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	Obsolete
Core Processor	PIC
Core Size	8-Bit
Speed	40MHz
Connectivity	EBI/EMI, I²C, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	69
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	1K x 8
RAM Size	3.8K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5.5V
Data Converters	A/D 16x10b
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic18lf8621t-i-pt

TABLE 1-2: PIC18F6525/6621/8525/8621 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number		Pin Type	Buffer Type	Description
	PIC18F6X2X	PIC18F8X2X			
					PORTJ is a bidirectional I/O port(6).
RJ0/ALE	—	62			
RJ0			I/O	ST	Digital I/O.
ALE			O	TTL	External memory address latch enable.
RJ1/$\overline{OE}$	—	61			
RJ1			I/O	ST	Digital I/O.
$\overline{OE}$			O	TTL	External memory output enable.
RJ2/$\overline{WRL}$	—	60			
RJ2			I/O	ST	Digital I/O.
$\overline{WRL}$			O	TTL	External memory write low control.
RJ3/$\overline{WRH}$	—	59			
RJ3			I/O	ST	Digital I/O.
$\overline{WRH}$			O	TTL	External memory write high control.
RJ4/BA0	—	39			
RJ4			I/O	ST	Digital I/O.
BA0			O	TTL	System bus byte address 0 control.
RJ5/$\overline{CE}$	—	40			
RJ5			I/O	ST	Digital I/O
$\overline{CE}$			O	TTL	External memory access indicator.
RJ6/$\overline{LB}$	—	41			
RJ6			I/O	ST	Digital I/O.
$\overline{LB}$			O	TTL	External memory low byte select.
RJ7/$\overline{UB}$	—	42			
RJ7			I/O	ST	Digital I/O.
$\overline{UB}$			O	TTL	External memory high byte select.
VSS	9, 25, 41, 56	11, 31, 51, 70	P	—	Ground reference for logic and I/O pins.
VDD	10, 26, 38, 57	12, 32, 48, 71	P	—	Positive supply for logic and I/O pins.
AVSS(8)	20	26	P	—	Ground reference for analog modules.
AVDD(8)	19	25	P	—	Positive supply for analog modules.

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)

Note 1: Alternate assignment for ECCP2/P2A in PIC18F8525/8621 devices when CCP2MX (CONFIG3H<0>) is not set (all Program Memory modes except Microcontroller).

2: Default assignment for ECCP2/P2A when CCP2MX is set (all devices).

3: External memory interface functions are only available on PIC18F8525/8621 devices.

4: Default assignment for P1B/P1C/P3B/P3C for PIC18F8525/8621 devices when ECCPMX (CONFIG3H<1>) is set and for all PIC18F6525/6621 devices.

5: Alternate assignment for ECCP2/P2A in PIC18F8525/8621 devices when CCP2MX is not set (Microcontroller mode).

6: PORTH and PORTJ (and their multiplexed functions) are only available on PIC18F8525/8621 devices.

7: Alternate assignment for P1B/P1C/P3B/P3C for PIC18F8525/8621 devices when ECCPMX (CONFIG3H<1>) is not set.

8: AVDD must be connected to a positive supply and AVSS must be connected to a ground reference for proper operation of the part in user or ICSP™ modes. See parameter D001 for details.

9: RG5 is multiplexed with $\overline{MCLR}$ and is only available when the $\overline{MCLR}$ Resets are disabled.

TABLE 3-3: INITIALIZATION CONDITIONS FOR ALL REGISTERS (CONTINUED)

Register	Applicable Devices		Power-on Reset, Brown-out Reset	MCLR Resets WDT Reset RESET Instruction Stack Resets	Wake-up via WDT or Interrupt
PORTG(7)	Feature1	Feature2	--xx xxxx	--uu uuuu	--uu uuuu
PORTF	Feature1	Feature2	x000 0000	u000 0000	uuuu uuuu
PORTE	Feature1	Feature2	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTD	Feature1	Feature2	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTC	Feature1	Feature2	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTB	Feature1	Feature2	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA(5,6)	Feature1	Feature2	-x0x 0000(5)	-u0u 0000(5)	-uuu uuuu(5)
SPBRGH1	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
BAUDCON1	Feature1	Feature2	-1-0 0-00	-1-0 0-00	-u-u u-uu
SPBRGH2	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
BAUDCON2	Feature1	Feature2	-1-0 0-00	-1-0 0-00	-u-1 u-uu
ECCP1DEL	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
TMR4	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
PR4	Feature1	Feature2	1111 1111	1111 1111	uuuu uuuu
T4CON	Feature1	Feature2	-000 0000	-000 0000	-uuu uuuu
CCPR4H	Feature1	Feature2	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR4L	Feature1	Feature2	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCP4CON	Feature1	Feature2	--00 0000	--00 0000	--uu uuuu
CCPR5H	Feature1	Feature2	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCPR5L	Feature1	Feature2	xxxx xxxx	xxxx xxxx	uuuu uuuu
CCP5CON	Feature1	Feature2	--00 0000	--00 0000	--uu uuuu
SPBRG2	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
RCREG2	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
TXREG2	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
TXSTA2	Feature1	Feature2	0000 0010	0000 0010	uuuu uuuu
RCSTA2	Feature1	Feature2	0000 000x	0000 000x	uuuu uuuu
ECCP3AS	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
ECCP3DEL	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
ECCP2AS	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu
ECCP2DEL	Feature1	Feature2	0000 0000	0000 0000	uuuu uuuu

Legend: u = unchanged, x = unknown, - = unimplemented bit, read as '0', q = value depends on condition. Shaded cells indicate conditions do not apply for the designated device.

Note 1: One or more bits in the INTCONx or PIRx registers will be affected (to cause wake-up).

2: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the PC is loaded with the interrupt vector (0008h or 0018h).

3: When the wake-up is due to an interrupt and the GIEL or GIEH bit is set, the TOSU, TOSH and TOSL are updated with the current value of the PC. The STKPTR is modified to point to the next location in the hardware stack.

4: See Table 3-2 for Reset value for specific condition.

5: Bit 6 of PORTA, LATA and TRISA are enabled in ECIO and RCIO Oscillator modes only. In all other oscillator modes, they are disabled and read '0'.

6: Bit 6 of PORTA, LATA and TRISA are not available on all devices. When unimplemented, they are read '0'.

7: If $\overline{\text{MCLR}}$ function is disabled, PORTG<5> is a read-only bit.

8: Enabled only in Microcontroller mode for PIC18F8525/8621 devices.

9: The MEMCON register is unimplemented and reads all '0's when the device is in Microcontroller mode.

4.12 Indirect Addressing, INDF and FSR Registers

Indirect addressing is a mode of addressing data memory, where the data memory address in the instruction is not fixed. An FSR register is used as a pointer to the data memory location that is to be read or written. Since this pointer is in RAM, the contents can be modified by the program. This can be useful for data tables in the data memory and for software stacks. Figure 4-9 shows the operation of indirect addressing. This shows the moving of the value to the data memory address specified by the value of the FSR register.

Indirect addressing is possible by using one of the INDF registers. Any instruction using the INDF register actually accesses the register pointed to by the File Select Register, FSR. Reading the INDF register itself indirectly (FSR = 0), will read 00h. Writing to the INDF register indirectly, results in a no operation (NOP). The FSR register contains a 12-bit address which is shown in Figure 4-10.

The INDFn register is not a physical register. Addressing INDFn actually addresses the register whose address is contained in the FSRn register (FSRn is a pointer). This is indirect addressing.

Example 4-5 shows a simple use of indirect addressing to clear the RAM in Bank 1 (locations 100h-1FFh) in a minimum number of instructions.

EXAMPLE 4-5: HOW TO CLEAR RAM (BANK 1) USING INDIRECT ADDRESSING

```
      LFSR   FSR0, 0x100  ;
NEXT  CLRF   POSTINC0     ; Clear INDF
                          ; register and
                          ; inc pointer
      BTFSS  FSR0H, 1     ; All done with
                          ; Bank1?
      GOTO   NEXT         ; NO, clear next
CONTINUE                  ; YES, continue
```

There are three indirect addressing registers. To address the entire data memory space (4096 bytes), these registers are 12 bits wide. To store the 12 bits of addressing information, two 8-bit registers are required. These indirect addressing registers are:

1. FSR0: composed of FSR0H:FSR0L
2. FSR1: composed of FSR1H:FSR1L
3. FSR2: composed of FSR2H:FSR2L

In addition, there are registers INDF0, INDF1 and INDF2, which are not physically implemented. Reading or writing to these registers activates indirect addressing, with the value in the corresponding FSR register being the address of the data. If an instruction writes a value to INDF0, the value will be written to the address pointed to by FSR0H:FSR0L. A read from INDF1 reads the data from the address pointed to by FSR1H:FSR1L. INDFn can be used in code anywhere an operand can be used.

If INDF0, INDF1 or INDF2 are read indirectly via an FSR, all '0's are read (zero bit is set). Similarly, if INDF0, INDF1 or INDF2 are written to indirectly, the operation will be equivalent to a NOP instruction and the Status bits are not affected.

4.12.1 INDIRECT ADDRESSING OPERATION

Each FSR register has an INDF register associated with it, plus four additional register addresses. Performing an operation on one of these five registers determines how the FSR will be modified during indirect addressing.

When data access is done to one of the five INDFn locations, the address selected will configure the FSRn register to:

- Do nothing to FSRn after an indirect access (no change) – INDFn.
- Auto-decrement FSRn after an indirect access (post-decrement) – POSTDECn.
- Auto-increment FSRn after an indirect access (post-increment) – POSTINCn.
- Auto-increment FSRn before an indirect access (pre-increment) – PREINCn.
- Use the value in the WREG register as an offset to FSRn. Do not modify the value of the WREG or the FSRn register after an indirect access (no change) – PLUSWn.

When using the auto-increment or auto-decrement features, the effect on the FSR is not reflected in the STATUS register. For example, if the indirect address causes the FSR to equal '0', the Z bit will not be set.

Incrementing or decrementing an FSR affects all 12 bits. That is, when FSRnL overflows from an increment, FSRnH will be incremented automatically.

Adding these features allows the FSRn to be used as a Stack Pointer in addition to its uses for table operations in data memory.

Each FSR has an address associated with it that performs an indexed indirect access. When a data access to this INDFn location (PLUSWn) occurs, the FSRn is configured to add the signed value in the WREG register and the value in FSR to form the address before an indirect access. The FSR value is not changed.

If an FSR register contains a value that points to one of the INDFn, an indirect read will read 00h (zero bit is set), while an indirect write will be equivalent to a NOP (Status bits are not affected).

If an indirect addressing operation is done where the target address is an FSRnH or FSRnL register, the write operation will dominate over the pre- or post-increment/decrement functions.

REGISTER 7-1: EECON1 REGISTER (ADDRESS FA6h)

R/W-x	R/W-x	U-0	R/W-0	R/W-x	R/W-0	R/S-0	R/S-0
EEPGD	CFGS	—	FREE	WRERR	WREN	WR	RD
bit 7							bit 0

bit 7 **EEPGD:** Flash Program/Data EEPROM Memory Select bit
1 = Access Flash program memory
0 = Access data EEPROM memory

bit 6 **CFGS:** Flash Program/Data EEPROM or Configuration Select bit
1 = Access Configuration or Calibration registers
0 = Access Flash program or data EEPROM memory

bit 5 **Unimplemented:** Read as '0'

bit 4 **FREE:** Flash Row Erase Enable bit
1 = Erase the program memory row addressed by TBLPTR on the next WR command (cleared by completion of erase operation)
0 = Perform write only

bit 3 **WRERR:** Flash Program/Data EEPROM Error Flag bit
1 = A write operation is prematurely terminated (any MCLR or any WDT Reset during self-timed programming in normal operation)
0 = The write operation completed

Note: When a WRERR occurs, the EEPGD or FREE bits are not cleared. This allows tracing of the error condition.

bit 2 **WREN:** Flash Program/Data EEPROM Write Enable bit
1 = Allows write cycles to Flash program/data EEPROM
0 = Inhibits write cycles to Flash program/data EEPROM

bit 1 **WR:** Write Control bit
1 = Initiates a data EEPROM erase/write cycle or a program memory erase cycle or write cycle (The operation is self-timed and the bit is cleared by hardware once write is complete. The WR bit can only be set (not cleared) in software.)
0 = Write cycle to the EEPROM is complete

bit 0 **RD:** Read Control bit
1 = Initiates an EEPROM read (Read takes one cycle. RD is cleared in hardware. The RD bit can only be set (not cleared) in software. RD bit cannot be set when EEPGD = 1.)
0 = Does not initiate an EEPROM read

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

8.0 8 x 8 HARDWARE MULTIPLIER

8.1 Introduction

An 8 x 8 hardware multiplier is included in the ALU of the PIC18F6525/6621/8525/8621 devices. By making the multiply a hardware operation, it completes in a single instruction cycle. This is an unsigned multiply that gives a 16-bit result. The result is stored in the 16-bit product register pair (PRODH:PRODL). The multiplier does not affect any flags in the ALUSTA register.

Making the 8 x 8 multiplier execute in a single cycle gives the following advantages:

- Higher computational throughput
- Reduces code size requirements for multiply algorithms

The performance increase allows the device to be used in applications previously reserved for Digital Signal Processors.

Table 8-1 shows a performance comparison between Enhanced devices using the single-cycle hardware multiply and performing the same function without the hardware multiply.

8.2 Operation

Example 8-1 shows the sequence to do an 8 x 8 unsigned multiply. Only one instruction is required when one argument of the multiply is already loaded in the WREG register.

Example 8-2 shows the sequence to do an 8 x 8 signed multiply. To account for the signed bits of the arguments, each argument's Most Significant bit (MSb) is tested and the appropriate subtractions are done.

EXAMPLE 8-1: 8 x 8 UNSIGNED MULTIPLY ROUTINE

```
MOVF   ARG1, W      ;
MULWF  ARG2         ; ARG1 * ARG2 ->
                    ;   PRODH:PRODL
```

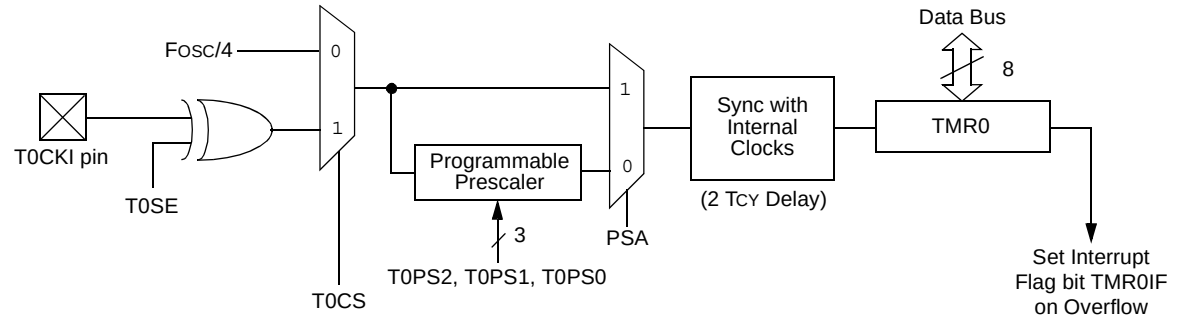
EXAMPLE 8-2: 8 x 8 SIGNED MULTIPLY ROUTINE

```
MOVF   ARG1, W          ;
MULWF  ARG2             ; ARG1 * ARG2 ->
                        ; PRODH:PRODL
BTFSC  ARG2, SB         ; Test Sign Bit
SUBWF  PRODH, F         ; PRODH = PRODH
                        ;         - ARG1
MOVF   ARG2, W          ;
BTFSC  ARG1, SB         ; Test Sign Bit
SUBWF  PRODH, F         ; PRODH = PRODH
                        ;         - ARG2
```

TABLE 8-1: PERFORMANCE COMPARISON

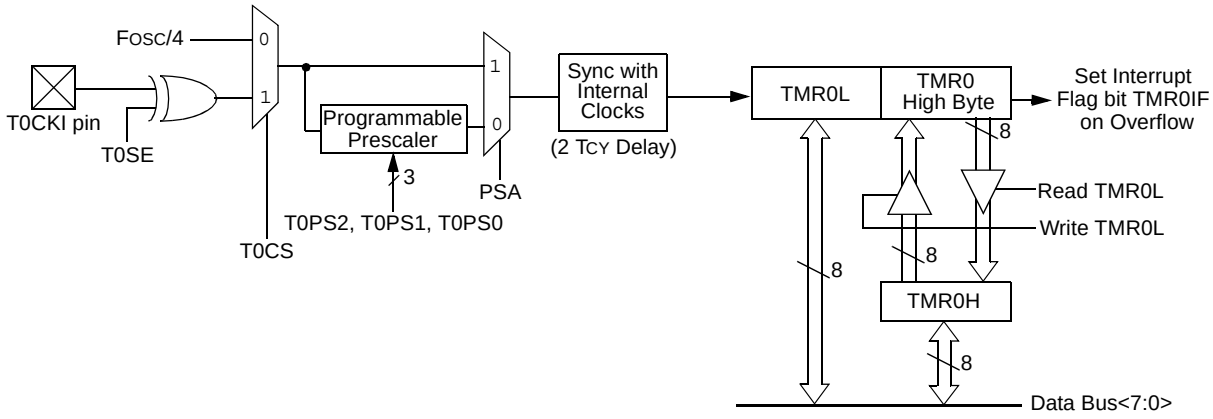
Routine	Multiply Method	Program Memory (Words)	Cycles (Max)	Time		
				@ 40 MHz	@ 10 MHz	@ 4 MHz
8 x 8 unsigned	Without hardware multiply	13	69	6.9 μs	27.6 μs	69 μs
	Hardware multiply	1	1	100 ns	400 ns	1 μs
8 x 8 signed	Without hardware multiply	33	91	9.1 μs	36.4 μs	91 μs
	Hardware multiply	6	6	600 ns	2.4 μs	6 μs
16 x 16 unsigned	Without hardware multiply	21	242	24.2 μs	96.8 μs	242 μs
	Hardware multiply	24	24	2.4 μs	9.6 μs	24 μs
16 x 16 signed	Without hardware multiply	52	254	25.4 μs	102.6 μs	254 μs
	Hardware multiply	36	36	3.6 μs	14.4 μs	36 μs

FIGURE 11-1: TIMER0 BLOCK DIAGRAM IN 8-BIT MODE



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

FIGURE 11-2: TIMER0 BLOCK DIAGRAM IN 16-BIT MODE



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI max. prescale.

NOTES:

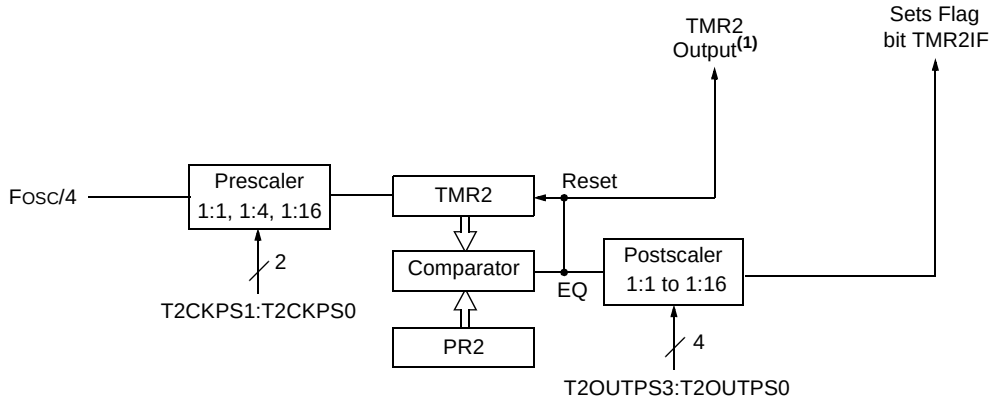
13.2 Timer2 Interrupt

The Timer2 module has an 8-bit period register, PR2. Timer2 increments from 00h until it matches PR2 and then resets to 00h on the next increment cycle. PR2 is a readable and writable register. The PR2 register is initialized to FFh upon Reset.

13.3 Output of TMR2

The output of TMR2 (before the postscaler) is fed to the synchronous serial port module which optionally uses it to generate the shift clock.

FIGURE 13-1: TIMER2 BLOCK DIAGRAM



Note 1: TMR2 register output can be software selected by the MSSP module as a baud clock.

TABLE 13-1: REGISTERS ASSOCIATED WITH TIMER2 AS A TIMER/COUNTER

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	0000 000x	0000 000u
PIR1	PSPIF(1)	ADIF	RC1IF	TX1IF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
PIE1	PSPIE(1)	ADIE	RC1IE	TX1IE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
IPR1	PSPIP(1)	ADIP	RC1IP	TX1IP	SSPIP	CCP1IP	TMR2IP	TMR1IP	1111 1111	1111 1111
TMR2	Timer2 Module Register								0000 0000	0000 0000
T2CON	—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0	-000 0000	-000 0000
PR2	Timer2 Period Register								1111 1111	1111 1111

Legend: x = unknown, u = unchanged, — = unimplemented, read as '0'. Shaded cells are not used by the Timer2 module.

Note 1: Enabled only in Microcontroller mode for PIC18F8525/8621 devices.

TABLE 17-2: PIN CONFIGURATIONS FOR ECCP2

ECCP Mode	CCP2CON Configuration	RB3	RC1	RE7	RE2	RE1	RE0
All devices, CCP2MX = 1, Microcontroller mode:							
Compatible CCP	00xx 11xx	RB3/INT3	ECCP2	RE7	RE2	RE1	RE0
Dual PWM	10xx 11xx	RB3/INT3	P2A	RE7	P2B	RE1	RE0
Quad PWM	x1xx 11xx	RB3/INT3	P2A	RE7	P2B	P2C	P2D
All devices, CCP2MX = 0, Microcontroller mode:							
Compatible CCP	00xx 11xx	RB3/INT3	RC1/T1OS1	ECCP2	RE2	RE1	RE0
Dual PWM	10xx 11xx	RB3/INT3	RC1/T1OS1	P2A	P2B	RE1	RE0
Quad PWM	x1xx 11xx	RB3/INT3	RC1/T1OS1	P2A	P2B	P2C	P2D
PIC18F8525/8621 devices, CCP2MX = 0, all other Program Memory modes:							
Compatible CCP	00xx 11xx	ECCP2	RC1/T1OS1	RE7/AD15	RE2/$\overline{CS}$	RE1/$\overline{WR}$	RE0/$\overline{RD}$
Dual PWM	10xx 11xx	P2A	RC1/T1OS1	RE7/AD15	P2B	RE1/$\overline{WR}$	RE0/$\overline{RD}$
Quad PWM	x1xx 11xx	P2A	RC1/T1OS1	RE7/AD15	P2B	P2C	P2D

Legend: x = Don't care. Shaded cells indicate pin assignments not used by ECCP2 in a given mode.

TABLE 17-3: PIN CONFIGURATIONS FOR ECCP3

ECCP Mode	CCP3CON Configuration	RG0	RE4	RE3	RG3	RH5	RH4
All PIC18F6525/6621 devices:							
Compatible CCP	00xx 11xx	ECCP3	RE4	RE3	RG3/CCP4	N/A	N/A
Dual PWM	10xx 11xx	P3A	P3B	RE3	RG3/CCP4	N/A	N/A
Quad PWM	x1xx 11xx	P3A	P3B	P3C	P3D	N/A	N/A
PIC18F8525/8621 devices, ECCPMX = 1, Microcontroller mode:							
Compatible CCP	00xx 11xx	ECCP3	RE4/AD12	RE3/AD11	RG3/CCP4	RH5/AN13	RH4/AN12
Dual PWM	10xx 11xx	P3A	P3B	RE3/AD11	RG3/CCP4	RH5/AN13	RH4/AN12
Quad PWM	x1xx 11xx	P3A	P3B	P3C	P3D	RH5/AN13	RH4/AN12
PIC18F8525/8621 devices, ECCPMX = 0, Microcontroller mode:							
Compatible CCP	00xx 11xx	ECCP3	RE6/AD14	RE5/AD13	RG3/CCP4	RH7/AN15	RH6/AN14
Dual PWM	10xx 11xx	P3A	RE6/AD14	RE5/AD13	RG3/CCP4	P3B	RH6/AN14
Quad PWM	x1xx 11xx	P3A	RE6/AD14	RE5/AD13	P3D	P3B	P3C
PIC18F8525/8621 devices, ECCPMX = 1, all other Program Memory modes:							
Compatible CCP	00xx 11xx	ECCP3	RE6/AD14	RE5/AD13	RG3/CCP4	RH7/AN15	RH6/AN14

Legend: x = Don't care, N/A = Not available. Shaded cells indicate pin assignments not used by ECCP3 in a given mode.

Note 1: With ECCP3 in Quad PWM mode, CCP4's output is overridden by P1D; otherwise CCP4 is fully operational.

18.3.2 OPERATION

When initializing the SPI, several options need to be specified. This is done by programming the appropriate control bits (SSPCON1<5:0>) and SSPSTAT<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)

The MSSP consists of a transmit/receive shift register (SSPSR) and a buffer register (SSPBUF). The SSPSR shifts the data in and out of the device, MSb first. The SSPBUF holds the data that was written to the SSPSR until the received data is ready. Once the 8 bits of data have been received, that byte is moved to the SSPBUF register. Then the buffer full detect bit, BF (SSPSTAT<0>) and the interrupt flag bit, SSPIF, are set. This double-buffering of the received data (SSPBUF) allows the next byte to start reception before reading the data that was just received. Any write to the SSPBUF register during transmission/reception of data will be ignored and the write collision detect bit, WCOL (SSPCON1<7>), will be set. User software must clear the WCOL bit so that it can be determined if the following write(s) to the SSPBUF register completed successfully.

When the application software is expecting to receive valid data, the SSPBUF should be read before the next byte of data to transfer is written to the SSPBUF. Buffer full bit, BF (SSPSTAT<0>), indicates when SSPBUF has been loaded with the received data (transmission is complete). When the SSPBUF is read, the BF bit is cleared. This data may be irrelevant if the SPI is only a transmitter. Generally, the MSSP interrupt is used to determine when the transmission/reception has completed. The SSPBUF must be read and/or written. If the interrupt method is not going to be used, then software polling can be done to ensure that a write collision does not occur. Example 18-1 shows the loading of the SSPBUF (SSPSR) for data transmission.

The SSPSR is not directly readable or writable and can only be accessed by addressing the SSPBUF register. Additionally, the MSSP Status register (SSPSTAT) indicates the various status conditions.

EXAMPLE 18-1: LOADING THE SSPBUF (SSPSR) REGISTER

```
LOOP    BTFSS   SSPSTAT, BF         ;Has data been received (transmit complete)?
        BRA     LOOP                ;No
        MOVF    SSPBUF, W           ;WREG reg = contents of SSPBUF
        MOVWF   RXDATA              ;Save in user RAM, if data is meaningful
        MOVF    TXDATA, W           ;W reg = contents of TXDATA
        MOVWF   SSPBUF              ;New data to xmit
```

FIGURE 18-10: I²C™ SLAVE MODE TIMING WITH SEN = 0 (RECEPTION, 10-BIT ADDRESS)

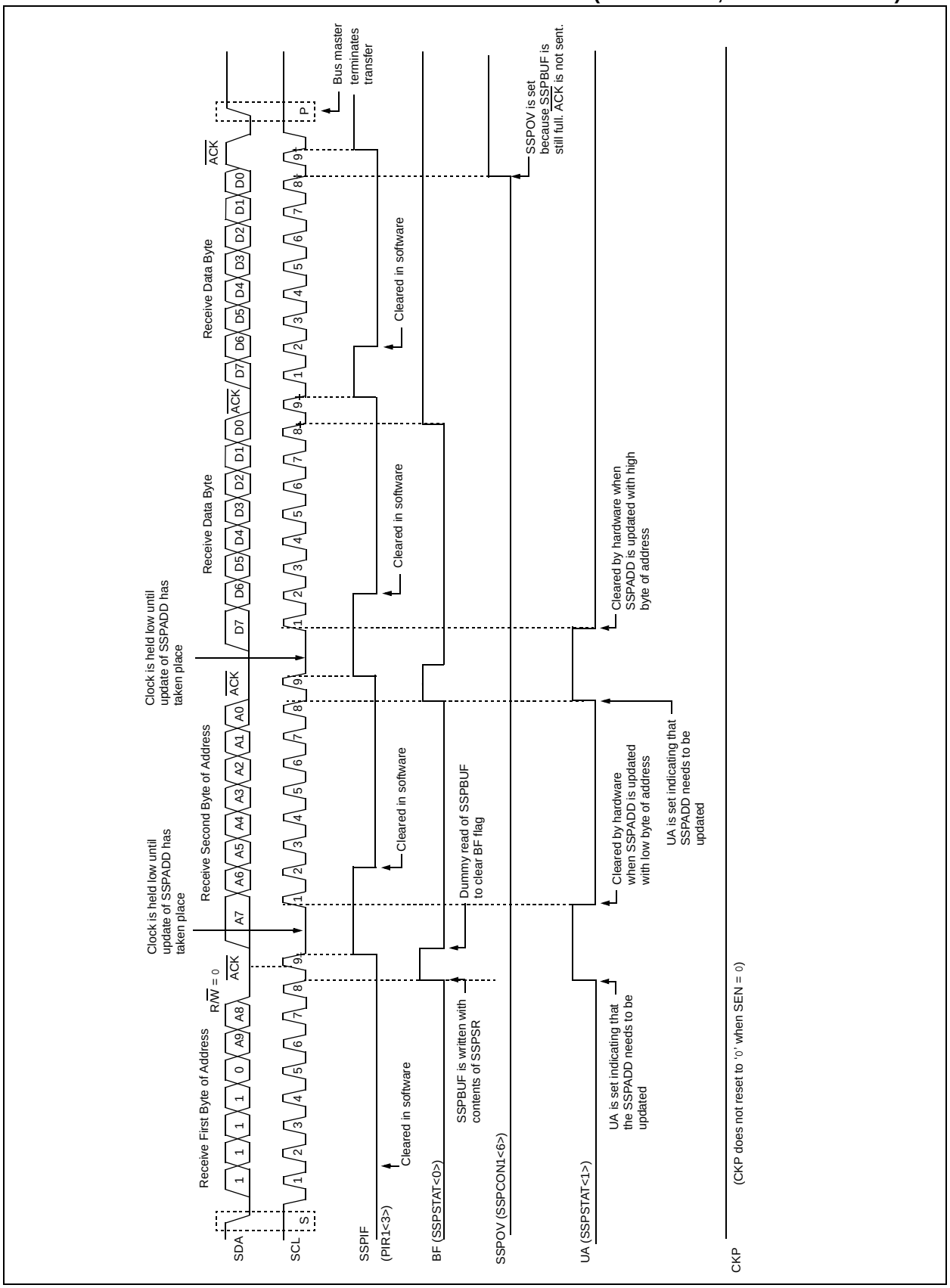
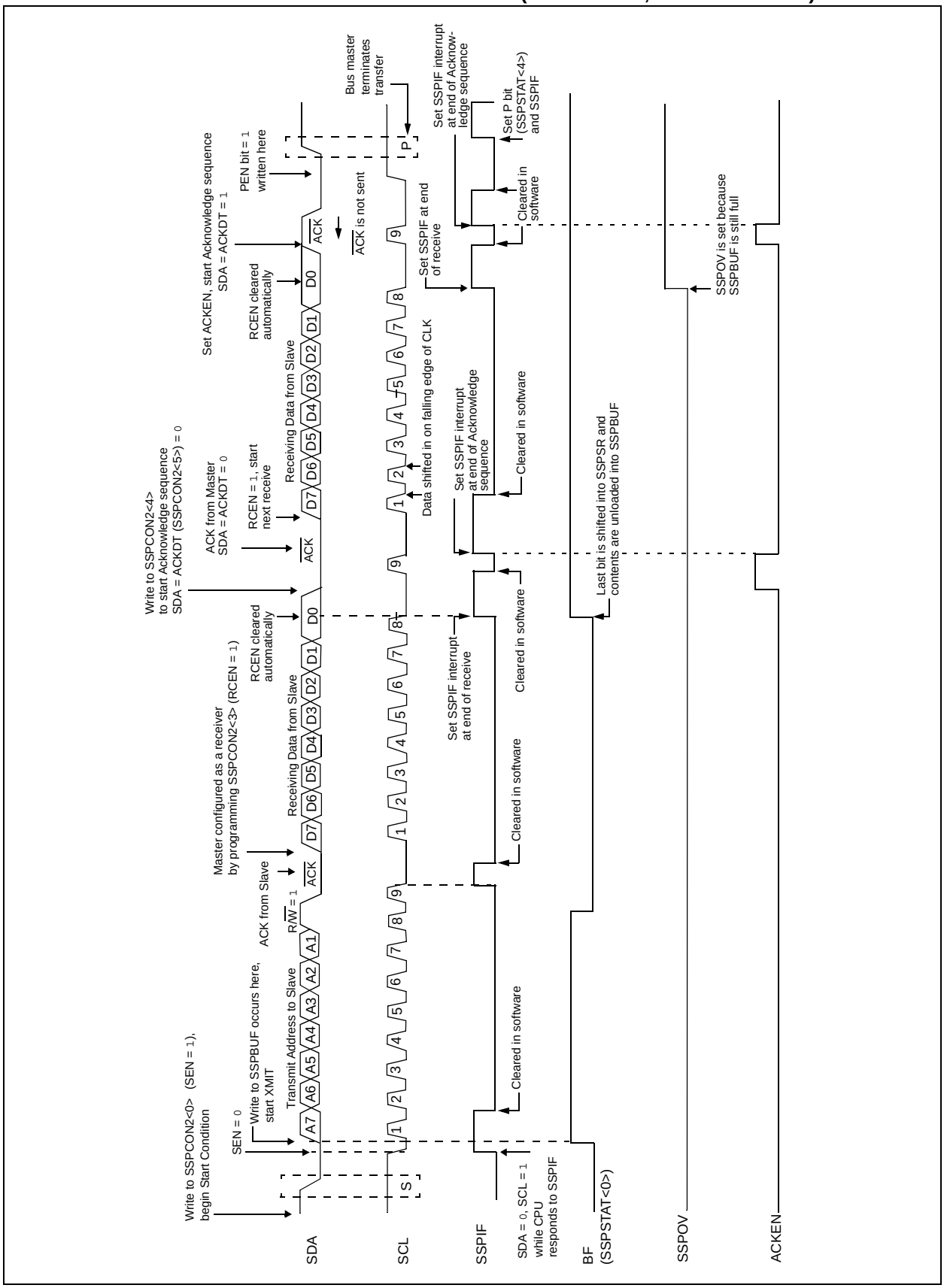


FIGURE 18-22: I²C™ MASTER MODE WAVEFORM (RECEPTION, 7-BIT ADDRESS)



REGISTER 19-1: TXSTAx: TRANSMIT STATUS AND CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-1	R/W-0
CSRC	TX9	TXEN	SYNC	SENDB	BRGH	TRMT	TX9D
bit 7							bit 0

bit 7 **CSRC:** Clock Source Select bit

Asynchronous mode:
Don't care.

Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)

bit 6 **TX9:** 9-bit Transmit Enable bit

1 = Selects 9-bit transmission
0 = Selects 8-bit transmission

bit 5 **TXEN:** Transmit Enable bit

1 = Transmit enabled
0 = Transmit disabled

Note: SREN/CREN overrides TXEN in Sync mode.

bit 4 **SYNC:** EUSART Mode Select bit

1 = Synchronous mode
0 = Asynchronous mode

bit 3 **SENDB:** Send Break Character bit

Asynchronous mode:
1 = Send sync break on next transmission (cleared by hardware upon completion)
0 = Sync break transmission completed

Synchronous mode:
Don't care.

bit 2 **BRGH:** High Baud Rate Select bit

Asynchronous mode:
1 = High speed
0 = Low speed

Synchronous mode:
Unused in this mode.

bit 1 **TRMT:** Transmit Shift Register Status bit

1 = TSR empty
0 = TSR full

bit 0 **TX9D:** 9th bit of Transmit Data

Can be address/data bit or a parity bit.

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

NOTES:

REGISTER 24-6: CONFIG4L: CONFIGURATION REGISTER 4 LOW (BYTE ADDRESS 300006h)

R/P-1	U-0	U-0	U-0	U-0	R/P-1	U-0	R/P-1
$\overline{\text{DEBUG}}$	—	—	—	—	LVP	—	STVREN
bit 7							bit 0

bit 7 **$\overline{\text{DEBUG}}$:** Background Debugger Enable bit
1 = Background debugger disabled. RB6 and RB7 configured as general purpose I/O pins.
0 = Background debugger enabled. RB6 and RB7 are dedicated to in-circuit debug.

bit 6-3 **Unimplemented:** Read as '0'

bit 2 **LVP:** Low-Voltage ICSP Enable bit
1 = Low-Voltage ICSP enabled
0 = Low-Voltage ICSP disabled

bit 1 **Unimplemented:** Read as '0'

bit 0 **STVREN:** Stack Full/Underflow Reset Enable bit
1 = Stack full/underflow will cause Reset
0 = Stack full/underflow will not cause Reset

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

REGISTER 24-7: CONFIG5L: CONFIGURATION REGISTER 5 LOW (BYTE ADDRESS 300008h)

U-0	U-0	U-0	U-0	R/C-1	R/C-1	R/C-1	R/C-1
—	—	—	—	CP3(1)	CP2	CP1	CP0
bit 7							bit 0

bit 7-4 **Unimplemented:** Read as '0'

bit 3 **CP3:** Code Protection bit(1)
1 = Block 3 (00C000-00FFFFh) not code-protected
0 = Block 3 (00C000-00FFFFh) code-protected

Note 1: Unimplemented in PIC18FX525 devices; maintain this bit set.

bit 2 **CP2:** Code Protection bit
1 = Block 2 (008000-00BFFFh) not code-protected
0 = Block 2 (008000-00BFFFh) code-protected

bit 1 **CP1:** Code Protection bit
1 = Block 1 (004000-007FFFh) not code-protected
0 = Block 1 (004000-007FFFh) code-protected

bit 0 **CP0:** Code Protection bit
1 = Block 0 (000800-003FFFh) not code-protected
0 = Block 0 (000800-003FFFh) code-protected

Legend:		
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

REGISTER 24-12: CONFIG7H: CONFIGURATION REGISTER 7 HIGH (BYTE ADDRESS 30000Dh)

U-0	R/C-1	U-0	U-0	U-0	U-0	U-0	U-0
—	EBTRB	—	—	—	—	—	—
bit 7							bit 0

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

bit 6 **EBTRB:** Boot Block Table Read Protection bit
1 = Boot block (000000-0007FFh) not protected from table reads executed in other blocks
0 = Boot block (000000-0007FFh) protected from table reads executed in other blocks

bit 5-0 **Unimplemented:** Read as '0'

Legend:		
R = Readable bit	C = Clearable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

REGISTER 24-13: DEVID1: DEVICE ID REGISTER 1 FOR PIC18F6525/6621/8525/8621 DEVICES (ADDRESS 3FFFFEh)

R	R	R	R	R	R	R	R
DEV2	DEV1	DEV0	REV4	REV3	REV2	REV1	REV0
bit 7							bit 0

bit 7-5 **DEV2:DEV0:** Device ID bits
100 = PIC18F8621
101 = PIC18F6621
110 = PIC18F8525
111 = PIC18F6525

bit 4-0 **REV4:REV0:** Revision ID bits
These bits are used to indicate the device revision.

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

REGISTER 24-14: DEVID2: DEVICE ID REGISTER 2 FOR PIC18F6525/6621/8525/8621 DEVICES (ADDRESS 3FFFFFh)

R-0	R-0	R-0	R-0	R-1	R-0	R-1	R-0
DEV10	DEV9	DEV8	DEV7	DEV6	DEV5	DEV4	DEV3
bit 7							bit 0

bit 7-0 **DEV10:DEV3:** Device ID bits
These bits are used with the DEV2:DEV0 bits in the Device ID Register 1 to identify the part number.
0000 1010 = PIC18F6525/6621/8525/8621

Legend:		
R = Readable bit	P = Programmable bit	U = Unimplemented bit, read as '0'
-n = Value when device is unprogrammed		u = Unchanged from programmed state

TABLE 27-4: MEMORY PROGRAMMING REQUIREMENTS

DC Characteristics			Standard Operating Conditions (unless otherwise stated) Operating temperature -40°C ≤ TA ≤ +85°C for industrial -40°C ≤ TA ≤ +125°C for extended				
Param No.	**Sym**	**Characteristic**	**Min**	**Typ†**	**Max**	**Units**	**Conditions**
		Internal Program Memory Programming Specifications					
D110	VPP	Voltage on $\overline{MCLR}$/VPP pin	9.00	—	13.25	V	**(Note 2)**
D112	IPP	Current into $\overline{MCLR}$/VPP pin	—	—	300	μA	
D113	IDDP	Supply Current during Programming	—	—	1.0	mA	
		Data EEPROM Memory					
D120	ED	Byte Endurance	100K 10K	1M 100K	— —	E/W E/W	-40°C to +85°C -40°C to +125°C
D121	VDRW	VDD for Read/Write	VMIN	—	5.5	V	Using EECON to read/write VMIN = Minimum operating voltage
D122	TDEW	Erase/Write Cycle Time	—	4	—	ms	
D123	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated
D124	TREF	Number of Total Erase/Write Cycles before Refresh(1)	1M 100K	10M 1M	— —	E/W E/W	-40°C to +85°C -40°C to +125°C
		Program Flash Memory					
D130	EP	Cell Endurance	10K 1K	100K 10K	— —	E/W E/W	-40°C to +85°C -40°C to +125°C
D131	VPR	VDD for Read	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D132	VIE	VDD for Block Erase	4.5	—	5.5	V	Using ICSP™ port
D132A	VIW	VDD for Externally Timed Erase or Write	4.5	—	5.5	V	Using ICSP port
D132B	VPEW	VDD for Self-Timed Write and Row Erase	VMIN	—	5.5	V	VMIN = Minimum operating voltage
D133	TIE	ICSP Block Erase Cycle Time	—	4	—	ms	VDD > 4.5V
D133A	TIW	ICSP Erase or Write Cycle Time (externally timed)	1	—	—	ms	VDD > 4.5V
D133A	TIW	Self-Timed Write Cycle Time	—	2	—	ms	
D134	TRETD	Characteristic Retention	40	—	—	Year	Provided no other specifications are violated

† Data in "Typ" column is at 5.0V, 25°C unless otherwise stated. These parameters are for design guidance only and are not tested.

Note 1: Refer to **Section 7.8 "Using the Data EEPROM"** for a more detailed discussion on data EEPROM endurance.

2: Required only if Low-Voltage Programming is disabled.

TABLE 27-25: A/D CONVERTER CHARACTERISTICS:PIC18F6525/6621/8525/8621 (INDUSTRIAL, EXTENDED)

				PIC18LF6X2X/8X2X (INDUSTRIAL)				
Param No.	**Symbol**	**Characteristic**		**Min**	**Typ**	**Max**	**Units**	**Conditions**
A01	NR	Resolution		— —	— —	10 TBD	bit bit	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A03	EIL	Integral Linearity Error		— —	— —	<±1 TBD	LSb LSb	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A04	EDL	Differential Linearity Error		— —	— —	<±1 TBD	LSb LSb	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A05	EFS	Full Scale Error		— —	— —	<±1 TBD	LSb LSb	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A06	EOFF	Offset Error		— —	— —	<±1 TBD	LSb LSb	VREF = VDD ≥ 3.0V VREF = VDD < 3.0V
A10	—	Monotonicity		guaranteed(3)			—	VSS ≤ VAIN ≤ VREF
A20	VREF	Reference Voltage (VREFH – VREFL)		0V	—	—	V	
A20A				3V	—	—	V	For 10-bit resolution
A21	VREFH	Reference Voltage High		AVSS	—	AVDD + 0.3V	V	
A22	VREFL	Reference Voltage Low		AVSS – 0.3V	—	AVDD	V	
A25	VAIN	Analog Input Voltage		AVSS – 0.3V	—	VREF + 0.3V	V	
A30	ZAIN	Recommended Impedance of Analog Voltage Source		—	—	10.0	kΩ	
A40	IAD	A/D Conversion Current (VDD)	PIC18F6525/6621/8525/8621	—	180	—	μA	Average current consumption when A/D is on **(Note 1)**
			PIC18LF6X2X/8X2X	—	90	—	μA	
A50	IREF	VREF Input Current **(Note 2)**		— —	— —	5 150	μA μA	During VAIN acquisition. During A/D conversion cycle.

Legend: TBD = To Be Determined

Note 1: When A/D is off, it will not consume any current other than minor leakage current. The power-down current spec includes any such leakage from the A/D module.
VREF current is from RA2/AN2/VREF- and RA3/AN3/VREF+ pins or AVDD and AVSS pins, whichever is selected as reference input.

2: VSS ≤ VAIN ≤ VREF

3: The A/D conversion result never decreases with an increase in the input voltage and has no missing codes.

NOTES:

FIGURE 28-29: A/D NONLINEARITY vs. VREFH (VDD = 5V, -40°C TO +125°C)