



Welcome to [E-XFL.COM](https://www.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	I <sup>2</sup> C, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	53
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	1.8V ~ 5.5V
Data Converters	A/D 16x12b
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	<a href="https://www.e-xfl.com/product-detail/microchip-technology/pic18f65k90t-i-pt">https://www.e-xfl.com/product-detail/microchip-technology/pic18f65k90t-i-pt</a>

# PIC18F87K90 FAMILY

**TABLE 1-4: PIC18F8XK90 PINOUT I/O DESCRIPTIONS (CONTINUED)**

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
RB0/INT0/SEG30/FLT0	58			PORTB is a bidirectional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs.
RB0		I/O	TTL	Digital I/O.
INT0		I	ST	External Interrupt 0.
SEG30		O	Analog	SEG30 output for LCD.
FLT0		I	ST	Enhanced PWM Fault input for ECCP1/2/3.
RB1/INT1/SEG8	57			
RB1		I/O	TTL	Digital I/O.
INT1		I	ST	External Interrupt 1.
SEG8		O	Analog	SEG8 output for LCD.
RB2/INT2/SEG9/CTED1	56			
RB2		I/O	TTL	Digital I/O.
INT2		I	ST	External Interrupt 2.
SEG9		O	Analog	SEG9 output for LCD.
CTED1		I	ST	CTMU Edge 1 input.
RB3/INT3/SEG10/CTED2/ECCP2/P2A	55			
RB3		I/O	TTL	Digital I/O.
INT3		I	ST	External Interrupt 3.
SEG10		O	Analog	SEG10 output for LCD.
CTED2		I	ST	CTMU Edge 2 input.
ECCP2		I/O	ST	Capture 2 input/Compare 2 output/PWM2 output.
P2A		O	ST	Enhanced PWM2 Output A.
RB4/KBI0/SEG11	54			
RB4		I/O	TTL	Digital I/O.
KBI0		I	TTL	Interrupt-on-change pin.
SEG11		O	Analog	SEG11 output for LCD.
RB5/KBI1/SEG29/T3CKI/T1G	53			
RB5		I/O	TTL	Digital I/O.
KBI1		I	TTL	Interrupt-on-change pin.
SEG29		O	Analog	SEG29 output for LCD.
T3CKI		I	ST	Timer3 clock input.
T1G		I	ST	Timer1 external clock gate input.

**Legend:** TTL = TTL compatible input CMOS = CMOS compatible input or output  
ST = Schmitt Trigger input with CMOS levels Analog = Analog input  
I = Input O = Output  
P = Power OD = Open-Drain (no P diode to VDD)  
I<sup>2</sup>C™ = I<sup>2</sup>C/SMBus

- Note 1:** Default assignment for ECCP2 when the CCP2MX Configuration bit is set.  
**Note 2:** Alternate assignment for ECCP2 when the CCP2MX Configuration bit is cleared.  
**Note 3:** Not available on PIC18F65K90 and PIC18F85K90 devices.  
**Note 4:** The CCP6, CCP7, CCP8 and CCP9 pin placement depends on the ECCPMX Configuration bit setting.

# PIC18F87K90 FAMILY

## REGISTER 3-1: OSCCON: OSCILLATOR CONTROL REGISTER (CONTINUED)

- bit 2      **HFIOFS:** INTOSC Frequency Stable bit  
1 = HF-INTOSC oscillator frequency is stable  
0 = HF-INTOSC oscillator frequency is not stable
- bit 1-0    **SCS<1:0>:** System Clock Select bits<sup>(4)</sup>  
1x = Internal oscillator block (LF-INTOSC, MF-INTOSC or HF-INTOSC)  
01 = SOSC oscillator  
00 = Default primary oscillator (OSC1/OSC2 or HF-INTOSC with or without PLL; defined by the OSC<3:0> Configuration bits, CONFIG1H<3:0>.)

- Note 1:** Reset state depends on the state of the IESO Configuration bit (CONFIG1H<7>).  
**2:** Modifying these bits will cause an immediate clock frequency switch if the internal oscillator is providing the device clocks.  
**3:** Source selected by the INTSRC bit (OSCTUNE<7>).  
**4:** Modifying these bits will cause an immediate clock source switch.  
**5:** INTSRC = OSCTUNE<7> and MFIOSEL = OSCCON2<0>.

## REGISTER 3-2: OSCCON2: OSCILLATOR CONTROL REGISTER 2

U-0	R-0	U-0	U-0	R/W-0	U-0	R-x	R/W-0
—	SOSCRUN	—	—	SOSCGO	—	MFIOFS	MFIOSEL
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 7      **Unimplemented:** Read as '0'
- bit 6      **SOSCRUN:** SOSC Run Status bit  
1 = System clock comes from a secondary SOSC  
0 = System clock comes from an oscillator other than SOSC
- bit 5-4    **Unimplemented:** Read as '0'
- bit 3      **SOSCGO:** Oscillator Start Control bit  
1 = Oscillator is running, even if no other sources are requesting it  
0 = Oscillator is shut off if no other sources are requesting it (When the SOSC is selected to run from a digital clock input, rather than an external crystal, this bit has no effect.)
- bit 2      **Unimplemented:** Read as '0'
- bit 1      **MFIOFS:** MF-INTOSC Frequency Stable bit  
1 = MF-INTOSC is stable  
0 = MF-INTOSC is not stable
- bit 0      **MFIOSEL:** MF-INTOSC Select bit  
1 = MF-INTOSC is used in place of HF-INTOSC frequencies of 500 kHz, 250 kHz and 31.25 kHz  
0 = MF-INTOSC is not used

## 4.0 POWER-MANAGED MODES

The PIC18F87K90 family of devices offers a total of seven operating modes for more efficient power management. These modes provide a variety of options for selective power conservation in applications where resources may be limited (such as battery-powered devices).

There are three categories of power-managed modes:

- Run modes
- Idle modes
- Sleep mode

There is an Ultra Low-Power Wake-up (ULPWU) for waking from the Sleep mode.

These categories define which portions of the device are clocked, and sometimes, at what speed. The Run and Idle modes may use any of the three available clock sources (primary, secondary or internal oscillator block). The Sleep mode does not use a clock source.

The ULPWU mode, on the RA0 pin, enables a slow falling voltage to generate a wake-up, even from Sleep, without excess current consumption. (See **Section 4.7 “Ultra Low-Power Wake-up”**.)

The power-managed modes include several power-saving features offered on previous PIC® devices. One is the clock switching feature, offered in other PIC18 devices. This feature allows the controller to use the SOSC oscillator instead of the primary one. Another power-saving feature is Sleep mode, offered by all PIC devices, where all device clocks are stopped.

### 4.1 Selecting Power-Managed Modes

Selecting a power-managed mode requires two decisions:

- Will the CPU be clocked or not
- What will be the clock source

The IDLEN bit (OSCCON<7>) controls CPU clocking, while the SCS<1:0> bits (OSCCON<1:0>) select the clock source. The individual modes, bit settings, clock sources and affected modules are summarized in Table 4-1.

#### 4.1.1 CLOCK SOURCES

The SCS<1:0> bits select one of three clock sources for power-managed modes. Those sources are:

- The primary clock, as defined by the OSC<3:0> Configuration bits
- The secondary clock (the SOSC oscillator)
- The internal oscillator block (for LF-INTOSC modes)

#### 4.1.2 ENTERING POWER-MANAGED MODES

Switching from one power-managed mode to another begins by loading the OSCCON register. The SCS<1:0> bits select the clock source and determine which Run or Idle mode is used. Changing these bits causes an immediate switch to the new clock source, assuming that it is running. The switch may also be subject to clock transition delays. These considerations are discussed in **Section 4.1.3 “Clock Transitions and Status Indicators”** and subsequent sections.

Entering the power-managed Idle or Sleep modes is triggered by the execution of a SLEEP instruction. The actual mode that results depends on the status of the IDLEN bit.

Depending on the current and impending mode, a change to a power-managed mode does not always require setting all of the previously discussed bits. Many transitions can be done by changing the oscillator select bits, or changing the IDLEN bit, prior to issuing a SLEEP instruction. If the IDLEN bit is already configured as desired, it may only be necessary to perform a SLEEP instruction to switch to the desired mode.

**TABLE 4-1: POWER-MANAGED MODES**

Mode	OSCCON Bits		Module Clocking		Available Clock and Oscillator Source
	IDLEN<7> <sup>(1)</sup>	SCS<1:0>	CPU	Peripherals	
Sleep	0	N/A	Off	Off	None – All clocks are disabled
PRI_RUN	N/A	00	Clocked	Clocked	Primary – XT, LP, HS, EC, RC and PLL modes. This is the normal, Full-Power Execution mode.
SEC_RUN	N/A	01	Clocked	Clocked	Secondary – SOSC Oscillator
RC_RUN	N/A	1x	Clocked	Clocked	Internal oscillator block <sup>(2)</sup>
PRI_IDLE	1	00	Off	Clocked	Primary – LP, XT, HS, RC, EC
SEC_IDLE	1	01	Off	Clocked	Secondary – SOSC oscillator
RC_IDLE	1	1x	Off	Clocked	Internal oscillator block <sup>(2)</sup>

**Note 1:** IDLEN reflects its value when the SLEEP instruction is executed.

**2:** Includes INTOSC (HF-INTOSC and MG-INTOSC) and INTOSC postscaler, as well as the LF-INTISC source.

# PIC18F87K90 FAMILY

---

NOTES:

# PIC18F87K90 FAMILY

## 6.3 Data Memory Organization

**Note:** The operation of some aspects of data memory are changed when the PIC18 extended instruction set is enabled. See **Section 6.6 “Data Memory and the Extended Instruction Set”** for more information.

The data memory in PIC18 devices is implemented as static RAM. Each register in the data memory has a 12-bit address, allowing up to 4,096 bytes of data memory. The memory space is divided into as many as 16 banks that contain 256 bytes each. PIC18FX6K90 and PIC18FX7K90 devices implement all 16 complete banks, for a total of 4 Kbytes. PIC18FX5K90 devices implement only the first eight complete banks, for a total of 2 Kbytes.

Figure 6-6 and Figure 6-7 show the data memory organization for the devices.

The data memory contains Special Function Registers (SFRs) and General Purpose Registers (GPRs). The SFRs are used for control and status of the controller and peripheral functions, while GPRs are used for data storage and scratchpad operations in the user's application. Any read of an unimplemented location will read as '0's.

The instruction set and architecture allow operations across all banks. The entire data memory may be accessed by Direct, Indirect or Indexed Addressing modes. Addressing modes are discussed later in this section.

To ensure that commonly used registers (select SFRs and select GPRs) can be accessed in a single cycle, PIC18 devices implement an Access Bank. This is a 256-byte memory space that provides fast access to select SFRs and the lower portion of GPR Bank 0 without using the Bank Select Register. For details on the Access RAM, see **Section 6.3.2 “Access Bank”**.

### 6.3.1 BANK SELECT REGISTER

Large areas of data memory require an efficient addressing scheme to make possible rapid access to any address. Ideally, this means that an entire address does not need to be provided for each read or write operation. For PIC18 devices, this is accomplished with a RAM banking scheme. This divides the memory space into 16 contiguous banks of 256 bytes. Depending on the instruction, each location can be addressed directly by its full 12-bit address, or an 8-bit, low-order address and a 4-bit Bank Pointer.

Most instructions in the PIC18 instruction set make use of the Bank Pointer, known as the Bank Select Register (BSR). This SFR holds the four Most Significant bits of a location's address. The instruction itself includes the eight Least Significant bits. Only the four lower bits of the BSR are implemented (BSR<3:0>). The upper four bits are unused, always read as '0' and cannot be written to. The BSR can be loaded directly by using the `MOVLB` instruction.

The value of the BSR indicates the bank in data memory. The eight bits in the instruction show the location in the bank and can be thought of as an offset from the bank's lower boundary. The relationship between the BSR's value and the bank division in data memory is shown in Figure 6-7.

Since up to 16 registers may share the same low-order address, the user must always be careful to ensure that the proper bank is selected before performing a data read or write. For example, writing what should be program data to an 8-bit address of F9h while the BSR is 0Fh, will end up resetting the Program Counter.

While any bank can be selected, only those banks that are actually implemented can be read or written to. Writes to unimplemented banks are ignored, while reads from unimplemented banks will return '0's. Even so, the STATUS register will still be affected as if the operation was successful. The data memory map in Figure 6-6 indicates which banks are implemented.

In the core PIC18 instruction set, only the `MOVFF` instruction fully specifies the 12-bit address of the source and target registers. When this instruction executes, it ignores the BSR completely. All other instructions include only the low-order address as an operand and must use either the BSR or the Access Bank to locate their target registers.

# PIC18F87K90 FAMILY

---

## 6.4.3.2 FSR Registers and POSTINC, POSTDEC, PREINC and PLUSW

In addition to the INDF operand, each FSR register pair also has four additional indirect operands. Like INDF, these are “virtual” registers that cannot be indirectly read or written to. Accessing these registers actually accesses the associated FSR register pair, but also performs a specific action on its stored value.

These operands are:

- POSTDEC – Accesses the FSR value, then automatically decrements it by ‘1’ afterwards
- POSTINC – Accesses the FSR value, then automatically increments it by ‘1’ afterwards
- PREINC – Increments the FSR value by ‘1’, then uses it in the operation
- PLUSW – Adds the signed value of the W register (range of -127 to 128) to that of the FSR and uses the new value in the operation

In this context, accessing an INDF register uses the value in the FSR registers without changing them. Similarly, accessing a PLUSW register gives the FSR value, offset by the value in the W register – with neither value actually changed in the operation. Accessing the other virtual registers changes the value of the FSR registers.

Operations on the FSRs with POSTDEC, POSTINC and PREINC affect the entire register pair. Rollovers of the FSRnL register, from FFh to 00h, carry over to the FSRnH register. On the other hand, results of these operations do not change the value of any flags in the STATUS register (for example, Z, N and OV bits).

The PLUSW register can be used to implement a form of Indexed Addressing in the data memory space. By manipulating the value in the W register, users can reach addresses that are fixed offsets from pointer addresses. In some applications, this can be used to implement some powerful program control structure, such as software stacks, inside of data memory.

## 6.4.3.3 Operations by FSRs on FSRs

Indirect Addressing operations that target other FSRs or virtual registers represent special cases. For example, using an FSR to point to one of the virtual registers will not result in successful operations.

As a specific case, assume that the FSR0H:FSR0L registers contain FE7h, the address of INDF1. Attempts to read the value of the INDF1, using INDF0 as an operand, will return 00h. Attempts to write to INDF1, using INDF0 as the operand, will result in a NOP.

On the other hand, using the virtual registers to write to an FSR pair may not occur as planned. In these cases, the value will be written to the FSR pair, but without any incrementing or decrementing. Thus, writing to INDF2 or POSTDEC2 will write the same value to the FSR2H:FSR2L.

Since the FSRs are physical registers mapped in the SFR space, they can be manipulated through all direct operations. Users should proceed cautiously when working on these registers, however, particularly if their code uses Indirect Addressing.

Similarly, operations by Indirect Addressing are generally permitted on all other SFRs. Users should exercise the appropriate caution, so that they do not inadvertently change settings that might affect the operation of the device.

## 6.5 Program Memory and the Extended Instruction Set

The operation of program memory is unaffected by the use of the extended instruction set.

Enabling the extended instruction set adds five additional two-word commands to the existing PIC18 instruction set: ADDFSR, CALLW, MOVSF, MOVSS and SUBFSR. These instructions are executed as described in **Section 6.2.4 “Two-Word Instructions”**.

# PIC18F87K90 FAMILY

**TABLE 11-7: PORTD FUNCTIONS**

Pin Name	Function	TRIS Setting	I/O	I/O Type	Description
RD0/SEG0/CTPLS	RD0	0	O	DIG	LATD<0> data output.
		1	I	ST	PORTD<0> data input.
	SEG0	1	O	ANA	LCD Segment 0 output; disables all other pin functions.
	CTPLS	x	O	DIG	CTMU pulse generator output.
RD1/SEG1/T5CKI/T7G	RD1	0	O	DIG	LATD<1> data output.
		1	I	ST	PORTD<1> data input.
	SEG1	1	O	ANA	LCD Segment 1 output; disables all other pin functions.
	T5CKI	x	I	ST	Timer5 clock input.
	T7G	x	I	ST	Timer7 external clock gate input.
RD2/SEG2	RD2	0	O	DIG	LATD<2> data output.
		1	I	ST	PORTD<2> data input.
	SEG2	1	O	ANA	LCD Segment 2 output; disables all other pin functions.
RD3/SEG3	RD3	0	O	DIG	LATD<3> data output.
		1	I	ST	PORTD<3> data input.
	SEG3	1	O	ANA	LCD Segment 3 output; disables all other pin functions.
RD4/SEG4/SDO2	RD4	0	O	DIG	LATD<4> data output.
		1	I	ST	PORTD<4> data input.
	SEG4	1	O	ANA	LCD Segment 4 output; disables all other pin functions.
	SDO2	0	P	DOG	SPI data output (MSSP module).
RD5/SEG5/SDI2/SDA2	RD5	0	O	DIG	LATD<5> data output.
		1	I	ST	PORTD<5> data input.
	SEG5	1	O	ANA	LCD Segment 5 output; disables all other pin functions.
	SDI2	1	I	ST	SPI data input (MSSP module).
	SDA2	0	O	I <sup>2</sup> C	I <sup>2</sup> C™ data input (MSSP module). Input type depends on module setting.
		1	I	ANA	LCD Segment 5 output; disables all other pin functions.
RD6/SEG6/SCK2/SCL2	RD6	0	O	DIG	LATD<6> data output.
		1	I	ST	PORTD<6> data input.
	SEG6	1	O	ANA	LCD Segment 6 output; disables all other pin functions.
	SCK2	0	O	DIG	SPI clock output (MSSP module); takes priority over port data.
		1	I	ST	SPI clock input (MSSP module).
	SCL2	0	O	DIG	I <sup>2</sup> C clock output (MSSP module); takes priority over port data.
		1	I	I <sup>2</sup> C	I <sup>2</sup> C clock input (MSSP module). Input type depends on module setting.
RD7/SEG7/SS2	RD7	0	O	DIG	LATD<7> data output.
		1	I	ST	PORTD<7> data input.
	SEG7	1	I	ANA	LCD Segment 7 output; disables all other pin functions.
	SS2	1	I	TTL	Slave select input for MSSP module.

**Legend:** O = Output, I = Input, ANA = Analog Signal, DIG = Digital Output, ST = Schmitt Trigger Buffer Input, I<sup>2</sup>C = I<sup>2</sup>C Buffer Input, x = Don't care (TRIS bit does not affect port direction or is overridden for this option).



# PIC18F87K90 FAMILY

TABLE 11-8: SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	78
LATD	LATD7	LATD6	LATD5	LATD4	LATD3	LATD2	LATD1	LATD0	78
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	78
LCDSE0	SE07	SE06	SE05	SE04	SE03	SE02	SE01	SE00	83
PADCFG1	RDPU	REPU	RJPU <sup>(1)</sup>	—	—	RTSECSEL1	RTSECSEL0	—	80

**Legend:** Shaded cells are not used by PORTD.

**Note 1:** This bit is not available in 64-pin devices.

# PIC18F87K90 FAMILY

## 11.10 PORTJ, TRISJ and LATJ Registers

**Note:** PORTJ is available only on 80-pin devices.

PORTJ is an 8-bit wide, bidirectional port. The corresponding Data Direction and Output Latch registers are TRISJ and LATJ.

All pins on PORTJ are implemented with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output.

**Note:** These pins are configured as digital inputs on any device Reset.

All PORTJ pins, except RJ0, are multiplexed with LCD segment drives controlled by the LCDSE4 register. I/O port functions are only available on these pins when the segments are disabled.

Each of the PORTJ pins has a weak internal pull-up. A single control bit can turn off all the pull-ups. This is performed by clearing bit RJPU (PADCFG1<5>). The weak pull-up is automatically turned off when the port pin is configured as an output. The pull-ups are disabled on any device Reset.

### EXAMPLE 11-9: INITIALIZING PORTJ

```
CLRF    PORTJ    ; Initialize PORTJ by
                  ; clearing output latches
CLRF    LATJ     ; Alternate method
                  ; to clear output latches
MOVLW   0CFh     ; Value used to
                  ; initialize data
                  ; direction
MOVWF   TRISJ    ; Set RJ3:RJ0 as inputs
                  ; RJ5:RJ4 as output
                  ; RJ7:RJ6 as inputs
```

# PIC18F87K90 FAMILY

## 13.1 Timer1 Gate Control Register

The Timer1 Gate Control register (T1GCON), displayed in Register 13-2, is used to control the Timer1 gate.

**REGISTER 13-2: T1GCON: TIMER1 GATE CONTROL REGISTER<sup>(1)</sup>**

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-x	R/W-0	R/W-0
TMR1GE	T1GPOL	T1GTM	T1GSPM	T1GGO/T1DONE	T1GVAL	T1GSS1	T1GSS0
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 7      **TMR1GE:** Timer1 Gate Enable bit  
If TMR1ON = 0:  
This bit is ignored.  
If TMR1ON = 1:  
1 = Timer1 counting is controlled by the Timer1 gate function  
0 = Timer1 counts regardless of the Timer1 gate function
- bit 6      **T1GPOL:** Timer1 Gate Polarity bit  
1 = Timer1 gate is active-high (Timer1 counts when gate is high)  
0 = Timer1 gate is active-low (Timer1 counts when gate is low)
- bit 5      **T1GTM:** Timer1 Gate Toggle Mode bit  
1 = Timer1 Gate Toggle mode is enabled  
0 = Timer1 Gate Toggle mode is disabled and toggle flip-flop is cleared  
Timer1 gate flip-flop toggles on every rising edge.
- bit 4      **T1GSPM:** Timer1 Gate Single Pulse Mode bit  
1 = Timer1 Gate Single Pulse mode is enabled and is controlling Timer1 gate  
0 = Timer1 Gate Single Pulse mode is disabled
- bit 3      **T1GGO/T1DONE:** Timer1 Gate Single Pulse Acquisition Status bit  
1 = Timer1 gate single pulse acquisition is ready, waiting for an edge  
0 = Timer1 gate single pulse acquisition has completed or has not been started  
This bit is automatically cleared when T1GSPM is cleared.
- bit 2      **T1GVAL:** Timer1 Gate Current State bit  
Indicates the current state of the Timer1 gate that could be provided to TMR1H:TMR1L; unaffected by the Timer1 Gate Enable (TMR1GE) bit.
- bit 1-0    **T1GSS<1:0>:** Timer1 Gate Source Select bits  
11 = Comparator 2 output  
10 = Comparator 1 output  
01 = TMR2 to match PR2 output  
00 = Timer1 gate pin

**Note 1:** Programming the T1GCON register prior to T1CON is recommended.

# PIC18F87K90 FAMILY

## REGISTER 17-11: HOUR: HOUR VALUE REGISTER<sup>(1)</sup>

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
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 7-6 **Unimplemented:** Read as '0'

bit 5-4 **HRTEN<1:0>:** Binary Coded Decimal Value of Hour's Tens Digit bits  
Contains a value from 0 to 2.

bit 3-0 **HRONE<3:0>:** Binary Coded Decimal Value of Hour's Ones Digit bits  
Contains a value from 0 to 9.

**Note 1:** A write to this register is only allowed when RTCWREN = 1.

## REGISTER 17-12: MINUTE: MINUTE VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
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 7 **Unimplemented:** Read as '0'

bit 6-4 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute's Tens Digit bits  
Contains a value from 0 to 5.

bit 3-0 **MINONE<3:0>:** Binary Coded Decimal Value of Minute's Ones Digit bits  
Contains a value from 0 to 9.

## REGISTER 17-13: SECOND: SECOND VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
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

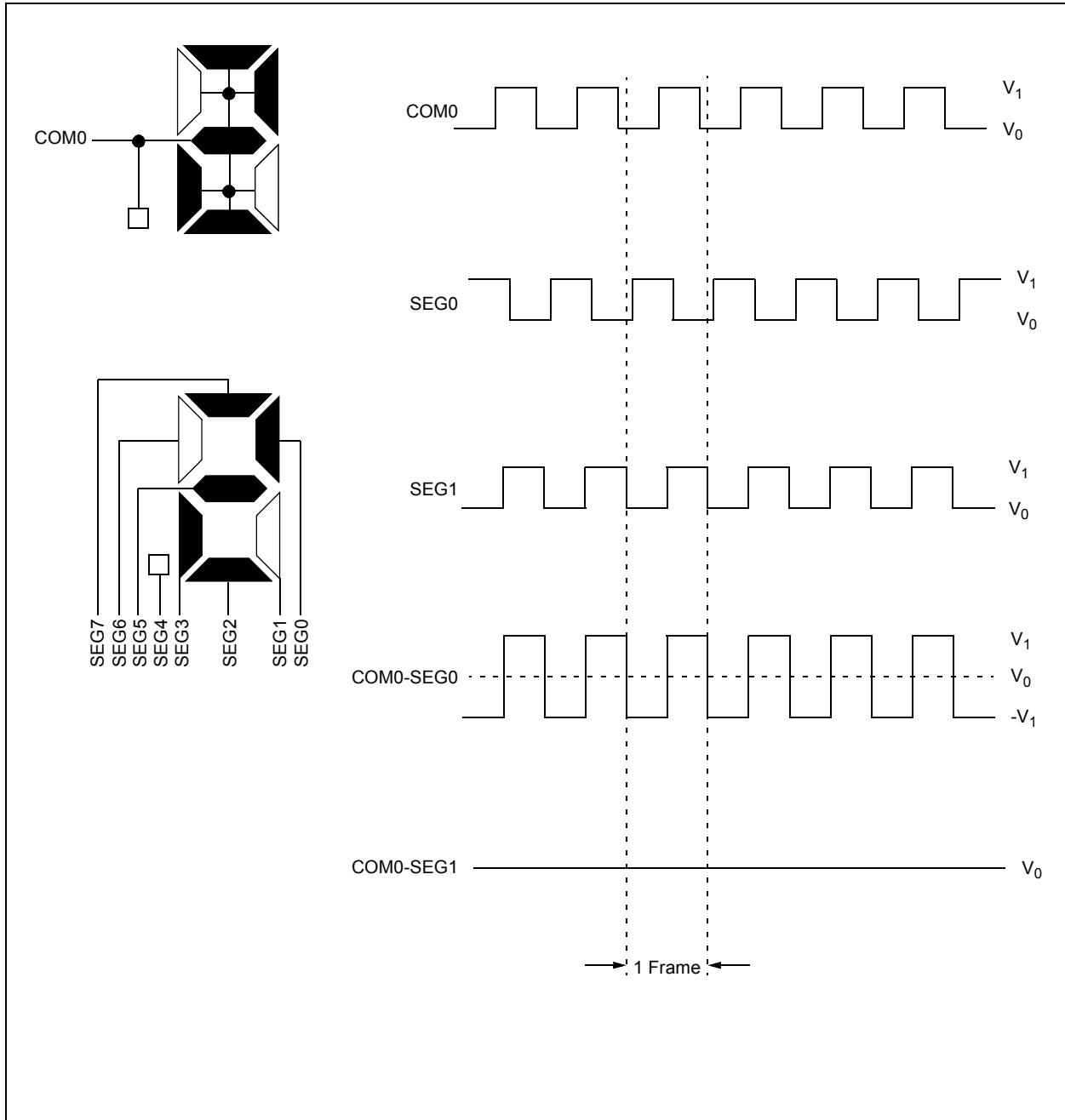
x = Bit is unknown

bit 7 **Unimplemented:** Read as '0'

bit 6-4 **SECTEN<2:0>:** Binary Coded Decimal Value of Second's Tens Digit bits  
Contains a value from 0 to 5.

bit 3-0 **SECONE<3:0>:** Binary Coded Decimal Value of Second's Ones Digit bits  
Contains a value from 0 to 9.

**FIGURE 20-7: TYPE-A/TYPE-B WAVEFORMS IN STATIC DRIVE**



# PIC18F87K90 FAMILY

**TABLE 21-2: REGISTERS ASSOCIATED WITH SPI OPERATION**

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on Page:
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	75
PIR1	—	ADIF	RC1IF	TX1IF	SSP1IF	TMR1GIF	TMR2IF	TMR1IF	77
PIE1	—	ADIE	RC1IE	TX1IE	SSP1IE	TMR1GIE	TMR2IE	TMR1IE	77
IPR1	—	ADIP	RC1IP	TX1IP	SSP1IP	TMR1GIP	TMR2IP	TMR1IP	77
PIR2	OSCFIF	—	SSP2IF	BCL2IF	BCL1IF	HLVDIF	TMR3IF	TMR3GIF	77
PIE2	OSCFIE	—	SSP2IE	BCL2IE	BCL1IE	HLVDIE	TMR3IE	TMR3GIE	77
IPR2	OSCFIP	—	SSP2IP	BCL2IP	BCL1IP	HLVDIP	TMR3IP	TMR3GIP	77
TRISC	TRISC7	TRISC6	TRISC5	TRISC4	TRISC3	TRISC2	TRISC1	TRISC0	78
TRISD	TRISD7	TRISD6	TRISD5	TRISD4	TRISD3	TRISD2	TRISD1	TRISD0	78
TRISF	TRISF7	TRISF6	TRISF5	TRISF4	TRISF3	TRISF2	TRISF1	—	78
SSP1BUF	MSSP1 Receive Buffer/Transmit Register								82
SSP1CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	76
SSP1CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	76
SSP1STAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	76
SSP2CON1	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	82
SSP2CON2	GCEN	ACKSTAT	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	83
SSP2STAT	SMP	CKE	D/ $\bar{A}$	P	S	R/ $\bar{W}$	UA	BF	82
SSP2BUF	MSSP2 Receive Buffer/Transmit Register								82
ODCON3	U2OD	U1OD	—	—	—	—	—	CTMUDS	81

**Legend:** Shaded cells are not used by the MSSP module in SPI mode.

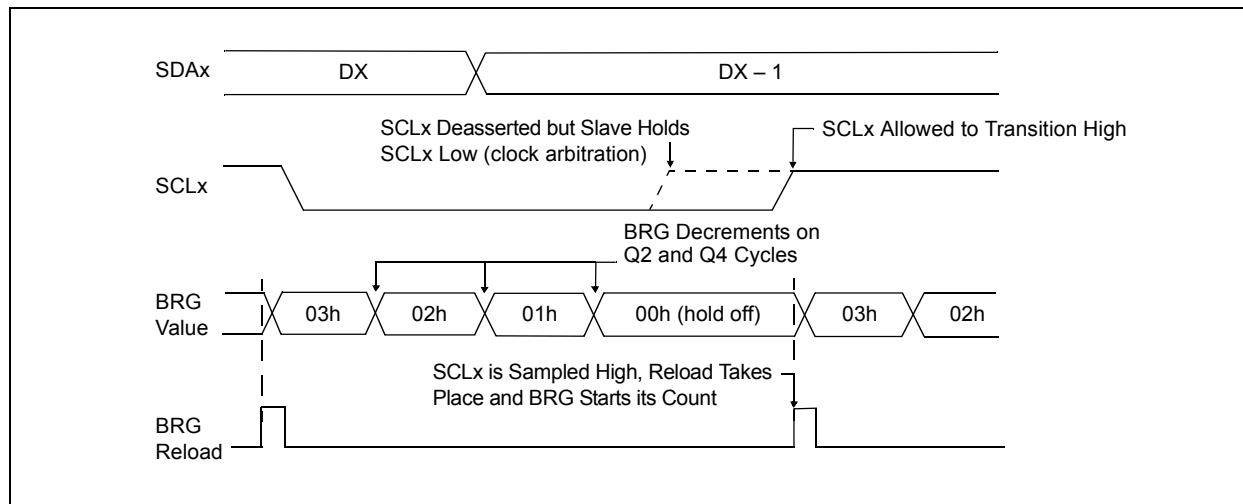
# PIC18F87K90 FAMILY

## 21.4.7.2 Clock Arbitration

Clock arbitration occurs when the master, during any receive, transmit or Repeated Start/Stop condition, deasserts the SCLx pin (SCLx allowed to float high). When the SCLx pin is allowed to float high, the Baud Rate Generator (BRG) is suspended from counting until the SCLx pin is actually sampled high. When the

SCLx pin is sampled high, the Baud Rate Generator is reloaded with the contents of SSPxADD<6:0> and begins counting. This ensures that the SCLx high time will always be at least one BRG rollover count in the event that the clock is held low by an external device (Figure 21-20).

**FIGURE 21-20: BAUD RATE GENERATOR TIMING WITH CLOCK ARBITRATION**



# PIC18F87K90 FAMILY

## REGISTER 23-6: ADRESH: A/D RESULT HIGH BYTE REGISTER, RIGHT JUSTIFIED (ADFM = 1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADSGN	ADSGN	ADSGN	ADSGN	ADRES11	ADRES10	ADRES9	ADRES8
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 7-4      **ADSGN:** A/D Result Sign bits

1 = A/D result is negative

0 = A/D result is positive

bit 3-0      **ADRES<11:8>:** A/D Result High Byte bits

## REGISTER 23-7: ADRESL: A/D RESULT LOW BYTE REGISTER, RIGHT JUSTIFIED (ADFM = 1)

R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
ADRES7	ADRES6	ADRES5	ADRES4	ADRES3	ADRES2	ADRES1	ADRES0
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 7-0      **ADRES<7:0>:** A/D Result Low Byte bits



# PIC18F87K90 FAMILY

**TABLE 29-1: OPCODE FIELD DESCRIPTIONS**

Field	Description
a	RAM access bit: a = 0: RAM location in Access RAM (BSR register is ignored) a = 1: RAM bank is specified by BSR register
bbb	Bit address within an 8-bit file register (0 to 7).
BSR	Bank Select Register. Used to select the current RAM bank.
C, DC, Z, OV, N	ALU Status bits: <b>C</b> arry, <b>D</b> igit <b>C</b> arry, <b>Z</b> ero, <b>O</b> verflow, <b>N</b> egative.
d	Destination select bit: d = 0: store result in WREG d = 1: store result in file register f
dest	Destination: either the WREG register or the specified register file location.
f	8-bit register file address (00h to FFh), or 2-bit FSR designator (0h to 3h).
f <sub>s</sub>	12-bit register file address (000h to FFFh). This is the source address.
f <sub>d</sub>	12-bit register file address (000h to FFFh). This is the destination address.
GIE	Global Interrupt Enable bit.
k	Literal field, constant data or label (may be either an 8-bit, 12-bit or a 20-bit value).
label	Label name.
mm	The mode of the TBLPTR register for the table read and table write instructions. Only used with table read and table write instructions:
*	No Change to register (such as TBLPTR with table reads and writes)
*+	Post-Increment register (such as TBLPTR with table reads and writes)
*-	Post-Decrement register (such as TBLPTR with table reads and writes)
++	Pre-Increment register (such as TBLPTR with table reads and writes)
n	The relative address (2's complement number) for relative branch instructions or the direct address for Call/Branch and Return instructions.
PC	Program Counter.
PCL	Program Counter Low Byte.
PCH	Program Counter High Byte.
PCLATH	Program Counter High Byte Latch.
PCLATU	Program Counter Upper Byte Latch.
PD	Power-Down bit.
PRODH	Product of Multiply High Byte.
PRODL	Product of Multiply Low Byte.
s	Fast Call/Return mode select bit: s = 0: do not update into/from shadow registers s = 1: certain registers loaded into/from shadow registers (Fast mode)
TBLPTR	21-bit Table Pointer (points to a Program Memory location).
TABLAT	8-bit Table Latch.
TO	Time-out bit.
TOS	Top-of-Stack.
u	Unused or Unchanged.
WDT	Watchdog Timer.
WREG	Working register (accumulator).
x	Don't care ('0' or '1'). The assembler will generate code with x = 0. It is the recommended form of use for compatibility with all Microchip software tools.
z <sub>s</sub>	7-bit offset value for Indirect Addressing of register files (source).
z <sub>d</sub>	7-bit offset value for Indirect Addressing of register files (destination).
{ }	Optional argument.
[text]	Indicates an Indexed Address.
(text)	The contents of text.
[expr]<n>	Specifies bit n of the register indicated by the pointer expr.
→	Assigned to.
< >	Register bit field.
∈	In the set of.
italics	User-defined term (font is Courier New).

# PIC18F87K90 FAMILY

CALLW		Subroutine Call Using WREG						
Syntax:	CALLW							
Operands:	None							
Operation:	(PC + 2) → TOS, (W) → PCL, (PCLATH) → PCH, (PCLATU) → PCU							
Status Affected:	None							
Encoding:	<table border="1"><tr><td>0000</td><td>0000</td><td>0001</td><td>0100</td></tr></table>				0000	0000	0001	0100
0000	0000	0001	0100					
Description	<p>First, the return address (PC + 2) is pushed onto the return stack. Next, the contents of W are written to PCL; the existing value is discarded. Then, the contents of PCLATH and PCLATU are latched into PCH and PCU, respectively. The second cycle is executed as a NOP instruction while the new next instruction is fetched.</p> <p>Unlike CALL, there is no option to update W, STATUS or BSR.</p>							
Words:	1							
Cycles:	2							

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Read WREG	Push PC to stack	No operation
No operation	No operation	No operation	No operation

**Example:**                      HERE                      CALLW

Before Instruction

PC = address (HERE)  
PCLATH = 10h  
PCLATU = 00h  
W = 06h

After Instruction

PC = 001006h  
TOS = address (HERE + 2)  
PCLATH = 10h  
PCLATU = 00h  
W = 06h

MOVSF		Move Indexed to f			
Syntax:	MOVSF [z <sub>s</sub> ], f <sub>d</sub>				
Operands:	0 ≤ z <sub>s</sub> ≤ 127 0 ≤ f <sub>d</sub> ≤ 4095				
Operation:	((FSR2) + z <sub>s</sub> ) → f <sub>d</sub>				
Status Affected:	None				
Encoding:					
1st word (source)	1110	1011	0zzz	zzzz <sub>s</sub>	
2nd word (destin.)	1111	ffff	ffff	ffff <sub>d</sub>	
Description:	<p>The contents of the source register are moved to destination register 'f<sub>d</sub>'. The actual address of the source register is determined by adding the 7-bit literal offset 'z<sub>s</sub>', in the first word, to the value of FSR2. The address of the destination register is specified by the 12-bit literal 'f<sub>d</sub>' in the second word. Both addresses can be anywhere in the 4096-byte data space (000h to FFFh).</p> <p>The MOVSF instruction cannot use the PCL, TOSU, TOSH or TOSL as the destination register.</p> <p>If the resultant source address points to an Indirect Addressing register, the value returned will be 00h.</p>				
Words:	2				
Cycles:	2				

Q Cycle Activity:

Q1	Q2	Q3	Q4
Decode	Determine source addr	Determine source addr	Read source reg
Decode	No operation 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

## 30.7 MPLAB SIM Software Simulator

The MPLAB SIM Software Simulator allows code development in a PC-hosted environment by simulating the PIC MCUs and dsPIC® DSCs on an instruction level. On any given instruction, the data areas can be examined or modified and stimuli can be applied from a comprehensive stimulus controller. Registers can be logged to files for further run-time analysis. The trace buffer and logic analyzer display extend the power of the simulator to record and track program execution, actions on I/O, most peripherals and internal registers.

The MPLAB SIM Software Simulator fully supports symbolic debugging using the MPLAB C Compilers, and the MPASM and MPLAB Assemblers. The software simulator offers the flexibility to develop and debug code outside of the hardware laboratory environment, making it an excellent, economical software development tool.

## 30.8 MPLAB REAL ICE In-Circuit Emulator System

MPLAB REAL ICE In-Circuit Emulator System is Microchip's next generation high-speed emulator for Microchip Flash DSC and MCU devices. It debugs and programs PIC® Flash MCUs and dsPIC® Flash DSCs with the easy-to-use, powerful graphical user interface of the MPLAB Integrated Development Environment (IDE), included with each kit.

The emulator is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with either a connector compatible with in-circuit debugger systems (RJ11) or with the new high-speed, noise tolerant, Low-Voltage Differential Signal (LVDS) interconnection (CAT5).

The emulator is field upgradable through future firmware downloads in MPLAB IDE. In upcoming releases of MPLAB IDE, new devices will be supported, and new features will be added. MPLAB REAL ICE offers significant advantages over competitive emulators including low-cost, full-speed emulation, run-time variable watches, trace analysis, complex breakpoints, a ruggedized probe interface and long (up to three meters) interconnection cables.

## 30.9 MPLAB ICD 3 In-Circuit Debugger System

MPLAB ICD 3 In-Circuit Debugger System is Microchip's most cost effective high-speed hardware debugger/programmer for Microchip Flash Digital Signal Controller (DSC) and microcontroller (MCU) devices. It debugs and programs PIC® Flash microcontrollers and dsPIC® DSCs with the powerful, yet easy-to-use graphical user interface of MPLAB Integrated Development Environment (IDE).

The MPLAB ICD 3 In-Circuit Debugger probe is connected to the design engineer's PC using a high-speed USB 2.0 interface and is connected to the target with a connector compatible with the MPLAB ICD 2 or MPLAB REAL ICE systems (RJ-11). MPLAB ICD 3 supports all MPLAB ICD 2 headers.

## 30.10 PICkit 3 In-Circuit Debugger/Programmer and PICkit 3 Debug Express

The MPLAB PICkit 3 allows debugging and programming of PIC® and dsPIC® Flash microcontrollers at a most affordable price point using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE). The MPLAB PICkit 3 is connected to the design engineer's PC using a full speed USB interface and can be connected to the target via an Microchip debug (RJ-11) connector (compatible with MPLAB ICD 3 and MPLAB REAL ICE). The connector uses two device I/O pins and the reset line to implement in-circuit debugging and In-Circuit Serial Programming™.

The PICkit 3 Debug Express include the PICkit 3, demo board and microcontroller, hookup cables and CDROM with user's guide, lessons, tutorial, compiler and MPLAB IDE software.

# PIC18F87K90 FAMILY

**TABLE 31-21: MSSP I<sup>2</sup>C™ BUS DATA REQUIREMENTS**

Param. No.	Symbol	Characteristic	Min	Max	Units	Conditions
100	THIGH	Clock High Time	100 kHz mode	2(Tosc)(BRG + 1)	—	—
			400 kHz mode	2(Tosc)(BRG + 1)	—	—
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	—	—
101	TLOW	Clock Low Time	100 kHz mode	2(Tosc)(BRG + 1)	—	—
			400 kHz mode	2(Tosc)(BRG + 1)	—	—
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	—	—
102	TR	SDAx and SCLx Rise Time	100 kHz mode	—	1000	ns
			400 kHz mode	20 + 0.1 C <sub>B</sub>	300	ns
			1 MHz mode <sup>(1)</sup>	—	300	ns
103	TF	SDAx and SCLx Fall Time	100 kHz mode	—	300	ns
			400 kHz mode	20 + 0.1 C <sub>B</sub>	300	ns
			1 MHz mode <sup>(1)</sup>	—	100	ns
90	TSU:STA	Start Condition Setup Time	100 kHz mode	2(Tosc)(BRG + 1)	—	—
			400 kHz mode	2(Tosc)(BRG + 1)	—	—
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	—	—
91	THD:STA	Start Condition Hold Time	100 kHz mode	2(Tosc)(BRG + 1)	—	—
			400 kHz mode	2(Tosc)(BRG + 1)	—	—
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	—	—
106	THD:DAT	Data Input Hold Time	100 kHz mode	0	—	ns
			400 kHz mode	0	0.9	μs
			1 MHz mode <sup>(1)</sup>	—	—	ns
107	TSU:DAT	Data Input Setup Time	100 kHz mode	250	—	ns
			400 kHz mode	100	—	ns
			1 MHz mode <sup>(1)</sup>	—	—	ns
92	TSU:STO	Stop Condition Setup Time	100 kHz mode	2(Tosc)(BRG + 1)	—	—
			400 kHz mode	2(Tosc)(BRG + 1)	—	—
			1 MHz mode <sup>(1)</sup>	2(Tosc)(BRG + 1)	—	—
109	TAA	Output Valid from Clock	100 kHz mode	—	3500	ns
			400 kHz mode	—	1000	ns
			1 MHz mode <sup>(1)</sup>	—	—	ns
110	TBUF	Bus Free Time	100 kHz mode	4.7	—	μs
			400 kHz mode	1.3	—	μs
			1 MHz mode <sup>(1)</sup>	—	—	μs
D102	CB	Bus Capacitive Loading	—	400	pF	

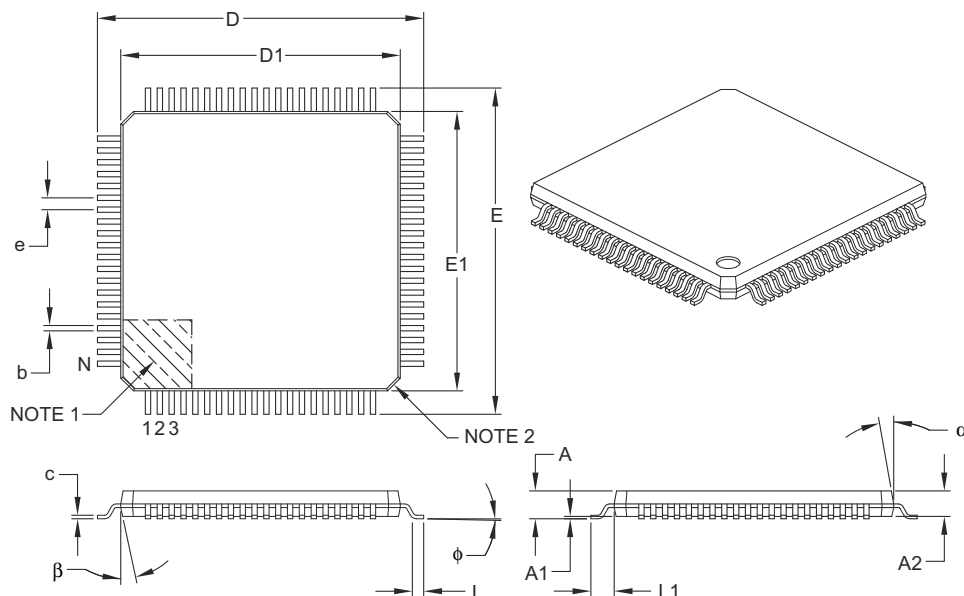
**Note 1:** Maximum pin capacitance = 10 pF for all I<sup>2</sup>C™ pins.

- 2:** A Fast mode I<sup>2</sup>C bus device can be used in a Standard mode I<sup>2</sup>C bus system, but Parameter #107 ≥ 250 ns must then be met. This will automatically be the case if the device does not stretch the LOW period of the SCLx signal. If such a device does stretch the LOW period of the SCLx signal, it must output the next data bit to the SDAx line, Parameter #102 + Parameter #107 = 1000 + 250 = 1250 ns (for 100 kHz mode), before the SCLx line is released.

# PIC18F87K90 FAMILY

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

**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 Leads	N	80		
Lead Pitch	e	0.50 BSC		
Overall Height	A	–	–	1.20
Molded Package Thickness	A2	0.95	1.00	1.05
Standoff	A1	0.05	–	0.15
Foot Length	L	0.45	0.60	0.75
Footprint	L1	1.00 REF		
Foot Angle	φ	0°	3.5°	7°
Overall Width	E	14.00 BSC		
Overall Length	D	14.00 BSC		
Molded Package Width	E1	12.00 BSC		
Molded Package Length	D1	12.00 BSC		
Lead Thickness	c	0.09	–	0.20
Lead Width	b	0.17	0.22	0.27
Mold Draft Angle Top	α	11°	12°	13°
Mold Draft Angle Bottom	β	11°	12°	13°

### Notes:

- Pin 1 visual index feature may vary, but must be located within the hatched area.
- Chamfers at corners are optional; size may vary.
- Dimensions D1 and E1 do not include mold flash or protrusions. Mold flash or protrusions shall not exceed 0.25 mm per side.
- 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 C04-092B