrupt P rity ity	riority bit						
bit 0	CCP1IP: ECC 1 = High prio 0 = Low prior	CP1 Interrupt P rity ity	riority bit						

REGISTER 14-29: IPR7: PERIPHERAL INTERRUPT PRIORITY REGISTER 7

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IOCBP	IOCBP7	IOCBP6	IOCBP5	IOCBP4	IOCBP3	IOCBP2	IOCBP1	IOCBP0
IOCBN	IOCBN7	IOCBN6	IOCBN5	IOCBN4	IOCBN3	IOCBN2	IOCBN1	IOCBN0
IOCBF	IOCBF7	IOCBF6	IOCBF5	IOCBF4	IOCBF3	IOCBF2	IOCBF1	IOCBF0
IOCCP	IOCCP7	IOCCP6	IOCCP5	IOCCP4	IOCCP3	IOCCP2	IOCCP1	IOCCP0
IOCCN	IOCCN7	IOCCN6	IOCCN5	IOCCN4	IOCCN3	IOCCN2	IOCCN1	IOCCN0
IOCCF	IOCCF7	IOCCF6	IOCCF5	IOCCF4	IOCCF3	IOCCF2	IOCCF1	IOCCF0
IOCEP	IOCEP7	IOCEP6	IOCEP5	IOCEP4	IOCEP3	IOCEP2	IOCEP1	IOCEP0
IOCEN	IOCEN7	IOCEN6	IOCEN5	IOCEN4	IOCEN3	IOCEN2	IOCEN1	IOCEN0
IOCEF	IOCEF7	IOCEF6	IOCEF5	IOCEF4	IOCEF3	IOCEF2	IOCEF1	IOCEF0
IOCGP	—	—	IOCGP5 ⁽¹⁾	_		—		
IOCGN			IOCGN5 ⁽¹⁾					
IOCGF	—	—	IOCGF5 ⁽¹⁾	—	—	—	_	_

TABLE 16-1: IOC REGISTERS

Note 1: If MCLRE = 1 or LVP = 1, RG5 port functionality is disabled and IOC on RG5 is not available.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	INT3EDG	INT2EDG	INT1EDG	INT0EDG	173
IOCxF	IOCxF7	IOCxF6	IOCxF5	IOCxF4	IOCxF3	IOCxF2	IOCxF1	IOCxF0	220
IOCxN	IOCxN7	IOCxN6	IOCxN5	IOCxN4	IOCxN3	IOCxN2	IOCxN1	IOCxN0	220
IOCxP	IOCxP7	IOCxP6	IOCxP5	IOCxP4	IOCxP3	IOCxP2	IOCxP1	IOCxP0	220

Legend: — = unimplemented location, read as '0'. Shaded cells are not used by interrupt-on-change.

20.5.4 LEVEL-TRIGGERED HARDWARE LIMIT MODE

In the Level-Triggered Hardware Limit Timer modes the counter is reset by high or low levels of the external signal TMRx_ers, as shown in Figure 20-7. Selecting MODE<4:0> = 00110 will cause the timer to reset on a low level external signal. Selecting MODE<4:0> = 00111 will cause the timer to reset on a high level external signal. In the example, the counter is reset while TMRx_ers = 1. ON is controlled by BSF and BCF instructions. When ON = 0 the external signal is ignored.

When the CCP uses the timer as the PWM time base then the PWM output will be set high when the timer starts counting and then set low only when the timer count matches the CCPRx value. The timer is reset when either the timer count matches the PRx value or two clock periods after the external Reset signal goes true and stays true.

The timer starts counting, and the PWM output is set high, on either the clock following the PRx match or two clocks after the external Reset signal relinquishes the Reset. The PWM output will remain high until the timer counts up to match the CCPRx pulse width value. If the external Reset signal goes true while the PWM output is high then the PWM output will remain high until the Reset signal is released allowing the timer to count up to match the CCPRx value.





Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on Page
CCPTMRS1	—	—	P7TSE	L<1:0>	P6TSE	L<1:0>	C5TSE	EL<1:0>	298
PWM6CON	EN	—	OUT	POL	_	_	-	—	297
PWM6DCH		DC<7:0>							299
PWM6DCL	PWM6I	DC<9:8>	_	—	—	—	—	—	299
PWM7CON	EN	—	PWM7OUT	PWM7POL	_	_	-	—	297
PWM7DCH				PWM7DC	H<7:0>				299
PWM7DCL	PWM7D)CL<7:6>	_	_	_	_	_	—	299
INTCON	GIE/GIEH	PEIE/GIEL	IPEN	—	INT3EDG	INT2EDG	INT1EDG	INT0EDG	173
PIE5	TMR8IE	TMR7IE	TMR6IE	TMR5IE	TMR4IE	TMR3IE	TMR2IE	TMR1IE	190
PIR5	TMR8IF	TMR7IF	TMR6IF	TMR5IF	TMR4IF	TMR3IF	TMR2IF	TMR1IF	179
IPR5	TMR8IP	TMR7IP	TMR6IP	TMR5IP	TMR4IP	TMR3IP	TMR2IP	TMR1IP	200
RxyPPS	—	—			RxyPPS	<5:0>			228
TMR2				TMR2<	7:0>				255*
PR2				PR2<7	':0>				255*
T2CON	T2ON		T2CKPS<2:0>			T2OUTF	PS<3:0>		273
T2HLT	T2PSYNC	T2CPOL	T2CSYNC		T2	2MODE<4:0>			274
T2CLKCON	_	_	_	_		T2CS	<3:0>		275
T2RST	—	—	—	—		T2RSE	L<3:0>		276
PMD4	_	PWM7MD	PWM6MD	CCP5MD	CCP4MD	CCP3MD	CCP2MD	CCP1MD	70

TABLE 22-4: SUMMARY OF REGISTERS ASSOCIATED WITH PWM

Legend: - = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the PWM.

* Not a physical location.



FIGURE 25-8: HIGH A

HIGH AND LOW MEASURE MODE REPEAT ACQUISITION TIMING DIAGRAM

PIC18(L)F65/66K40

25.7.10 GATED COUNTER MODE

This mode counts pulses on the SMTx_signal input, gated by the SMTxWIN input. It begins incrementing the timer upon seeing a rising edge of the SMTxWIN input and updates the SMTxCPW register upon a falling edge on the SMTxWIN input. See Figure 25-19 and Figure 25-20.

27.5 SPI Mode Operation

Transmissions involve two shift registers, eight bits in size, one in the master and one in the slave. With either the master or the slave device, data is always shifted out one bit at a time, with the Most Significant bit (MSb) shifted out first. At the same time, a new Least Significant bit (LSb) is shifted into the same register.

Figure 27-3 shows a typical connection between two processors configured as master and slave devices.

Data is shifted out of both shift registers on the programmed clock edge and latched on the opposite edge of the clock.

The master device transmits information out on its SDO output pin which is connected to, and received by, the slave's SDI input pin. The slave device transmits information out on its SDO output pin, which is connected to, and received by, the master's SDI input pin.

To begin communication, the master device first sends out the clock signal. Both the master and the slave devices should be configured for the same clock polarity.

The master device starts a transmission by sending out the MSb from its shift register. The slave device reads this bit from that same line and saves it into the LSb position of its shift register.

During each SPI clock cycle, a full-duplex data transmission occurs. This means that while the master device is sending out the MSb from its shift register (on its SDO pin) and the slave device is reading this bit and saving it as the LSb of its shift register, that the slave device is also sending out the MSb from its shift register (on its SDO pin) and the master device is reading this bit and saving it as the LSb of its shift register.

After eight bits have been shifted out, the master and slave have exchanged register values.

If there is more data to exchange, the shift registers are loaded with new data and the process repeats itself.

Whether the data is meaningful or not (dummy data), depends on the application software. This leads to three scenarios for data transmission:

- Master sends useful data and slave sends dummy data.
- Master sends useful data and slave sends useful data.
- Master sends dummy data and slave sends useful data.

Transmissions may involve any number of clock cycles. When there is no more data to be transmitted, the master stops sending the clock signal and it deselects the slave.

Every slave device connected to the bus that has not been selected through its slave select line must disregard the clock and transmission signals and must not transmit out any data of its own. When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPxCON1<5:0> and SSPxSTAT<7:6>). These control bits allow the following to be specified:

- Master mode (SCK is the clock output)
- Slave mode (SCK is the clock input)
- Clock Polarity (Idle state of SCK)
- Data Input Sample Phase (middle or end of data output time)
- Clock Edge (output data on rising/falling edge of SCK)
- Clock Rate (Master mode only)
- Slave Select mode (Slave mode only)

To enable the serial port, SSP Enable bit, SSPEN of the SSPxCON1 register, must be set. To reset or reconfigure SPI mode, clear the SSPEN bit, re-initialize the SSPxCONx registers and then set the SSPEN bit. This configures the SDI, SDO, SCK and SS pins as serial port pins. For the pins to behave as the serial port function, some must have their data direction bits (in the TRIS register) appropriately programmed as follows:

- · SDI must have corresponding TRIS bit set
- SDO must have corresponding TRIS bit cleared
- SCK (Master mode) must have corresponding
 TRIS bit cleared
- SCK (Slave mode) must have corresponding TRIS bit set
- SS must have corresponding TRIS bit set

Any serial port function that is not desired may be overridden by programming the corresponding data direction (TRIS) register to the opposite value.

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPxBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPxBUF holds the data that was written to the SSPSR until the received data is ready. Once the eight bits of data have been received, that byte is moved to the SSPxBUF register. Then, the Buffer Full Detect bit, BF of the SSPxSTAT register, and the interrupt flag bit, SSPxIF, are set. This double-buffering of the received data (SSPxBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPxBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL of the SSPxCON1 register, will be set. User software must clear the WCOL bit to allow the following write(s) to the SSPxBUF register to complete successfully.

27.9.8 GENERAL CALL ADDRESS SUPPORT

The addressing procedure for the I^2C bus is such that the first byte after the Start condition usually determines which device will be the slave addressed by the master device. The exception is the general call address which can address all devices. When this address is used, all devices should, in theory, respond with an acknowledge.

The general call address is a reserved address in the I²C protocol, defined as address 0x00. When the GCEN bit of the SSPxCON2 register is set, the slave module will automatically ACK the reception of this address regardless of the value stored in SSPxADD. After the slave clocks in an address of all zeros with the R/W bit clear, an interrupt is generated and slave software can read SSPxBUF and respond. Figure 27-24 shows reception а general call sequence.

In 10-bit Address mode, the UA bit will not be set on the reception of the general call address. The slave will prepare to receive the second byte as data, just as it would in 7-bit mode. If the AHEN bit of the SSPxCON3 register is set, just as with any other address reception, the slave hardware will stretch the clock after the eighth falling edge of SCL. The slave must then set its ACKDT value and release the clock with communication progressing as it would normally.

27.9.9 SSP MASK REGISTER

An SSP Mask (SSPxMSK) register (Register 27-12) is available in I²C Slave mode as a mask for the value held in the SSPSR register during an address comparison operation. A zero ('0') bit in the SSPxMSK register has the effect of making the corresponding bit of the received address a "don't care".

This register is reset to all '1's upon any Reset condition and, therefore, has no effect on standard SSP operation until written with a mask value.

The SSP Mask register is active during:

• 7-bit Address mode: address compare of A<7:1>.

10-bit Address mode: address compare of A<7:0> only. The SSP mask has no effect during the reception of the first (high) byte of the address.





27.10.5 I²C MASTER MODE REPEATED START CONDITION TIMING

A Repeated Start condition (Figure 27-27) occurs when the RSEN bit of the SSPxCON2 register is programmed high and the master state machine is no longer active. When the RSEN bit is set, the SCL pin is asserted low. When the SCL pin is sampled low, the Baud Rate Generator is loaded and begins counting. The SDA pin is released (brought high) for one Baud Rate Generator count (TBRG). When the Baud Rate Generator times out, if SDA is sampled high, the SCL pin will be deasserted (brought high). When SCL is sampled high, the Baud Rate Generator is reloaded and begins counting. SDA and SCL must be sampled high for one TBRG. This action is then followed by assertion of the SDA pin (SDA = 0) for one TBRG while SCL is high. SCL is asserted low. Following this, the RSEN bit of the SSPxCON2 register will be automatically cleared and the Baud Rate Generator will not be reloaded, leaving the SDA pin held low. As soon as a Start condition is detected on the SDA and SCL pins, the S bit of the SSPxSTAT register will be set. The SSPxIF bit will not be set until the Baud Rate Generator has timed out.

- **Note 1:** If RSEN is programmed while any other event is in progress, it will not take effect.
 - **2:** A bus collision during the Repeated Start condition occurs if:
 - SDA is sampled low when SCL goes from low-to-high.
 - SCL goes low before SDA is asserted low. This may indicate that another master is attempting to transmit a data '1'.

27.10.6 I²C MASTER MODE TRANSMISSION

Transmission of a data byte, a 7-bit address or the other half of a 10-bit address is accomplished by simply writing a value to the SSPxBUF register. This action will set the Buffer Full flag bit, BF, and allow the Baud Rate Generator to begin counting and start the next transmission. Each bit of address/data will be shifted out onto the SDA pin after the falling edge of SCL is asserted. SCL is held low for one Baud Rate Generator rollover count (TBRG). Data should be valid before SCL is released high. When the SCL pin is released high, it is held that way for TBRG. The data on the SDA pin must remain stable for that duration and some hold time after the next falling edge of SCL. After the eighth bit is shifted out (the falling edge of the eighth clock), the BF flag is cleared and the master releases SDA. This allows the slave device being addressed to respond with an \overline{ACK} bit during the ninth bit time if an address match occurred, or if data was received properly. The status of ACK is written into the ACKSTAT bit on the rising edge of the ninth clock. If the master receives an Acknowledge, the Acknowledge Status bit, ACKSTAT, is cleared. If not, the bit is set. After the ninth clock, the SSPxIF bit is set and the master clock (Baud Rate Generator) is suspended until the next data byte is loaded into the SSPxBUF, leaving SCL low and SDA unchanged (Figure 27-28).

FIGURE 27-27: REPEATED START CONDITION WAVEFORM







© 2016-2017 Microchip Technology Inc.

PIC18(L)F65/66K40

27.10.13.1 Bus Collision During a Start Condition

During a Start condition, a bus collision occurs if:

- a) SDA or SCL are sampled low at the beginning of the Start condition (Figure 27-33).
- b) SCL is sampled low before SDA is asserted low (Figure 27-34).

During a Start condition, both the SDA and the SCL pins are monitored.

If the SDA pin is already low, or the SCL pin is already low, then all of the following occur:

- · the Start condition is aborted,
- the BCLxIF flag is set and
- the MSSP module is reset to its Idle state (Figure 27-33).

The Start condition begins with the SDA and SCL pins deasserted. When the SDA pin is sampled high, the Baud Rate Generator is loaded and counts down. If the SCL pin is sampled low while SDA is high, a bus collision occurs because it is assumed that another master is attempting to drive a data '1' during the Start condition.

If the SDA pin is sampled low during this count, the BRG is reset and the SDA line is asserted early (Figure 27-35). If, however, a '1' is sampled on the SDA pin, the SDA pin is asserted low at the end of the BRG count. The Baud Rate Generator is then reloaded and counts down to zero; if the SCL pin is sampled as '0' during this time, a bus collision does not occur. At the end of the BRG count, the SCL pin is asserted low.

Note: The reason that bus collision is not a factor during a Start condition is that no two bus masters can assert a Start condition at the exact same time. Therefore, one master will always assert SDA before the other. This condition does not cause a bus collision because the two masters must be allowed to arbitrate the first address following the Start condition. If the address is the same, arbitration must be allowed to continue into the data portion, Repeated Start or Stop conditions.





R-0/0	R-1/1	U-0	R/W-0/0	R/W-0/0	U-0	R/W-0/0	R/W-0/0
ABDOVF	RCIDL	_	SCKP	BRG16		WUE	ABDEN
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable	bit	U = Unimplei	mented bit, read	l as '0'	
u = Bit is unch	anged	x = Bit is unkr	nown	-n/n = Value	at POR and BO	R/Value at all o	ther Resets
'1' = Bit is set		'0' = Bit is cle	ared				
bit /	ABDOVF: Au	to-Baud Detec	t Overflow bit				
	Asynchronous	<u>s mode</u> : 1 timer overflov	ved				
	0 = Auto-bauc	d timer did not	overflow				
	Synchronous	mode:					
	Don't care						
bit 6	RCIDL: Recei	ive Idle Flag bi	t				
	Asynchronous	<u>s mode</u> :					
	\perp = Receiver 1 0 = Start bit bit	IS IQIE as been receiv	ed and the rea	ceiver is receiv	vina		
	Synchronous	mode:			ing .		
	Don't care						
bit 5	Unimplemen	ted: Read as '	0'				
bit 4	SCKP: Synch	ronous Clock I	Polarity Select	bit			
	Asynchronous	<u>s mode</u> :					
	1 = Idle state 0 = Idle state	for transmit (T) for transmit (T)	X) is a low lev X) is a high lev	el (transmit da vel (transmit da	ta inverted) ata is non-invert	ed)	
	Synchronous	mode:					
	1 = Data is clo 0 = Data is clo	ocked on rising ocked on falling) edge of the c g edge of the (lock clock			
bit 3	BRG16: 16-bi	it Baud Rate G	enerator bit				
	1 = 16-bit Ba	ud Rate Gener	ator is used				
	0 = 8-bit Bau	d Rate Genera	itor is used				
bit 2	Unimplemen	ted: Read as '	0'				
bit 1	WUE: Wake-u	up Enable bit					
	Asynchronous	<u>s mode</u> :	6 11: 1				
	1 = Receiver	is waiting for a	falling edge. I	No character v	vill be received,	byte RCxIF wil	I be set. WUE
	0 = Receiver	is operating no	rmally				
	<u>Synchronous</u>	mode:	,				
	Don't care						
bit 0	ABDEN: Auto	-Baud Detect	Enable bit				
	Asynchronous	<u>s mode</u> :					
	1 = Auto-Bau	d Detect mode	e is enabled (c	lears when au	to-baud is comp	olete)	
	0 = Auto-Bau	d Detect mode	e is disabled				
	Don't care	<u>moue</u> .					

29.3 Register Definitions: FVR Control

R/W-0/0	R-q/q	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0	R/W-0/0
FVREN	FVRRDY ⁽¹⁾	TSEN ⁽³⁾	TSRNG ⁽³⁾	CDAFVR<1:0> ADFVR<1		R<1:0>	
bit 7							bit (
Legend:							
R = Readable bitW = Writable bitU = Unimplemented bit, read as '0'							
u = Bit is unc	hanged	x = Bit is unk	nown	-n/n = Value a	at POR and BO	R/Value at all c	other Resets
'1' = Bit is set	t	'0' = Bit is cle	ared	q = Value dep	pends on condit	ion	
bit 7	FVREN: Fixed 1 ⁼ Fixed Vol 0 ⁼ Fixed Vol	d Voltage Refe Itage Referenc Itage Referenc	rence Enable e is enabled e is disabled	bit			
bit 6	FVRRDY: Fix 1 = Fixed Vol 0 = Fixed Vol	ed Voltage Re Itage Referenc Itage Referenc	ference Ready e output is rea e output is not	/ Flag bit ⁽¹⁾ ady for use t ready or not e	nabled		
bit 5	TSEN: Temperation 1 = Temperation 0 = Temperation	erature Indicator ture Indicator i ture Indicator i	or Enable bit ⁽³⁾ s enabled s disabled)			
bit 4	TSRNG: Tem 1 = Vout = V 0 = Vout = V	perature Indica ′DD - 4V⊤ (Higł ′DD - 2V⊤ (Low	ator Range Se I Range) Range)	lection bit ⁽³⁾			
bit 3-2	CDAFVR<1:0 11 = Compar 10 = Compar 01 = Compar 00 = Compar	D>: Comparato ator FVR Buffe ator FVR Buffe ator FVR Buffe ator FVR Buffe	r FVR Buffer (er Gain is 4x, (er Gain is 2x, (er Gain is 1x, (er is off	Gain Selection 4.096V) ⁽²⁾ 2.048V) ⁽²⁾ 1.024V)	bits		
bit 1-0	ADFVR<1:0> 11 = ADC FV 10 = ADC FV 01 = ADC FV 00 = ADC FV	ADC FVR Burner R Buffer Gain R Buffer Gain R Buffer Gain R Buffer is off	iffer Gain Sele is 4x, (4.096V is 2x, (2.048V is 1x, (1.024V	ection bit ₎ (2))(2))			
Note 1: FV	/RRDY is always	sʻ1'.					

	EVECON, EVED VOLTAGE DEFEDENCE CONTROL DECISTED
REGISTER 23-1.	FURGON. FIXED VOLTAGE REFERENCE CONTROL REGISTER

2: Fixed Voltage Reference output cannot exceed VDD.

3: See Section 30.0 "Temperature Indicator Module" for additional information.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Register on page
FVRCON	FVREN	FVRRDY	TSEN	TSRNG	CDAFV	′R<1:0>	ADFVR<1:0>		479
ADCON0	ADON	ADCONT	_	ADCS	_	ADFM	—	ADGO	504
CMxNCH	—	_	—	—	—		CxNCH<2:0	>	525
CMxPCH	—	_	—		— CxPCH<2:0>			526	
DAC1CON1	_		_		DAC1R<4:0>				

TABLE 29-1: SUMMARY OF REGISTERS ASSOCIATED WITH FIXED VOLTAGE REFERENCE

Legend: — = Unimplemented location, read as '0'. Shaded cells are not used with the Fixed Voltage Reference.

REGISTER 32-13: ADCNT: ADC REPEAT COUNTER REGISTER

R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u	R/W-x/u
			ADCN	T<7:0>			
bit 7							bit 0
Legend:							
R = Readable	bit	W = Writable b	oit	U = Unimpler	nented bit, read	d as '0'	
u = Bit is unchanged x =		x = Bit is unkno	own	-n/n = Value a	at POR and BC	R/Value at all	other Resets
'1' = Bit is set		'0' = Bit is clea	red				

bit 7-0 **ADCNT<7:0>**: ADC Repeat Count bits Determines the number of times that the ADC is triggered before the threshold is checked when the computation is Low-pass Filter, Burst Average, or Average modes. See Table 32-2 for more details.

REGISTER 32-14: ADFLTRH: ADC FILTER HIGH BYTE REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x	
ADFLTR<15:8>								
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 7-0 ADFLTR<15:8>: ADC Filter Output Most Significant bits In Accumulate, Average, and Burst Average mode, this is equal to ADACC right shifted by the ADCRS bits of ADCON2. In LPF mode, this is the output of the Low-pass Filter.

REGISTER 32-15: ADFLTRL: ADC FILTER LOW BYTE REGISTER

R-x	R-x	R-x	R-x	R-x	R-x	R-x	R-x	
ADFLTR<7:0>								
bit 7 k								

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 7-0 ADFLTR<7:0>: ADC Filter Output Least Significant bits In Accumulate, Average, and Burst Average mode, this is equal to ADACC right shifted by the ADCRS bits of ADCON2. In LPF mode, this is the output of the Low-pass Filter.

© 2016-2017 Microchip Technology Inc.

36.0 INSTRUCTION SET SUMMARY

PIC18(L)F6xK40 devices incorporate the standard set of 75 PIC18 core instructions, as well as an extended set of eight new instructions, for the optimization of code that is recursive or that utilizes a software stack. The extended set is discussed later in this section.

36.1 Standard Instruction Set

The standard PIC18 instruction set adds many enhancements to the previous PIC^{\circledast} MCU instruction sets, while maintaining an easy migration from these PIC^{\circledast} MCU instruction sets. Most instructions are a single program memory word (16 bits), but there are four instructions that require two program memory locations.

Each single-word instruction is a 16-bit word divided into an opcode, which specifies the instruction type and one or more operands, which further specify the operation of the instruction.

The instruction set is highly orthogonal and is grouped into four basic categories:

- Byte-oriented operations
- **Bit-oriented** operations
- · Literal operations
- · Control operations

The PIC18 instruction set summary in Table 36-2 lists **byte-oriented**, **bit-oriented**, **literal** and **control** operations. Table 36-1 shows the opcode field descriptions.

Most byte-oriented instructions have three operands:

- 1. The file register (specified by 'f')
- 2. The destination of the result (specified by 'd')
- 3. The accessed memory (specified by 'a')

The file register designator 'f' specifies which file register is to be used by the instruction. The destination designator 'd' specifies where the result of the operation is to be placed. If 'd' is zero, the result is placed in the WREG register. If 'd' is one, the result is placed in the file register specified in the instruction.

All **bit-oriented** instructions have three operands:

- 1. The file register (specified by 'f')
- 2. The bit in the file register (specified by 'b')
- 3. The accessed memory (specified by 'a')

The bit field designator 'b' selects the number of the bit affected by the operation, while the file register designator 'f' represents the number of the file in which the bit is located. The literal instructions may use some of the following operands:

- A literal value to be loaded into a file register (specified by 'k')
- The desired FSR register to load the literal value into (specified by 'f')
- No operand required (specified by '—')

The control instructions may use some of the following operands:

- A program memory address (specified by 'n')
- The mode of the CALL or RETURN instructions (specified by 's')
- The mode of the table read and table write instructions (specified by 'm')
- No operand required (specified by '—')

All instructions are a single word, except for four double-word instructions. These instructions were made double-word to contain the required information in 32 bits. In the second word, the four MSbs are '1's. If this second word is executed as an instruction (by itself), it will execute as a NOP.

All single-word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles, with the additional instruction cycle(s) executed as a NOP.

The double-word instructions execute in two instruction cycles.

One instruction cycle consists of four oscillator periods. Thus, for an oscillator frequency of 4 MHz, the normal instruction execution time is 1 μ s. If a conditional test is true, or the program counter is changed as a result of an instruction, the instruction execution time is 2 μ s. Two-word branch instructions (if true) would take 3 μ s.

Figure 36-1 shows the general formats that the instructions can have. All examples use the convention 'nnh' to represent a hexadecimal number.

The Instruction Set Summary, shown in Table 36-2, lists the standard instructions recognized by the Microchip Assembler (MPASMTM).

Section 36.1.1 "Standard Instruction Set" provides a description of each instruction.

PIC18(L)F65/66K40

ADE	OWF	ADD W to (Indexed	o Indexe Literal (ed Offset m	iode)					
Synt	ax:	ADDWF	[k] {,d}							
Oper	ands:	$\begin{array}{l} 0 \leq k \leq 95 \\ d \in [0,1] \end{array}$	$\begin{array}{l} 0 \leq k \leq 95 \\ d \ \in \ [0,1] \end{array}$							
Oper	ration:	(W) + ((FS	(W) + ((FSR2) + k) \rightarrow dest							
Statu	is Affected:	N, OV, C, I	N, OV, C, DC, Z							
Enco	oding:	0010	01d0	kkkk	kkkk					
Desc	cription:	The conter contents o FSR2, offs If 'd' is '0', is '1', the r register 'f'	The contents of W are added to the contents of the register indicated by FSR2, offset by the value 'k'. If 'd' is '0', the result is stored in W. If 'd' is '1', the result is stored back in register 'f' (default).							
Word	ds:	1								
Cycle	es:	1								
QC	ycle Activity:									
	Q1	Q2	Q3	1	Q4					
	Decode	Read 'k'	Proce Dat	ess \ a de	Write to estination					
Exar	nple:	ADDWF	[OFST]	, 0						
	Before Instruct	ion								
W OFST FSR2 Contents of 0A2Ch		= = =	17h 2Ch 0A00h 20h	1						
	After Instructio	n								
	W	=	37h							
	of 0A2Ch	=	20h							

BSF		Bit Set (Index	t Ind ed I	dexed Literal	Offse	ət m	ode)		
Synta	ax:	BSF [k	k], b						
Operands:		$0 \le f \le 9$ $0 \le b \le 1$	$\begin{array}{l} 0 \leq f \leq 95 \\ 0 \leq b \leq 7 \end{array}$						
Oper	ration:	$1 \rightarrow ((F$	SR2	<u>2)</u> + k) <b< td=""><td>></td><td></td><td></td></b<>	>				
Statu	is Affected:	None							
Enco	oding:	1000		bbb0	kk}	ck	kkkk		
Description:		Bit 'b' of offset b	Bit 'b' of the register indicated by FSR2, offset by the value 'k', is set.						
Word	ds:	1							
Cycle	es:	1							
QC	ycle Activity:								
	Q1	Q2		Q3		Q4			
	Decode	Read register	'f'	Proce Data	Process Data		Vrite to stination		
<u>Exar</u>	nple:	BSF	[FLAG_O	FST]	, 7			
	Before Instruc	tion							
	FLAG_OFS FSR2		=	0Ah 0A00h	n				
	of 0A0Ah	1	=	55h					
	After Instructio	n							
	of 0A0Ah		=	D5h					

SETF	Set Indexed (Indexed Literal Offset mode)							
Syntax:	SETF [k]							
Operands:	$0 \leq k \leq 95$							
Operation:	$FFh \rightarrow ((FS))$	SR2) + k)						
Status Affected:	None							
Encoding:	0110	1000	kkk	k	kkkk			
Description:	The content FSR2, offse	The contents of the register indicated by FSR2, offset by 'k', are set to FFh.						
Words:	1							
Cycles:	1	1						
Q Cycle Activity:								
Q1	Q2	Q3		Q4				
Decode	Read 'k'	Process Data		Write register				
Example:	SETF [OFST]						
Before Instruct	ion							
OFST FSR2 Contents	= 2C = 0A	h .00h						
of 0A2Ch After Instruction	= 00 n	h						

Contents of 0A2Ch

= FFh