dsPIC

31. března 2005

Obsah

1 Základ

1.1 Úvod

MICROCHIP

dsPIC30F2010

28-pin dsPIC30F2010 Enhanced Flash 16-bit Digital Signal Controller

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 *dsPIC30F Programmer's Reference Manual* (DS70030).

High-Performance Modified RISC CPU:

- Modified Harvard architecture
- C compiler optimized instruction set architecture
- 84 base instructions with flexible addressing modes
- 24-bit wide instructions, 16-bit wide data path
- 12 Kbytes on-chip Flash program space
- 512 bytes on-chip data RAM
- 1 Kbyte non-volatile data EEPROM
- 16 x 16-bit working register array
- Up to 30 MIPs operation:
  - DC to 40 MHz external clock input
  - 4 MHz-10 MHz oscillator input with PLL active (4x, 8x, 16x)
- 27 interrupt sources
- Three external interrupt sources
- 8 user selectable priority levels for each interrupt
- 4 processor exceptions and software traps

DSP Engine Features:

- Modulo and Bit-Reversed modes
- Two, 40-bit wide accumulators with optional saturation logic
- 17-bit x 17-bit single cycle hardware fractional/integer multiplier
- Single cycle Multiply-Accumulate (MAC) operation
- 40-stage Barrel Shifter
- Dual data fetch

Peripheral Features:

- High current sink/source I/O pins: 25 mA/25 mA
- Three 16-bit timers/counters; optionally pair up 16-bit timers into 32-bit timer modules
- Four 16-bit Capture input functions
- Two 16-bit Compare/PWM output functions
  - Dual Compare mode available
- 3-wire SPI™ modules (supports 4 Frame modes)
- I^2C™ module supports Multi-Master/Slave mode and 7-bit/10-bit addressing
- Addressable UART modules with FIFO buffers

Motor Control PWM Module Features:

- 6 PWM output channels
  - Complementary or Independent Output modes
  - Edge and Center Aligned modes
- 4 duty cycle generators
- Dedicated time base with 4 modes
- Programmable output polarity
- Dead time control for Complementary mode
- Manual output control
- Trigger for synchronized A/D conversions

Quadrature Encoder Interface Module Features:

- Phase A, Phase B and Index Pulse input
- 16-bit up/down position counter
- Count direction status
- Position Measurement (x2 and x4) mode
- Programmable digital noise filters on inputs
- Alternate 16-bit Timer/Counter mode
- Interrupt on position counter rollover/underflow

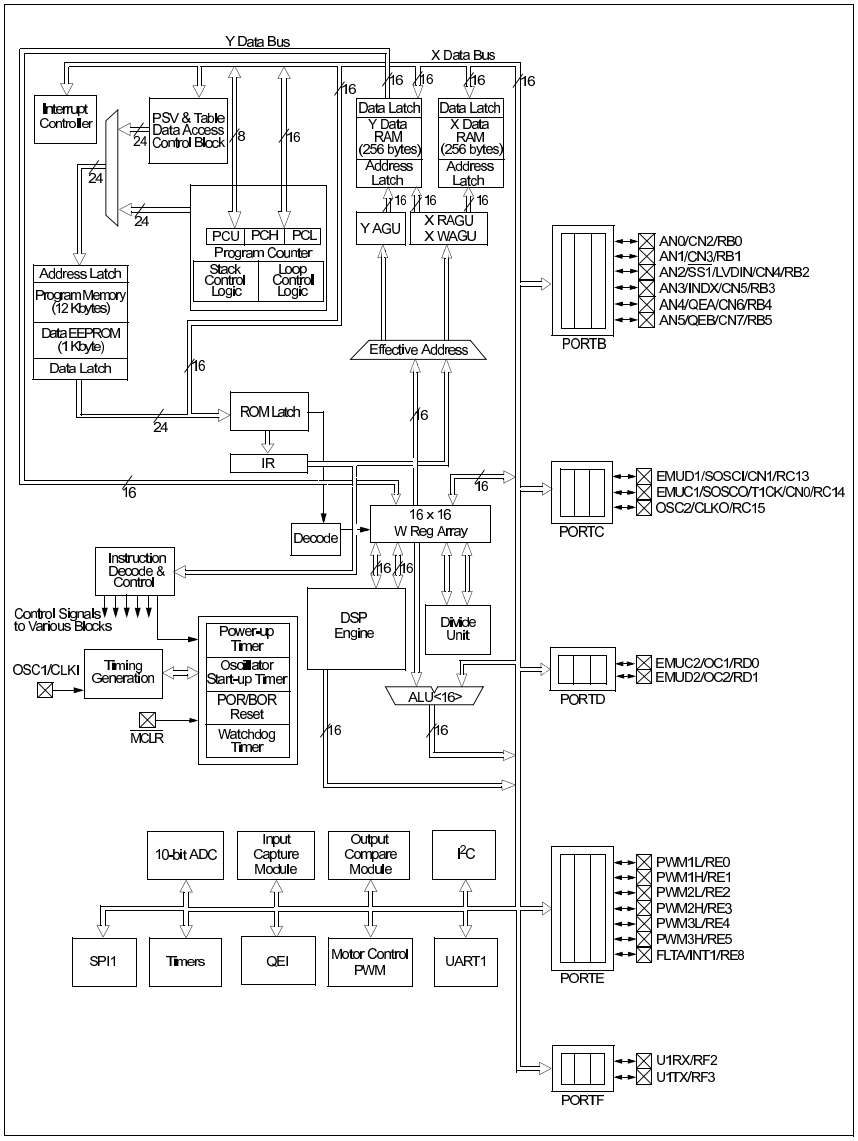
Analog Features:

- 10-bit Analog-to-Digital Converter (A/D) with:
  - 500 Ksps (for 10-bit A/D) conversion rate
  - Six input channels
  - Conversion available during Sleep and Idle
- Programmable Brown-out Detection and Reset generation

1.2 Vnitřní struktura

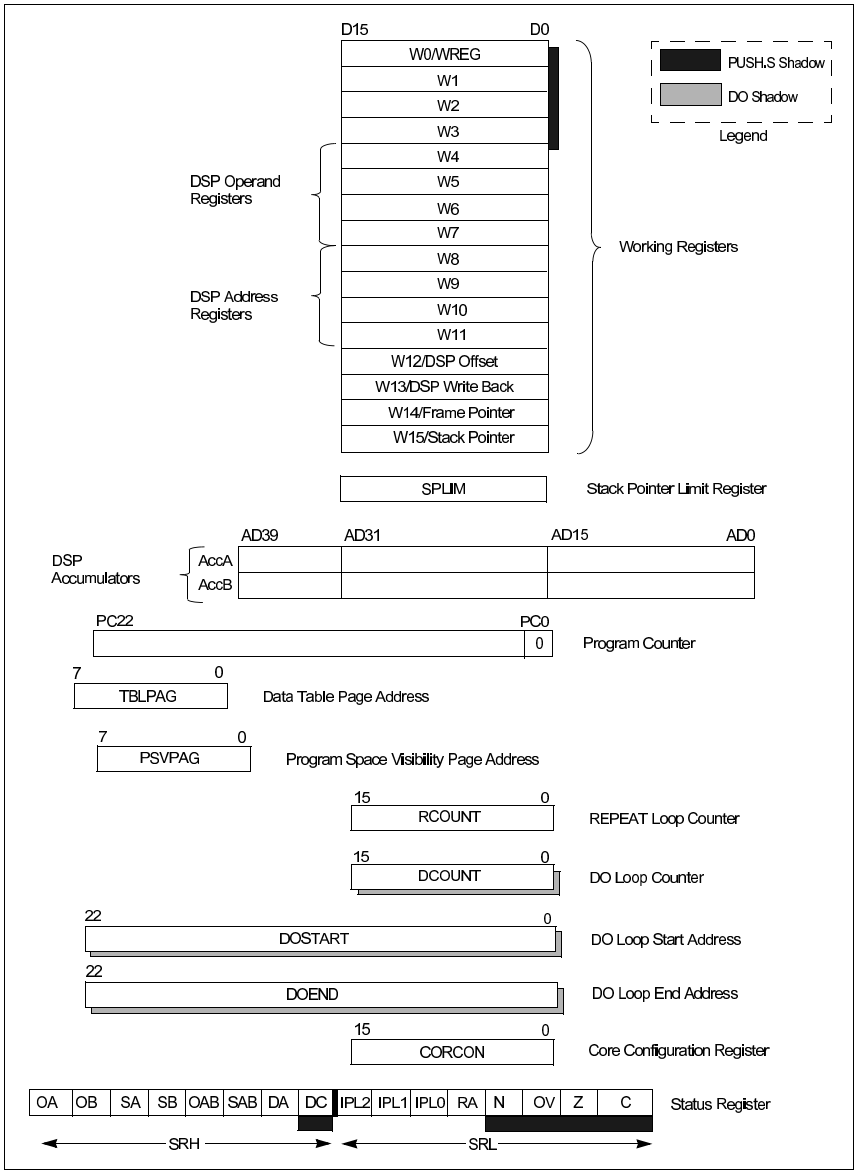
dsPIC30F2010

FIGURE 1-1: dsPIC30F2010 BLOCK DIAGRAM



1.3 Programátorský model

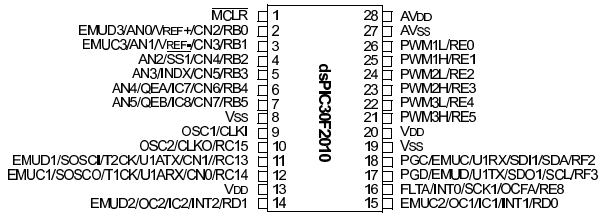
FIGURE 2-1: PROGRAMMER'S MODEL



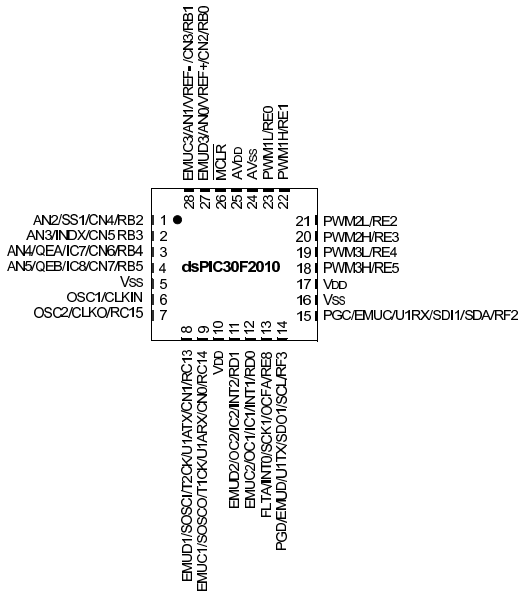
1.4 Zapojení vývodů

Pin Diagrams

28-Pin SDIP and SOIC



28-Pin QFN



1.5 Význam vývodů

Table 1-1 provides a brief description of device I/O pinouts and the functions that may be multiplexed to a port pin. Multiple functions may exist on one port pin. When multiplexing occurs, the peripheral module's functional requirements may force an override of the data direction of the port pin.

TABLE 1-1: PINOUT I/O DESCRIPTIONS

Pin Name	Pin Type	Buffer Type	Description
AN0-AN5	I	Analog	Analog input channels.
AVDD	P	P	Positive supply for analog module.
AVSS	P	P	Ground reference for analog module.
CLKI CLKO	I O	ST/CMOS —	External clock source input. Always associated with OSC1 pin function. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes. Always associated with OSC2 pin function.
CN0-CN7	I	ST	Input change notification inputs. Can be software programmed for internal weak pull-ups on all inputs.
EMUD EMUC EMUD1 EMUC1 EMUD2 EMUC2 EMUD3 EMUC3	I/O I/O I/O I/O I/O I/O I/O I/O	ST ST ST ST ST ST ST ST	ICD Primary Communication Channel data input/output pin. ICD Primary Communication Channel clock input/output pin. ICD Secondary Communication Channel data input/output pin. ICD Secondary Communication Channel clock input/output pin. ICD Tertiary Communication Channel data input/output pin. ICD Tertiary Communication Channel clock input/output pin. ICD Quaternary Communication Channel data input/output pin. ICD Quaternary Communication Channel clock input/output pin.
IC1, IC2, IC7, IC8	I	ST	Capture inputs. The dsPIC30F2010 has 4 capture inputs. The inputs are numbered for consistency with the inputs on larger device variants.
INDX QEA QEB	I I I	ST ST ST	Quadrature Encoder Index Pulse input. Quadrature Encoder Phase A input in QEI mode. Auxiliary Timer External Clock/Gate input in Timer mode. Quadrature Encoder Phase A input in QEI mode. Auxiliary Timer External Clock/Gate input in Timer mode.
INT0 INT1 INT2	I I I	ST ST ST	External interrupt 0 External interrupt 1 External interrupt 2
FLTA PWM1L PWM1H PWM2L PWM2H PWM3L PWM3H	I O O O O O O	ST — — — — — —	PWM Fault A input PWM 1 Low output PWM 1 High output PWM 2 Low output PWM 2 High output PWM 3 Low output PWM 3 High output
$\overline{MCLR}$	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low Reset to the device.
OCFA OC1-OC2	I O	ST —	Compare Fault A input (for Compare channels 1, 2, 3 and 4). Compare outputs.
OSC1 OSC2	I I/O	ST/CMOS —	Oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. Optionally functions as CLKO in RC and EC modes.

Legend: CMOS = CMOS compatible input or output Analog = Analog input
ST = Schmitt Trigger input with CMOS levels O = Output
I = Input P = Power

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

Pin Name	Pin Type	Buffer Type	Description
PGD PGC	I/O I	ST ST	In-Circuit Serial Programming data input/output pin. In-Circuit Serial Programming clock input pin.
RB0-RB5	I/O	ST	PORTB is a bidirectional I/O port.
RC13-RC14	I/O	ST	PORTC is a bidirectional I/O port.
RD0-RD1	I/O	ST	PORTD is a bidirectional I/O port.
RE0-RE5, RE8	I/O	ST	PORTE is a bidirectional I/O port.
RF2, RF3	I/O	ST	PORTF is a bidirectional I/O port.
SCK1 SDI1 SDO1 SS1	I/O I O I	ST ST — ST	Synchronous serial clock input/output for SPI™ #1. SPI #1 Data In. SPI #1 Data Out. SPI #1 Slave Synchronization.
SCL SDA	I/O I/O	ST ST	Synchronous serial clock input/output for I^2C. Synchronous serial data input/output for I^2C.
SOSCO SOSCI	O I	— ST/CMOS	32 kHz low power oscillator crystal output. 32 kHz low power oscillator crystal input. ST buffer when configured in RC mode; CMOS otherwise.
T1CK T2CK	I I	ST ST	Timer1 external clock input. Timer2 external clock input.
U1RX U1TX U1ARX U1ATX	I O I O	ST — ST —	UART1 Receive. UART1 Transmit. UART1 Alternate Receive. UART1 Alternate Transmit.
VDD	P	—	Positive supply for logic and I/O pins.
VSS	P	—	Ground reference for logic and I/O pins.
VREF+	I	Analog	Analog Voltage Reference (High) input.
VREF-	I	Analog	Analog Voltage Reference (Low) input.

Legend: CMOS = CMOS compatible input or output Analog = Analog input
ST = Schmitt Trigger input with CMOS levels O = Output
I = Input P = Power

TABLE 3-3: CORE REGISTER MAP

SFR Name	Address (Home)	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
W0	0000	W0 / WREG																0000 0000 0000 0000
W1	0002	W1																0000 0000 0000 0000
W2	0004	W2																0000 0000 0000 0000
W3	0006	W3																0000 0000 0000 0000
W4	0008	W4																0000 0000 0000 0000
W5	000A	W5																0000 0000 0000 0000
W6	000C	W6																0000 0000 0000 0000
W7	000E	W7																0000 0000 0000 0000
W8	0010	W8																0000 0000 0000 0000
W9	0012	W9																0000 0000 0000 0000
W10	0014	W10																0000 0000 0000 0000
W11	0016	W11																0000 0000 0000 0000
W12	0018	W12																0000 0000 0000 0000
W13	001A	W13																0000 0000 0000 0000
W14	001C	W14																0000 0000 0000 0000
W15	001E	W15																0000 1000 0000 0000
SPLIM	0020	SPLIM																0000 0000 0000 0000
ACCAL	0022	ACCAL																0000 0000 0000 0000
ACCAH	0024	ACCAH																0000 0000 0000 0000
ACCAU	0026	Sign-Extension (ACCA<39>)							ACCAU									0000 0000 0000 0000
ACCBL	0028	ACCBL																0000 0000 0000 0000
ACCBH	002A	ACCBH																0000 0000 0000 0000
ACCBU	002C	Sign-Extension (ACCB<39>)							ACCBU									0000 0000 0000 0000
PCL	002E	PCL																0000 0000 0000 0000
PCH	0030	—	—	—	—	—	—	—	—	—	—	PCH						0000 0000 0000 0000
TBLPAG	0032	—	—	—	—	—	—	—	—	—	—	TBLPAG						0000 0000 0000 0000
PSVPAG	0034	—	—	—	—	—	—	—	—	—	—	PSVPAG						0000 0000 0000 0000
RCOUNT	0036	RCOUNT																uuuu uuuu uuuu uuuu
DCOUNT	0038	DCOUNT																uuuu uuuu uuuu uuuu
DOSTARTL	003A	DOSTARTL															0	uuuu uuuu uuuu uuu0
DOSTARTH	003C	—	—	—	—	—	—	—	—	—	—	DOSTARTH						0000 0000 0uuu uuuu
DOENDL	003E	DOENDL															0	uuuu uuuu uuuu uuu0
DOENDH	0040	—	—	—	—	—	—	—	—	—	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000 0000 0uuu uuuu
SR	0042	OA	OB	SA	SB	OAB	SAB	DA	DC	IPL2	IPL1	IPL0	RA	N	OV	Z	C	0000 0000 0000 0000
CORCON	0044	—	—	—	US	EDT	DL2	DL1	DL0	SATA	SATB	SATDW	ACCSAT	IPL3	PSV	RND	IF	0000 0000 0010 0000
MODCON	0046	XMODEN	YMODEN	—	—	BWM<3:0>				YWM<3:0>				XWM<3:0>				0000 0000 0000 0000

Legend: u = uninitialized bit

TABLE 3-3: CORE REGISTER MAP (CONTINUED)

SFR Name	Address (Home)	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
XMODSRT	0048	XS<15:1>															0	uuuu uuuu uuuu uuu0
XMODEND	004A	XE<15:1>															1	uuuu uuuu uuuu uuu1
YMODSRT	004C	YS<15:1>															0	uuuu uuuu uuuu uuu0
YMODEND	004E	YE<15:1>															1	uuuu uuuu uuuu uuu1
XBREV	0050	BREN	XB<14:0>															uuuu uuuu uuuu uuuu
DISICNT	0052	—	—	DISICNT<13:0>														0000 0000 0000 0000

Legend: u = uninitialized bit

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

3.0 MEMORY ORGANIZATION

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 *dsPIC30F Programmer's Reference Manual* (DS70030).

3.1 Program Address Space

The program address space is 4M instruction words. It is addressable by a 24-bit value from either the 23-bit PC, table instruction Effective Address (EA), or data space EA, when program space is mapped into data space, as defined by Table 3-1. Note that the program space address is incremented by two between successive program words, in order to provide compatibility with data space addressing.

User program space access is restricted to the lower 4M instruction word address range (0x000000 to 0x7FFFFE), for all accesses other than `TBLRD/TBLWT`, which use TBLPAG<7> to determine user or configuration space access. In Table 3-1, Read/Write instructions, bit 23 allows access to the Device ID, the User ID and the configuration bits. Otherwise, bit 23 is always clear.

Note:	The address map shown in Figure 3-1 is conceptual, and the actual memory configuration may vary across individual devices depending on available memory.

FIGURE 3-1: PROGRAM SPACE MEMORY MAP FOR dsPIC30F2010

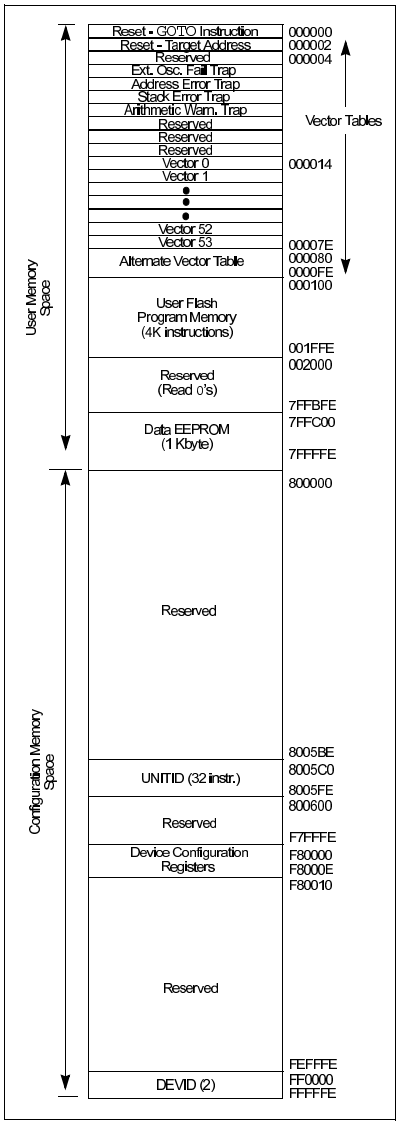
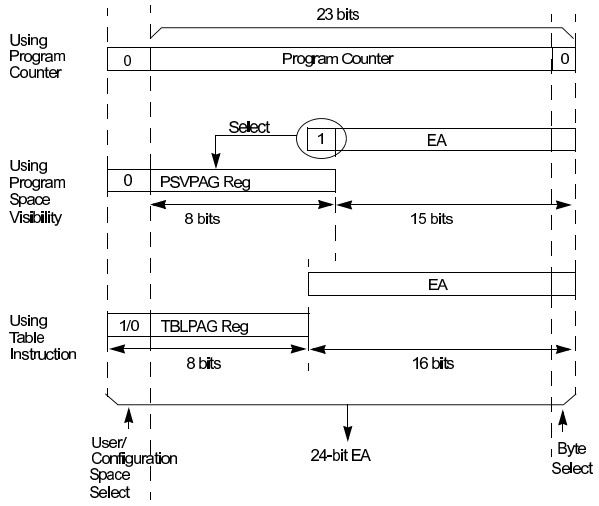


TABLE 3-1: PROGRAM SPACE ADDRESS CONSTRUCTION

Access Type	Access Space	Program Space Address				
		<23>	<22:16>	<15>	<14:1>	<0>
Instruction Access	User	0	PC<22:1>			0
TBLRD/TBLWT	User (TBLPAG<7> = 0)	TBLPAG<7:0>		Data EA <15:0>		
TBLRD/TBLWT	Configuration (TBLPAG<7> = 1)	TBLPAG<7:0>		Data EA <15:0>		
Program Space Visibility	User	0	PSVPAG<7:0>		Data EA <14:0>	

FIGURE 3-2: DATA ACCESS FROM PROGRAM SPACE ADDRESS GENERATION



Note: Program Space Visibility cannot be used to access bits <23:16> of a word in program memory.

3.1.1 DATA ACCESS FROM PROGRAM MEMORY USING TABLE INSTRUCTIONS

This architecture fetches 24-bit wide program memory. Consequently, instructions are always aligned. However, as the architecture is modified Harvard, data can also be present in program space.

There are two methods by which program space can be accessed; via special table instructions, or through the remapping of a 16K word program space page into the upper half of data space (see Section 3.1.2). The TBLRDL and TBLWTL instructions offer a direct method of reading or writing the LS Word of any address within program space, without going through data space. The TBLRDH and TBLWTH instructions are the only method whereby the upper 8 bits of a program space word can be accessed as data.

The PC is incremented by two for each successive 24-bit program word. This allows program memory addresses to directly map to data space addresses. Program memory can thus be regarded as two 16-bit word wide address spaces, residing side by side, each with the same address range. TBLRDL and TBLWTL access the space which contains the LS Data Word, and TBLRDH and TBLWTH access the space which contains the MS Data Byte.

Figure 3-2 shows how the EA is created for table operations and data space accesses (PSV = 1). Here, P<23:0> refers to a program space word, whereas D<15:0> refers to a data space word.

A set of Table Instructions are provided to move byte or word sized data to and from program space.

1. **TBLRDL:** Table Read Low
   Word: Read the LS Word of the program address;
   P<15:0> maps to D<15:0>.
   Byte: Read one of the LS Bytes of the program address;
   P<7:0> maps to the destination byte when byte select = 0;
   P<15:8> maps to the destination byte when byte select = 1.
2. **TBLWTL:** Table Write Low (refer to Section 6.0 for details on Flash Programming).
3. **TBLRDH:** Table Read High
   Word: Read the MS Word of the program address;
   P<23:16> maps to D<7:0>; D<15:8> always be = 0.
   Byte: Read one of the MS Bytes of the program address;
   P<23:16> maps to the destination byte when byte select = 0;
   The destination byte will always be = 0 when byte select = 1.
4. **TBLWTH:** Table Write High (refer to Section 6.0 for details on Flash Programming).

FIGURE 3-3: PROGRAM DATA TABLE ACCESS (LS WORD)

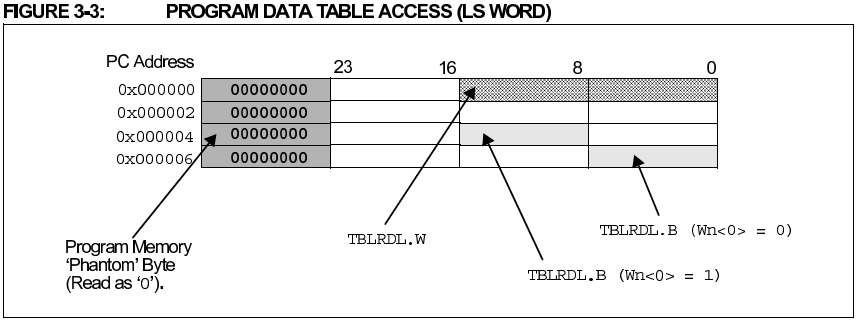
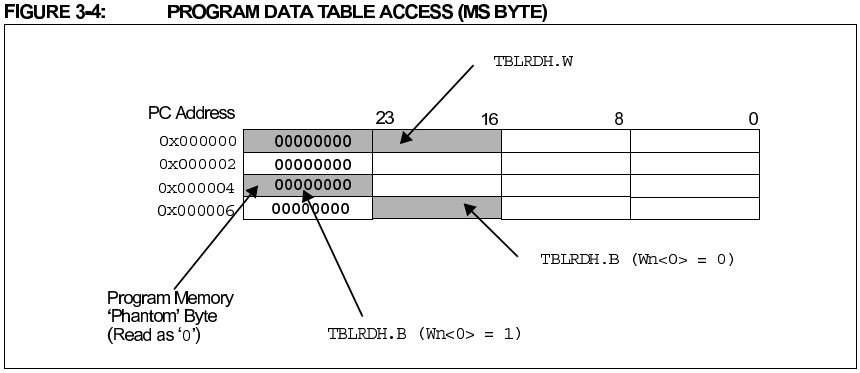


FIGURE 3-4: PROGRAM DATA TABLE ACCESS (MS BYTE)



3.1.2 DATA ACCESS FROM PROGRAM MEMORY USING PROGRAM SPACE VISIBILITY

The upper 32 Kbytes of data space may optionally be mapped into any 16K word program space page. This provides transparent access of stored constant data from X data space, without the need to use special instructions (i.e., `TBLRDL/H`, `TBLWTL/H` instructions).

Program space access through the data space occurs if the MS bit of the data space EA is set and program space visibility is enabled, by setting the PSV bit in the Core Control register (CORCON). The functions of CORCON are discussed in Section 2.4, DSP Engine.

Data accesses to this area add an additional cycle to the instruction being executed, since two program memory fetches are required.

Note that the upper half of addressable data space is always part of the X data space. Therefore, when a DSP operation uses program space mapping to access this memory region, Y data space should typically contain state (variable) data for DSP operations, whereas X data space should typically contain coefficient (constant) data.

Although each data space address, 0x8000 and higher, maps directly into a corresponding program memory address (see Figure 3-5), only the lower 16-bits of the 24-bit program word are used to contain the data. The upper 8 bits should be programmed to force an illegal instruction to maintain machine robustness. Refer to the Programmer's Reference Manual (DS70030) for details on instruction encoding.

Note that by incrementing the PC by 2 for each program memory word, the LS 15 bits of data space addresses directly map to the LS 15 bits in the corresponding program space addresses. The remaining bits are provided by the Program Space Visibility Page register, PSVPAG<7:0>, as shown in Figure 3-5.

Note:	PSV access is temporarily disabled during Table Reads/Writes.

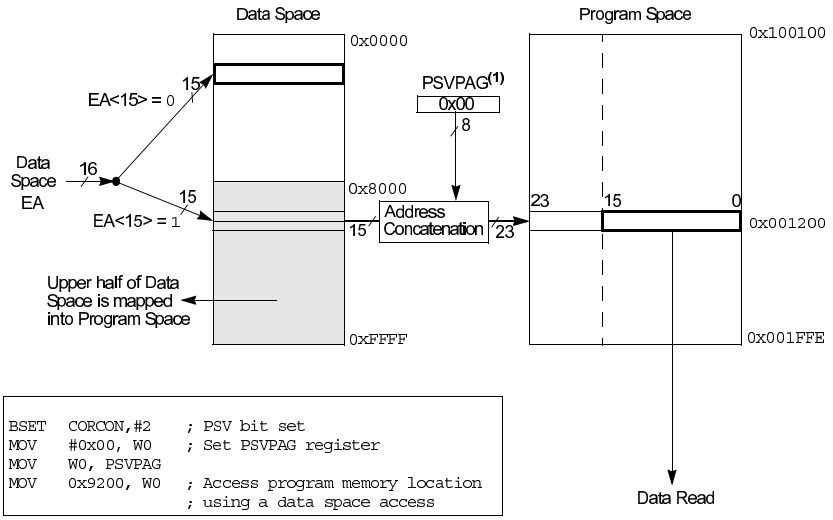
For instructions that use PSV which are executed outside a REPEAT loop:

- The following instructions will require one instruction cycle in addition to the specified execution time:
  - `MAC` class of instructions with data operand pre-fetch
  - `MOV` instructions
  - `MOV.D` instructions
- All other instructions will require two instruction cycles in addition to the specified execution time of the instruction.

For instructions that use PSV which are executed inside a REPEAT loop:

- The following instances will require two instruction cycles in addition to the specified execution time of the instruction:
  - Execution in the first iteration
  - Execution in the last iteration
  - Execution prior to exiting the loop due to an interrupt
  - Execution upon re-entering the loop after an interrupt is serviced
- Any other iteration of the REPEAT loop will allow the instruction, accessing data using PSV, to execute in a single cycle.

FIGURE 3-5: DATA SPACE WINDOW INTO PROGRAM SPACE OPERATION



Note: PSVPAG is an 8-bit register, containing bits <22:15> of the program space address (i.e., it defines the page in program space to which the upper half of data space is being mapped).

3.2 Data Address Space

The core has two data spaces. The data spaces can be considered either separate (for some DSP instructions), or as one unified linear address range (for MCU instructions). The data spaces are accessed using two Address Generation Units (AGUs) and separate data paths.

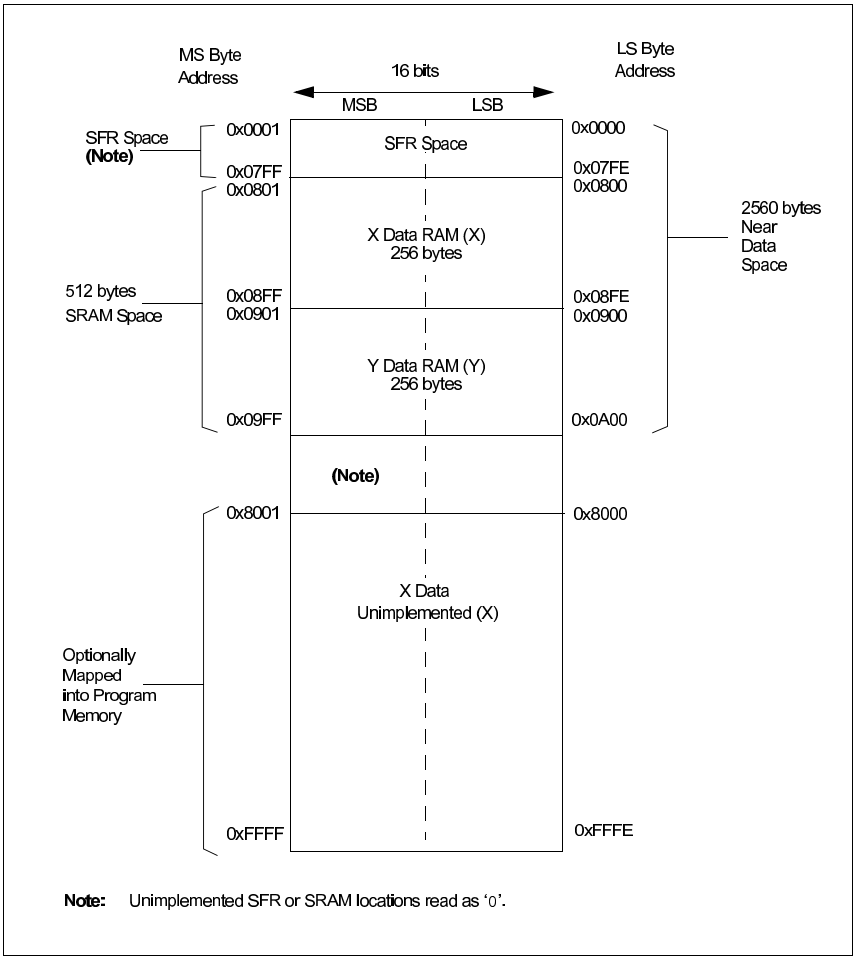
3.2.1 DATA SPACE MEMORY MAP

The data space memory is split into two blocks, X and Y data space. A key element of this architecture is that Y space is a subset of X space, and is fully contained within X space. In order to provide an apparent linear addressing space, X and Y spaces have contiguous addresses.

When executing any instruction other than one of the MAC class of instructions, the X block consists of the 256 byte data address space (including all Y addresses). When executing one of the MAC class of instructions, the X block consists of the 256 bytes data address space excluding the Y address block (for data reads only). In other words, all other instructions regard the entire data memory as one composite address space. The MAC class instructions extract the Y address space from data space and address it using EAs sourced from W10 and W11. The remaining X data space is addressed using W8 and W9. Both address spaces are concurrently accessed only with the MAC class instructions.

A data space memory map is shown in Figure 3-6.

FIGURE 3-6: DATA SPACE MEMORY MAP



Note: Unimplemented SFR or SRAM locations read as '0'.

1.8 Přerušení

5.0 INTERRUPTS

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 *dsPIC30F Programmer's Reference Manual* (DS70030).

The dsPIC30F2010 has 24 interrupt sources and 4 processor exceptions (traps), which must be arbitrated based on a priority scheme.

The CPU is responsible for reading the Interrupt Vector Table (IVT) and transferring the address contained in the interrupt vector to the program counter. The interrupt vector is transferred from the program data bus into the program counter, via a 24-bit wide multiplexer on the input of the program counter.

The Interrupt Vector Table (IVT) and Alternate Interrupt Vector Table (AIVT) are placed near the beginning of program memory (0x000004). The IVT and AIVT are shown in Figure 5-1.

The interrupt controller is responsible for preprocessing the interrupts and processor exceptions, prior to their being presented to the processor core. The peripheral interrupts and traps are enabled, prioritized and controlled using centralized special function registers:

- IFS0<15:0>, IFS1<15:0>, IFS2<15:0>
  All interrupt request flags are maintained in these three registers. The flags are set by their respective peripherals or external signals, and they are cleared via software.
- IEC0<15:0>, IEC1<15:0>, IEC2<15:0>
  All interrupt enable control bits are maintained in these three registers. These control bits are used to individually enable interrupts from the peripherals or external signals.
- IPC0<15:0>... IPC11<7:0>
  The user assignable priority level associated with each of these interrupts is held centrally in these twelve registers.
- IPL<3:0> The current CPU priority level is explicitly stored in the IPL bits. IPL<3> is present in the CORCON register, whereas IPL<2:0> are present in the status register (SR) in the processor core.
- INTCON1<15:0>, INTCON2<15:0>
  Global interrupt control functions are derived from these two registers. INTCON1 contains the control and status flags for the processor exceptions. The INTCON2 register controls the external interrupt request signal behavior and the use of the alternate vector table.

Note:	Interrupt flag bits get set when an interrupt condition occurs, regardless of the state of its corresponding enable bit. User software should ensure the appropriate interrupt flag bits are clear prior to enabling an interrupt.

All interrupt sources can be user assigned to one of 7 priority levels, 1 through 7, via the IPCx registers. Each interrupt source is associated with an interrupt vector, as shown in Figure 5-1. Levels 7 and 1 represent the highest and lowest maskable priorities, respectively.

Note:	Assigning a priority level of 0 to an interrupt source is equivalent to disabling that interrupt.

If the NSTDIS bit (INTCON1<15>) is set, nesting of interrupts is prevented. Thus, if an interrupt is currently being serviced, processing of a new interrupt is prevented, even if the new interrupt is of higher priority than the one currently being serviced.

Note:	The IPL bits become read-only whenever the NSTDIS bit has been set to '1'.

Certain interrupts have specialized control bits for features like edge or level triggered interrupts, interrupt-on-change, etc. Control of these features remains within the peripheral module which generates the interrupt.

The `DISI` instruction can be used to disable the processing of interrupts of priorities 6 and lower for a certain number of instructions, during which the DISI bit (INTCON2<14>) remains set.

When an interrupt is serviced, the PC is loaded with the address stored in the vector location in Program Memory that corresponds to the interrupt. There are 63 different vectors within the IVT (refer to Figure 5-1). These vectors are contained in locations 0x000004 through 0x0000FE of program memory (refer to Figure 5-1). These locations contain 24-bit addresses, and in order to preserve robustness, an address error trap will take place should the PC attempt to fetch any of these words during normal execution. This prevents execution of random data as a result of accidentally decrementing a PC into vector space, accidentally mapping a data space address into vector space, or the PC rolling over to 0x000000 after reaching the end of implemented program memory space. Execution of a `GOTO` instruction to this vector space will also generate an address error trap.

5.1 Interrupt Priority

The user assignable Interrupt Priority (IP<2:0>) bits for each individual interrupt source are located in the LS 3-bits of each nibble, within the IPCx register(s). Bit 3 of each nibble is not used and is read as a '0'. These bits define the priority level assigned to a particular interrupt by the user.

Note:	The user selectable priority levels start at 0, as the lowest priority, and level 7, as the highest priority.

Since more than one interrupt request source may be assigned to a specific user specified priority level, a means is provided to assign priority within a given level. This method is called "Natural Order Priority" and is final.

Natural Order Priority is determined by the position of an interrupt in the vector table, and only affects interrupt operation when multiple interrupts with the same user-assigned priority become pending at the same time.

Table 5-1 lists the interrupt numbers and interrupt sources for the dsPIC devices and their associated vector numbers.

Note 1:	The natural order priority scheme has 0 as the highest priority and 53 as the lowest priority.
2:	The natural order priority number is the same as the INT number.

The ability for the user to assign every interrupt to one of seven priority levels implies that the user can assign a very high overall priority level to an interrupt with a low natural order priority. For example, the PLVD (Low Voltage Detect) can be given a priority of 7. The INT0 (external interrupt 0) may be assigned to priority level 1, thus giving it a very low effective priority.

TABLE 5-1: dsPIC30F2010 INTERRUPT VECTOR TABLE

INT Number	Vector Number	Interrupt Source
Highest Natural Order Priority		
0	8	INT0 - External Interrupt 0
1	9	IC1 - Input Capture 1
2	10	OC1 - Output Compare 1
3	11	T1 - Timer 1
4	12	IC2 - Input Capture 2
5	13	OC2 - Output Compare 2
6	14	T2 - Timer 2
7	15	T3 - Timer 3
8	16	SPI1
9	17	U1RX - UART1 Receiver
10	18	U1TX - UART1 Transmitter
11	19	ADC - ADC Convert Done
12	20	NVM - NVM Write Complete
13	21	SI2C - I^2C Slave Interrupt
14	22	MI2C - I^2C Master Interrupt
15	23	Input Change Interrupt
16	24	INT1 - External Interrupt 1
17	25	IC7 - Input Capture 7
18	26	IC8 - Input Capture 8
19	27	Reserved
20	28	Reserved
21	29	Reserved
22	30	Reserved
23	31	INT2 - External Interrupt 2
24	32	Reserved
25	33	Reserved
26	34	Reserved
27	35	Reserved
28	36	Reserved
29	37	Reserved
30	38	Reserved
31	39	Reserved
32	40	Reserved
33	41	Reserved
34	42	Reserved
35	43	Reserved
36	44	INT3 - External Interrupt 3
37	45	Reserved
38	46	Reserved
39	47	PWM - PWM Period Match
40	48	QEI - QEI Interrupt
41	49	Reserved
42	50	Reserved
43	51	FLTA - PWM Fault A
44	52	Reserved
45-53	53-61	Reserved
Lowest Natural Order Priority		

5.2 Reset Sequence

A Reset is not a true exception, because the interrupt controller is not involved in the Reset process. The processor initializes its registers in response to a Reset, which forces the PC to zero. The processor then begins program execution at location 0x000000. A GOTO instruction is stored in the first program memory location, immediately followed by the address target for the GOTO instruction. The processor executes the GOTO to the specified address and then begins operation at the specified target (start) address.

5.2.1 RESET SOURCES

In addition to External Reset and Power-on Reset (POR), there are 6 sources of error conditions which 'trap' to the Reset vector.

- Watchdog Time-out:
  The watchdog has timed out, indicating that the processor is no longer executing the correct flow of code.
- Uninitialized W Register Trap:
  An attempt to use an uninitialized W register as an address pointer will cause a Reset.
- Illegal Instruction Trap:
  Attempted execution of any unused opcodes will result in an illegal instruction trap. Note that a fetch of an illegal instruction does not result in an illegal instruction trap if that instruction is flushed prior to execution due to a flow change.
- Brown-out Reset (BOR):
  A momentary dip in the power supply to the device has been detected, which may result in malfunction.
- Trap Lockout:
  Occurrence of multiple Trap conditions simultaneously will cause a Reset.

5.3 Traps

Traps can be considered as non-maskable interrupts indicating a software or hardware error, which adhere to a predefined priority as shown in Figure 5-1. They are intended to provide the user a means to correct erroneous operation during debug and when operating within the application.

Note:	If the user does not intend to take corrective action in the event of a trap error condition, these vectors must be loaded with the address of a default handler that simply contains the RESET instruction. If, on the other hand, one of the vectors containing an invalid address is called, an address error trap is generated.

Note that many of these trap conditions can only be detected when they occur. Consequently, the questionable instruction is allowed to complete prior to trap exception processing. If the user chooses to recover from the error, the result of the erroneous action that caused the trap may have to be corrected.

There are 8 fixed priority levels for traps: Level 8 through Level 15, which implies that the IPL3 is always set during processing of a trap.

If the user is not currently executing a trap, and he sets the IPL<3:0> bits to a value of '0111' (Level 7), then all interrupts are disabled, but traps can still be processed.

5.3.1 TRAP SOURCES

The following traps are provided with increasing priority. However, since all traps can be nested, priority has little effect.

Math Error Trap:

The Math Error trap executes under the following three circumstances:

1. Should an attempt be made to divide by zero, the divide operation will be aborted on a cycle boundary and the trap taken.
2. If enabled, a Math Error trap will be taken when an arithmetic operation on either accumulator A or B causes an overflow from bit 31 and the accumulator guard bits are not utilized.
3. If enabled, a Math Error trap will be taken when an arithmetic operation on either accumulator A or B causes a catastrophic overflow from bit 39 and all saturation is disabled.
4. If the shift amount specified in a shift instruction is greater than the maximum allowed shift amount, a trap will occur.

Address Error Trap:

This trap is initiated when any of the following circumstances occurs:

1. A misaligned data word access is attempted.
2. A data fetch from our unimplemented data memory location is attempted.
3. A data access of an unimplemented program memory location is attempted.
4. An instruction fetch from vector space is attempted.

Note:	In the MAC class of instructions, wherein the data space is split into X and Y data space, unimplemented X space includes all of Y space, and unimplemented Y space includes all of X space.

5. Execution of a "BRA #literal" instruction or a "GOTO #literal" instruction, where literal is an unimplemented program memory address.
6. Executing instructions after modifying the PC to point to unimplemented program memory addresses. The PC may be modified by loading a value into the stack and executing a RETURN instruction.

Stack Error Trap:

This trap is initiated under the following conditions:

1. The stack pointer is loaded with a value which is greater than the (user programmable) limit value written into the SPLIM register (stack overflow).
2. The stack pointer is loaded with a value which is less than 0x0800 (simple stack underflow).

Oscillator Fail Trap:

This trap is initiated if the external oscillator fails and operation becomes reliant on an internal RC backup.

5.3.2 HARD AND SOFT TRAPS

It is possible that multiple traps can become active within the same cycle (e.g., a misaligned word stack write to an overflowed address). In such a case, the fixed priority shown in Figure 5-1 is implemented, which may require the user to check if other traps are pending, in order to completely correct the fault.

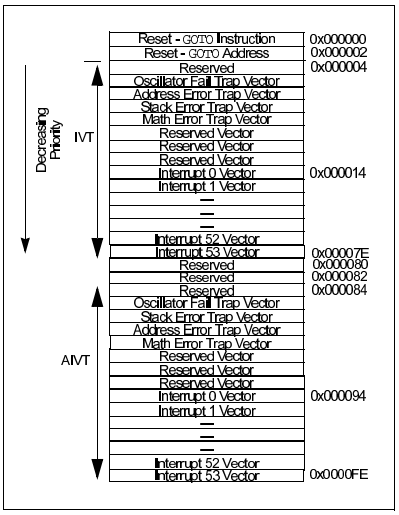
'Soft' traps include exceptions of priority level 8 through level 11, inclusive. The arithmetic error trap (level 11) falls into this category of traps.

'Hard' traps include exceptions of priority level 12 through level 15, inclusive. The address error (level 12), stack error (level 13) and oscillator error (level 14) traps fall into this category.

Each hard trap that occurs must be acknowledged before code execution of any type may continue. If a lower priority hard trap occurs while a higher priority trap is pending, acknowledged, or is being processed, a hard trap conflict will occur.

The device is automatically Reset in a hard trap conflict condition. The TRAPR status bit (RCON<15>) is set when the Reset occurs, so that the condition may be detected in software.

FIGURE 5-1: TRAP VECTORS



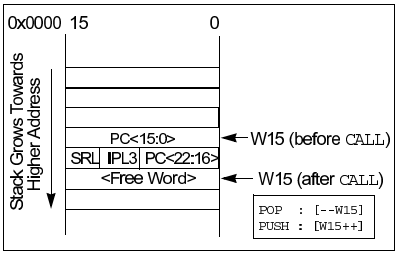
5.4 Interrupt Sequence

All interrupt event flags are sampled in the beginning of each instruction cycle by the IFSx registers. A pending interrupt request (IRQ) is indicated by the flag bit being equal to a '1' in an IFSx register. The IRQ will cause an interrupt to occur if the corresponding bit in the interrupt enable (IECx) register is set. For the remainder of the instruction cycle, the priorities of all pending interrupt requests are evaluated.

If there is a pending IRQ with a priority level greater than the current processor priority level in the IPL bits, the processor will be interrupted.

The processor then stacks the current program counter and the low byte of the processor status register (SRL), as shown in Figure 5-2. The low byte of the status register contains the processor priority level at the time, prior to the beginning of the interrupt cycle. The processor then loads the priority level for this interrupt into the status register. This action will disable all lower priority interrupts until the completion of the Interrupt Service Routine.

FIGURE 5-2: INTERRUPT STACK FRAME



Note 1:	The user can always lower the priority level by writing a new value into SR. The Interrupt Service Routine must clear the interrupt flag bits in the IFSx register before lowering the processor interrupt priority, in order to avoid recursive interrupts.
2:	The IPL3 bit (CORCON<3>) is always clear when interrupts are being processed. It is set only during execution of traps.

The RETFIE (Return from Interrupt) instruction will unstack the program counter and status registers to return the processor to its state prior to the interrupt sequence.

5.5 Alternate Vector Table

In Program Memory, the Interrupt Vector Table (IVT) is followed by the Alternate Interrupt Vector Table (AIVT), as shown in Figure 5-1. Access to the Alternate Vector Table is provided by the ALTIVT bit in the INTCON2 register. If the ALTIVT bit is set, all interrupt and exception processes will use the alternate vectors instead of the default vectors. The alternate vectors are organized in the same manner as the default vectors. The AIVT supports emulation and debugging efforts by providing a means to switch between an application and a support environment, without requiring the interrupt vectors to be reprogrammed. This feature also enables switching between applications for evaluation of different software algorithms at run time.

If the AIVT is not required, the program memory allocated to the AIVT may be used for other purposes. AIVT is not a protected section and may be freely programmed by the user.

5.6 Fast Context Saving

A context saving option is available using shadow registers. Shadow registers are provided for the DC, N, OV, Z and C bits in SR, and the registers W0 through W3. The shadows are only one level deep. The shadow registers are accessible using the PUSH.S and POP.S instructions only.

When the processor vectors to an interrupt, the PUSH.S instruction can be used to store the current value of the aforementioned registers into their respective shadow registers.

If an ISR of a certain priority uses the PUSH.S and POP.S instructions for fast context saving, then a higher priority ISR should not include the same instructions. Users must save the key registers in software during a lower priority interrupt, if the higher priority ISR uses fast context saving.

5.7 External Interrupt Requests

The interrupt controller supports five external interrupt request signals, INT0-INT4. These inputs are edge sensitive; they require a low-to-high or a high-to-low transition to generate an interrupt request. The INTCON2 register has five bits, INT0EP-INT4EP, that select the polarity of the edge detection circuitry.

5.8 Wake-up from Sleep and Idle

The interrupt controller may be used to wake up the processor from either Sleep or Idle modes, if Sleep or Idle mode is active when the interrupt is generated.

If an enabled interrupt request of sufficient priority is received by the interrupt controller, then the standard interrupt request is presented to the processor. At the same time, the processor will wake-up from Sleep or Idle and begin execution of the Interrupt Service Routine (ISR) needed to process the interrupt request.

TABLE 5-2: INTERRUPT CONTROLLER REGISTER MAP

SFR Name	ADR	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
INTCON1	0080	NSTDIS	—	—	—	—	OVATE	OVBTE	COVTE	—	—	—	MATHERR	ADDRERR	STKERR	OSCFAIL	—	0000 0000 0000 0000
INTCON2	0082	ALTIVT	—	—	—	—	—	—	—	—	—	—	—	—	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0IF	0000 0000 0000 0000
IFS1	0086	—	—	—	—	—	—	—	—	INT2IF	—	—	—	—	IC8IF	IC7IF	INT1IF	0000 0000 0000 0000
IFS2	0088	—	—	—	—	FLTAIF	—	—	QEIIF	PWMIF	—	—	—	—	—	—	—	0000 0000 0000 0000
IEC0	008C	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IEC1	008E	—	—	—	—	—	—	—	—	INT2IE	—	—	—	—	IC8IE	IC7IE	INT1IE	0000 0000 0000 0000
IEC2	0090	—	—	—	—	FLTAIE	—	—	QEIIE	PWMIE	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC0	0094	—	T1IP<2:0>			—	OC1IP<2:0>			—	IC1IP<2:0>			—	INT0IP<2:0>			0100 0100 0100 0100
IPC1	0096	—	T3IP<2:0>			—	T2IP<2:0>			—	OC2IP<2:0>			—	IC2IP<2:0>			0100 0100 0100 0100
IPC2	0098	—	ADIP<2:0>			—	U1TXIP<2:0>			—	U1RXIP<2:0>			—	SPI1IP<2:0>			0100 0100 0100 0100
IPC3	009A	—	CNIP<2:0>			—	MI2CIP<2:0>			—	SI2CIP<2:0>			—	NVMIP<2:0>			0100 0100 0100 0100
IPC4	009C	—	—	—	—	—	IC8IP<2:0>			—	IC7IP<2:0>			—	INT1IP<2:0>			0100 0100 0100 0100
IPC5	009E	—	INT2IP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	0100 0000 0000 0000
IPC6	00A0	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC7	00A2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC8	00A4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC9	00A6	—	PWMIP<2:0>			—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
IPC10	00A8	—	FLTAIP<2:0>			—	—	—	—	—	—	—	—	—	QEIIP<2:0>			0100 0000 0000 0100
IPC11	00AA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000

Legend: u = uninitialized bit

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

TABLE 6-1: NVM REGISTER MAP

File 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	All RESETS
NVMCON	0760	WR	WREN	WRERR	—	—	—	—	TWRI	—	PROGOP<6:0>							0000 0000 0000 0000
NVMADR	0762	NVMADR<15:0>																uuuu uuuu uuuu uuuu
NVMADRU	0764	—	—	—	—	—	—	—	—	—	NVMADR<23:16>							0000 0000 uuuu uuuu
NVMKEY	0766	—	—	—	—	—	—	—	—	—	KEY<7:0>							0000 0000 0000 0000

Legend: u = uninitialized bit

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

3 Porty

3.1 Hardware

8.0 I/O PORTS

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).

All of the device pins (except VDD, VSS, $\overline{\text{MCLR}}$ and OSC1/CLKIN) are shared between the peripherals and the parallel I/O ports.

All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

8.1 Parallel I/O (PIO) Ports

When a peripheral is enabled and the peripheral is actively driving an associated pin, the use of the pin as a general purpose output pin is disabled. The I/O pin may be read, but the output driver for the parallel port bit will be disabled. If a peripheral is enabled, but the peripheral is not actively driving a pin, that pin may be driven by a port.

All port pins have three registers directly associated with the operation of the port pin. The data direction register (TRISx) determines whether the pin is an input or an output. If the data direction bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the latch (LATx), read the latch. Writes to the latch, write the latch (LATx). Reads from the port (PORTx), read the port pins, and writes to the port pins, write the latch (LATx).

Any bit and its associated data and control registers that are not valid for a particular device will be disabled. That means the corresponding LATx and TRISx registers and the port pin will read as zeros.

When a pin is shared with another peripheral or function that is defined as an input only, it is nevertheless regarded as a dedicated port because there is no other competing source of outputs. An example is the INT4 pin.

A parallel I/O (PIO) port that shares a pin with a peripheral is, in general, subservient to the peripheral. The peripheral's output buffer data and control signals are provided to a pair of multiplexers. The multiplexers select whether the peripheral or the associated port has ownership of the output data and control signals of the I/O pad cell. Figure 8-1 shows how ports are shared with other peripherals, and the associated I/O cell (pad) to which they are connected. Table 8-1 shows the formats of the registers for the shared ports, PORTB through PORTG.

FIGURE 8-1: BLOCK DIAGRAM OF A SHARED PORT STRUCTURE

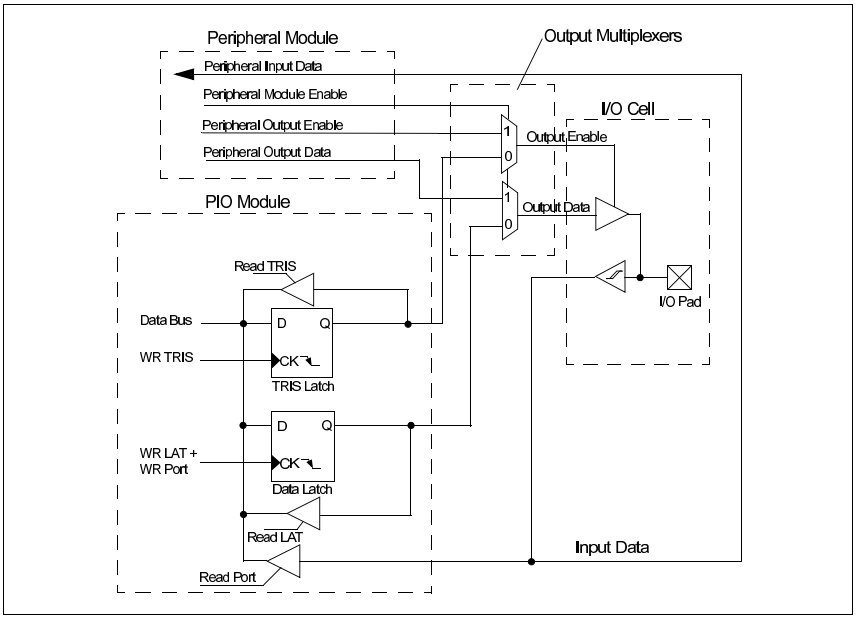


TABLE 8-1: dsPIC30F2010 PORT REGISTER MAP

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
TRISB	02C6	—	—	—	—	—	—	—	—	—	—	TRISB5	TRISB4	TRISB3	TRISB2	TRISB1	TRISB0	0000 0000 0011 1111
PORTB	02C8	—	—	—	—	—	—	—	—	—	—	RB5	RB4	RB3	RB2	RB1	RB0	0000 0000 0000 0000
LATB	02CB	—	—	—	—	—	—	—	—	—	—	LATB5	LATB4	LATB3	LATB2	LATB1	LATB0	0000 0000 0000 0000
TRISC	02CC	TRISC15	TRISC14	TRISC13	—	—	—	—	—	—	—	—	—	—	—	—	—	1110 0000 0000 0000
PORTC	02CE	RC15	RC14	RC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
LATC	02D0	LATC15	LATC14	LATC13	—	—	—	—	—	—	—	—	—	—	—	—	—	0000 0000 0000 0000
TRISD	02D2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRISD1	TRISD0	0000 0000 0000 0111
PORTD	02D4	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RD1	RD0	0000 0000 0000 0000
LATD	02D6	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LATD1	LATD0	0000 0000 0000 0000
TRISE	02D8	—	—	—	—	—	—	—	TRISE8	—	—	TRISE5	TRISE4	TRISE3	TRISE2	TRISE1	TRISE0	0000 0001 0011 1111
PORTE	02DA	—	—	—	—	—	—	—	RE8	—	—	RE5	RE4	RE3	RE2	RE1	RE0	0000 0000 0000 0000
LATE	02DC	—	—	—	—	—	—	—	LATE8	—	—	LATE5	LATE4	LATE3	LATE2	LATE1	LATE0	0000 0000 0000 0000
TRISF	02EE	—	—	—	—	—	—	—	—	—	—	—	—	TRISF3	TRISF2	—	—	0000 0000 0000 1100
PORTF	02E0	—	—	—	—	—	—	—	—	—	—	—	—	RF3	RF2	—	—	0000 0000 0000 0000
LATF	02E2	—	—	—	—	—	—	—	—	—	—	—	—	LATF3	LATF2	—	—	0000 0000 0000 0000

Legend: u = uninitialized bit

TABLE 8-2: INPUT CHANGE NOTIFICATION REGISTER MAP (BITS 15-0)

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
CNEN1	00C0	CN15IE	CN14IE	CN13IE	CN12IE	CN11IE	CN10IE	CN9IE	CN8IE	CN7IE	CN6IE	CN5IE	CN4IE	CN3IE	CN2IE	CN1IE	CN0IE	0000 0000 0000 0000
CNEN2	00C2	—	—	—	—	—	—	—	—	—	—	CN21IE	CN20IE	CN19IE	CN18IE	CN17IE	CN16IE	0000 0000 0000 0000
CNPU1	00C4	CN15PUE	CN14PUE	CN13PUE	CN12PUE	CN11PUE	CN10PUE	CN9PUE	CN8PUE	CN7PUE	CN6PUE	CN5PUE	CN4PUE	CN3PUE	CN2PUE	CN1PUE	CN0PUE	0000 0000 0000 0000
CNPU2	00C6	—	—	—	—	—	—	—	—	—	—	CN21PUE	CN20PUE	CN19PUE	CN18PUE	CN17PUE	CN16PUE	0000 0000 0000 0000

Legend: u = uninitialized bit

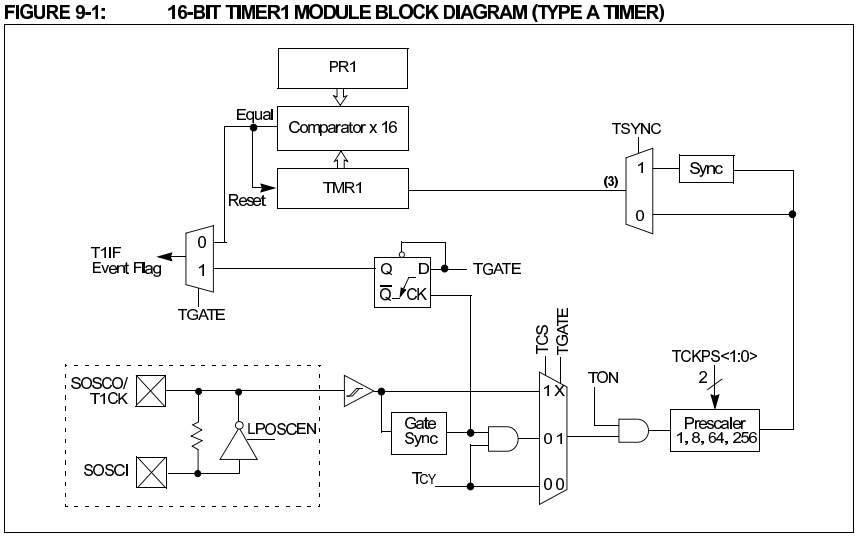
Note: Refer to *dsPIC30F Family Reference Manual* (DS70046) for descriptions of register bit fields.

4 Čítače

4.1 Typ A

dsPIC30F2010

FIGURE 9-1: 16-BIT TIMER1 MODULE BLOCK DIAGRAM (TYPE A TIMER)



9.1 Timer Gate Operation

The 16-bit timer can be placed in the Gated Time Accumulation mode. This mode allows the internal TCY to increment the respective timer when the gate input signal (T1CK pin) is asserted high. Control bit TGATE (T1CON<6>) must be set to enable this mode. The timer must be enabled (TON = 1) and the timer clock source set to internal (TCS = 0).

When the CPU goes into the Idle mode, the timer will stop incrementing, unless TSIDL = 0. If TSIDL = 1, the timer will resume the incrementing sequence upon termination of the CPU Idle mode.

9.2 Timer Prescaler

The input clock (FOSC/4 or external clock) to the 16-bit Timer, has a prescale option of 1:1, 1:8, 1:64, and 1:256 selected by control bits TCKPS<1:0> (T1CON<5:4>). The prescaler counter is cleared when any of the following occurs:

- a write to the TMR1 register
- clearing of the TON bit (T1CON<15>)
- device Reset such as POR and BOR

However, if the timer is disabled (TON = 0), then the timer prescaler cannot be reset since the prescaler clock is halted.

TMR1 is not cleared when T1CON is written. It is cleared by writing to the TMR1 register.

9.3 Timer Operation During Sleep Mode

During CPU Sleep mode, the timer will operate if:

- The timer module is enabled (TON = 1) and
- The timer clock source is selected as external (TCS = 1) and
- The TSYNC bit (T1CON<2>) is asserted to a logic 0, which defines the external clock source as asynchronous

When all three conditions are true, the timer will continue to count up to the period register and be reset to 0x0000.

When a match between the timer and the period register occurs, an interrupt can be generated, if the respective timer interrupt enable bit is asserted.

4.2 Typy B a C

FIGURE 10-2: 16-BIT TIMER2 BLOCK DIAGRAM (TYPE B TIMER)

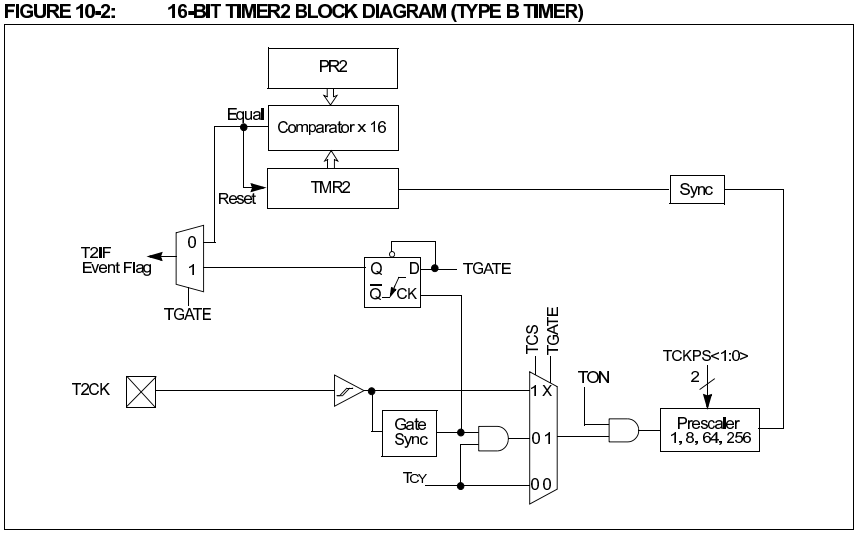
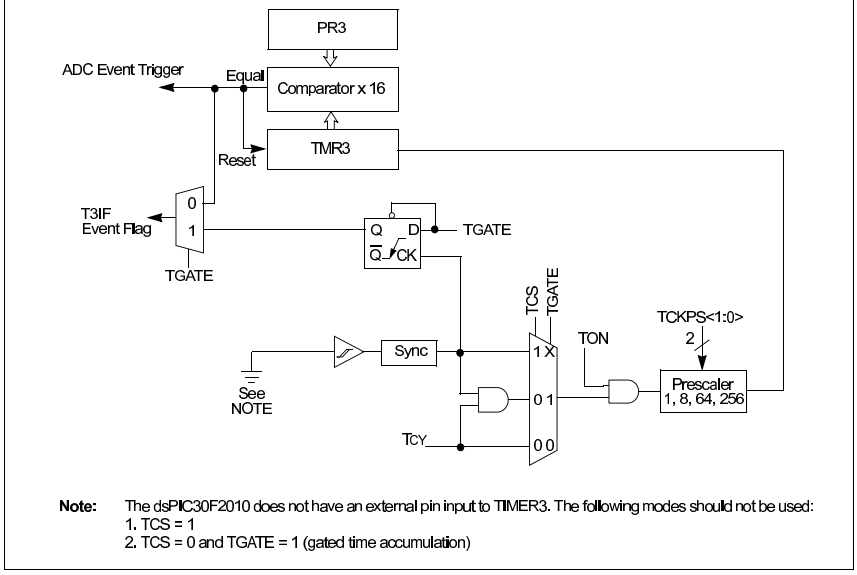


FIGURE 10-3: 16-BIT TIMER3 BLOCK DIAGRAM (TYPE C TIMER)

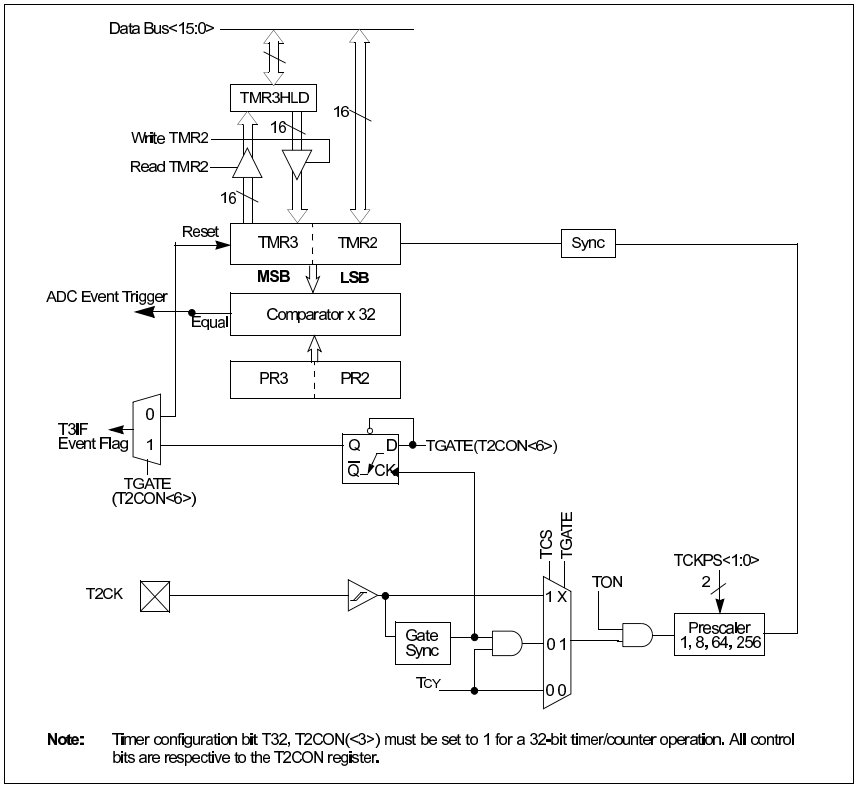


Note: The dsPIC30F2010 does not have an external pin input to TIMER3. The following modes should not be used:
1. TCS = 1
2. TCS = 0 and TGATE = 1 (gated time accumulation)

4.3 32-bitový

dsPIC30F2010

FIGURE 10-1: 32-BIT TIMER2/3 BLOCK DIAGRAM



Note: Timer configuration bit T32, T2CON(<3>) must be set to 1 for a 32-bit timer/counter operation. All control bits are respective to the T2CON register.

4.4 Řídící registry

12.3 Control Registers

Register 12-1: TxCON: Type A Time Base Register

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		—	TSYNC	TCS	—
bit 7							bit 0

bit 15 **TON:** Timer On Control bit
1 = Starts the timer
0 = Stops the timer

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue timer operation when device enters Idle mode
0 = Continue timer operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timer Gated Time Accumulation Enable bit
1 = Gated time accumulation enabled
0 = Gated time accumulation disabled
(TCS must be set to '0' when TGATE = 1. Reads as '0' if TCS = 1)

bit 5-4 **TCKPS<1:0>:** Timer Input Clock Prescale Select bits
11 = 1:256 prescale value
10 = 1:64 prescale value
01 = 1:8 prescale value
00 = 1:1 prescale value

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

bit 2 **TSYNC:** Timer External Clock Input Synchronization Select bit
When TCS = 1:
1 = Synchronize external clock input
0 = Do not synchronize external clock input
When TCS = 0:
This bit is ignored. Read as '0'. Timer1 uses the internal clock when TCS = 0.

bit 1 **TCS:** Timer Clock Source Select bit
1 = External clock from pin TxCK
0 = Internal clock (FOSC/4)

bit 0 **Unimplemented:** Read as '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

Register 12-2: TxCON: Type B Time Base Register

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		T32	—	TCS	—
bit 7							bit 0

bit 15 **TON:** Timer On bit
When T32 = 1 (in 32-bit Timer mode):
1 = Starts 32-bit TMRx:TMRy timer pair
0 = Stops 32-bit TMRx:TMRy timer pair
When T32 = 0 (in 16-bit Timer mode):
1 = Starts 16-bit timer
0 = Stops 16-bit timer

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue timer operation when device enters Idle mode
0 = Continue timer operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timer Gated Time Accumulation Enable bit
1 = Timer gated time accumulation enabled
0 = Timer gated time accumulation disabled
(TCS must be set to logic '0' when TGATE = 1)

bit 5-4 **TCKPS<1:0>:** Timer Input Clock Prescale Select bits
11 = 1:256 prescale value
10 = 1:64 prescale value
01 = 1:8 prescale value
00 = 1:1 prescale value

bit 3 **T32:** 32-bit Timer Mode Select bits
1 = TMRx and TMRy form a 32-bit timer
0 = TMRx and TMRy form separate 16-bit timer

bit 2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timer Clock Source Select bit
1 = External clock from pin TxCK
0 = Internal clock (FOSC/4)

bit 0 **Unimplemented:** Read as '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

Register 12-3: TxCON: Type C Time Base Register

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
TON	—	TSIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	U-0
—	TGATE	TCKPS<1:0>		—	—	TCS	—
bit 7							bit 0

bit 15 **TON:** Timer On bit
1 = Starts 16-bit TMRx
0 = Stops 16-bit TMRx

bit 14 **Unimplemented:** Read as '0'

bit 13 **TSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **TGATE:** Timer Gated Time Accumulation Enable bit
1 = Timer gated time accumulation enabled
0 = Timer gated time accumulation disabled (Read as '0' if TCS = 1)
(TCS must be set to logic '0' when TGATE = 1)

bit 5-4 **TCKPS<1:0>:** Timer Input Clock Prescale Select bits
11 = 1:256 prescale value
10 = 1:64 prescale value
01 = 1:8 prescale value
00 = 1:1 prescale value

bit 3-2 **Unimplemented:** Read as '0'

bit 1 **TCS:** Timer Clock Source Select bit
1 = External clock from pin TxCK
0 = Internal clock (FOSC/4)

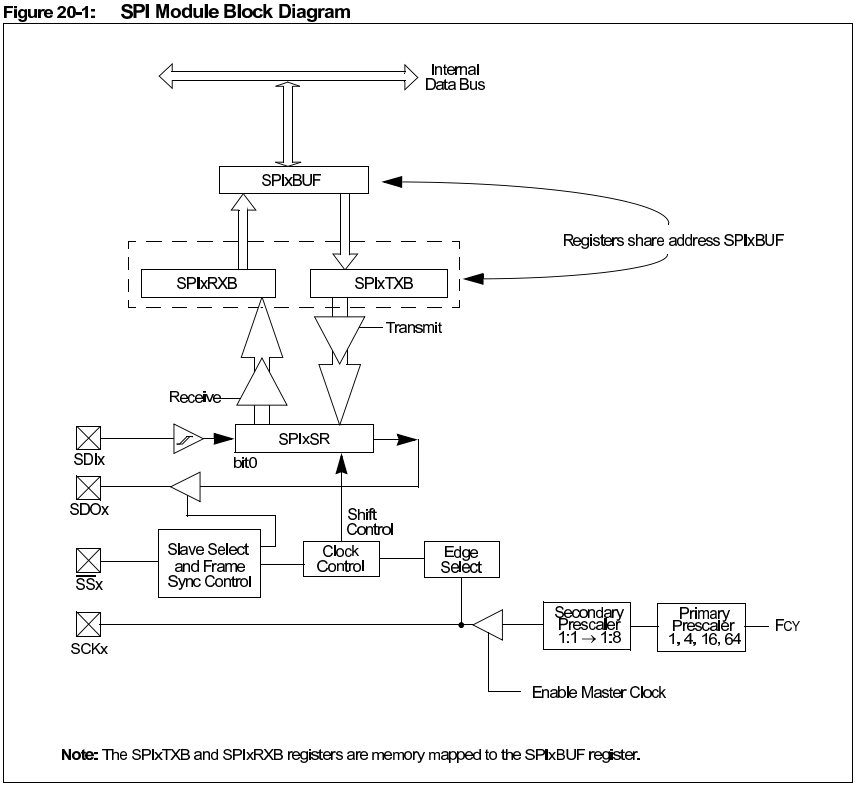
bit 0 **Unimplemented:** Read as '0'

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

5 SPI

5.1 Struktura

Figure 20-1: SPI Module Block Diagram



Note: The SPIxTXB and SPIxRXB registers are memory mapped to the SPIxBUF register.

20.2 Status and Control Registers

Register 20-2: SPIxSTAT: SPI Status and Control Register

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	U-0	U-0
SPIEN	—	SPISIDL	—	—	—	—	—
bit 15							bit 8

Lower Byte:							
U-0	R/W-0 HS	U-0	U-0	U-0	U-0	R-0	R-0
—	SPIROV	—	—	—	—	SPITBF	SPIRBF
bit 7							bit 0

bit 15 **SPIEN:** SPI Enable bit
1 = Enables module and configures SCKx, SDOx, SDIx and SSx as serial port pins
0 = Disables module

bit 14 **Unimplemented:** Read as '0'

bit 13 **SPISIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-7 **Unimplemented:** Read as '0'

bit 6 **SPIROV:** Receive Overflow Flag bit
1 = A new byte/word is completely received and discarded. The user software has not read the previous data in the SPIxBUF register.
0 = No overflow has occurred

bit 5-2 **Unimplemented:** Read as '0'

bit 1 **SPITBF:** SPI Transmit Buffer Full Status bit
1 = Transmit not yet started, SPIxTXB is full
0 = Transmit started, SPIxTXB is empty
Automatically set in hardware when CPU writes SPIxBUF location, loading SPIxTXB.
Automatically cleared in hardware when SPIx module transfers data from SPIxTXB to SPIxSR.

bit 0 **SPIRBF:** SPI Receive Buffer Full Status bit
1 = Receive complete, SPIxRXB is full
0 = Receive is not complete, SPIxRXB is empty
Automatically set in hardware when SPIx transfers data from SPIxSR to SPIxRXB.
Automatically cleared in hardware when core reads SPIxBUF location, reading SPIxRXB.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
HC = Cleared by Hardware	HS = Set by Hardware		
-n = Value at Reset	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 20-2: SPIxCON: SPIx Control Register

Upper Byte:							
U-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SSEN	CKP	MSTEN	SPRE<2:0>			PPRE<1:0>	
bit 7							bit 0

bit 15 **Unimplemented:** Read as '0'

bit 14 **FRMEN:** Framed SPI Support bit
1 = Framed SPI support enabled
0 = Framed SPI support disabled

bit 13 **SPIFSD:** Frame Sync Pulse Direction Control on $\overline{SSx}$ pin bit
1 = Frame sync pulse input (slave)
0 = Frame sync pulse output (master)

bit 12 **Unimplemented:** Read as '0'

bit 11 **DISSDO:** Disable SDOx pin bit
1 = SDOx pin is not used by module. Pin is controlled by associated port register.
0 = SDOx pin is controlled by the module

bit 10 **MODE16:** Word/Byte Communication Select bit
1 = Communication is word-wide (16 bits)
0 = Communication is byte-wide (8 bits)

bit 9 **SMP:** SPI Data Input Sample Phase bit
Master mode:
1 = Input data sampled at end of data output time
0 = Input data sampled at middle of data output time
Slave mode:
SMP must be cleared when SPI is used in Slave mode.

bit 8 **CKE:** SPI Clock Edge Select bit
1 = Serial output data changes on transition from active clock state to Idle clock state (see bit 6)
0 = Serial output data changes on transition from Idle clock state to active clock state (see bit 6)

Note: The CKE bit is not used in the Framed SPI modes. The user should program this bit to '0' for the Framed SPI modes (FRMEN = 1).

bit 7 **SSEN:** Slave Select Enable (Slave mode) bit
1 = $\overline{SS}$ pin used for Slave mode
0 = $\overline{SS}$ pin not used by module. Pin controlled by port function.

bit 6 **CKP:** Clock Polarity Select bit
1 = Idle state for clock is a high level; active state is a low level
0 = Idle state for clock is a low level; active state is a high level

bit 5 **MSTEN:** Master Mode Enable bit
1 = Master mode
0 = Slave mode

Register 20-2: SPIxCON: SPIx Control Register (Continued)

bit 4-2 **SPRE<2:0>:** Secondary Prescale (Master Mode) bits
(Supported settings: 1:1, 2:1 through 8:1, all inclusive)
111 = Secondary prescale 1:1
110 = Secondary prescale 2:1
...
000 = Secondary prescale 8:1

bit 1-0 **PPRE<1:0>:** Primary Prescale (Master Mode) bits
11 = Primary prescale 1:1
10 = Primary prescale 4:1
01 = Primary prescale 16:1
00 = Primary prescale 64:1

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

20.6 Special Function Registers Associated with SPI Modules

Table 20-3: SPI1 Register Map

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
SPI1STAT	0220	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI1CON	0222	—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000 0000 0000 0000
SPI1BUF	0224	Transmit and Receive Buffer Address shared by SPI1TXB and SPI1RXB registers																0000 0000 0000 0000

Table 20-4: SPI2 Register Map

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
SPI2STAT	0226	SPIEN	—	SPISIDL	—	—	—	—	—	—	SPIROV	—	—	—	—	SPITBF	SPIRBF	0000 0000 0000 0000
SPI2CON	0228	—	FRMEN	SPIFSD	—	DISSDO	MODE16	SMP	CKE	SSEN	CKP	MSTEN	SPRE2	SPRE1	SPRE0	PPRE1	PPRE0	0000 0000 0000 0000
SPI2BUF	022A	Transmit and Receive Buffer Address shared by SPI2TXB and SPI2RXB registers																0000 0000 0000 0000

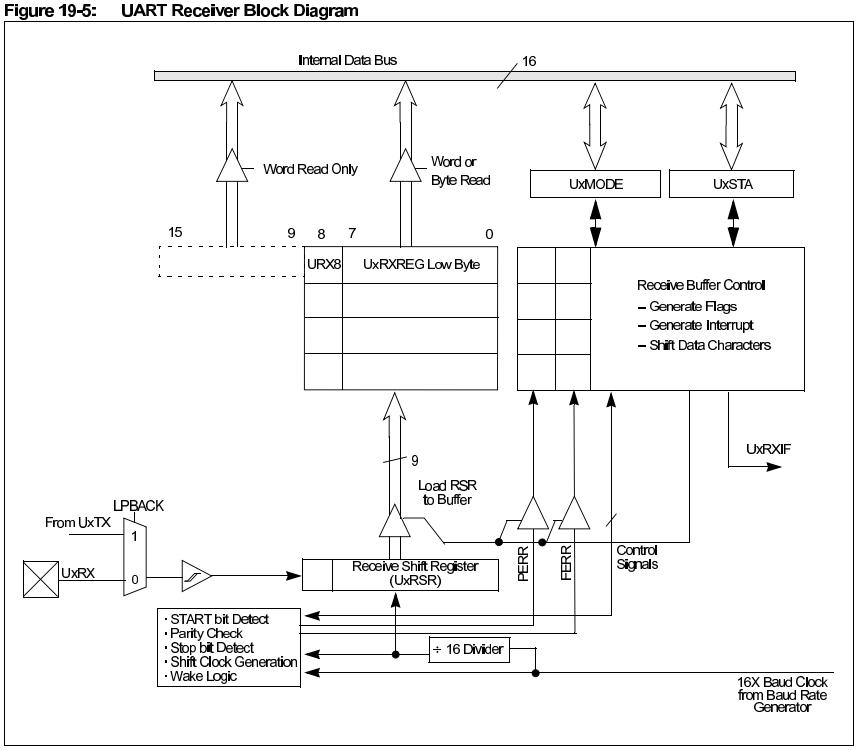
Table 20-5: SPI Module Related Interrupt Registers

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
INTCON1	0080	NSTDIS	—	—	—	—	OVATE	OVBTE	COVTE	—	—	—	SWTRAP	OVRFLOW	ADDRERR	STKERR	—	0000 0000 0000 0000
INTCON2	0082	ALTIVT	DISI	—	—	—	—	LEVB	—	—	—	—	INT4EP	INT3EP	INT2EP	INT1EP	INT0EP	0000 0000 0000 0000
IFS0	0084	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0	0000 0000 0000 0000
IFS1	0086	IC8IF	IC7IF	IC4IF	IC3IF	C1IF	SPI2IF	U2TXIF	U2RXIF	INT2IF	T5IF	T4IF	OC4IF	OC3IF	IC8IF	IC7IF	INT1IF	0000 0000 0000 0000
IEC0	008C	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IEC1	008E	IC8IE	IC7IE	IC5IE	IC4IE	IC3IE	C1IE	SPI2IE	U2TXIE	U2RXIE	INT2IE	T5IE	T4IE	OC4IE	OC3IE	IC8IE	IC7IE	INT1IE	0000 0000 0000 0000
IPC2	0098	—	ADIP<2:0>			—	U1TXIP<2:0>			—	U1RXIP<2:0>			—	SPI1IP<2:0>			0100 0100 0100 0100
IPC6	00A0	—	C1IP<2:0>			—	SPI2IP<2:0>			—	U2TXIP<2:0>			—	U2RXIP<2:0>			0100 0100 0100 0100

6 UART

6.1 Přijímač

Figure 19-5: UART Receiver Block Diagram



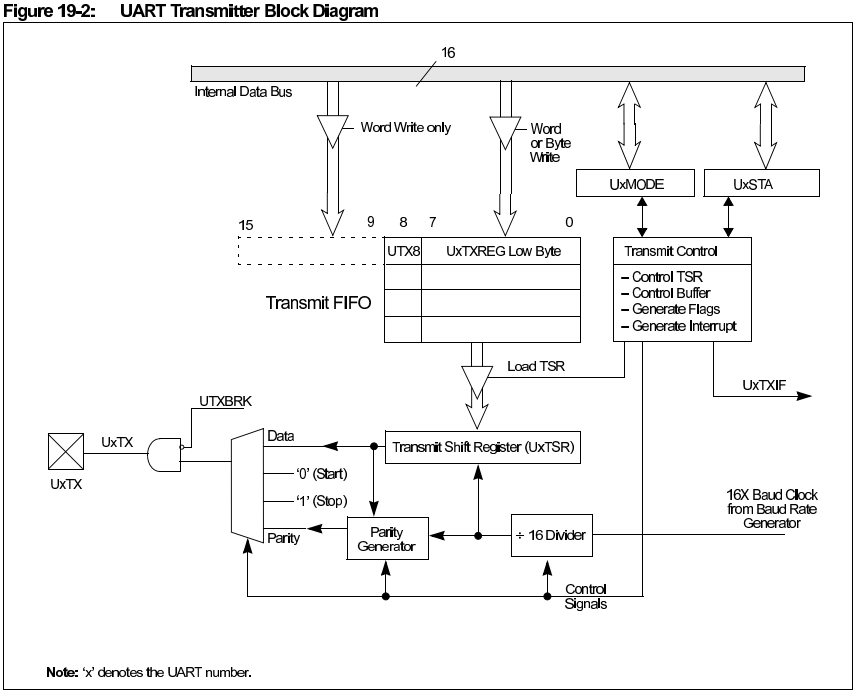
6.2 Vysílač

19.5 UART Transmitter

The UART transmitter block diagram is shown in Figure 19-2. The heart of the transmitter is the Transmit Shift register (UxTSR). The Shift register obtains its data from the transmit FIFO buffer, UxTXREG. The UxTXREG register is loaded with data in software. The UxTSR register is not loaded until the Stop bit has been transmitted from the previous load. As soon as the Stop bit is transmitted, the UxTSR is loaded with new data from the UxTXREG register (if available).

Note:	The UxTSR register is not mapped in data memory, so it is not available to the user.

Figure 19-2: UART Transmitter Block Diagram



Note: 'x' denotes the UART number.

Transmission is enabled by setting the UTXEN enable bit (UxSTA<10>). The actual transmission will not occur until the UxTXREG register has been loaded with data and the Baud Rate Generator (UxBRG) has produced a shift clock (Figure 19-2). The transmission can also be started by first loading the UxTXREG register and then setting the UTXEN enable bit. Normally when transmission is first started, the UxTSR register is empty, so a transfer to the UxTXREG register will result in an immediate transfer to UxTSR. Clearing the UTXEN bit during a transmission will cause the transmission to be aborted and will reset the transmitter. As a result, the UxTX pin will revert to a high-impedance state.

In order to select 9-bit transmission, the PDSEL<1:0> bits (UxMODE<2:1>) should be set to '11' and the ninth bit should be written to the UTX9 bit (UxTXREG<8>). A word write should be performed to UxTXREG so that all nine bits are written at the same time.

Note:	There is no parity in the case of 9-bit data transmission.

19.2 Control Registers

Register 19-1: UxMODE: UARTx Mode Register

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	R/W-0	U-0	U-0
UARTEN	—	USIDL	—	reserved	ALTIO	reserved	reserved
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
WAKE	LPBACK	ABAUD	—	—	PDSEL<1:0>		STSEL
bit 7							bit 0

bit 15 **UARTEN:** UART Enable bit
1 = UART is enabled. UART pins are controlled by UART as defined by UEN<1:0> and UTXEN control bits.
0 = UART is disabled. UART pins are controlled by corresponding PORT, LAT, and TRIS bits.

bit 14 **Unimplemented:** Read as '0'

bit 13 **USIDL:** Stop in Idle Mode bit
1 = Discontinue operation when device enters Idle mode
0 = Continue operation in Idle mode

bit 12 **Unimplemented:** Read as '0'

bit 11 **Reserved:** Write '0' to this location

bit 10 **ALTIO:** UART Alternate I/O Selection bit
1 = UART communicates using UxATX and UxARX I/O pins
0 = UART communicates using UxTX and UxRX I/O pins

Note: The alternate UART I/O pins are not available on all devices. See device data sheet for details.

bit 9-8 **Reserved:** Write '0' to these locations

bit 7 **WAKE:** Enable Wake-up on Start bit Detect During Sleep Mode bit
1 = Wake-up enabled
0 = Wake-up disabled

bit 6 **LPBACK:** UART Loopback Mode Select bit
1 = Enable Loopback mode
0 = Loopback mode is disabled

bit 5 **ABAUD:** Auto Baud Enable bit
1 = Input to Capture module from UxRX pin
0 = Input to Capture module from ICx pin

bit 4-3 **Unimplemented:** Read as '0'

bit 2-1 **PDSEL<1:0>:** Parity and Data Selection bits
11 = 9-bit data, no parity
10 = 8-bit data, odd parity
01 = 8-bit data, even parity
00 = 8-bit data, no parity

bit 0 **STSEL:** Stop Selection bit
1 = 2 Stop bits
0 = 1 Stop 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

Register 19-2: UxSTA: UARTx Status and Control Register

Upper Byte:							
R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

bit 15 **UTXISEL:** Transmission Interrupt Mode Selection bit
1 = Interrupt when a character is transferred to the Transmit Shift register and as result, the transmit buffer becomes empty
0 = Interrupt when a character is transferred to the Transmit Shift register (this implies that there is at least one character open in the transmit buffer)

bit 14-12 **Unimplemented:** Read as '0'

bit 11 **UTXBRK:** Transmit Break bit
1 = UxTX pin is driven low, regardless of transmitter state
0 = UxTX pin operates normally

bit 10 **UTXEN:** Transmit Enable bit
1 = UART transmitter enabled, UxTX pin controlled by UART (if UARTEN = 1)
0 = UART transmitter disabled, any pending transmission is aborted and buffer is reset. UxTX pin controlled by PORT.

bit 9 **UTXBF:** Transmit Buffer Full Status bit (Read Only)
1 = Transmit buffer is full
0 = Transmit buffer is not full, at least one more data word can be written

bit 8 **TRMT:** Transmit Shift Register is Empty bit (Read Only)
1 = Transmit shift register is empty and transmit buffer is empty (the last transmission has completed)
0 = Transmit shift register is not empty, a transmission is in progress or queued in the transmit buffer

bit 7-6 **URXISEL<1:0>:** Receive Interrupt Mode Selection bit
11 = Interrupt flag bit is set when Receive Buffer is full (i.e., has 4 data characters)
10 = Interrupt flag bit is set when Receive Buffer is 3/4 full (i.e., has 3 data characters)
0x = Interrupt flag bit is set when a character is received

bit 5 **ADDEN:** Address Character Detect (bit 8 of received data = 1)
1 = Address Detect mode enabled. If 9-bit mode is not selected, this control bit has no effect.
0 = Address Detect mode disabled

bit 4 **RIDLE:** Receiver Idle bit (Read Only)
1 = Receiver is Idle
0 = Data is being received

bit 3 **PERR:** Parity Error Status bit (Read Only)
1 = Parity error has been detected for the current character
0 = Parity error has not been detected

bit 2 **FERR:** Framing Error Status bit (Read Only)
1 = Framing Error has been detected for the current character
0 = Framing Error has not been detected

Register 19-2: UxSTA: UARTx Status and Control Register (Continued)

bit 1 **OERR:** Receive Buffer Overrun Error Status bit (Read/Clear Only)
1 = Receive buffer has overflowed
0 = Receive buffer has not overflowed

bit 0 **URXDA:** Receive Buffer Data Available bit (Read Only)
1 = Receive buffer has data, at least one more character can be read
0 = Receive buffer is empty

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	C = Bit can be cleared
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 19-3: UxRXREG: UARTx Receive Register

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
—	—	—	—	—	—	—	URX8
bit 15							bit 8

Lower Byte:							
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
URX<7:0>							
bit 7							bit 0

bit 15-9 **Unimplemented:** Read as '0'
bit 8 **URX8:** Data bit 8 of the Received Character (in 9-bit mode)
bit 7-0 **URX<7:0>:** Data bits 7-0 of the Received Character

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

Register 19-4: UxTXREG: UARTx Transmit Register (Write Only)

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	W-x
—	—	—	—	—	—	—	UTX8
bit 15							bit 8

Lower Byte:							
W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
UTX<7:0>							
bit 7							bit 0

bit 15-9 **Unimplemented:** Read as '0'
bit 8 **UTX8:** Data bit 8 of the Character to be Transmitted (in 9-bit mode)
bit 7-0 **UTX<7:0>:** Data bits 7-0 of the Character to be Transmitted

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

Register 19-5: UxBRG: UARTx Baud Rate Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<15:8>							
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BRG<7:0>							
bit 7							bit 0

bit 15-0 **BRG<15:0>:** Baud Rate Divisor bits

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

19.12 Registers Associated with UART Module

Table 19-3: Registers Associated with UART1

SFR Name	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
U1MODE	UARTEN	—	USIDL	—	reserved	ALTIO	reserved	reserved	WAKE	LPBACK	ABAUD	—	—	PDSEL<1:0>		STSEL	0000 0000 0000 0000
U1STA	UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA	0000 0001 0001 0000
U1TXREG	—	—	—	—	—	—	—	UTX8	Transmit Register								0000 0000 0000 0000
U1RXREG	—	—	—	—	—	—	—	URX8	Receive Register								0000 0000 0000 0000
U1BRG	Baud Rate Generator Prescaler																0000 0000 0000 0000
IFS0	CNIF	MI2CIF	SI2CIF	NVMIF	ADIF	U1TXIF	U1RXIF	SPI1IF	T3IF	T2IF	OC2IF	IC2IF	T1IF	OC1IF	IC1IF	INT0	0000 0000 0000 0000
IEC0	CNIE	MI2CIE	SI2CIE	NVMIE	ADIE	U1TXIE	U1RXIE	SPI1IE	T3IE	T2IE	OC2IE	IC2IE	T1IE	OC1IE	IC1IE	INT0IE	0000 0000 0000 0000
IPC2	—	ADIP<2:0>				U1TXIP<2:0>			—	U1RXIP<2:0>			—	SPI1IP<2:0>			0100 0100 0100 0100

Note: The registers associated with UART1 are shown for reference. See the device data sheet for the registers associated with other UART modules.

7 A/D Převodník

7.1 Struktura

18.0 10-BIT HIGH SPEED ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

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).

The10-bit high-speed analog-to-digital converter (A/D) allows conversion of an analog input signal to a 10-bit digital number. This module is based on a Successive Approximation Register (SAR) architecture, and provides a maximum sampling rate of 500 ksps. The A/D module has up to 16 analog inputs which are multiplexed into four sample and hold amplifiers. The output of the sample and hold is the input into the converter, which generates the result. The analog reference voltages are software selectable to either the device supply voltage (AVDD/AVSS) or the voltage level on the (VREF+/VREF-) pin. The A/D converter has a unique feature of being able to operate while the device is in Sleep mode.

The A/D module has six 16-bit registers:

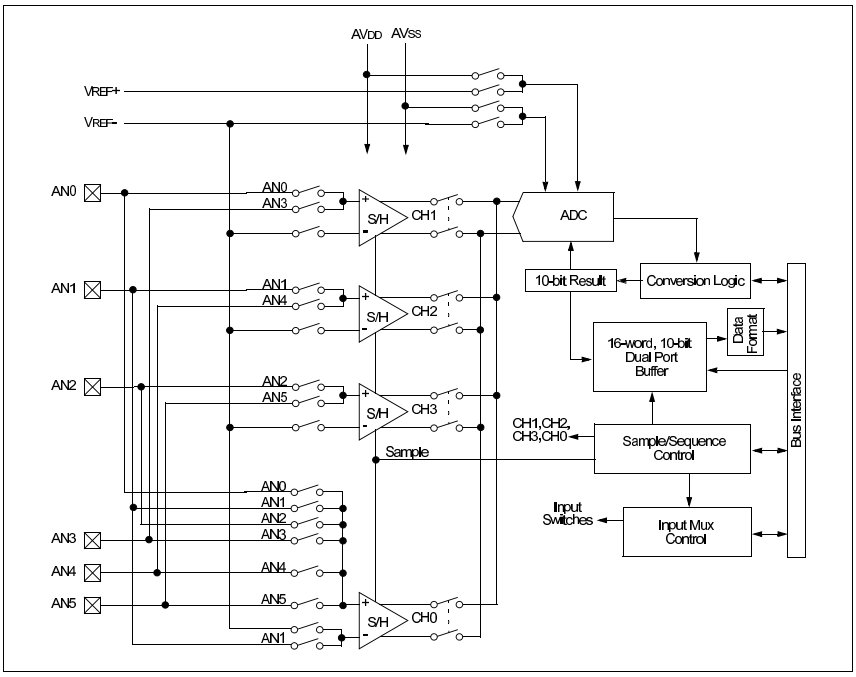
- A/D Control Register1 (ADCON1)
- A/D Control Register2 (ADCON2)
- A/D Control Register3 (ADCON3)
- A/D Input Select Register (ADCHS)
- A/D Port Configuration Register (ADPCFG)
- A/D Input Scan Selection Register (ADCSSL)

The ADCON1, ADCON2 and ADCON3 registers control the operation of the A/D module. The ADCHS register selects the input channels to be converted. The ADPCFG register configures the port pins as analog inputs or as digital I/O. The ADCSSL register selects inputs for scanning.

Note:	The SSRC<2:0>, ASAM, SIMSAM, SMPI<3:0>, BUFM and ALTS bits, as well as the ADCON3 and ADCSSL registers, must not be written to while ADON = 1. This would lead to indeterminate results.

The block diagram of the A/D module is shown in Figure 18-1.

FIGURE 18-1: 10-BIT HIGH SPEED A/D FUNCTIONAL BLOCK DIAGRAM



Register 17-1: ADCON1: A/D Control Register 1

Upper Byte:							
R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON	—	ADSIDL	—	—	—	FORM<1:0>	
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0 HC, HS	R/C-0 HC, HS
SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE
bit 7							bit 0

bit 15 **ADON:** A/D Operating Mode bit
1 = A/D converter module is operating
0 = A/D converter is off

bit 14 **Unimplemented:** Read as '0'

bit 13 **ADSIDL:** Stop in Idle Mode bit
1 = Discontinue module operation when device enters Idle mode
0 = Continue module operation in Idle mode

bit 12-10 **Unimplemented:** Read as '0'

bit 9-8 **FORM<1:0>:** Data Output Format bits
11 = Signed Fractional (DOUT = sddd dddd dd00 0000)
10 = Fractional (DOUT = dddd dddd dd00 0000)
01 = Signed Integer (DOUT = ssss sssd dddd dddd)
00 = Integer (DOUT = 0000 00dd dddd dddd)

bit 7-5 **SSRC<2:0>:** Conversion Trigger Source Select bits
111 = Internal counter ends sampling and starts conversion (auto convert)
110 = Reserved
101 = Reserved
100 = Reserved
011 = Motor Control PWM interval ends sampling and starts conversion
010 = GP Timer3 compare ends sampling and starts conversion
001 = Active transition on INT0 pin ends sampling and starts conversion
000 = Clearing SAMP bit ends sampling and starts conversion

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

bit 3 **SIMSAM:** Simultaneous Sample Select bit (only applicable when CHPS = 01 or 1x)
1 = Samples CH0, CH1, CH2, CH3 simultaneously (when CHPS = 1x)
or
Samples CH0 and CH1 simultaneously (when CHPS = 01)
0 = Samples multiple channels individually in sequence

bit 2 **ASAM:** A/D Sample Auto-Start bit
1 = Sampling begins immediately after last conversion completes. SAMP bit is auto set.
0 = Sampling begins when SAMP bit set

Register 17-1: ADCON1: A/D Control Register 1 (Continued)

bit 1 **SAMP:** A/D Sample Enable bit
1 = At least one A/D sample/hold amplifier is sampling
0 = A/D sample/hold amplifiers are holding
When ASAM = 0, writing '1' to this bit will start sampling.
When SSRC = 000, writing '0' to this bit will end sampling and start conversion.

bit 0 **DONE:** A/D Conversion Status bit (Rev. B silicon or later)
1 = A/D conversion is done
0 = A/D conversion is NOT done
Cleared by software or start of a new conversion.
Clearing this bit will not effect any operation in progress.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
HC = Hardware clear	HS = Hardware set	C = Clearable by software	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

Register 17-2: ADCON2: A/D Control Register 2

Upper Byte:							
R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
VCFG<2:0>			reserved	—	CSCNA	CHPS<1:0>	
bit 15							bit 8

Lower Byte:							
R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	—	SMPI<3:0>				BUFM	ALTS
bit 7							bit 0

bit 15-13 **VCFG<2:0>:** Voltage Reference Configuration bits

	A/D VREFH	A/D VREFL
000	AVDD	AVSS
001	External VREF+ pin	AVSS
010	AVDD	External VREF- pin
011	External VREF+ pin	External VREF- pin
1XX	AVDD	AVSS

bit 12 **Reserved:** User should write '0' to this location

bit 11 **Unimplemented:** Read as '0'

bit 10 **CSCNA:** Scan Input Selections for CH0+ S/H Input for MUX A Input Multiplexer Setting bit
1 = Scan inputs
0 = Do not scan inputs

bit 9-8 **CHPS<1:0>:** Selects Channels Utilized bits
1x = Converts CH0, CH1, CH2 and CH3
01 = Converts CH0 and CH1
00 = Converts CH0
When SIMSAM bit (ADCON1<3>) = 0 multiple channels sampled simultaneously.
When SMSAM bit (ADCON1<3>) = 1 multiple channels sampled as in CHPS<1:0>.

bit 7 **BUFS:** Buffer Fill Status bit
Only valid when BUFM = 1 (ADRES split into 2 x 8-word buffers).
1 = A/D is currently filling buffer 0x8-0xF, user should access data in 0x0-0x7
0 = A/D is currently filling buffer 0x0-0x7, user should access data in 0x8-0xF

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

bit 5-2 **SMPI<3:0>:** Sample/Convert Sequences Per Interrupt Selection bits
1111 = Interrupts at the completion of conversion for each 16th sample/convert sequence
1110 = Interrupts at the completion of conversion for each 15th sample/convert sequence
.....
0001 = Interrupts at the completion of conversion for each 2nd sample/convert sequence
0000 = Interrupts at the completion of conversion for each sample/convert sequence

bit 1 **BUFM:** Buffer Mode Select bit
1 = Buffer configured as two 8-word buffers ADCBUF(15...8), ADCBUF(7...0)
0 = Buffer configured as one 16-word buffer ADCBUF(15...0.)

bit 0 **ALTS:** Alternate Input Sample Mode Select bit
1 = Uses MUX A input multiplexer settings for first sample, then alternate between MUX B and MUX A input multiplexer settings for all subsequent samples
0 = Always use MUX A input multiplexer settings

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

Register 17-3: ADCON3: A/D Control Register 3

Upper Byte:							
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	SAMC<4:0>				
bit 15							bit 8

Lower Byte:							
R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	ADCS<5:0>					
bit 7							bit 0

bit 15-13 **Unimplemented:** Read as '0'

bit 12-8 **SAMC<4:0>:** Auto-Sample Time bits
11111 = 31 TAD
• • • • •
00001 = 1 TAD
00000 = 0 TAD (only allowed if performing sequential conversions using more than one S/H amplifier)

bit 7 **ADRC:** A/D Conversion Clock Source bit
1 = A/D internal RC clock
0 = Clock derived from system clock

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

bit 5-0 **ADCS<5:0>:** A/D Conversion Clock Select bits
111111 = TCY/2 • (ADCS<5:0> + 1) = 32 • TCY
• • • • • •
000001 = TCY/2 • (ADCS<5:0> + 1) = TCY
000000 = TCY/2 • (ADCS<5:0> + 1) = TCY/2

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

Register 17-4: ADCHS: A/D Input Select Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH123NB<1:0>		CH123SB	CH0NB	CH0SB<3:0>			
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH123NA<1:0>		CH123SA	CH0NA	CH0SA<3:0>			
bit 7							bit 0

bit 15-14 **CH123NB<1:0>:** Channel 1, 2, 3 Negative Input Select for MUX B Multiplexer Setting bits
Same definition as bits 6-7 (see Note)

bit 13 **CH123SB:** Channel 1, 2, 3 Positive Input Select for MUX B Multiplexer Setting bit
Same definition as bit 5 (see Note)

bit 12 **CH0NB:** Channel 0 Negative Input Select for MUX B Multiplexer Setting bit
Same definition as bit 4 (see Note)

bit 11-8 **CH0SB<3:0>:** Channel 0 Positive Input Select for MUX B Multiplexer Setting bits
Same definition as bits 3-0 (see Note)

bit 7-6 **CH123NA<1:0>:** Channel 1, 2, 3 Negative Input Select for MUX A Multiplexer Setting bits
11 = CH1 negative input is AN9, CH2 negative input is AN10, CH3 negative input is AN11
10 = CH1 negative input is AN6, CH2 negative input is AN7, CH3 negative input is AN8
0x = CH1, CH2, CH3 negative input is VREF-

bit 5 **CH123SA:** Channel 1, 2, 3 Positive Input Select for MUX A Multiplexer Setting bit
1 = CH1 positive input is AN3, CH2 positive input is AN4, CH3 positive input is AN5
0 = CH1 positive input is AN0, CH2 positive input is AN1, CH3 positive input is AN2

bit 4 **CH0NA:** Channel 0 Negative Input Select for MUX A Multiplexer Setting bit
1 = Channel 0 negative input is AN1
0 = Channel 0 negative input is VREF-

bit 3-0 **CH0SA<3:0>:** Channel 0 Positive Input Select for MUX A Multiplexer Setting bits
1111 = Channel 0 positive input is AN15
1110 = Channel 0 positive input is AN14
1101 = Channel 0 positive input is AN13
||
||
||
0001 = Channel 0 positive input is AN1
0000 = Channel 0 positive input is AN0

Note: The analog input multiplexer supports two input setting configurations, denoted MUX A and MUX B. ADCHS<15:8> determine the settings for MUX B, and ADCHS<7:0> determine the settings for MUX A. Both sets of control bits function identically.

Note: The ADCHS register description and functionality will vary depending on the number of A/D inputs available on the selected device. Please refer to the specific device data sheet for additional details on this register.

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

17 10-bit A/D Converter

Register 17-5: ADPCFG: A/D Port Configuration Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

bit 15-0 **PCFG<15:0>:** Analog Input Pin Configuration Control bits
1 = Analog input pin in Digital mode, port read input enabled, A/D input multiplexer input connected to AVSS
0 = Analog input pin in Analog mode, port read input disabled, A/D samples pin voltage

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

Register 17-6: ADCSSL: A/D Input Scan Select Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
bit 7							bit 0

bit 15-0 **CSSL<15:0>:** A/D Input Pin Scan Selection bits
1 = Select ANx for input scan
0 = Skip ANx for input scan

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

TABLE 18-1: ADC REGISTER MAP

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
ADCBUF0	0280											ADC Data Buffer 0						0000 0000 0000 0000
ADCBUF1	0282											ADC Data Buffer 1						0000 0000 0000 0000
ADCBUF2	0284											ADC Data Buffer 2						0000 0000 0000 0000
ADCBUF3	0286											ADC Data Buffer 3						0000 0000 0000 0000
ADCBUF4	0288											ADC Data Buffer 4						0000 0000 0000 0000
ADCBUF5	028A											ADC Data Buffer 5						0000 0000 0000 0000
ADCBUF6	028C											ADC Data Buffer 6						0000 0000 0000 0000
ADCBUF7	028E											ADC Data Buffer 7						0000 0000 0000 0000
ADCBUF8	0290											ADC Data Buffer 8						0000 0000 0000 0000
ADCBUF9	0292											ADC Data Buffer 9						0000 0000 0000 0000
ADCBUFA	0294											ADC Data Buffer 10						0000 0000 0000 0000
ADCBUFB	0296											ADC Data Buffer 11						0000 0000 0000 0000
ADCBUFC	0298											ADC Data Buffer 12						0000 0000 0000 0000
ADCBUFD	029A											ADC Data Buffer 13						0000 0000 0000 0000
ADCBUFE	029C											ADC Data Buffer 14						0000 0000 0000 0000
ADCBUFF	029E											ADC Data Buffer 15						0000 0000 0000 0000
ADCON1	02A0	ADON	—	ADSIDL	—	—	—	FORM<1:0>		SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE	0000 0000 0000 0000
ADCON2	02A2	VCFG<2:0>			—	—	CSCNA	CHPS<1:0>		BUFS	—	SMPI<3:0>				BUFM	ALTS	0000 0000 0000 0000
ADCON3	02A4	—	—	—	SAMC<4:0>					ADRC	—	ADCS<5:0>						0000 0000 0000 0000
ADCHS	02A6	CH123NB<1:0>		CH123SB	CH0NB	CH0SB<3:0>				CH123NA<1:0>		CH123SA	CH0NA	CH0SA<3:0>				0000 0000 0000 0000
ADPCFG	02A8	—	—	—	—	—	—	—	—	—	—	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0	0000 0000 0000 0000
ADCSSL	02AA	—	—	—	—	—	—	—	—	—	—	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0	0000 0000 0000 0000

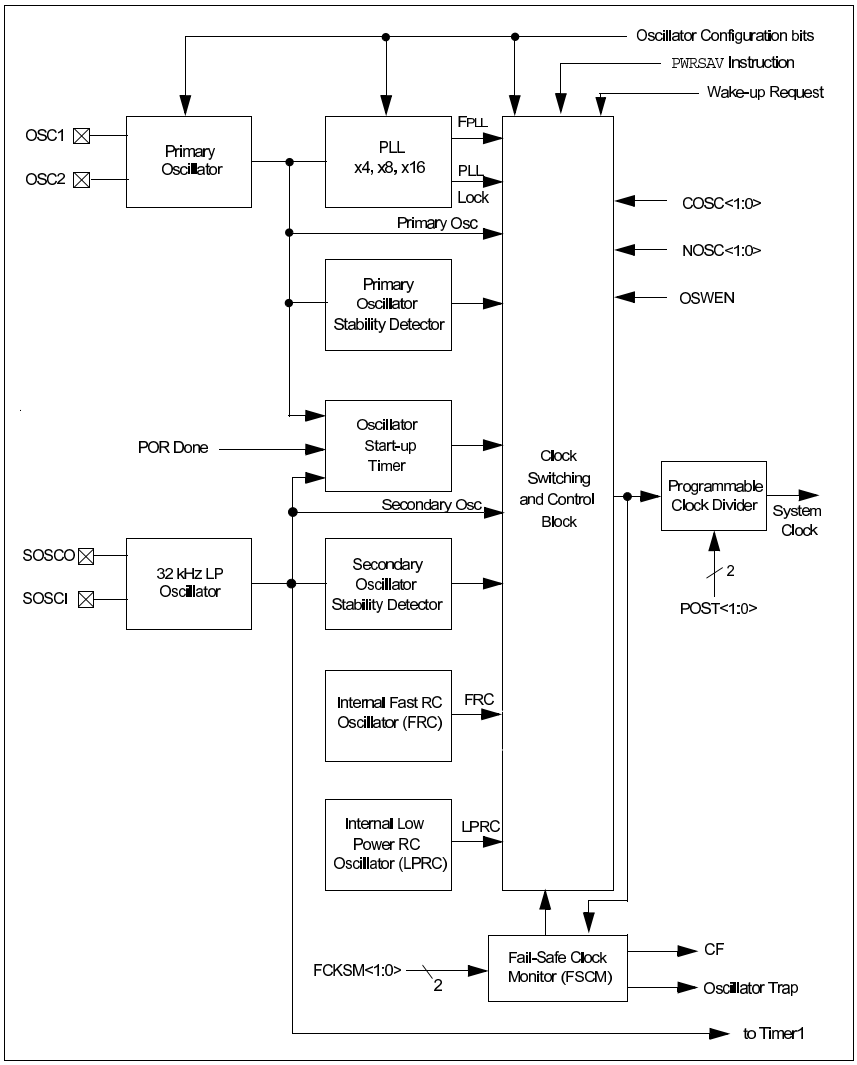
Legend: u = uninitialized bit

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

8 Oscilátor

8.1 Struktura

FIGURE 19-1: OSCILLATOR SYSTEM BLOCK DIAGRAM



19.2 Oscillator Configurations

19.2.1 INITIAL CLOCK SOURCE SELECTION

While coming out of Power-on Reset or Brown-out Reset, the device selects its clock source based on:

a) FOS<1:0> configuration bits that select one of four oscillator groups.

b) AND FPR<3:0> configuration bits that select one of 13 oscillator choices within the primary group.

The selection is as shown in Table 19-2.

19.2.2 OSCILLATOR START-UP TIMER (OST)

In order to ensure that a crystal oscillator (or ceramic resonator) has started and stabilized, an oscillator start-up timer is included. It is a simple 10-bit counter that counts 1024 TOSC cycles before releasing the oscillator clock to the rest of the system. The time-out period is designated as TOST. The TOST time is involved every time the oscillator has to restart (i.e., on POR, BOR and wake-up from Sleep). The oscillator start-up timer is applied to the LP Oscillator, XT, XTL, and HS modes (upon wake-up from Sleep, POR and BOR) for the primary oscillator.

TABLE 19-2: CONFIGURATION BIT VALUES FOR CLOCK SELECTION

Oscillator Mode	Oscillator Source	FOS1	FOS0	FPR3	FPR2	FPR1	FPR0	OSC2 Function
EC	Primary	1	1	1	0	1	1	CLKO
ECIO	Primary	1	1	1	1	0	0	I/O
EC w/ PLL 4x	Primary	1	1	1	1	0	1	I/O
EC w/ PLL 8x	Primary	1	1	1	1	1	0	I/O
EC w/ PLL 16x	Primary	1	1	1	1	1	1	I/O
ERC	Primary	1	1	1	0	0	1	CLKO
ERCIO	Primary	1	1	1	0	0	0	I/O
XT	Primary	1	1	0	1	0	0	OSC2
XT w/ PLL 4x	Primary	1	1	0	1	0	1	OSC2
XT w/ PLL 8x	Primary	1	1	0	1	1	0	OSC2
XT w/ PLL 16x	Primary	1	1	0	1	1	1	OSC2
XTL	Primary	1	1	0	0	0	X	OSC2
HS	Primary	1	1	0	0	1	X	OSC2
LP	Secondary	0	0	—	—	—	—	**(Notes 1, 2)**
FRC	Internal FRC	0	1	—	—	—	—	**(Notes 1, 2)**
LPRC	Internal LPRC	1	0	—	—	—	—	**(Notes 1, 2)**

Note 1: OSC2 pin function is determined by the Primary Oscillator mode selection (FPR<3:0>).

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.

Register 7-1: OSCCON: Oscillator Control Register

Upper Byte:							
R/W-0	R/W-0	R-y	R-y	U-0	U-0	R/W-y	R/W-y
TUN3	**TUN2**	COSC<1:0>		**TUN1**	**TUN0**	NOSC<1:0>	
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R-0	U-0	R/W-0	U-0	R/W-0	R/W-0
POST<1:0>		LOCK	—	CF	—	LPOSCEN	OSWEN
bit 7							bit 0

bit 15-14 **TUN<3:2>:** Upper 2 bits of the TUN bit-field. Refer to the description of TUN<1:0> (OSCCON<11:10>) bits for details.

bit 13-12 **COSC<1:0>:** Current Oscillator Source Status bits
11 = Primary oscillator
10 = Internal LPRC oscillator
01 = Internal FRC oscillator
00 = Low Power 32 kHz Crystal oscillator (Timer1)

bit 11-10 **TUN<1:0>:** Lower 2 bits of the TUN bit-field.
The four bit field specified by TUN<3:0> allows the user to tune the internal fast RC oscillator which has a nominal frequency of 8 MHz. The user may be able to tune the frequency of the FRC oscillator within a range of +/- 12% (or 960 kHz) in steps of 1.5% around the factory-calibrated frequency setting, as follows:
TUN<3:0> = 0111 provides the highest frequency
.....
TUN<3:0> = 0000 provides the factory-calibrated frequency
.....
TUN<3:0> = 1000 provides the lowest frequency

Note: Refer to the device-specific data sheet for the tuning range and tuning step size for the FRC oscillator on your device.

bit 9-8 **NOSC<1:0>:** New Oscillator Group Selection bits
11 = Primary oscillator
10 = Internal LPRC oscillator
01 = Internal FRC oscillator
00 = Low Power 32 kHz Crystal oscillator (Timer1)

bit 7-6 **POST<1:0>:** Oscillator Postscaler Selection bits
11 = Oscillator postscaler divides clock by 64
10 = Oscillator postscaler divides clock by 16
01 = Oscillator postscaler divides clock by 4
00 = Oscillator postscaler does not alter clock

bit 5 **LOCK:** PLL Lock Status bit
1 = Indicates that PLL is in lock
0 = Indicates that PLL is out of lock (or disabled)

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

bit 3 **CF:** Clock Fail Status bit
1 = FSCM has detected a clock failure
0 = FSCM has not detected a clock failure

bit 2 **Unimplemented:** Read as '0'

Register 7-1: OSCCON: Oscillator Control Register (Continued)

bit 1 **LPOSCEN:** 32 kHz LP Oscillator Enable bit
1 = LP oscillator is enabled
0 = LP oscillator is disabled

bit 0 **OSWEN:** Oscillator Switch Enable bit
1 = Request oscillator switch to selection specified by NOSC<1:0> bits
0 = Oscillator switch is complete

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
y = Value set from configuration bits on POR or BOR			

Note: The OSCCON register description and functionality may vary depending on the clock sources available on the selected device. Please refer to the specific device data sheet for additional details on this register.

TABLE 19-7: SYSTEM INTEGRATION REGISTER MAP

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
RCON	0740	TRAPR	IOPUWR	BGST	—	—	—	—	—	EXTR	SWR	SWDTEN	WDTO	SLEEP	IDLE	—	BOR	POR	Depends on type of Reset.
OSCCON	0742	TUN3	TUN2	COSC<1:0>		TUN1	TUN0	NOSC<1:0>		POST<1:0>		LOCK	—	CF	—	LPOSCEN	OSWEN		Depends on configuration bits.

Legend: u = uninitialized bit

TABLE 19-8: DEVICE CONFIGURATION REGISTER MAP

File Name	Addr.	Bits 23-16	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
FOSC	F80000	—	FCKSM<1:0>		—	—	—	—	FOS<1:0>		—	—	—	—	FPR<3:0>			
FWDT	F80002	—	FWDTEN	—	—	—	—	—	—	—	—	—	FWPSA<1:0>		FWPSB<3:0>			
FBORPOR	F80004	—	MCLREN	—	—	—	—	PWMPIN	HPOL	LPOL	BOREN	—	BORV<1:0>		—	—	FPWRT<1:0>	
FGS	F8000A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GCP	GWRP

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

10 Instrukční soubor

dsPIC30F2010

Most single word instructions are executed in a single instruction cycle, unless a conditional test is true or the program counter is changed as a result of the instruction. In these cases, the execution takes two instruction cycles with the additional instruction cycle(s) executed as a NOP. Notable exceptions are the BRA (unconditional/computed branch), indirect CALL/GOTO, all Table Reads and Writes and RETURN/RETFIE instructions, which are single word instructions, but take two or three cycles. Certain instructions that involve skipping over the subsequent instruction, require either two or three cycles if the skip is performed, depending on whether the instruction being skipped is a single word or two-word instruction. Moreover, double-word moves require two cycles. The double-word instructions execute in two instruction cycles.

Note: For more details on the instruction set, refer to the *dsPIC30F Programmer's Reference Manual* (DS70030).

TABLE 20-1: SYMBOLS USED IN OPCODE DESCRIPTIONS

Field	Description
#text	Means literal defined by "text"
(text)	Means "content of text"
[text]	Means "the location addressed by text"
{ }	Optional field or operation
<n:m>	Register bit field
.b	Byte mode selection
.d	Double-word mode selection
.S	Shadow register select
.w	Word mode selection (default)
Acc	One of two accumulators {A, B}
AWB	Accumulator write back destination address register ∈ {W13, [W13]+=2}
bit4	4-bit bit selection field (used in word addressed instructions) ∈ {0...15}
C, DC, N, OV, Z	MCU status bits: Carry, Digit Carry, Negative, Overflow, Zero
Expr	Absolute address, label or expression (resolved by the linker)
f	File register address ∈ {0x0000...0x1FFF}
lit1	1-bit unsigned literal ∈ {0,1}
lit4	4-bit unsigned literal ∈ {0...15}
lit5	5-bit unsigned literal ∈ {0...31}
lit8	8-bit unsigned literal ∈ {0...255}
lit10	10-bit unsigned literal ∈ {0...255} for Byte mode, {0:1023} for Word mode
lit14	14-bit unsigned literal ∈ {0...16384}
lit16	16-bit unsigned literal ∈ {0...65535}
lit23	23-bit unsigned literal ∈ {0...8388608}; LSB must be 0
None	Field does not require an entry, may be blank
OA, OB, SA, SB	DSP status bits: AccA Overflow, AccB Overflow, AccA Saturate, AccB Saturate
PC	Program Counter
Slit10	10-bit signed literal ∈ {-512...511}
Slit16	16-bit signed literal ∈ {-32768...32767}
Slit6	6-bit signed literal ∈ {-16...16}

TABLE 20-1: SYMBOLS USED IN OPCODE DESCRIPTIONS (CONTINUED)

Field	Description
Wb	Base W register ∈ {W0..W15}
Wd	Destination W register ∈ { Wd, [Wd], [Wd++], [Wd–], [++Wd], [–Wd] }
Wdo	Destination W register ∈ { Wnd, [Wnd], [Wnd++], [Wnd–], [++Wnd], [--Wnd], [Wnd+Wb] }
Wm,Wn	Dividend, Divisor working register pair (direct addressing)
Wm*Wm	Multiplicand and Multiplier working register pair for Square instructions ∈ {W4*W4,W5*W5,W6*W6,W7*W7}
Wm*Wn	Multiplicand and Multiplier working register pair for DSP instructions ∈ {W4*W5,W4*W6,W4*W7,W5*W6,W5*W7,W6*W7}
Wn	One of 16 working registers ∈ {W0..W15}
Wnd	One of 16 destination working registers ∈ {W0..W15}
Wns	One of 16 source working registers ∈ {W0..W15}
WREG	W0 (working register used in file register instructions)
Ws	Source W register ∈ { Ws, [Ws], [Ws++], [Ws–], [++Ws], [–Ws] }
Wso	Source W register ∈ { Wns, [Wns], [Wns++], [Wns–], [++Wns], [–Wns], [Wns+Wb] }
Wx	X data space pre-fetch address register for DSP instructions ∈ {[W8]+=6, [W8]+=4, [W8]+=2, [W8], [W8]–=6, [W8]–=4, [W8]-=2, [W9]+=6, [W9]+=4, [W9]+=2, [W9], [W9]–=6, [W9]–=4, [W9]-=2, [W9+W12],none}
Wxd	X data space pre-fetch destination register for DSP instructions ∈ {W4..W7}
Wy	Y data space pre-fetch address register for DSP instructions ∈ {[W10]+=6, [W10]+=4, [W10]+=2, [W10], [W10]–=6, [W10]-=4, [W10]-=2, [W11]+=6, [W11]+=4, [W11]+=2, [W11], [W11]-=6, [W11]–=4, [W11]-=2, [W11+W12], none}
Wyd	Y data space pre-fetch destination register for DSP instructions ∈ {W4..W7}

TABLE 20-2: INSTRUCTION SET OVERVIEW

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycles	Status Flags Affected
1	ADD	ADD	Acc	Add Accumulators	1	1	OA,OB,SA,SB
		ADD	f	f = f + WREG	1	1	C,DC,N,OV,Z
		ADD	f,WREG	WREG = f + WREG	1	1	C,DC,N,OV,Z
		ADD	#lit10,Wn	Wd = lit10 + Wd	1	1	C,DC,N,OV,Z
		ADD	Wb,Ws,Wd	Wd = Wb + Ws	1	1	C,DC,N,OV,Z
		ADD	Wb,#lit5,Wd	Wd = Wb + lit5	1	1	C,DC,N,OV,Z
		ADD	Wso,#Slit4,Acc	16-bit Signed Add to Accumulator	1	1	OA,OB,SA,SB
2	ADDC	ADDC	f	f = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	f,WREG	WREG = f + WREG + (C)	1	1	C,DC,N,OV,Z
		ADDC	#lit10,Wn	Wd = lit10 + Wd + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,Ws,Wd	Wd = Wb + Ws + (C)	1	1	C,DC,N,OV,Z
		ADDC	Wb,#lit5,Wd	Wd = Wb + lit5 + (C)	1	1	C,DC,N,OV,Z
3	AND	AND	f	f = f .AND. WREG	1	1	N,Z
		AND	f,WREG	WREG = f .AND. WREG	1	1	N,Z
		AND	#lit10,Wn	Wd = lit10 .AND. Wd	1	1	N,Z
		AND	Wb,Ws,Wd	Wd = Wb .AND. Ws	1	1	N,Z
		AND	Wb,#lit5,Wd	Wd = Wb .AND. lit5	1	1	N,Z
4	ASR	ASR	f	f = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	f,WREG	WREG = Arithmetic Right Shift f	1	1	C,N,OV,Z
		ASR	Ws,Wd	Wd = Arithmetic Right Shift Ws	1	1	C,N,OV,Z
		ASR	Wb,Wns,Wnd	Wnd = Arithmetic Right Shift Wb by Wns	1	1	N,Z
		ASR	Wb,#lit5,Wnd	Wnd = Arithmetic Right Shift Wb by lit5	1	1	N,Z
5	BCLR	BCLR	f,#bit4	Bit Clear f	1	1	None
		BCLR	Ws,#bit4	Bit Clear Ws	1	1	None
6	BRA	BRA	C,Expr	Branch if Carry	1	1 (2)	None
		BRA	GE,Expr	Branch if greater than or equal	1	1 (2)	None
		BRA	GEU,Expr	Branch if unsigned greater than or equal	1	1 (2)	None
		BRA	GT,Expr	Branch if greater than	1	1 (2)	None
		BRA	GTU,Expr	Branch if unsigned greater than	1	1 (2)	None
		BRA	LE,Expr	Branch if less than or equal	1	1 (2)	None
		BRA	LEU,Expr	Branch if unsigned less than or equal	1	1 (2)	None
		BRA	LT,Expr	Branch if less than	1	1 (2)	None
		BRA	LTU,Expr	Branch if unsigned less than	1	1 (2)	None
		BRA	N,Expr	Branch if Negative	1	1 (2)	None
		BRA	NC,Expr	Branch if Not Carry	1	1 (2)	None
		BRA	NN,Expr	Branch if Not Negative	1	1 (2)	None
		BRA	NOV,Expr	Branch if Not Overflow	1	1 (2)	None
		BRA	NZ,Expr	Branch if Not Zero	1	1 (2)	None
		BRA	OA,Expr	Branch if accumulator A overflow	1	1 (2)	None
		BRA	OB,Expr	Branch if accumulator B overflow	1	1 (2)	None
		BRA	OV,Expr	Branch if Overflow	1	1 (2)	None
		BRA	SA,Expr	Branch if accumulator A saturated	1	1 (2)	None
		BRA	SB,Expr	Branch if accumulator B saturated	1	1 (2)	None
		BRA	Expr	Branch Unconditionally	1	2	None
		BRA	Z,Expr	Branch if Zero	1	1 (2)	None
		BRA	Wn	Computed Branch	1	2	None
7	BSET	BSET	f,#bit4	Bit Set f	1	1	None
		BSET	Ws,#bit4	Bit Set Ws	1	1	None
8	BSW	BSW.C	Ws,Wb	Write C bit to Ws<Wb>	1	1	None
		BSW.Z	Ws,Wb	Write Z bit to Ws<Wb>	1	1	None
9	BTG	BTG	f,#bit4	Bit Toggle f	1	1	None
		BTG	Ws,#bit4	Bit Toggle Ws	1	1	None

TABLE 20-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycles	Status Flags Affected
10	BTSC	BTSC	f,#bit4	Bit Test f, Skip if Clear	1	1 (2 or 3)	None
		BTSC	Ws,#bit4	Bit Test Ws, Skip if Clear	1	1 (2 or 3)	None
11	BTSS	BTSS	f,#bit4	Bit Test f, Skip if Set	1	1 (2 or 3)	None
		BTSS	Ws,#bit4	Bit Test Ws, Skip if Set	1	1 (2 or 3)	None
12	BTST	BTST	f,#bit4	Bit Test f	1	1	Z
		BTST.C	Ws,#bit4	Bit Test Ws to C	1	1	C
		BTST.Z	Ws,#bit4	Bit Test Ws to Z	1	1	Z
		BTST.C	Ws,Wb	Bit Test Ws<Wb> to C	1	1	C
		BTST.Z	Ws,Wb	Bit Test Ws<Wb> to Z	1	1	Z
13	BTSTS	BTSTS	f,#bit4	Bit Test then Set f	1	1	Z
		BTSTS.C	Ws,#bit4	Bit Test Ws to C, then Set	1	1	C
		BTSTS.Z	Ws,#bit4	Bit Test Ws to Z, then Set	1	1	Z
14	CALL	CALL	lit23	Call subroutine	2	2	None
		CALL	Wn	Call indirect subroutine	1	2	None
15	CLR	CLR	f	f = 0x0000	1	1	None
		CLR	WREG	WREG = 0x0000	1	1	None
		CLR	Ws	Ws = 0x0000	1	1	None
		CLR	Acc,Wx,Wxd,Wy,Wyd,AWB	Clear Accumulator	1	1	OA,OB,SA,SB
16	CLRWDT	CLRWDT		Clear Watchdog Timer	1	1	WDTO,Sleep
17	COM	COM	f	$f = \bar{f}$	1	1	N,Z
		COM	f,WREG	$WREG = \bar{f}$	1	1	N,Z
		COM	Ws,Wd	$Wd = \overline{Ws}$	1	1	N,Z
18	CP	CP	f	Compare f with WREG	1	1	C,DC,N,OV,Z
		CP	Wb,#lit5	Compare Wb with lit5	1	1	C,DC,N,OV,Z
		CP	Wb,Ws	Compare Wb with Ws (Wb - Ws)	1	1	C,DC,N,OV,Z
19	CP0	CP0	f	Compare f with 0x0000	1	1	C,DC,N,OV,Z
		CP0	Ws	Compare Ws with 0x0000	1	1	C,DC,N,OV,Z
20	CP1	CP1	f	Compare f with 0xFFFF	1	1	C,DC,N,OV,Z
		CP1	Ws	Compare Ws with 0xFFFF	1	1	C,DC,N,OV,Z
21	CPB	CPB	f	Compare f with WREG, with Borrow	1	1	C,DC,N,OV,Z
		CPB	Wb,#lit5	Compare Wb with lit5, with Borrow	1	1	C,DC,N,OV,Z
		CPB	Wb,Ws	Compare Wb with Ws, with Borrow (Wb - Ws - $\bar{C}$)	1	1	C,DC,N,OV,Z
22	CPSEQ	CPSEQ	Wb, Wn	Compare Wb with Wn, skip if =	1	1 (2 or 3)	None
23	CPSGT	CPSGT	Wb, Wn	Compare Wb with Wn, skip if >	1	1 (2 or 3)	None
24	CPSLT	CPSLT	Wb, Wn	Compare Wb with Wn, skip if <	1	1 (2 or 3)	None
25	CPSNE	CPSNE	Wb, Wn	Compare Wb with Wn, skip if ≠	1	1 (2 or 3)	None
26	DAW	DAW	Wn	Wn = decimal adjust Wn	1	1	C
27	DEC	DEC	f	f = f -1	1	1	C,DC,N,OV,Z
		DEC	f,WREG	WREG = f -1	1	1	C,DC,N,OV,Z
		DEC	Ws,Wd	Wd = Ws - 1	1	1	C,DC,N,OV,Z
28	DEC2	DEC2	f	f = f -2	1	1	C,DC,N,OV,Z
		DEC2	f,WREG	WREG = f -2	1	1	C,DC,N,OV,Z
		DEC2	Ws,Wd	Wd = Ws - 2	1	1	C,DC,N,OV,Z

TABLE 20-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycles	Status Flags Affected
29	DISI	DISI	#lit14	Disable Interrupts for k instruction cycles	1	1	None
30	DIV	DIV.S	Wm,Wn	Signed 16/16-bit Integer Divide	1	18	N,Z,C, OV
		DIV.SD	Wm,Wn	Signed 32/16-bit Integer Divide	1	18	N,Z,C, OV
		DIV.U	Wm,Wn	Unsigned 16/16-bit Integer Divide	1	18	N,Z,C, OV
		DIV.UD	Wm,Wn	Unsigned 32/16-bit Integer Divide	1	18	N,Z,C, OV
31	DIVF	DIVF	Wm,Wn	Signed 16/16-bit Fractional Divide	1	18	N,Z,C, OV
32	DO	DO	#lit14,Expr	Do code to PC+Expr, lit14+1 times	2	2	None
		DO	Wn,Expr	Do code to PC+Expr, (Wn)+1 times	2	2	None
33	ED	ED	Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean Distance (no accumulate)	1	1	OA,OB,OAB, SA,SB,SAB
34	EDAC	EDAC	Wm*Wm,Acc,Wx,Wy,Wxd	Euclidean Distance	1	1	OA,OB,OAB, SA,SB,SAB
35	EXCH	EXCH	Wns,Wnd	Swap Wns with Wnd	1	1	None
36	FBCL	FBCL	Ws,Wnd	Find Bit Change from Left (MSb) Side	1	1	C
37	FF1L	FF1L	Ws,Wnd	Find First One from Left (MSb) Side	1	1	C
38	FF1R	FF1R	Ws,Wnd	Find First One from Right (LSb) Side	1	1	C
39	GOTO	GOTO	Expr	Go to address	2	2	None
		GOTO	Wn	Go to indirect	1	2	None
40	INC	INC	f	f = f + 1	1	1	C,DC,N,OV,Z
		INC	f,WREG	WREG = f + 1	1	1	C,DC,N,OV,Z
		INC	Ws,Wd	Wd = Ws + 1	1	1	C,DC,N,OV,Z
41	INC2	INC2	f	f = f + 2	1	1	C,DC,N,OV,Z
		INC2	f,WREG	WREG = f + 2	1	1	C,DC,N,OV,Z
		INC2	Ws,Wd	Wd = Ws + 2	1	1	C,DC,N,OV,Z
42	IOR	IOR	f	f = f .IOR. WREG	1	1	N,Z
		IOR	f,WREG	WREG = f .IOR. WREG	1	1	N,Z
		IOR	#lit10,Wn	Wd = lit10 .IOR. Wd	1	1	N,Z
		IOR	Wb,Ws,Wd	Wd = Wb .IOR. Ws	1	1	N,Z
		IOR	Wb,#lit5,Wd	Wd = Wb .IOR. lit5	1	1	N,Z
43	LAC	LAC	Wso,#Slit4,Acc	Load Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
44	LNK	LNK	#lit14	Link frame pointer	1	1	None
45	LSR	LSR	f	f = Logical Right Shift f	1	1	C,N,OV,Z
		LSR	f,WREG	WREG = Logical Right Shift f	1	1	C,N,OV,Z
		LSR	Ws,Wd	Wd = Logical Right Shift Ws	1	1	C,N,OV,Z
		LSR	Wb,Wns,Wnd	Wnd = Logical Right Shift Wb by Wns	1	1	N,Z
		LSR	Wb,#lit5,Wnd	Wnd = Logical Right Shift Wb by lit5	1	1	N,Z
46	MAC	MAC	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd, AWB	Multiply and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB
		MAC	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square and Accumulate	1	1	OA,OB,OAB, SA,SB,SAB
47	MOV	MOV	f,Wn	Move f to Wn	1	1	None
		MOV	f	Move f to f	1	1	N,Z
		MOV	f,WREG	Move f to WREG	1	1	N,Z
		MOV	#lit16,Wn	Move 16-bit literal to Wn	1	1	None
		MOV.b	#lit8,Wn	Move 8-bit literal to Wn	1	1	None
		MOV	Wn,f	Move Wn to f	1	1	None
		MOV	Wso,Wdo	Move Ws to Wd	1	1	None
		MOV	WREG,f	Move WREG to f	1	1	N,Z
		MOV.D	Wns,Wd	Move Double from W(ns):W(ns+1) to Wd	1	2	None
		MOV.D	Ws,Wnd	Move Double from Ws to W(nd+1):W(nd)	1	2	None
48	MOVSAC	MOVSAC	Acc,Wx,Wxd,Wy,Wyd,AWB	Pre-fetch and store accumulator	1	1	None
49	MPY	MPY	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	Multiply Wm by Wn to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		MPY	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd	Square Wm to Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
50	MPY.N	MPY.N	Wm*Wn,Acc,Wx,Wxd,Wy,Wyd	-(Multiply Wm by Wn) to Accumulator	1	1	None

TABLE 20-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycles	Status Flags Affected
51	MSC	MSC	Wm*Wm,Acc,Wx,Wxd,Wy,Wyd, AWB	Multiply and Subtract from Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
52	MUL	MUL.SS	Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * signed(Ws)	1	1	None
		MUL.SU	Wb,Ws,Wnd	{Wnd+1, Wnd} = signed(Wb) * unsigned(Ws)	1	1	None
		MUL.US	Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * signed(Ws)	1	1	None
		MUL.UU	Wb,Ws,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(Ws)	1	1	None
		MUL.SU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = signed(Wb) * unsigned(lit5)	1	1	None
		MUL.UU	Wb,#lit5,Wnd	{Wnd+1, Wnd} = unsigned(Wb) * unsigned(lit5)	1	1	None
		MUL	f	W3:W2 = f * WREG	1	1	None
53	NEG	NEG	Acc	Negate Accumulator	1	1	OA,OB,OAB, SA,SB,SAB
		NEG	f	f = $\bar{f}$ + 1	1	1	C,DC,N,OV,Z
		NEG	f,WREG	WREG = $\bar{f}$ + 1	1	1	C,DC,N,OV,Z
		NEG	Ws,Wd	Wd = $\overline{Ws}$ + 1	1	1	C,DC,N,OV,Z
54	NOP	NOP		No Operation	1	1	None
		NOPR		No Operation	1	1	None
55	POP	POP	f	Pop f from top-of-stack (TOS)	1	1	None
		POP	Wdo	Pop from top-of-stack (TOS) to Wdo	1	1	None
		POP.D	Wnd	Pop from top-of-stack (TOS) to W(nd):W(nd+1)	1	2	None
		POP.S		Pop Shadow Registers	1	1	All
56	PUSH	PUSH	f	Push f to top-of-stack (TOS)	1	1	None
		PUSH	Wso	Push Wso to top-of-stack (TOS)	1	1	None
		PUSH.D	Wns	Push W(ns):W(ns+1) to top-of-stack (TOS)	1	2	None
		PUSH.S		Push Shadow Registers	1	1	None
57	PWRSAV	PWRSAV	#lit1	Go into Sleep or Idle mode	1	1	WDTO,Sleep
58	RCALL	RCALL	Expr	Relative Call	1	2	None
		RCALL	Wn	Computed Call	1	2	None
59	REPEAT	REPEAT	#lit14	Repeat Next Instruction lit14+1 times	1	1	None
		REPEAT	Wn	Repeat Next Instruction (Wn)+1 times	1	1	None
60	RESET	RESET		Software device Reset	1	1	None
61	RETFIE	RETFIE		Return from interrupt	1	3 (2)	None
62	RETLW	RETLW	#lit10,Wn	Return with literal in Wn	1	3 (2)	None
63	RETURN	RETURN		Return from Subroutine	1	3 (2)	None
64	RLC	RLC	f	f = Rotate Left through Carry f	1	1	C,N,Z
		RLC	f,WREG	WREG = Rotate Left through Carry f	1	1	C,N,Z
		RLC	Ws,Wd	Wd = Rotate Left through Carry Ws	1	1	C,N,Z
65	RLNC	RLNC	f	f = Rotate Left (No Carry) f	1	1	N,Z
		RLNC	f,WREG	WREG = Rotate Left (No Carry) f	1	1	N,Z
		RLNC	Ws,Wd	Wd = Rotate Left (No Carry) Ws	1	1	N,Z
66	RRC	RRC	f	f = Rotate Right through Carry f	1	1	C,N,Z
		RRC	f,WREG	WREG = Rotate Right through Carry f	1	1	C,N,Z
		RRC	Ws,Wd	Wd = Rotate Right through Carry Ws	1	1	C,N,Z
67	RRNC	RRNC	f	f = Rotate Right (No Carry) f	1	1	N,Z
		RRNC	f,WREG	WREG = Rotate Right (No Carry) f	1	1	N,Z
		RRNC	Ws,Wd	Wd = Rotate Right (No Carry) Ws	1	1	N,Z
68	SAC	SAC	Acc,#Slit4,Wdo	Store Accumulator	1	1	None
		SAC.R	Acc,#Slit4,Wdo	Store Rounded Accumulator	1	1	None
69	SE	SE	Ws,Wnd	Wnd = sign extended Ws	1	1	C,N,Z
70	SETM	SETM	f	f = 0xFFFF	1	1	None
		SETM	WREG	WREG = 0xFFFF	1	1	None
		SETM	Ws	Ws = 0xFFFF	1	1	None

TABLE 20-2: INSTRUCTION SET OVERVIEW (CONTINUED)

Base Instr #	Assembly Mnemonic	Assembly Syntax		Description	# of words	# of cycles	Status Flags Affected
71	SFTAC	SFTAC	Acc,Wn	Arithmetic Shift Accumulator by (Wn)	1	1	OA,OB,OAB, SA,SB,SAB
		SFTAC	Acc,#Slit6	Arithmetic Shift Accumulator by Slit6	1	1	OA,OB,OAB, SA,SB,SAB
72	SL	SL	f	f = Left Shift f	1	1	C,N,OV,Z
		SL	f,WREG	WREG = Left Shift f	1	1	C,N,OV,Z
		SL	Ws,Wd	Wd = Left Shift Ws	1	1	C,N,OV,Z
		SL	Wb,Wns,Wnd	Wnd = Left Shift Wb by Wns	1	1	N,Z
		SL	Wb,#lit5,Wnd	Wnd = Left Shift Wb by lit5	1	1	N,Z
73	SUB	SUB	Acc	Subtract Accumulators	1	1	OA,OB,OAB, SA,SB,SAB
		SUB	f	f = f - WREG	1	1	C,DC,N,OV,Z
		SUB	f,WREG	WREG = f - WREG	1	1	C,DC,N,OV,Z
		SUB	#lit10,Wn	Wn = Wn - lit10	1	1	C,DC,N,OV,Z
		SUB	Wb,Ws,Wd	Wd = Wb - Ws	1	1	C,DC,N,OV,Z
		SUB	Wb,#lit5,Wd	Wd = Wb - lit5	1	1	C,DC,N,OV,Z
74	SUBB	SUBB	f	f = f - WREG - (C)	1	1	C,DC,N,OV,Z
		SUBB	f,WREG	WREG = f - WREG - (C)	1	1	C,DC,N,OV,Z
		SUBB	#lit10,Wn	Wn = Wn - lit10 - (C)	1	1	C,DC,N,OV,Z
		SUBB	Wb,Ws,Wd	Wd = Wb - Ws - (C)	1	1	C,DC,N,OV,Z
		SUBB	Wb,#lit5,Wd	Wd = Wb - lit5 - (C)	1	1	C,DC,N,OV,Z
75	SUBR	SUBR	f	f = WREG - f	1	1	C,DC,N,OV,Z
		SUBR	f,WREG	WREG = WREG - f	1	1	C,DC,N,OV,Z
		SUBR	Wb,Ws,Wd	Wd = Ws - Wb	1	1	C,DC,N,OV,Z
		SUBR	Wb,#lit5,Wd	Wd = lit5 - Wb	1	1	C,DC,N,OV,Z
76	SUBBR	SUBBR	f	f = WREG - f - (C)	1	1	C,DC,N,OV,Z
		SUBBR	f,WREG	WREG = WREG -f - (C)	1	1	C,DC,N,OV,Z
		SUBBR	Wb,Ws,Wd	Wd = Ws - Wb - (C)	1	1	C,DC,N,OV,Z
		SUBBR	Wb,#lit5,Wd	Wd = lit5 - Wb - (C)	1	1	C,DC,N,OV,Z
77	SWAP	SWAP.b	Wn	Wn = nibble swap Wn	1	1	None
		SWAP	Wn	Wn = byte swap Wn	1	1	None
78	TBLRDH	TBLRDH	Ws,Wd	Read Prog<23:16> to Wd<7:0>	1	2	None
79	TBLRDL	TBLRDL	Ws,Wd	Read Prog<15:0> to Wd	1	2	None
80	TBLWTH	TBLWTH	Ws,Wd	Write Ws<7:0> to Prog<23:16>	1	2	None
81	TBLWTL	TBLWTL	Ws,Wd	Write Ws to Prog<15:0>	1	2	None
82	ULNK	ULNK		Unlink frame pointer	1	1	None
83	XOR	XOR	f	f = f .XOR. WREG	1	1	N,Z
		XOR	f,WREG	WREG = f .XOR. WREG	1	1	N,Z
		XOR	#lit10,Wn	Wd = lit10 .XOR. Wd	1	1	N,Z
		XOR	Wb,Ws,Wd	Wd = Wb .XOR. Ws	1	1	N,Z
		XOR	Wb,#lit5,Wd	Wd = Wb .XOR. lit5	1	1	N,Z
84	ZE	ZE	Ws,Wnd	Wnd = Zero-Extend Ws	1	1	C,Z,N

A MCP4921/4922

A.1 Úvod

MICROCHIP

MCP4921/4922

12-Bit DAC with SPI™ Interface

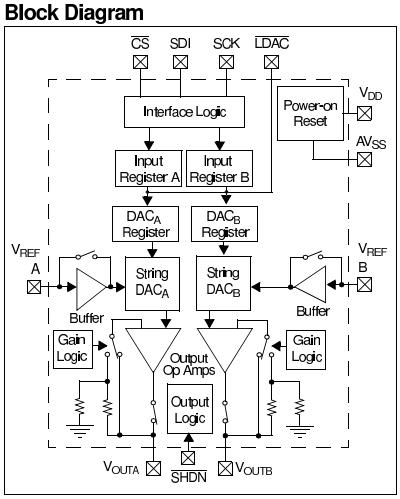
Features

- 12-Bit Resolution
- ±0.2 LSB DNL (typ)
- ±2 LSB INL (typ)
- Single or Dual Channel
- Rail-to-Rail Output
- SPI™ Interface with 20 MHz Clock Support
- Simultaneous Latching of the Dual DACs w/$\overline{\text{LDAC}}$
- Fast Settling Time of 4.5 µs
- Selectable Unity or 2x Gain Output
- 450 kHz Multiplier Mode
- External V_{REF} Input
- 2.7V to 5.5V Single-Supply Operation
- Extended Temperature Range: -40°C to +125°C

Applications

- Set Point or Offset Trimming
- Sensor Calibration
- Digitally-Controlled Multiplier/Divider
- Portable Instrumentation (Battery-Powered)
- Motor Feedback Loop Control

Block Diagram



Description

The Microchip Technology Inc. MCP492X are 2.7–5.5V, low-power, low DNL, 12-Bit Digital-to-Analog Converters (DACs) with optional 2x buffered output and SPI interface.

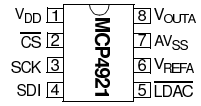
The MCP492X are DACs that provide high accuracy and low noise performance for industrial applications where calibration or compensation of signals (such as temperature, pressure and humidity) are required.

The MCP492X are available in the extended temperature range and PDIP, SOIC, MSOP and TSSOP packages.

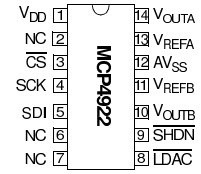
The MCP492X devices utilize a resistive string architecture, with its inherent advantages of low DNL error, low ratio metric temperature coefficient and fast settling time. These devices are specified over the extended temperature range. The MCP492X include double-buffered inputs, allowing simultaneous updates using the $\overline{\text{LDAC}}$ pin. These devices also incorporate a Power-On Reset (POR) circuit to ensure reliable power-up.

Package Types

8-Pin PDIP, SOIC, MSOP



14-Pin PDIP, SOIC, TSSOP



A.2 Vývody

3.0 PIN DESCRIPTIONS

The descriptions of the pins are listed in Table 3-1.

TABLE 3-1: PIN FUNCTION TABLE

MCP4921 Pin No.	MCP4922 Pin No.	Symbol	Function
1	1	V_{DD}	Positive Power Supply Input (2.7V to 5.5V)
—	2	NC	No Connection
2	3	$\overline{CS}$	Chip Select Input
3	4	SCK	Serial Clock Input
4	5	SDI	Serial Data Input
—	6	NC	No Connection
—	7	NC	No Connection
5	8	$\overline{LDAC}$	Syncronization input used to transfer DAC settings from serial latches to the output latches.
—	9	$\overline{SHDN}$	Hardware Shutdown Input
—	10	V_{OUTB}	DAC_B Output
—	11	V_{REFB}	DAC_B Voltage Input (AV_{SS} to V_{DD})
7	12	AV_{SS}	Analog ground
6	13	V_{REFA}	DAC_A Voltage Input (AV_{SS} to V_{DD})
8	14	V_{OUTA}	DAC_A Output

3.1 Positive Power Supply Input (V_{DD})

V_{DD} is the positive power supply input. The input power supply is relative to AV_{SS} and can range from 2.7V to 5.5V. A decoupling capacitor on V_{DD} is recommended to achieve maximum performance.

3.2 Chip Select ($\overline{CS}$)

$\overline{CS}$ is the chip select input, which requires an active-low signal to enable serial clock and data functions.

3.3 Serial Clock Input (SCK)

SCK is the SPI compatible serial clock input.

3.4 Serial Data Input (SDI)

SDI is the SPI compatible serial data input.

3.5 Latch DAC Input ($\overline{LDAC}$)

$\overline{LDAC}$ (the latch DAC syncronization input) transfers the input latch registers to the DAC registers (output latches) when low. Can also be tied low if transfer on the rising edge of $\overline{CS}$ is desired.

3.6 Hardware Shutdown Input ($\overline{SHDN}$)

$\overline{SHDN}$ is the hardware shutdown input that requires an active-low input signal to configure the DACs in their low-power Standby mode.

3.7 DAC_X Outputs (V_{OUTA}, V_{OUTB})

V_{OUTA} and V_{OUTB} are DAC outputs. The DAC output amplifier drives these pins with a range of AV_{SS} to V_{DD}.

3.8 DAC_X Voltage Reference Inputs (V_{REFA}, V_{REFB})

V_{REFA} and V_{REFB} are DAC voltage reference inputs. The analog signal on these pins is utilized to set the reference voltage on the string DAC. The input signal can range from AV_{SS} to V_{DD}.

3.9 Analog Ground (AV_{SS})

AV_{SS} is the analog ground pin.

5.0 SERIAL INTERFACE

5.1 Overview

The MCP492X family is designed to interface directly with the Serial Peripheral Interface (SPI) port, available on many microcontrollers, and supports Mode 0,0 and Mode 1,1. Commands and data are sent to the device via the SDI pin, with data being clocked-in on the rising edge of SCK. The communications are unidirectional and, thus, data cannot be read out of the MCP492X. The $\overline{CS}$ pin must be held low for the duration of a write command. The write command consists of 16 bits and is used to configure the DAC's control and data latches. Register 5-1 details the input registers used to configure and load the DAC_A and DAC_B registers. Refer to Figure 1-1 and **Section 1.0 "Electrical Characteristics"** AC Electrical Characteristics table for detailed input and output timing specifications for both Mode 0,0 and Mode 1,1 operation.

5.2 Write Command

The write command is initiated by driving the $\overline{CS}$ pin low, followed by clocking the four configuration bits and the 12 data bits into the SDI pin on the rising edge of SCK. The $\overline{CS}$ pin is then raised, causing the data to be latched into the selected DAC's input registers. The MCP492X utilizes a double-buffered latch structure to allow both DAC_A's and DAC_B's outputs to be syncronized with the $\overline{LDAC}$ pin, if desired. Upon the $\overline{LDAC}$ pin achieving a low state, the values held in the DAC's input registers are transferred into the DACs' output registers. The outputs will transition to the value and held in the DAC_X register.

All writes to the MCP492X are 16-bit words. Any clocks past 16 will be ignored. The most significant four bits are configuration bits. The remaining 12 bits are data bits. No data can be transferred into the device with $\overline{CS}$ high. This transfer will only occur if 16 clocks have been transferred into the device. If the rising edge of $\overline{CS}$ occurs prior, shifting of data into the input registers will be aborted.

REGISTER 5-1: WRITE COMMAND REGISTER

Upper Half:							
W-x	W-x	W-x	W-0	W-x	W-x	W-x	W-x
$\overline{A}$/B	BUF	$\overline{GA}$	$\overline{SHDN}$	D11	D10	D9	D8
bit 15							bit 8

Lower Half:							
W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
D7	D6	D5	D4	D3	D2	D1	D0
bit 7							bit 0

bit 15 **$\overline{A}$/B:** DAC_A or DAC_B Select bit
1 = Write to DAC_B
0 = Write to DAC_A

bit 14 **BUF:** V_{REF} Input Buffer Control bit
1 = Buffered
0 = Unbuffered

bit 13 **$\overline{GA}$:** Output Gain Select bit
1 = 1x ($V_{OUT} = V_{REF} * D/4096$)
0 = 2x ($V_{OUT} = 2 * V_{REF} * D/4096$)

bit 12 **$\overline{SHDN}$:** Output Power Down Control bit
1 = Output Power Down Control bit
0 = Output buffer disabled, Output is high impedance

bit 11-0 **D11:D0:** DAC Data bits
12 bit number "D" which sets the output value. Contains a value between 0 and 4095.

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

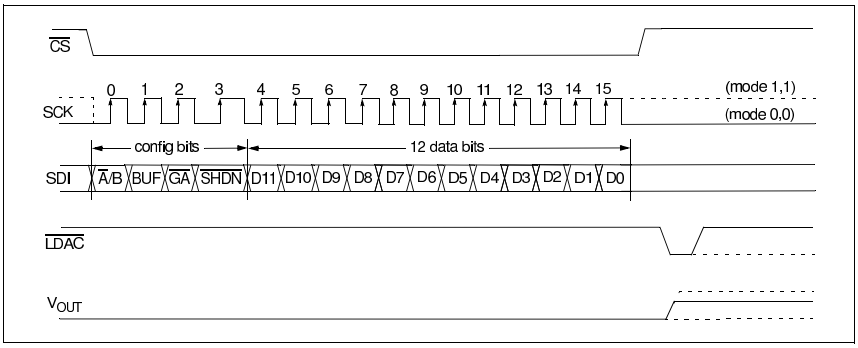


FIGURE 5-1: *Write Command.*