Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	dsPIC
Core Size	16-Bit
Speed	30 MIPs
Connectivity	CANbus, I²C, SPI, UART/USART
Peripherals	AC'97, Brown-out Detect/Reset, I²S, LVD, POR, PWM, WDT
Number of I/O	68
Program Memory Size	144KB (48K x 24)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.5V ~ 5.5V
Data Converters	A/D 16x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	80-TQFP
Supplier Device Package	80-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/dspic30f6014a-30i-pf

FIGURE 2-1: PROGRAMMER'S MODEL

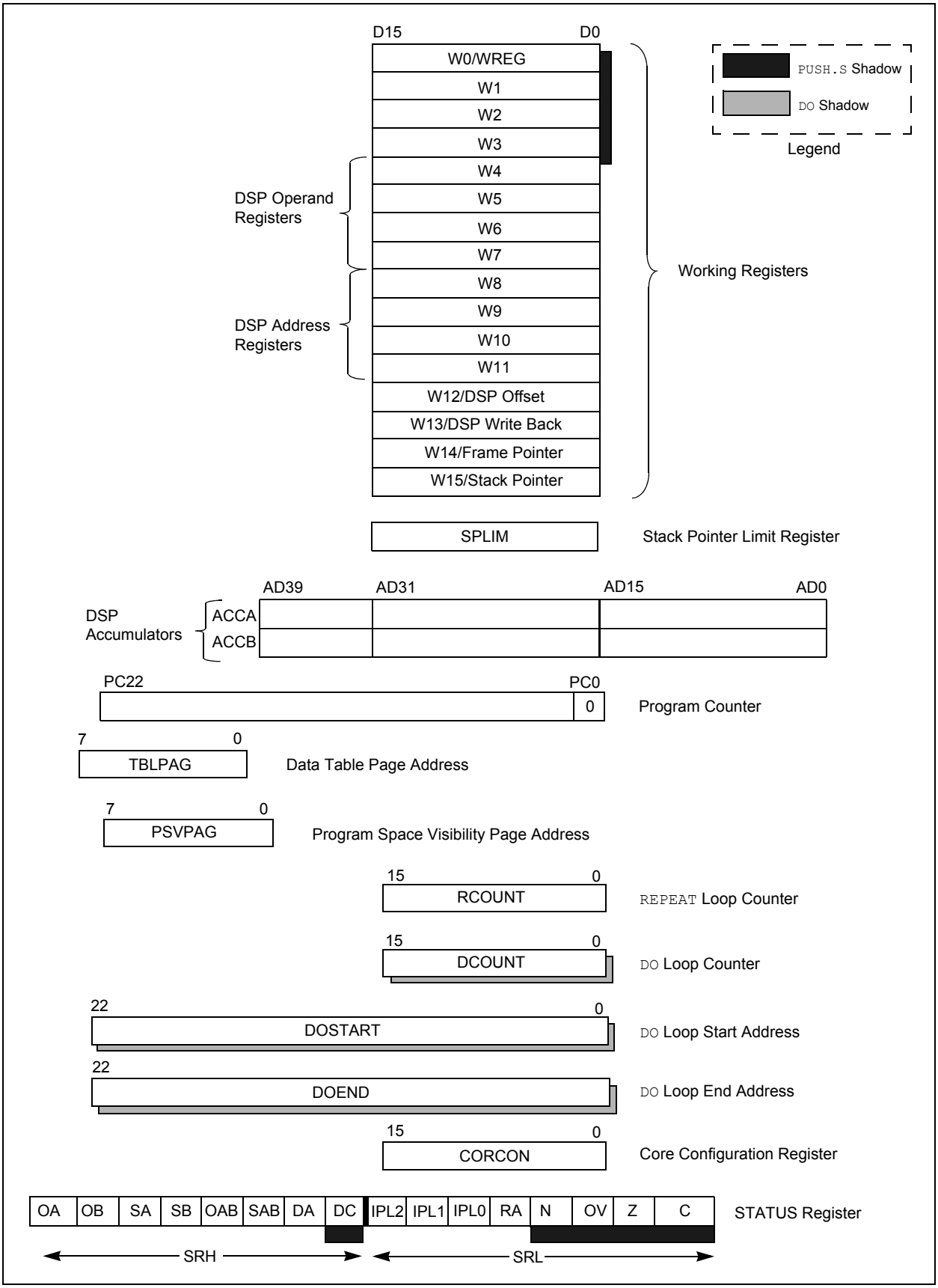
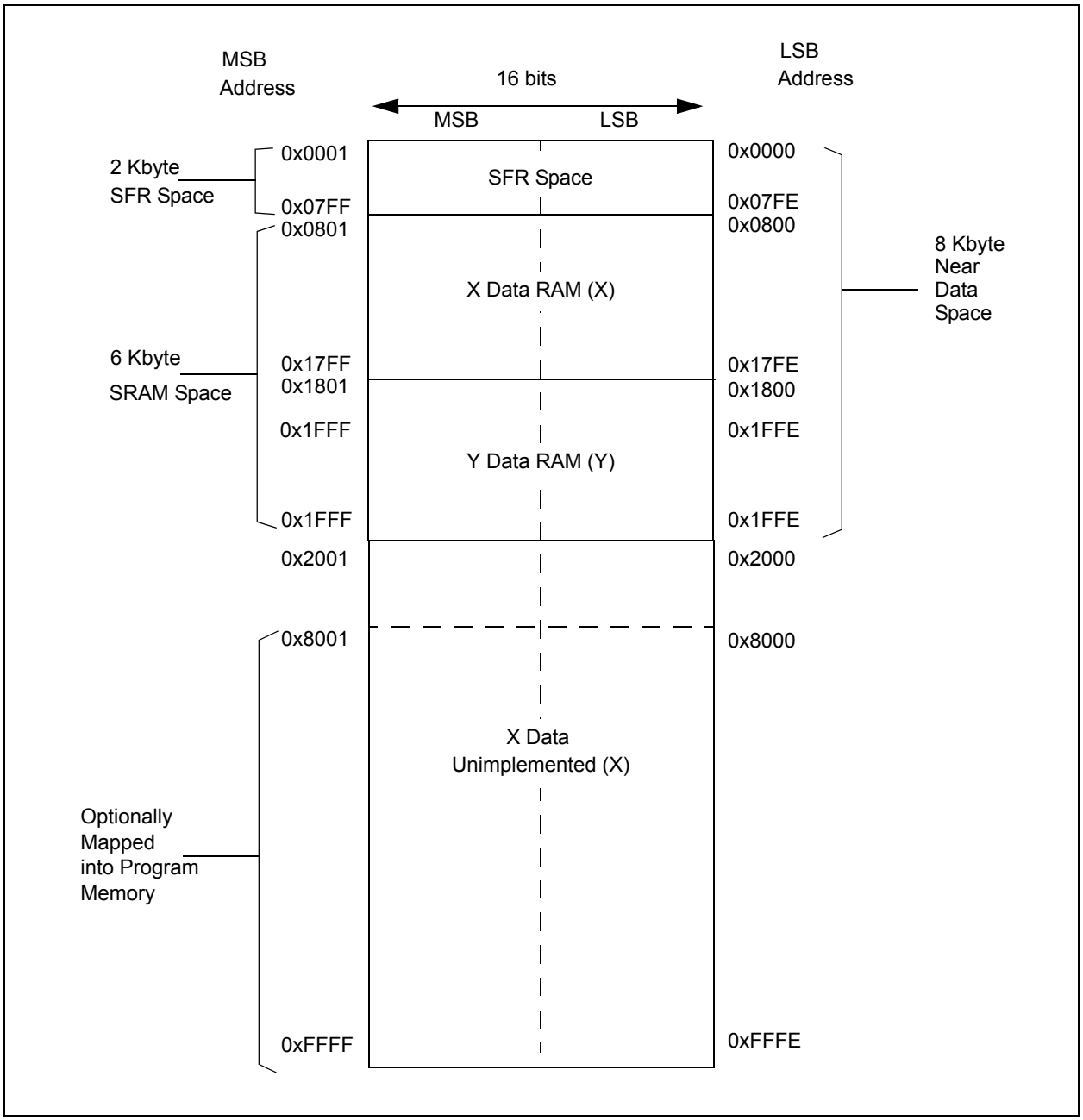
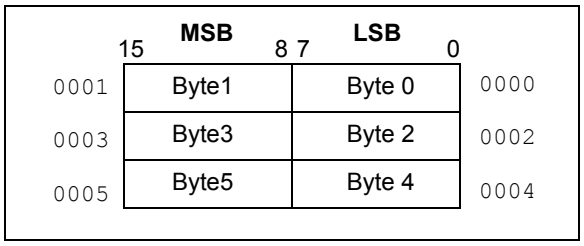


FIGURE 3-7: DATA SPACE MEMORY MAP FOR dsPIC30F6011A/6013A



All word accesses must be aligned to an even address. Misaligned word data fetches are not supported so care must be taken when mixing byte and word operations, or translating from 8-bit MCU code. Should a misaligned read or write be attempted, an address error trap will be generated. If the error occurred on a read, the instruction underway is completed, whereas if it occurred on a write, the instruction will be executed but the write will not occur. In either case, a trap will then be executed, allowing the system and/or user to examine the machine state prior to execution of the address fault.

FIGURE 3-10: DATA ALIGNMENT



All byte loads into any W register are loaded into the LSB. The MSB is not modified.

A sign-extend (SE) instruction is provided to allow users to translate 8-bit signed data to 16-bit signed values. Alternatively, for 16-bit unsigned data, users can clear the MSB of any W register by executing a zero-extend (ZE) instruction on the appropriate address.

Although most instructions are capable of operating on word or byte data sizes, it should be noted that some instructions, including the DSP instructions, operate only on words.

3.2.5 NEAR DATA SPACE

An 8 Kbyte 'near' data space is reserved in X address memory space between 0x0000 and 0x1FFF, which is directly addressable via a 13-bit absolute address field within all memory direct instructions. The remaining X address space and all of the Y address space is addressable indirectly. Additionally, the whole of X data space is addressable using MOV instructions, which support memory direct addressing with a 16-bit address field.

3.2.6 SOFTWARE STACK

The dsPIC DSC devices contain a software stack. W15 is used as the Stack Pointer.

The Stack Pointer always points to the first available free word and grows from lower addresses towards higher addresses. It pre-decrements for stack pops and post-increments for stack pushes as shown in Figure 3-11. Note that for a PC push during any CALL instruction, the MSB of the PC is zero-extended before the push, ensuring that the MSB is always clear.

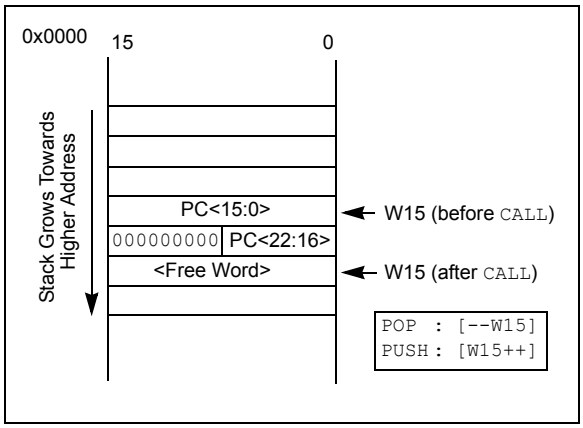
Note:	A PC push during exception processing will concatenate the SRL register to the MSB of the PC prior to the push.

There is a Stack Pointer Limit register (SPLIM) associated with the Stack Pointer. SPLIM is uninitialized at Reset. As is the case for the Stack Pointer, SPLIM<0> is forced to '0' because all stack operations must be word aligned. Whenever an Effective Address (EA) is generated using W15 as a source or destination pointer, the address thus generated is compared with the value in SPLIM. If the contents of the Stack Pointer (W15) and the SPLIM register are equal and a push operation is performed, a Stack Error Trap will not occur. The Stack Error Trap will occur on a subsequent push operation. Thus, for example, if it is desirable to cause a Stack Error Trap when the stack grows beyond address 0x2000 in RAM, initialize the SPLIM with the value, 0x1FFE.

Similarly, a Stack Pointer underflow (stack error) trap is generated when the Stack Pointer address is found to be less than 0x0800, thus preventing the stack from interfering with the Special Function Register (SFR) space.

A write to the SPLIM register should not be immediately followed by an indirect read operation using W15.

FIGURE 3-11: CALL STACK FRAME



4.0 ADDRESS GENERATOR UNITS

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "*dsPIC30F Family Reference Manual*" (DS70046). For more information on the device instruction set and programming, refer to the *"16-bit MCU and DSC Programmer's Reference Manual"* (DS70157).

The dsPIC DSC core contains two independent address generator units: the X AGU and Y AGU. The Y AGU supports word sized data reads for the DSP MAC class of instructions only. The dsPIC30F AGUs support:

- Linear Addressing
- Modulo (Circular) Addressing
- Bit-Reversed Addressing

Linear and Modulo Data Addressing modes can be applied to data space or program space. Bit-Reversed Addressing is only applicable to data space addresses.

4.1 Instruction Addressing Modes

The Addressing modes in Table 4-1 form the basis of the Addressing modes optimized to support the specific features of individual instructions. The Addressing modes provided in the MAC class of instructions are somewhat different from those in the other instruction types.

4.1.1 FILE REGISTER INSTRUCTIONS

Most File register instructions use a 13-bit address field (f) to directly address data present in the first 8192 bytes of data memory (Near data space). Most File register instructions employ a working register, W0, which is denoted as WREG in these instructions. The destination is typically either the same File register or WREG (with the exception of the MUL instruction), which writes the result to a register or register pair. The MOV instruction allows additional flexibility and can access the entire data space.

4.1.2 MCU INSTRUCTIONS

The three-operand MCU instructions are of the form:

```
Operand 3 = Operand 1 <function> Operand 2
```

where:

Operand 1 is always a working register (i.e., the Addressing mode can only be Register Direct) which is referred to as Wb.

Operand 2 can be a W register, fetched from data memory or a 5-bit literal. The result location can be either a W register or a data memory location. The following addressing modes are supported by MCU instructions:

- Register Direct
- Register Indirect
- Register Indirect Post-modified
- Register Indirect Pre-modified
- 5-bit or 10-bit Literal

Note: Not all instructions support all the addressing modes given above. Individual instructions may support different subsets of these addressing modes.

TABLE 4-1: FUNDAMENTAL ADDRESSING MODES SUPPORTED

Addressing Mode	Description
File Register Direct	The address of the File register is specified explicitly.
Register Direct	The contents of a register are accessed directly.
Register Indirect	The contents of Wn forms the EA.
Register Indirect Post-modified	The contents of Wn forms the EA. Wn is post-modified (incremented or decremented) by a constant value.
Register Indirect Pre-modified	Wn is pre-modified (incremented or decremented) by a signed constant value to form the EA.
Register Indirect with Register Offset	The sum of Wn and Wb forms the EA.
Register Indirect with Literal Offset	The sum of Wn and a literal forms the EA.

NOTES:

6.0 FLASH PROGRAM MEMORY

Note: This data sheet summarizes features of this group of dsPIC30F devices and is not intended to be a complete reference source. For more information on the CPU, peripherals, register descriptions and general device functionality, refer to the "*dsPIC30F Family Reference Manual*" (DS70046). For more information on the device instruction set and programming, refer to the *"16-bit MCU and DSC Programmer's Reference Manual"* (DS70157).

The dsPIC30F family of devices contains internal program Flash memory for executing user code. There are two methods by which the user can program this memory:

- Run-Time Self-Programming (RTSP)
- In-Circuit Serial Programming™ (ICSP™)

6.1 In-Circuit Serial Programming (ICSP)

dsPIC30F devices can be serially programmed while in the end application circuit. This is simply done with two lines for Programming Clock and Programming Data (which are named PGC and PGD respectively), and three other lines for Power (V_{DD}), Ground (V_{SS}) and Master Clear ($\overline{MCLR}$). this allows customers to manufacture boards with unprogrammed devices, and then program the microcontroller just before shipping the product. This also allows the most recent firmware or a custom firmware to be programmed.

6.2 Run-Time Self-Programming (RTSP)

RTSP is accomplished using `TBLRD` (table read) and `TBLWT` (table write) instructions.

With RTSP, the user may erase program memory, 32 instructions (96 bytes) at a time and can write program memory data, 32 instructions (96 bytes) at a time.

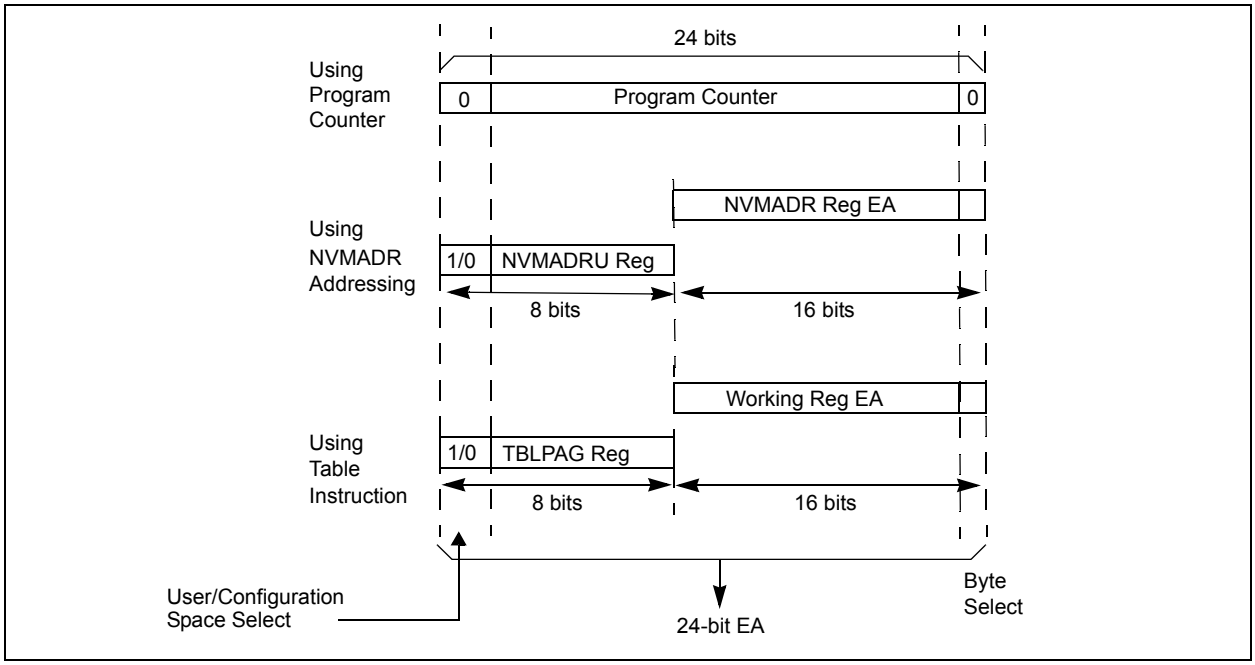
6.3 Table Instruction Operation Summary

The `TBLRDL` and the `TBLWTL` instructions are used to read or write to bits<15:0> of program memory. `TBLRDL` and `TBLWTL` can access program memory in Word or Byte mode.

The `TBLRDH` and `TBLWTH` instructions are used to read or write to bits<23:16> of program memory. `TBLRDH` and `TBLWTH` can access program memory in Word or Byte mode.

A 24-bit program memory address is formed using bits<7:0> of the TBLPAG register and the Effective Address (EA) from a W register specified in the table instruction, as shown in Figure 6-1.

FIGURE 6-1: ADDRESSING FOR TABLE AND NVM REGISTERS



6.4 RTSP Operation

The dsPIC30F Flash program memory is organized into rows and panels. Each row consists of 32 instructions or 96 bytes. Each panel consists of 128 rows or 4K x 24 instructions. RTSP allows the user to erase one row (32 instructions) at a time and to program four instructions at one time. RTSP may be used to program multiple program memory panels, but the table pointer must be changed at each panel boundary.

Each panel of program memory contains write latches that hold 32 instructions of programming data. Prior to the actual programming operation, the write data must be loaded into the panel write latches. The data to be programmed into the panel is loaded in sequential order into the write latches: instruction 0, instruction 1, etc. The instruction words loaded must always be from a group of 32 boundary.

The basic sequence for RTSP programming is to set up a table pointer, then do a series of `TBLWT` instructions to load the write latches. Programming is performed by setting the special bits in the NVMCON register. Four `TBLWTL` and four `TBLWTH` instructions are required to load the four instructions. If multiple panel programming is required, the table pointer needs to be changed and the next set of multiple write latches written.

All of the table write operations are single word writes (2 instruction cycles) because only the table latches are written. A programming cycle is required for programming each row.

The Flash program memory is readable, writable, and erasable during normal operation over the entire VDD range.

6.5 Control Registers

The four SFRs used to read and write the program Flash memory are:

- NVMCON
- NVMADR
- NVMADRU
- NVMKEY

6.5.1 NVMCON REGISTER

The NVMCON register controls which blocks are to be erased, which memory type is to be programmed and start of the programming cycle.

6.5.2 NVMADR REGISTER

The NVMADR register is used to hold the lower two bytes of the Effective Address. The NVMADR register captures the EA<15:0> of the last table instruction that has been executed and selects the row to write.

6.5.3 NVMADRU REGISTER

The NVMADRU register is used to hold the upper byte of the Effective Address. The NVMADRU register captures the EA<23:16> of the last table instruction that has been executed.

6.5.4 NVMKEY REGISTER

NVMKEY is a write only register that is used for write protection. To start a programming or an erase sequence, the user must consecutively write 0x55 and 0xAA to the NVMKEY register. Refer to **Section 6.6 "Programming Operations"** for further details.

Note:	The user can also directly write to the NVMADR and NVMADRU registers to specify a program memory address for erasing or programming.

6.6 Programming Operations

A complete programming sequence is necessary for programming or erasing the internal Flash in RTSP mode. A programming operation is nominally 2 msec in duration and the processor stalls (waits) until the operation is finished. Setting the WR bit (NVMCON<15>) starts the operation, and the WR bit is automatically cleared when the operation is finished.

6.6.1 PROGRAMMING ALGORITHM FOR PROGRAM FLASH

The user can erase and program one row of program Flash memory at a time. The general process is:

1. Read one row of program Flash (32 instruction words) and store into data RAM as a data “image”.
2. Update the data image with the desired new data.
3. Erase program Flash row.
   a) Set up NVMCON register for multi-word, program Flash, erase, and set WREN bit.
   b) Write address of row to be erased into NVMADRU/NVMADR.
   c) Write 0x55 to NVMKEY.
   d) Write 0xAA to NVMKEY.
   e) Set the WR bit. This will begin erase cycle.
   f) CPU will stall for the duration of the erase cycle.
   g) The WR bit is cleared when erase cycle ends.
4. Write 32 instruction words of data from data RAM “image” into the program Flash write latches.
5. Program 32 instruction words into program Flash.
   a) Set up NVMCON register for multi-word, program Flash, program, and set WREN bit.
   b) Write 0x55 to NVMKEY.
   c) Write 0xAA to NVMKEY.
   d) Set the WR bit. This will begin program cycle.
   e) CPU will stall for duration of the program cycle.
   f) The WR bit is cleared by the hardware when program cycle ends.
6. Repeat steps 1 through 5 as needed to program desired amount of program Flash memory.

6.6.2 ERASING A ROW OF PROGRAM MEMORY

Example 6-1 shows a code sequence that can be used to erase a row (32 instructions) of program memory.

EXAMPLE 6-1: ERASING A ROW OF PROGRAM MEMORY

```
; Setup NVMCON for erase operation, multi word write
; program memory selected, and writes enabled
       MOV     #0x4041,W0                  ;
       MOV     W0,NVMCON                   ; Init NVMCON SFR
; Init pointer to row to be ERASED
       MOV     #tblpage(PROG_ADDR),W0      ;
       MOV     W0,NVMADRU                  ; Initialize PM Page Boundary SFR
       MOV     #tbloffset(PROG_ADDR),W0    ; Intialize in-page EA[15:0] pointer
       MOV     W0, NVMADR                  ; Initialize NVMADR SFR
       DISI    #5                          ; Block all interrupts with priority <7 for
                                           ; next 5 instructions
       MOV     #0x55,W0
       MOV     W0,NVMKEY                   ; Write the 0x55 key
       MOV     #0xAA,W1                    ;
       MOV     W1,NVMKEY                   ; Write the 0xAA key
       BSET    NVMCON,#WR                  ; Start the erase sequence
       NOP                                 ; Insert two NOPs after the erase
       NOP                                 ; command is asserted
```

TABLE 15-2: I²C™ REGISTER MAP(1)

SFR Name	Addr.	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset State
I2CRCV	0200	—	—	—	—	—	—	—	—	Receive Register								0000 0000 0000 0000
I2CTRN	0202	—	—	—	—	—	—	—	—	Transmit Register								0000 0000 1111 1111
I2CBRG	0204	—	—	—	—	—	—	—	Baud Rate Generator									0000 0000 0000 0000
I2CCON	0206	I2CEN	—	I2CSIDL	SCLREL	IPMIEN	A10M	DISSLW	SMEN	GCEN	STREN	ACKDT	ACKEN	RCEN	PEN	RSEN	SEN	0001 0000 0000 0000
I2CSTAT	0208	ACKSTAT	TRSTAT	—	—	—	BCL	GCSTAT	ADD10	IWCOL	I2COV	D_A	P	S	R_W	RBF	TBF	0000 0000 0000 0000
I2CADD	020A	—	—	—	—	—	—	Address Register										0000 0000 0000 0000

Legend: — = unimplemented bit, read as '0'

Note 1: Refer to the *"dsPIC30F Family Reference Manual"* (DS70046) for descriptions of register bit fields.

In the I^2S mode, a frame sync signal having a 50% duty cycle is generated. The period of the I^2S frame sync signal in CSCK cycles is determined by the word size and frame sync generator control bits. A new I^2S data transfer boundary is marked by a high-to-low or a low-to-high transition edge on the COFS pin.

18.3.6 SLAVE FRAME SYNC OPERATION

When the DCI module is operating as a frame sync slave (COFSD = 1), data transfers are controlled by the Codec device attached to the DCI module. The COFSM control bits control how the DCI module responds to incoming COFS signals.

In the Multi-Channel mode, a new data frame transfer will begin one CSCK cycle after the COFS pin is sampled high (see Figure 18-2). The pulse on the COFS pin resets the frame sync generator logic.

In the I^2S mode, a new data word will be transferred one CSCK cycle after a low-to-high or a high-to-low transition is sampled on the COFS pin. A rising or falling edge on the COFS pin resets the frame sync generator logic.

In the AC-Link mode, the tag slot and subsequent data slots for the next frame will be transferred one CSCK cycle after the COFS pin is sampled high.

The COFSG and WS bits must be configured to provide the proper frame length when the module is operating in the Slave mode. Once a valid frame sync pulse has been sampled by the module on the COFS pin, an entire data frame transfer will take place. The module will not respond to further frame sync pulses until the data frame transfer has completed.

FIGURE 18-2: FRAME SYNC TIMING, MULTI-CHANNEL MODE

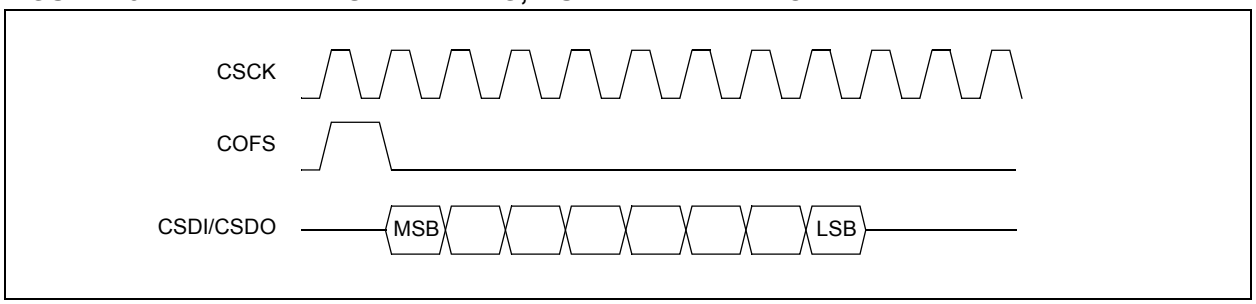


FIGURE 18-3: FRAME SYNC TIMING, AC-LINK START OF FRAME

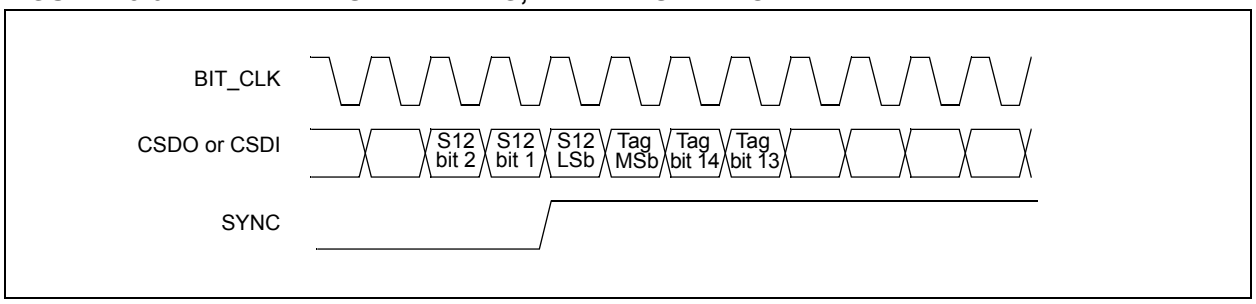
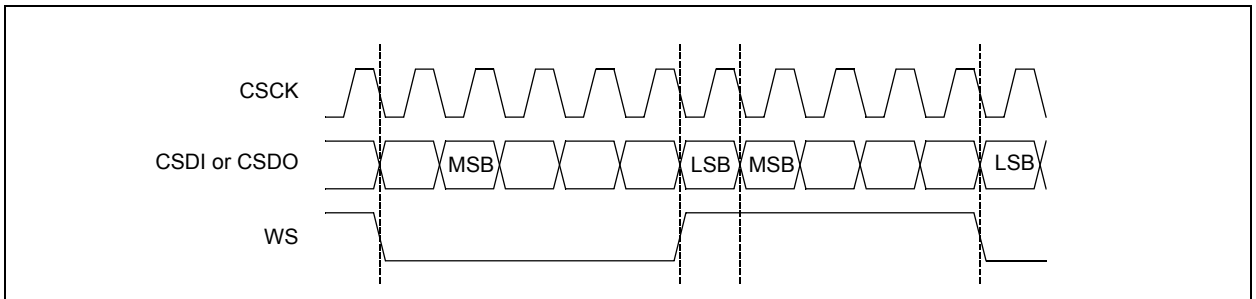


FIGURE 18-4: I^2S INTERFACE FRAME SYNC TIMING



Note: A 5-bit transfer is shown here for illustration purposes. The I^2S protocol does not specify word length – this will be system dependent.

19.1 ADC Result Buffer

The module contains a 16-word dual port read only buffer, called ADCBUF0...ADCBUFF, to buffer the ADC results. The RAM is 12 bits wide but the data obtained is represented in one of four different 16-bit data formats. The contents of the sixteen ADC Result Buffer registers, ADCBUF0 through ADCBUFF, cannot be written by user software.

19.2 Conversion Operation

After the ADC module has been configured, the sample acquisition is started by setting the SAMP bit. Various sources, such as a programmable bit, timer time-outs and external events, will terminate acquisition and start a conversion. When the A/D conversion is complete, the result is loaded into ADCBUF0...ADCBUFF, and the DONE bit and the ADC interrupt flag ADIF are set after the number of samples specified by the SMPI bit. The ADC module can be configured for different interrupt rates as described in **Section 19.3 "Selecting the Conversion Sequence"**.

Use the following steps to perform an Analog-to-Digital conversion:

1. Configure the ADC module:
   a) Configure the analog pins, voltage reference and digital I/O.
   b) Select the ADC input channels.
   c) Select the ADC conversion clock.
   d) Select the ADC conversion trigger.
   e) Turn on the ADC module.
2. Configure ADC interrupt (if required):
   a) Clear the ADIF bit.
   b) Select the ADC interrupt priority.
3. Start sampling.
4. Wait the required acquisition time.
5. Trigger acquisition end, start conversion:
6. Wait for ADC conversion to complete, by either:
   • Waiting for the ADC interrupt, or
   • Waiting for the DONE bit to get set.
7. Read ADC result buffer, clear ADIF if required.

19.3 Selecting the Conversion Sequence

Several groups of control bits select the sequence in which the ADC connects inputs to the sample/hold channel, converts a channel, writes the buffer memory and generates interrupts.

The sequence is controlled by the sampling clocks.

The SMPI bits select the number of acquisition/conversion sequences that would be performed before an interrupt occurs. This can vary from 1 sample per interrupt to 16 samples per interrupt.

The BUFM bit will split the 16-word results buffer into two 8-word groups. Writing to the 8-word buffers will be alternated on each interrupt event.

Use of the BUFM bit will depend on how much time is available for the moving of the buffers after the interrupt.

If the processor can quickly unload a full buffer within the time it takes to acquire and convert one channel, the BUFM bit can be '0' and up to 16 conversions (corresponding to the 16 input channels) may be done per interrupt. The processor will have one acquisition and conversion time to move the sixteen conversions.

If the processor cannot unload the buffer within the acquisition and conversion time, the BUFM bit should be '1'. For example, if SMPI<3:0> (ADCON2<5:2>) = 0111, then eight conversions will be loaded into 1/2 of the buffer, following which an interrupt occurs. The next eight conversions will be loaded into the other 1/2 of the buffer. The processor will have the entire time between interrupts to move the eight conversions.

The ALTS bit can be used to alternate the inputs selected during the sampling sequence. The input multiplexer has two sets of sample inputs: MUX A and MUX B. If the ALTS bit is '0', only the MUX A inputs are selected for sampling. If the ALTS bit is '1' and SMPI<3:0> = 0000 on the first sample/convert sequence, the MUX A inputs are selected and, on the next acquire/convert sequence, the MUX B inputs are selected.

The CSCNA bit (ADCON2<10>) will allow the multiplexer input to be alternately scanned across a selected number of analog inputs for the MUX A group. The inputs are selected by the ADCSSL register. If a particular bit in the ADCSSL register is '1', the corresponding input is selected. The inputs are always scanned from lower to higher numbered inputs, starting after each interrupt. If the number of inputs selected is greater than the number of samples taken per interrupt, the higher numbered inputs are unused.

TABLE 20-1: OSCILLATOR OPERATING MODES

Oscillator Mode	Description
XTL	200 kHz-4 MHz crystal on OSC1:OSC2
XT	4 MHz-10 MHz crystal on OSC1:OSC2
XT w/PLL 4x	4 MHz-10 MHz crystal on OSC1:OSC2, 4x PLL enabled
XT w/PLL 8x	4 MHz-10 MHz crystal on OSC1:OSC2, 8x PLL enabled
XT w/PLL 16x	4 MHz-7.5 MHz crystal on OSC1:OSC2, 16x PLL enabled(1)
LP	32 kHz crystal on SOSCO:SOSCI(2)
HS	10 MHz-25 MHz crystal.
HS/2 w/PLL 4x	10 MHz-20 MHz crystal, divide by 2, 4x PLL enabled(3)
HS/2 w/PLL 8x	10 MHz-20 MHz crystal, divide by 2, 8x PLL enabled(3)
HS/2 w/PLL 16x	10 MHz-15 MHz crystal, divide by 2, 16x PLL enabled(1)
HS/3 w/PLL 4x	12 MHz-25 MHz crystal, divide by 3, 4x PLL enabled(4)
HS/3 w/PLL 8x	12 MHz-25 MHz crystal, divide by 3, 8x PLL enabled(4)
HS/3 w/PLL 16x	12 MHz-22.5 MHz crystal, divide by 3, 16x PLL enabled(1)(4)
EC	External clock input (0-40 MHz)
ECIO	External clock input (0-40 MHz), OSC2 pin is I/O
EC w/PLL 4x	External clock input (4-10 MHz), OSC2 pin is I/O, 4x PLL enabled
EC w/PLL 8x	External clock input (4-10 MHz), OSC2 pin is I/O, 8x PLL enabled
EC w/PLL 16x	External clock input (4-7.5 MHz), OSC2 pin is I/O, 16x PLL enabled(1)
ERC	External RC oscillator, OSC2 pin is FOSC/4 output(5)
ERCIO	External RC oscillator, OSC2 pin is I/O(5)
FRC	7.37 MHz internal RC oscillator
FRC w/PLL 4x	7.37 MHz internal RC oscillator, 4x PLL enabled
FRC w/PLL 8x	7.37 MHz internal RC oscillator, 8x PLL enabled
FRC w/PLL 16x	7.37 MHz internal RC oscillator, 16x PLL enabled
LPRC	512 kHz internal RC oscillator

Note 1: Any higher will violate device operating frequency range.

2: LP oscillator can be conveniently shared as system clock, as well as Real-Time Clock for Timer1.

3: Any higher will violate PLL input range.

4: Any lower will violate PLL input range.

5: Requires external R and C. Frequency operation up to 4 MHz.

20.2.6 LOW-POWER RC OSCILLATOR (LPRC)

The LPRC oscillator is a component of the Watchdog Timer (WDT) and oscillates at a nominal frequency of 512 kHz. The LPRC oscillator is the clock source for the Power-up Timer (PWRT) circuit, WDT and clock monitor circuits. It may also be used to provide a low frequency clock source option for applications where power consumption is critical, and timing accuracy is not required.

The LPRC oscillator is always enabled at a Power-on Reset, because it is the clock source for the PWRT. After the PWRT expires, the LPRC oscillator will remain ON if one of the following is TRUE:

- The Fail-Safe Clock Monitor is enabled
- The WDT is enabled
- The LPRC oscillator is selected as the system clock via the COSC<2:0> control bits in the OSCCON register

If one of the above conditions is not true, the LPRC will shut-off after the PWRT expires.

Note 1:	OSC2 pin function is determined by the Primary Oscillator mode selection (FPR<4:0>).
2:	Note that OSC1 pin cannot be used as an I/O pin, even if the secondary oscillator or an internal clock source is selected at all times.

20.2.7 FAIL-SAFE CLOCK MONITOR

The Fail-Safe Clock Monitor (FSCM) allows the device to continue to operate even in the event of an oscillator failure. The FSCM function is enabled by appropriately programming the FCKSM Configuration bits (Clock Switch and Monitor Selection bits) in the FOSC device Configuration register. If the FSCM function is enabled, the LPRC internal oscillator will run at all times (except during Sleep mode) and will not be subject to control by the SWDTEN bit.

In the event of an oscillator failure, the FSCM will generate a clock failure trap event and will switch the system clock over to the FRC oscillator. The user will then have the option to either attempt to restart the oscillator or execute a controlled shutdown. The user may decide to treat the trap as a warm Reset by simply loading the Reset address into the oscillator fail trap vector. In this event, the CF (Clock Fail) status bit (OSCCON<3>) is also set whenever a clock failure is recognized.

In the event of a clock failure, the WDT is unaffected and continues to run on the LPRC clock.

If the oscillator has a very slow start-up time coming out of POR, BOR or Sleep, it is possible that the PWRT timer will expire before the oscillator has started. In such cases, the FSCM will be activated and the FSCM will initiate a clock failure trap, and the COSC<2:0> bits are loaded with FRC oscillator selection. This will effectively shut-off the original oscillator that was trying to start.

The user may detect this situation and restart the oscillator in the clock fail trap ISR.

Upon a clock failure detection, the FSCM module will initiate a clock switch to the FRC oscillator as follows:

1. The COSC bits (OSCCON<14:12>) are loaded with the FRC oscillator selection value.
2. CF bit is set (OSCCON<3>).
3. OSWEN control bit (OSCCON<0>) is cleared.

For the purpose of clock switching, the clock sources are sectioned into four groups:

- Primary
- Secondary
- Internal FRC
- Internal LPRC

The user can switch between these functional groups, but cannot switch between options within a group. If the primary group is selected, then the choice within the group is always determined by the FPR<4:0> Configuration bits.

The OSCCON register holds the control and status bits related to clock switching.

- COSC<2:0>: Read-only status bits always reflect the current oscillator group in effect
- NOSC<2:0>: Control bits which are written to indicate the new oscillator group of choice
  - On POR and BOR, COSC<2:0> and NOSC<2:0> are both loaded with the Configuration bit values FOS<2:0>
- LOCK: The LOCK status bit indicates a PLL lock
- CF: Read-only status bit indicating if a clock fail detect has occurred
- OSWEN: Control bit changes from a '0' to a '1' when a clock transition sequence is initiated. Clearing the OSWEN control bit will abort a clock transition in progress (used for hang-up situations).

If Configuration bits FCKSM<1:0> = 1x, then the clock switching and Fail-Safe Clock Monitor functions are disabled. This is the default Configuration bit setting.

If clock switching is disabled, then the FOS<2:0> and FPR<4:0> bits directly control the oscillator selection and the COSC<2:0> bits do not control the clock selection. However, these bits will reflect the clock source selection.

Note:	The application should not attempt to switch to a clock of frequency lower than 100 kHz when the Fail-Safe Clock Monitor is enabled. If clock switching is performed, the device may generate an oscillator fail trap and switch to the fast RC oscillator.

FIGURE 20-3: TIME-OUT SEQUENCE ON POWER-UP (MCLR TIED TO VDD)

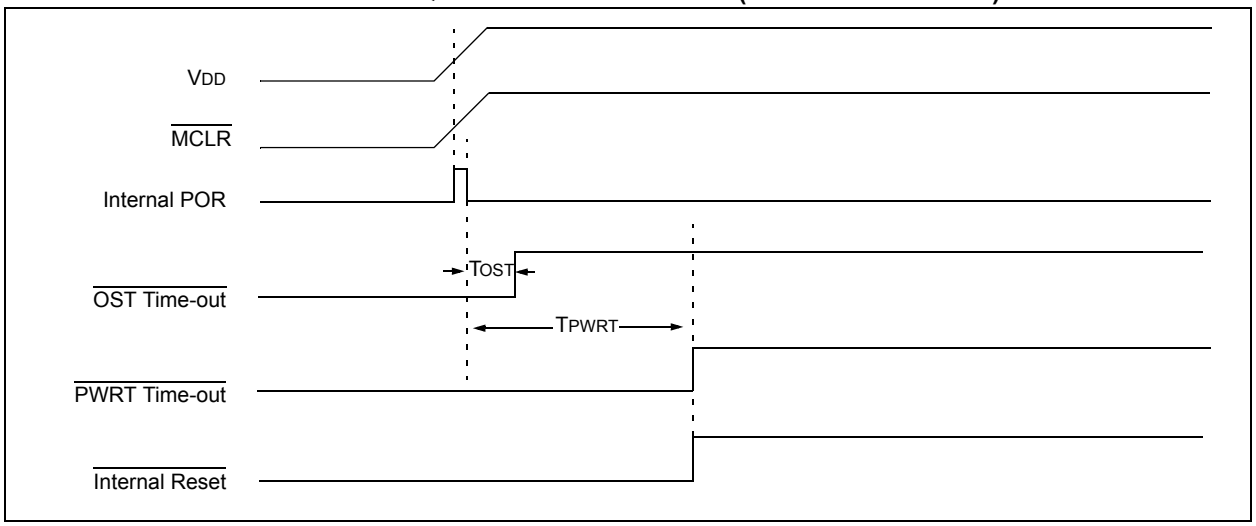


FIGURE 20-4: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 1

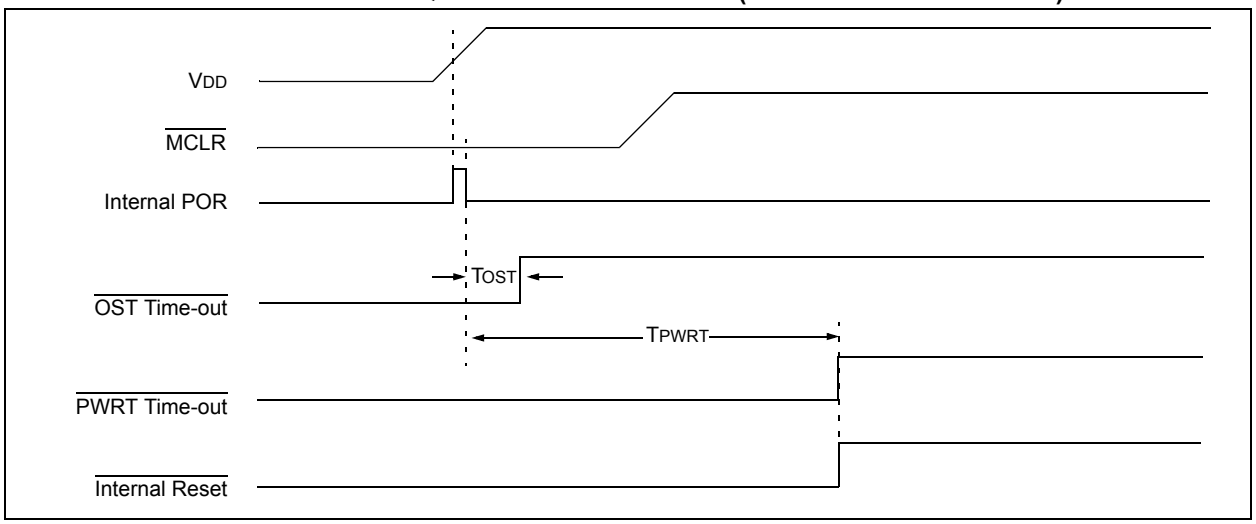


FIGURE 20-5: TIME-OUT SEQUENCE ON POWER-UP (MCLR NOT TIED TO VDD): CASE 2

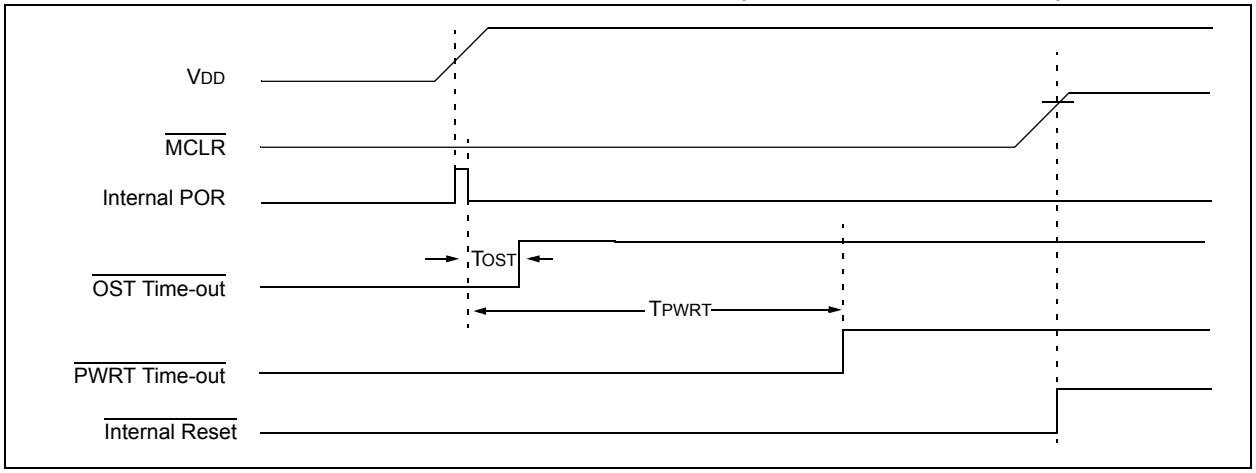


Table 20-5 shows the Reset conditions for the RCON register. Since the control bits within the RCON register are R/W, the information in the table implies that all the bits are negated prior to the action specified in the condition column.

TABLE 20-5: INITIALIZATION CONDITION FOR RCON REGISTER CASE 1

Condition	Program Counter	TRAPR	IOPUWR	EXTR	SWR	WDTO	IDLE	SLEEP	POR	BOR
Power-on Reset	0x000000	0	0	0	0	0	0	0	1	1
Brown-out Reset	0x000000	0	0	0	0	0	0	0	0	1
$\overline{MCLR}$ Reset during normal operation	0x000000	0	0	1	0	0	0	0	0	0
Software Reset during normal operation	0x000000	0	0	0	1	0	0	0	0	0
$\overline{MCLR}$ Reset during Sleep	0x000000	0	0	1	0	0	0	1	0	0
$\overline{MCLR}$ Reset during Idle	0x000000	0	0	1	0	0	1	0	0	0
WDT Time-out Reset	0x000000	0	0	0	0	1	0	0	0	0
WDT Wake-up	PC + 2	0	0	0	0	1	0	1	0	0
Interrupt Wake-up from Sleep	PC + 2(1)	0	0	0	0	0	0	1	0	0
Clock Failure Trap	0x000004	0	0	0	0	0	0	0	0	0
Trap Reset	0x000000	1	0	0	0	0	0	0	0	0
Illegal Operation Trap	0x000000	0	1	0	0	0	0	0	0	0

Note 1: When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

Table 20-6 shows a second example of the bit conditions for the RCON register. In this case, it is not assumed the user has set/cleared specific bits prior to action specified in the condition column.

TABLE 20-6: INITIALIZATION CONDITION FOR RCON REGISTER CASE 2

Condition	Program Counter	TRAPR	IOPUWR	EXTR	SWR	WDTO	IDLE	SLEEP	POR	BOR
Power-on Reset	0x000000	0	0	0	0	0	0	0	1	1
Brown-out Reset	0x000000	u	u	u	u	u	u	u	0	1
$\overline{MCLR}$ Reset during normal operation	0x000000	u	u	1	0	0	0	0	u	u
Software Reset during normal operation	0x000000	u	u	0	1	0	0	0	u	u
$\overline{MCLR}$ Reset during Sleep	0x000000	u	u	1	u	0	0	1	u	u
$\overline{MCLR}$ Reset during Idle	0x000000	u	u	1	u	0	1	0	u	u
WDT Time-out Reset	0x000000	u	u	0	0	1	0	0	u	u
WDT Wake-up	PC + 2	u	u	u	u	1	u	1	u	u
Interrupt Wake-up from Sleep	PC + 2(1)	u	u	u	u	u	u	1	u	u
Clock Failure Trap	0x000004	u	u	u	u	u	u	u	u	u
Trap Reset	0x000000	1	u	u	u	u	u	u	u	u
Illegal Operation Reset	0x000000	u	1	u	u	u	u	u	u	u

Legend: u = unchanged

Note 1: When the wake-up is due to an enabled interrupt, the PC is loaded with the corresponding interrupt vector.

TABLE 23-7: DC CHARACTERISTICS: POWER-DOWN CURRENT (I_{PD})

DC CHARACTERISTICS			Standard Operating Conditions: 2.5V to 5.5V (unless otherwise stated) Operating temperature -40°C ≤TA ≤+85°C for Industrial -40°C ≤TA ≤+125°C for Extended			
Parameter No.	**Typical**	**Max**	**Units**	**Conditions**		
Power Down Current (I_{PD})						
DC60a	0.5	—	µA	25°C	3.3V	Base Power Down Current(1)
DC60b	1	40	µA	85°C		
DC60c	24	65	µA	125°C		
DC60e	0.7	—	µA	25°C	5V	
DC60f	4	55	µA	85°C		
DC60g	35	90	µA	125°C		
DC61a	9	20	µA	25°C	3.3V	Watchdog Timer Current: ΔI_{WDT}(2)
DC61b	9	20	µA	85°C		
DC61c	8	20	µA	125°C		
DC61e	18	40	µA	25°C	5V	
DC61f	16	40	µA	85°C		
DC61g	15	40	µA	125°C		
DC62a	4	10	µA	25°C	3.3V	Timer 1 w/32 kHz Crystal: ΔI_{TI32}(2)
DC62b	5	10	µA	85°C		
DC62c	4	10	µA	125°C		
DC62e	4	15	µA	25°C	5V	
DC62f	6	15	µA	85°C		
DC62g	5	15	µA	125°C		
DC63a	30	55	µA	25°C	3.3V	BOR On: ΔI_{BOR}(2)
DC63b	34	55	µA	85°C		
DC63c	35	55	µA	125°C		
DC63e	36	60	µA	25°C	5V	
DC63f	39	60	µA	85°C		
DC63g	40	60	µA	125°C		
DC66a	20	35	µA	25°C	3.3V	Low Voltage Detect: ΔI_{LVD}(2)
DC66b	22	35	µA	85°C		
DC66c	23	35	µA	125°C		
DC66e	24	40	µA	25°C	5V	
DC66f	26	40	µA	85°C		
DC66g	26	40	µA	125°C		

Note 1: Base I_{PD} is measured with all peripherals and clocks shut down. All I/Os are configured as inputs and pulled high. LVD, BOR, WDT, etc. are all switched off.

2: The Δ current is the additional current consumed when the module is enabled. This current should be added to the base I_{PD} current.

23.2 AC Characteristics and Timing Parameters

The information contained in this section defines dsPIC30F AC characteristics and timing parameters.

TABLE 23-13: TEMPERATURE AND VOLTAGE SPECIFICATIONS – AC

AC CHARACTERISTICS	**Standard Operating Conditions: 2.5V to 5.5V** **(unless otherwise stated)** Operating temperature -40°C ≤TA ≤+85°C for Industrial -40°C ≤TA ≤+125°C for Extended Operating voltage VDD range as described in Table 23-1.

FIGURE 23-3: LOAD CONDITIONS FOR DEVICE TIMING SPECIFICATIONS

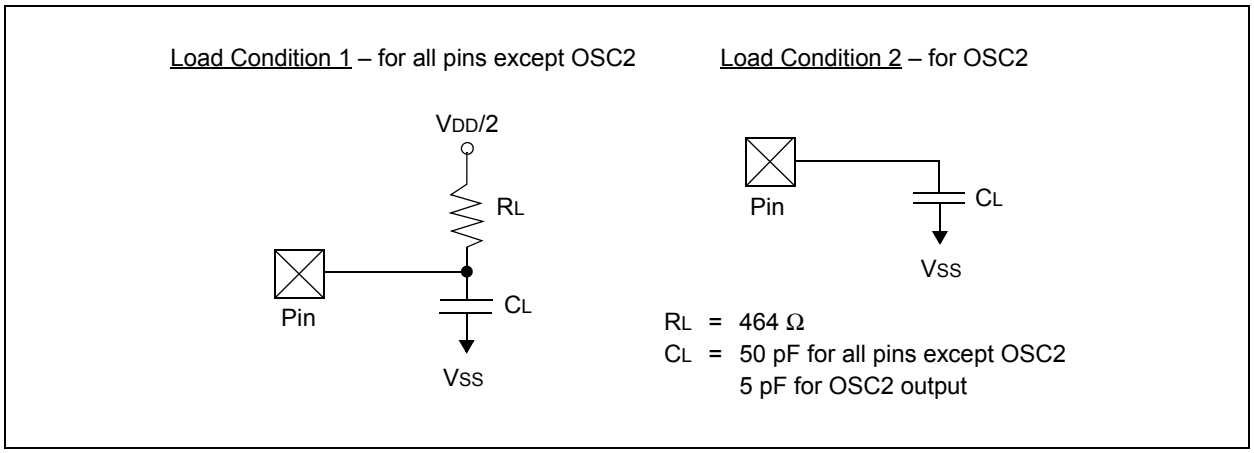


FIGURE 23-4: EXTERNAL CLOCK TIMING

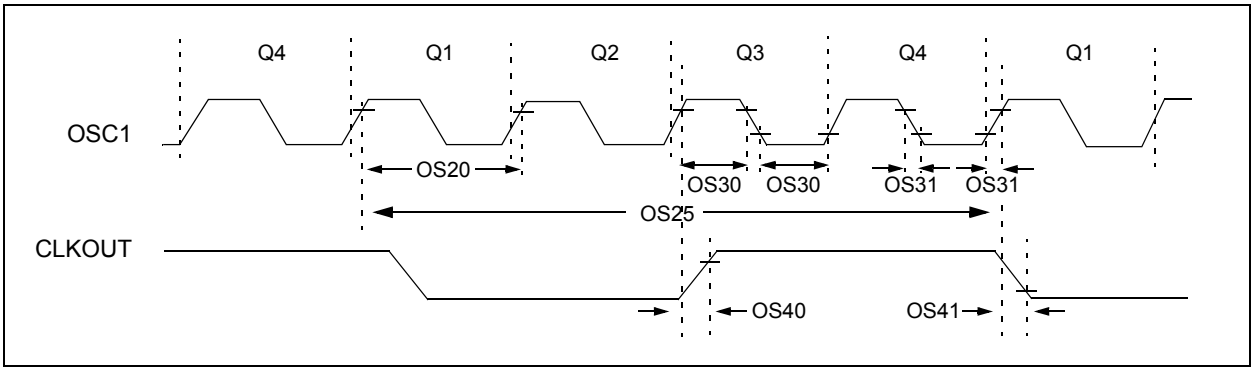
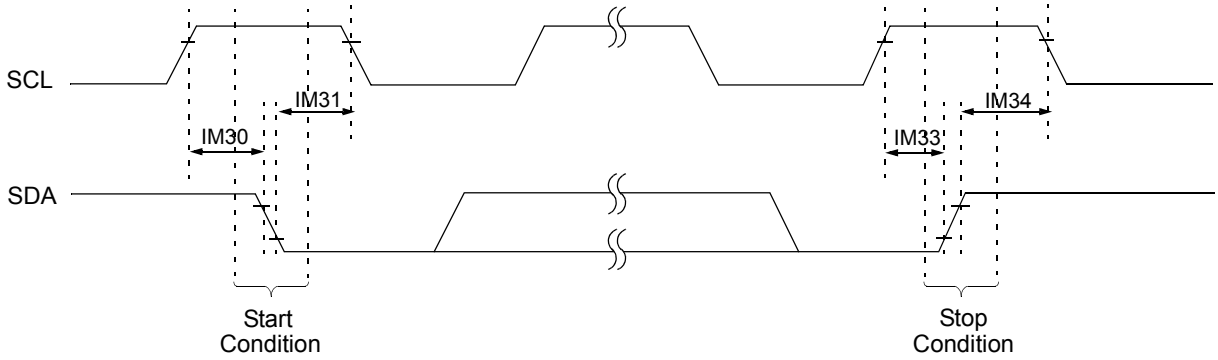
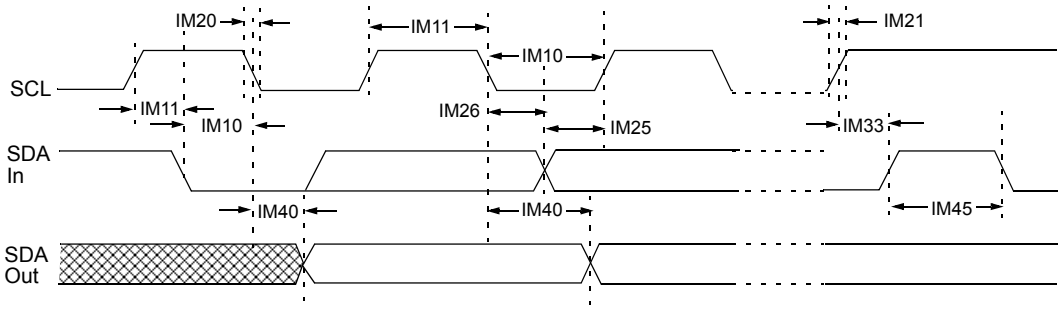


FIGURE 23-18: I²C™ BUS START/STOP BITS TIMING CHARACTERISTICS (MASTER MODE)



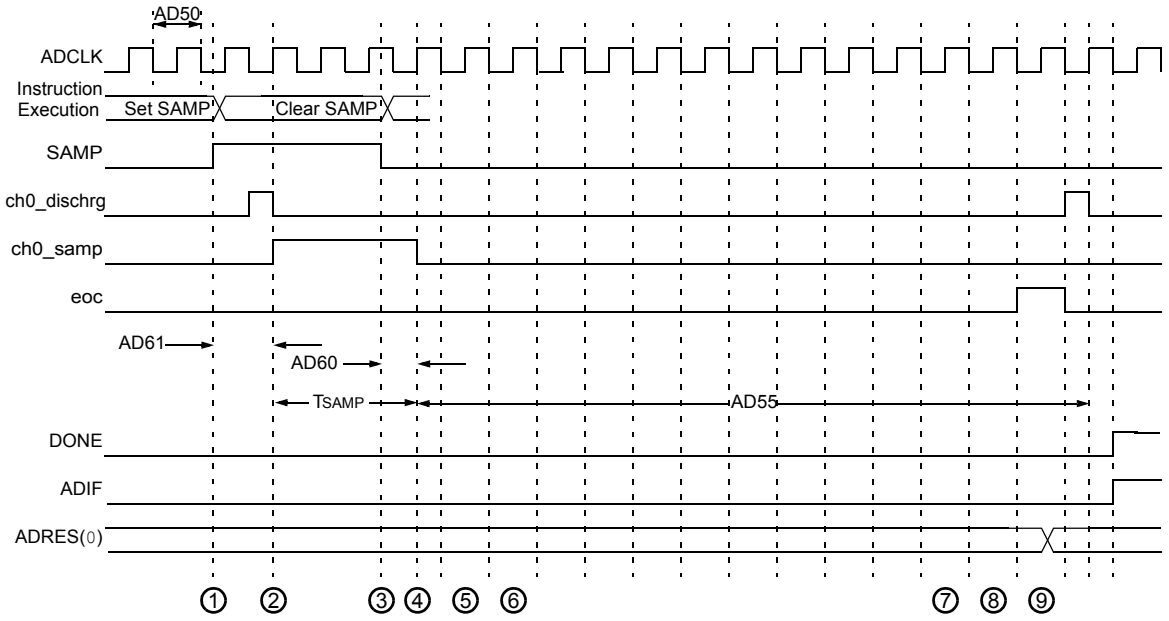
Note: Refer to Figure 23-3 for load conditions.

FIGURE 23-19: I²C™ BUS DATA TIMING CHARACTERISTICS (MASTER MODE)



Note: Refer to Figure 23-3 for load conditions.

FIGURE 23-23: 12-BIT ADC TIMING CHARACTERISTICS (ASAM = 0, SSRC = 000)



1 – Software sets ADCON. SAMP to start sampling.

2 – Sampling starts after discharge period.
TSAMP is described in **Section 18. "12-bit A/D Converter"** (DS70046) of the *"dsPIC30F Family Reference Manual"*.

3 – Software clears ADCON. SAMP to start conversion.

4 – Sampling ends, conversion sequence starts.

5 – Convert bit 11.

6 – Convert bit 10.

7 – Convert bit 1.

8 – Convert bit 0.

9 – One TAD for end of conversion.

L

M

N

O