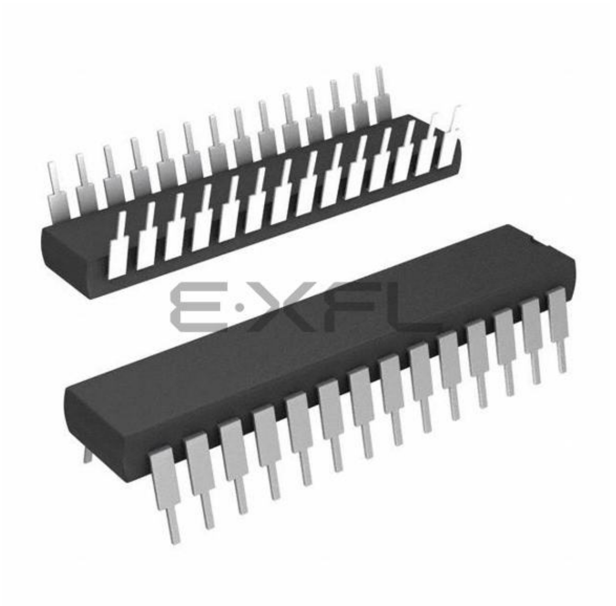


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	PIC
Core Size	16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, LVD, POR, PWM, WDT
Number of I/O	23
Program Memory Size	8KB (2.75K x 24)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	2K x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 5V
Data Converters	A/D 19x10b/12b; D/A 2x8b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Through Hole
Package / Case	28-DIP (0.300", 7.62mm)
Supplier Device Package	28-SPDIP
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/pic24fv08km202-i-sp

Peripheral Features

- High-Current Sink/Source, 18 mA/18 mA All Ports
- Independent Ultra Low-Power, 32 kHz Timer Oscillator
- Up to Two Master Synchronous Serial Ports (MSSPs) with SPI and I^2C™ modes:

  In SPI mode:
  - User-configurable SCKx and SDOx pin outputs
  - Daisy-chaining of SPI slave devices

  In I^2C mode:
  - Serial clock synchronization (clock stretching)
  - Bus collision detection and will arbitrate accordingly
  - Support for 16-bit read/write interface
- Up to Two Enhanced Addressable UARTs:
  - LIN/J2602 bus support (auto-wake-up, Auto-Baud Detect, Break character support)
  - High and low speed (SCI)
  - IrDA® mode (hardware encoder/decoder function)
- Two External Interrupt Pins
- Hardware Real-Time Clock and Calendar (RTCC)
- Configurable Reference Clock Output (REFO)
- Two Configurable Logic Cells (CLC)
- Up to Two Single Output Capture/Compare/PWM (SCCP) modules and up to Three Multiple Output Capture/Compare/PWM (MCCP) modules

Special Microcontroller Features

- Wide Operating Voltage Range Options:
  - 1.8V to 3.6V (PIC24F devices)
  - 2.0V to 5.0V (PIC24FV devices)
- Selectable Power Management modes:
  - Idle: CPU shuts down, allowing for significant power reduction
  - Sleep: CPU and peripherals shut down for substantial power reduction and fast wake-up
  - Retention Sleep mode: PIC24FV devices can enter Sleep mode, employing the Retention Regulator, further reducing power consumption
  - Doze: CPU can run at a lower frequency than peripherals, a user-programmable feature
  - Alternate Clock modes allow on-the-fly switching to a lower clock speed for selective power reduction
- Fail-Safe Clock Monitor:
  - Detects clock failure and switches to on-chip, low-power RC Oscillator
- Ultra Low-Power Wake-up Pin Provides an External Trigger for Wake from Sleep
- 10,000 Erase/Write Cycle Endurance Flash Program Memory, Typical
- 100,000 Erase/Write Cycle Endurance Data EEPROM, Typical
- Flash and Data EEPROM Data Retention: 20 Years Minimum
- Self-Programmable under Software Control
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its Own On-Chip RC Oscillator for Reliable Operation
- On-Chip Regulator for 5V Operation
- Selectable Windowed WDT Feature
- Selectable Oscillator Options including:
  - 4x Phase Locked Loop (PLL)
- 8 MHz (FRC) Internal RC Oscillator:
  - HS/EC, High-Speed Crystal/Resonator Oscillator or External Clock
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Emulation (ICE) – via Two Pins
- In-Circuit Debugging
- Programmable High/Low-Voltage Detect (HLVD) module
- Programmable Brown-out Reset (BOR):
  - Software enable feature
  - Configurable shutdown in Sleep
  - Auto-configures power mode and sensitivity based on device operating speed
  - LPBOR available for re-arming of the POR

1.1.4 EASY MIGRATION

The PIC24FV16KM204 family devices have two variants. The KM20X variant provides the full feature set of the device, while the KM10X offers a reduced peripheral set, allowing for the balance of features and cost (refer to Table 1-1). Both variants allow for a smooth migration path as applications grow and evolve.

The consistent pinout scheme used throughout the entire family also helps in migrating to the next larger device. This is true when moving between devices with the same pin count, different die variants, or even moving from 20-pin or 28-pin devices to 44-pin/48-pin devices.

The PIC24F family is pin compatible with devices in the dsPIC33 family, and shares some compatibility with the pinout schema for PIC18 and dsPIC30. This extends the ability of applications to grow from the relatively simple to the powerful and complex, yet still selecting a Microchip device.

1.2 Other Special Features

- Communications: The PIC24FV16KM204 family incorporates a range of serial communication peripherals to handle a range of application requirements. There is an MSSP module which implements both SPI and I^{2}C™ protocols, and supports both Master and Slave modes of operation for each. Devices also include one of two UARTs with built-in IrDA® encoders/decoders.
- Analog Features: Select members of the PIC24FV16KM204 family include two 8-bit Digital-to-Analog Converters which offer support in Idle mode, and left and right justified input data, as well as up to two operational amplifiers with selectable power and speed modes.
- Real-Time Clock/Calendar (RTCC): This module implements a full-featured clock and calendar with alarm functions in hardware, freeing up timer resources and program memory space for use of the core application.
- 12-Bit A/D Converter: This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period, and faster sampling speed. The 16-deep result buffer can be used either in Sleep, to reduce power, or in Active mode to improve throughput.
- Charge Time Measurement Unit (CTMU) Interface: The PIC24FV16KM204 family includes the new CTMU interface module, which can be used for capacitive touch sensing, proximity sensing, and also for precision time measurement and pulse generation. The CTMU can also be connected to the operational amplifiers to provide active guarding, which provides increased robustness in the presence of noise in capacitive touch applications.

1.3 Details on Individual Family Members

Devices in the PIC24FV16KM204 family are available in 20-pin, 28-pin, 44-pin and 48-pin packages. The general block diagram for all devices is shown in Figure 1-1.

Members of the PIC24FV16KM204 family are available as both standard and high-voltage devices. High-voltage devices, designated with an “FV” in the part number (such as PIC24FV16KM204), accommodate an operating VDD range of 2.0V to 5.5V and have an on-board voltage regulator that powers the core. Peripherals operate at VDD.

Standard devices, designated by “F” (such as PIC24F16KM204), function over a lower VDD range of 1.8V to 3.6V. These parts do not have an internal regulator, and both the core and peripherals operate directly from VDD.

The PIC24FV16KM204 family may be thought of as two different device groups, both offering slightly different sets of features. These differ from each other in multiple ways:

- The size of the Flash program memory
- The number of external analog channels available
- The number of Digital-to-Analog Converters
- The number of operational amplifiers
- The number of analog comparators
- The presence of a Real-Time Clock and Calendar (RTCC)
- The number and type of CCP modules (i.e., MCCP vs. SCCP)
- The number of serial communication modules (both MSSPs and UARTs)
- The number of Configurable Logic Cell (CLC) modules

The general differences between the different sub-families are shown in Table 1-1 and Table 1-2.

A list of the pin features available on the PIC24FV16KM204 family devices, sorted by function, is provided in Table 1-5.

TABLE 1-4: DEVICE FEATURES FOR THE PIC24FV16KM104 FAMILY

Features	PIC24FV16KM104	PIC24FV16KM102	PIC24FV08KM102	PIC24FV08KM101
Operating Frequency	DC-32 MHz			
Program Memory (bytes)	16K	16K	8K	8K
Program Memory (instructions)	5632	5632	2816	2816
Data Memory (bytes)	1024			
Data EEPROM Memory (bytes)	512			
Interrupt Sources (soft vectors/NMI traps)	25 (21/4)			
Voltage Range	2.0-5.5V			
I/O Ports	PORTA<11:7,5:0> PORTB<15:0> PORTC<9:0>	PORTA<7,5:0> PORTB<15:0>		PORTA<5:0> PORTB<15:12,9:7, 4,2:0>
Total I/O Pins	37	23		17
Timers	5 (One 16-bit timer, two MCCPs/SCCPs with up to two 16/32 timers each)			
Capture/Compare/PWM modules MCCP SCCP	 1 1			
Serial Communications MSSP UART	 1 1			
Input Change Notification Interrupt	36	22		16
12-Bit Analog-to-Digital Module (input channels)	22	19		16
Analog Comparators	1			
8-Bit Digital-to-Analog Converters	—			
Operational Amplifiers	—			
Charge Time Measurement Unit (CTMU)	Yes			
Real-Time Clock and Calendar (RTCC)	—			
Configurable Logic Cell (CLC)	1			
Resets (and delays)	POR, BOR, RESET Instruction, $\overline{\text{MCLR}}$, WDT, Illegal Opcode, REPEAT Instruction, Hardware Traps, Configuration Word Mismatch (PWRT, OST, PLL Lock)			
Instruction Set	76 Base Instructions, Multiple Addressing Mode Variations			
Packages	44-Pin QFN/TQFP, 48-Pin UQFN	28-Pin SPDIP/SSOP/SOIC/QFN		20-Pin SOIC/SSOP/PDIP

TABLE 1-5: PIC24FV16KM204 FAMILY PINOUT DESCRIPTION (CONTINUED)

Function	F					FV					I/O	Buffer	Description
	Pin Number					Pin Number							
	20-Pin PDIP/ SSOP/ SOIC	28-Pin PDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN	20-Pin PDIP/ SSOP/ SOIC	28-Pin PDIP/ SSOP/ SOIC	28-Pin QFN	44-Pin QFN/ TQFP	48-Pin UQFN			
RB9	13	18	15	1	1	13	18	15	1	1	I/O	ST	PORTB Pins
RB10	—	21	18	8	9	—	21	18	8	9	I/O	ST	PORTB Pins
RB11	—	22	19	9	10	—	22	19	9	10	I/O	ST	PORTB Pins
RB12	15	23	20	10	11	15	23	20	10	11	I/O	ST	PORTB Pins
RB13	16	24	21	11	12	16	24	21	11	12	I/O	ST	PORTB Pins
RB14	17	25	22	14	15	17	25	22	14	15	I/O	ST	PORTB Pins
RB15	18	26	23	15	16	18	26	23	15	16	I/O	ST	PORTB Pins
RC0	—	—	—	25	27	—	—	—	25	27	I/O	ST	PORTC Pins
RC1	—	—	—	26	28	—	—	—	26	28	I/O	ST	PORTC Pins
RC2	—	—	—	27	29	—	—	—	27	29	I/O	ST	PORTC Pins
RC3	—	—	—	36	39	—	—	—	36	39	I/O	ST	PORTC Pins
RC4	—	—	—	37	40	—	—	—	37	40	I/O	ST	PORTC Pins
RC5	—	—	—	38	41	—	—	—	38	41	I/O	ST	PORTC Pins
RC6	—	—	—	2	2	—	—	—	2	2	I/O	ST	PORTC Pins
RC7	—	—	—	3	3	—	—	—	3	3	I/O	ST	PORTC Pins
RC8	—	—	—	4	4	—	—	—	4	4	I/O	ST	PORTC Pins
RC9	—	—	—	5	5	—	—	—	5	5	I/O	ST	PORTC Pins
REFO	18	26	23	15	16	18	26	23	15	16	O	—	Reference Clock Output
RTCC	—	25	22	14	15	—	25	22	14	15	O	—	Real-Time Clock/Calendar Output
SCK1	15	22	19	9	10	15	22	19	9	10	I/O	ST	MSSP1 SPI Clock
SDI1	17	21	18	8	9	17	21	18	8	9	I	ST	MSSP1 SPI Data Input
SDO1	16	24	21	11	12	16	24	21	11	12	O	—	MSSP1 SPI Data Output
$\overline{SS1}$	18	26	23	15	16	18	26	23	15	16	I	ST	MSSP1 SPI Slave Select Input
SCK2	—	14	11	38	41	—	14	11	38	41	I/O	ST	MSSP2 SPI Clock
SDI2	—	19	16	36	39	—	19	16	36	39	I	ST	MSSP2 SPI Data Input
SDO2	—	15	12	37	40	—	15	12	37	40	O	—	MSSP2 SPI Data Output
$\overline{SS2}$	—	23	20	35	38	—	23	20	35	38	I	ST	MSSP2 SPI Slave Select Input

Legend: ANA = Analog level input/output, ST = Schmitt Trigger input buffer, I^2C™ = I^2C/SMBus input buffer

NOTES:

TABLE 4-12: SCCP5 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
CCP5CON1L(1)	1D0h	CCPON	—	CCPSIDL	r	TMRSYNC	CLKSEL2	CLKSEL1	CLKSEL0	TMRPS1	TMRPS0	T32	CCSEL	MOD3	MOD2	MOD1	MOD0	0000
CCP5CON1H(1)	1D2h	OPSSRC	RTRGEN	—	—	IOPS3	IOPS2	IOPS1	IOPS0	TRIGEN	ONESHOT	ALTSYNC	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0	0000
CCP5CON2L(1)	1D4h	PWMRSEN	ASDGM	—	SSDG	—	—	—	—	ASDG7	ASDG6	ASDG5	ASDG4	ASDG3	ASDG2	ASDG1	ASDG0	0000
CCP5CON2H(1)	1D6h	OENSYNC	—	—	—	—	—	—	OCAEN	ICGSM1	ICGSM0	—	AUXOUT1	AUXOUT0	ICSEL2	ICSEL1	ICSEL0	0100
CCP5CON3H(1)	1DAh	OETRIG	OSCNT2	OSCNT1	OSCNT0	—	—	—	—	—	—	POLACE	—	PSSACE1	PSSACE0	—	—	0000
CCP5STATL(1)	1DCh	—	—	—	—	—	—	—	—	CCPTRIG	TRSET	TRCLR	ASEVT	SCEVT	ICDIS	ICOV	ICBNE	0000
CCP5TMRL(1)	1E0h	SCCP5 Time Base Register Low Word																0000
CCP5TMRH(1)	1E2h	SCCP5 Time Base Register High Word																0000
CCP5PRL(1)	1E4h	SCCP5 Time Base Period Register Low Word																FFFF
CCP5PRH(1)	1E6h	SCCP5 Time Base Period Register High Word																FFFF
CCP5RAL(1)	1E8h	Output Compare 5 Data Word A																0000
CCP5RBL(1)	1ECh	Output Compare 5 Data Word B																0000
CCP5BUFL(1)	1F0h	Input Capture 5 Data Buffer Low Word																0000
CCP5BUFH(1)	1F2h	Input Capture 5 Data Buffer High Word																0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved.
Note 1: These registers are available only on PIC24F(V)16KM2XX devices.

TABLE 4-17: OP AMP 1 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
AMP1CON(1)	24Ah	AMPEN	—	AMPSIDL	AMPSLP	—	—	—	—	SPDSEL	—	NINSEL2	NINSEL1	NINSEL0	PINSEL2	PINSEL1	PINSEL0	0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved.
Note 1: This registers are available only on PIC24F(V)16KM2XX devices.

TABLE 4-18: OP AMP 2 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
AMP2CON(1)	24Ch	AMPEN	—	AMPSIDL	AMPSLP	—	—	—	—	SPDSEL	—	NINSEL2	NINSEL1	NINSEL0	PINSEL2	PINSEL1	PINSEL0	0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved.
Note 1: This registers are available only on PIC24F(V)16KM2XX devices.

TABLE 4-19: DAC1 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
DAC1CON(1)	274h	DACEN	—	DACSIDL	DACSLP	DACFM	—	SRDIS	DACTRIG	DACOE	DACTSEL4	DACTSEL3	DACTSEL2	DACTSEL1	DACTSEL0	DACREF1	DACREF0	0000
DAC1DAT(1)	276h	DACDAT15(2)	DACDAT14(2)	DACDAT13(2)	DACDAT12(2)	DACDAT11(2)	DACDAT10(2)	DACDAT9(2)	DACDAT8(2)	DACDAT7(2)	DACDAT6(2)	DACDAT5(2)	DACDAT4(2)	DACDAT3(2)	DACDAT2(2)	DACDAT1(2)	DACDAT0(2)	0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved.
Note 1: These registers are available only on PIC24F(V)16KM1XX devices.
2: The 8-bit result format depends on the value of the DACFM control bit.

TABLE 4-20: DAC2 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
DAC2CON(1)	278h	DACEN	—	DACSIDL	DACSLP	DACFM	—	SRDIS	DACTRIG	DACOE	DACTSEL4	DACTSEL3	DACTSEL2	DACTSEL1	DACTSEL0	DACREF1	DACREF0	0000
DAC2DAT(1)	27Ah	DACDAT15(2)	DACDAT14(2)	DACDAT13(2)	DACDAT12(2)	DACDAT11(2)	DACDAT10(2)	DACDAT9(2)	DACDAT8(2)	DACDAT7(2)	DACDAT6(2)	DACDAT5(2)	DACDAT4(2)	DACDAT3(2)	DACDAT2(2)	DACDAT1(2)	DACDAT0(2)	0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved.
Note 1: These registers are available only on PIC24F(V)16KM2XX devices.
2: The 8-bit result format depends on the value of the DACFM control bit.

TABLE 4-24: PAD CONFIGURATION 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
PADCFG1	2FCh	—	—	—	—	SDO2DIS(1)	SCK2DIS(1)	SDO1DIS	SCK1DIS	—	—	—	—	—	—	—	—	0000

Legend: x = unknown, u = unchanged, — = unimplemented, q = value depends on condition, r = reserved.

Note 1: These bits are not available on the PIC24F(V)08KM101 device, read as '0'.

6.4.1.1 Data EEPROM Bulk Erase

To erase the entire data EEPROM (bulk erase), the address registers do not need to be configured because this operation affects the entire data EEPROM. The following sequence helps in performing a bulk erase:

1. Configure NVMCON to Bulk Erase mode.
2. Clear the NVMIF status bit and enable the NVM interrupt (optional).
3. Write the key sequence to NVMKEY.
4. Set the WR bit to begin the erase cycle.
5. Either poll the WR bit or wait for the NVM interrupt (NVMIF is set).

A typical bulk erase sequence is provided in Example 6-3.

6.4.2 SINGLE-WORD WRITE

To write a single word in the data EEPROM, the following sequence must be followed:

1. Erase one data EEPROM word (as mentioned in the previous section) if the PGMONLY bit (NVMCON<12>) is set to '1'.
2. Write the data word into the data EEPROM latch.
3. Program the data word into the EEPROM:
   - Configure the NVMCON register to program one EEPROM word (NVMCON<5:0> = `0001xx`).
   - Clear the NVMIF status bit and enable the NVM interrupt (optional).
   - Write the key sequence to NVMKEY.
   - Set the WR bit to begin the erase cycle.
   - Either poll the WR bit or wait for the NVM interrupt (NVMIF is set).
   - To get cleared, wait until NVMIF is set.

A typical single-word write sequence is provided in Example 6-4.

EXAMPLE 6-3: DATA EEPROM BULK ERASE

```
    // Set up NVMCON to bulk erase the data EEPROM
    NVMCON = 0x4050;

    // Disable Interrupts For 5 Instructions
    asm volatile ("disi #5");

    // Issue Unlock Sequence and Start Erase Cycle
    __builtin_write_NVM();
```

EXAMPLE 6-4: SINGLE-WORD WRITE TO DATA EEPROM

```
int __attribute__ ((space(eedata))) eeData = 0x1234;
   int newData;                                      // New data to write to EEPROM
/*----------------------------------------------------------------------------------------------
The variable eeData must be a Global variable declared outside of any method

the code following this comment can be written inside the method that will execute the write
-----------------------------------------------------------------------------------------------
*/
   unsigned int offset;

   // Set up NVMCON to erase one word of data EEPROM
   NVMCON = 0x4004;

   // Set up a pointer to the EEPROM location to be erased
   TBLPAG = __builtin_tblpage(&eeData);              // Initialize EE Data page pointer
   offset = __builtin_tbloffset(&eeData);            // Initizlize lower word of address
   __builtin_tblwtl(offset, newData);                // Write EEPROM data to write latch

   asm volatile ("disi #5");                         // Disable Interrupts For 5 Instructions
   __builtin_write_NVM();                            // Issue Unlock Sequence & Start Write Cycle
   while(NVMCONbits.WR=1);                           // Optional: Poll WR bit to wait for
                                                     // write sequence to complete
```

10.0 POWER-SAVING FEATURES

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *"PIC24F Family Reference Manual"*, **"Power-Saving Features with VBAT"** (DS30622).
>
> This FRM describes some features which are not implemented in this device. Sections related to the VBAT pin and Deep Sleep do not apply to the PIC24FV16KM204 family.

The PIC24FV16KM204 family of devices provides the ability to manage power consumption by selectively managing clocking to the CPU and the peripherals. In general, a lower clock frequency and a reduction in the number of circuits being clocked constitutes lower consumed power. All PIC24F devices manage power consumption in four different ways:

- Clock Frequency
- Instruction-Based Sleep and Idle modes
- Software Controlled Doze mode
- Selective Peripheral Control in Software

Combinations of these methods can be used to selectively tailor an application's power consumption, while still maintaining critical application features, such as timing-sensitive communications.

10.1 Clock Frequency and Clock Switching

PIC24F devices allow for a wide range of clock frequencies to be selected under application control. If the system clock configuration is not locked, users can choose low-power or high-precision oscillators by simply changing the NOSCx bits. The process of changing a system clock during operation, as well as limitations to the process, are discussed in more detail in **Section 9.0 "Oscillator Configuration"**.

10.2 Instruction-Based Power-Saving Modes

PIC24F devices have two special power-saving modes that are entered through the execution of a special PWRSAV instruction. Sleep mode stops clock operation and halts all code execution; Idle mode halts the CPU and code execution, but allows peripheral modules to continue operation.

The 'C' syntax of the PWRSAV instruction is shown in Example 10-1.

> **Note:** SLEEP_MODE and IDLE_MODE are constants defined in the assembler include file for the selected device.

Sleep and Idle modes can be exited as a result of an enabled interrupt, WDT time-out or a device Reset. When the device exits these modes, it is said to "wake-up".

10.2.1 SLEEP MODE

Sleep mode includes these features:

- The system clock source is shut down. If an on-chip oscillator is used, it is turned off.
- The device current consumption will be reduced to a minimum provided that no I/O pin is sourcing current.
- The I/O pin directions and states are frozen.
- The Fail-Safe Clock Monitor does not operate during Sleep mode since the system clock source is disabled.
- The LPRC clock will continue to run in Sleep mode if the WDT or RTCC with LPRC as the clock source is enabled.
- The WDT, if enabled, is automatically cleared prior to entering Sleep mode.
- Some device features or peripherals may continue to operate in Sleep mode. This includes items, such as the Input Change Notification on the I/O ports or peripherals that use an External Clock input. Any peripheral that requires the system clock source for its operation will be disabled in Sleep mode.

The device will wake-up from Sleep mode on any of these events:

- On any interrupt source that is individually enabled
- On any form of device Reset
- On a WDT time-out

On wake-up from Sleep, the processor will restart with the same clock source that was active when Sleep mode was entered.

EXAMPLE 10-1: 'C' POWER-SAVING ENTRY

```
Sleep();                //Put the device into Sleep mode
Idle();                 //Put the device into Idle mode
```

11.0 I/O PORTS

> **Note:** This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information on the I/O ports, refer to the *"PIC24F Family Reference Manual"*, **"I/O Ports with Peripheral Pin Select (PPS)"** (DS39711). Note that the PIC24FV16KM204 family devices do not support Peripheral Pin Select features.

All of the device pins (except VDD and VSS) are shared between the peripherals and the parallel I/O ports. All I/O input ports feature Schmitt Trigger inputs for improved noise immunity.

11.1 Parallel I/O (PIO) Ports

A Parallel I/O 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 pin. The logic also prevents "loop through", in which a port's digital output can drive the input of a peripheral that shares the same pin. Figure 11-1 illustrates how ports are shared with other peripherals and the associated I/O pin to which they are connected.

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 their operation as digital I/O. The Data Direction register (TRISx) determines whether the pin is an input or an output. If the Data Direction register bit is a '1', then the pin is an input. All port pins are defined as inputs after a Reset. Reads from the Data Latch register (LATx), read the latch. Writes to the latch, write the latch. Reads from the port (PORTx), read the port pins; writes to the port pins, write the latch.

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.

FIGURE 11-1: BLOCK DIAGRAM OF A TYPICAL SHARED PORT STRUCTURE

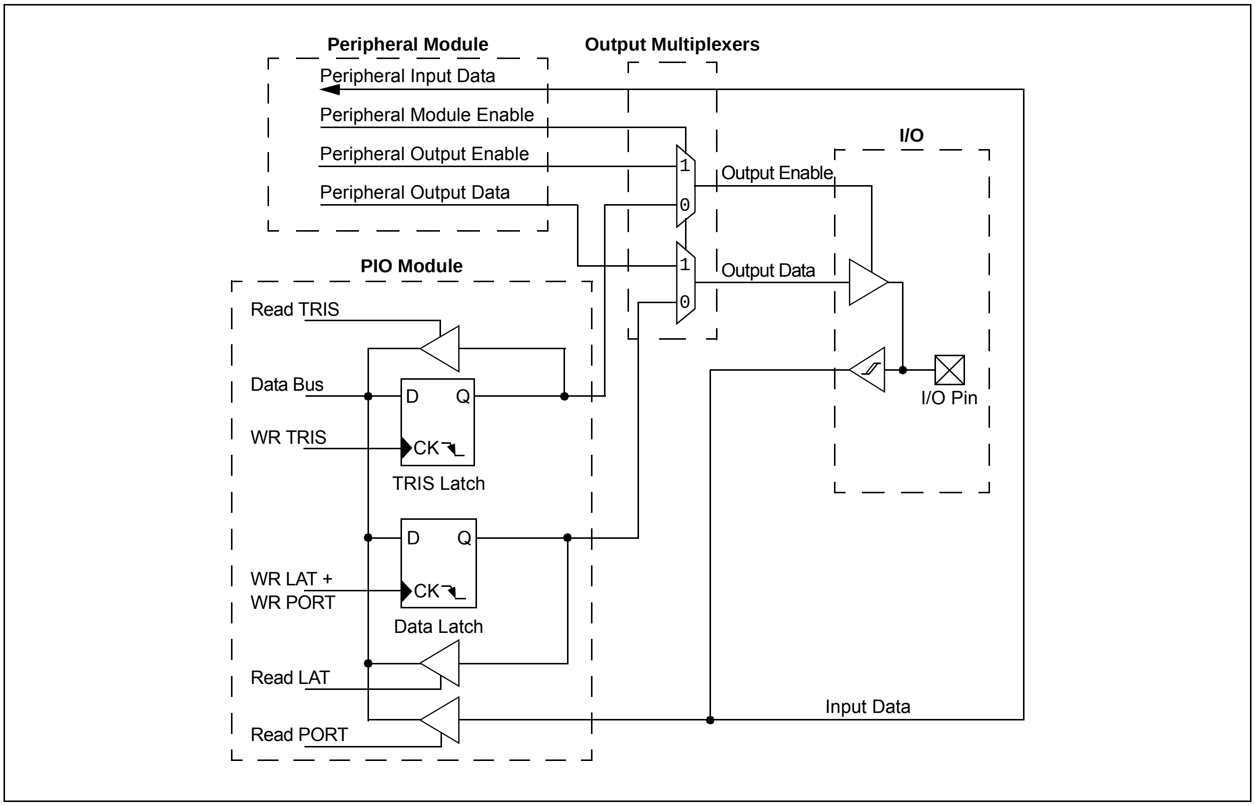
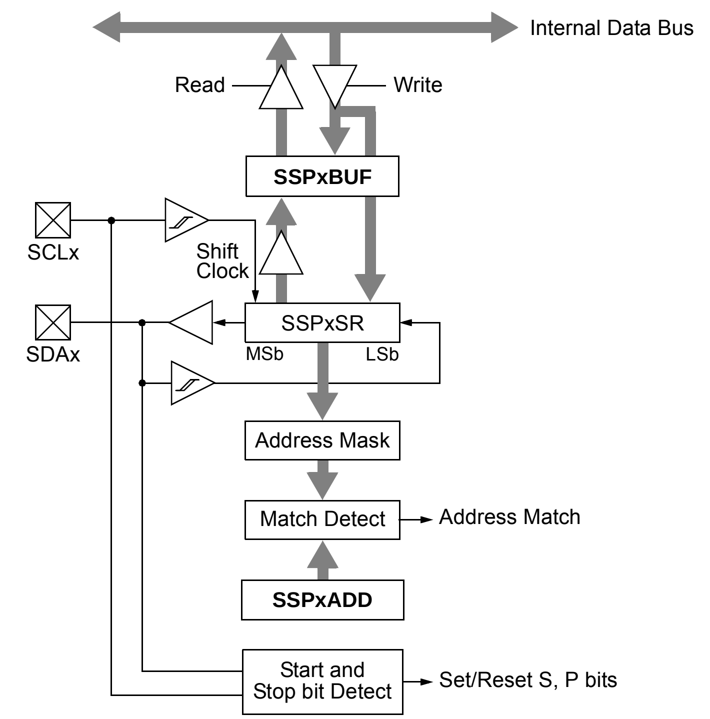
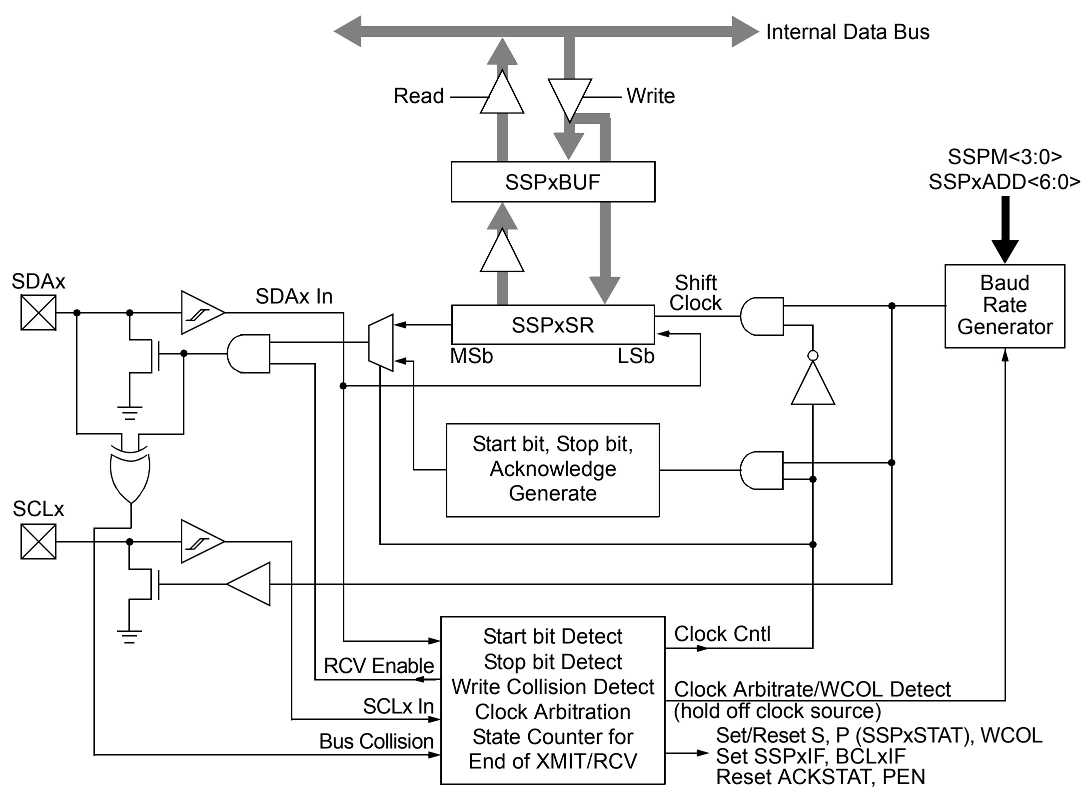


FIGURE 14-3: MSSPx BLOCK DIAGRAM (I^2C™ MODE)



Note: Only port I/O names are shown in this diagram. Refer to the text for a full list of multiplexed functions.

FIGURE 14-4: MSSPx BLOCK DIAGRAM (I^2C™ MASTER MODE)



16.2.6 ALRMVAL REGISTER MAPPINGS

REGISTER 16-8: ALMTHDY: ALARM MONTH AND DAY VALUE REGISTER[(1)]

U-0	U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	—	MTHTEN0	MTHONE3	MTHONE2	MTHONE1	MTHONE0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	DAYTEN1	DAYTEN0	DAYONE3	DAYONE2	DAYONE1	DAYONE0
bit 7							bit 0

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

bit 15-13 **Unimplemented:** Read as ‘0’

bit 12 **MTHTEN0:** Binary Coded Decimal Value of Month’s Tens Digit bit
Contains a value of ‘0’ or ‘1’.

bit 11-8 **MTHONE<3:0>:** Binary Coded Decimal Value of Month’s Ones Digit bits
Contains a value from 0 to 9.

bit 7-6 **Unimplemented:** Read as ‘0’

bit 5-4 **DAYTEN<1:0>:** Binary Coded Decimal Value of Day’s Tens Digit bits
Contains a value from 0 to 3.

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

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

REGISTER 16-9: ALWDHR: ALARM WEEKDAY AND HOURS VALUE REGISTER[(1)]

U-0	U-0	U-0	U-0	U-0	R/W-x	R/W-x	R/W-x
—	—	—	—	—	WDAY2	WDAY1	WDAY0
bit 15							bit 8

U-0	U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	—	HRTEN1	HRTEN0	HRONE3	HRONE2	HRONE1	HRONE0
bit 7							bit 0

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

bit 15-11 **Unimplemented:** Read as ‘0’

bit 10-8 **WDAY<2:0>:** Binary Coded Decimal Value of Weekday Digit bits
Contains a value from 0 to 6.

bit 7-6 **Unimplemented:** Read as ‘0’

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

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

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

REGISTER 16-10: ALMINSEC: ALARM MINUTES AND SECONDS VALUE REGISTER

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	MINTEN2	MINTEN1	MINTEN0	MINONE3	MINONE2	MINONE1	MINONE0
bit 15							bit 8

U-0	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x	R/W-x
—	SECTEN2	SECTEN1	SECTEN0	SECONE3	SECONE2	SECONE1	SECONE0
bit 7							bit 0

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

bit 15 **Unimplemented:** Read as ‘0’

bit 14-12 **MINTEN<2:0>:** Binary Coded Decimal Value of Minute’s Tens Digit bits
Contains a value from 0 to 5.

bit 11-8 **MINONE<3:0>:** Binary Coded Decimal Value of Minute’s Ones Digit bits
Contains a value from 0 to 9.

bit 7 **Unimplemented:** Read as ‘0’

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

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

16.3 Calibration

The real-time crystal input can be calibrated using the periodic auto-adjust feature. When properly calibrated, the RTCC can provide an error of less than 3 seconds per month. This is accomplished by finding the number of error clock pulses and storing the value into the lower half of the RCFGCAL register. The 8-bit signed value, loaded into the lower half of RCFGCAL, is multiplied by four and will be either added or subtracted from the RTCC timer, once every minute. Refer to the steps below for RTCC calibration:

1. Using another timer resource on the device, the user must find the error of the 32.768 kHz crystal.
2. Once the error is known, it must be converted to the number of error clock pulses per minute.
3. a) If the oscillator is faster than ideal (negative result from Step 2), the RCFGCAL register value must be negative. This causes the specified number of clock pulses to be subtracted from the timer counter, once every minute.

   b) If the oscillator is slower than ideal (positive result from Step 2), the RCFGCAL register value must be positive. This causes the specified number of clock pulses to be subtracted from the timer counter, once every minute.

EQUATION 16-1:

(Ideal Frequency† – Measured Frequency) * 60 = Clocks per Minute

† Ideal Frequency = 32,768 Hz

Writes to the lower half of the RCFGCAL register should only occur when the timer is turned off, or immediately after the rising edge of the seconds pulse, except when SECONDS = 00, 15, 30 or 45. This is due to the auto-adjust of the RTCC at 15 second intervals.

Note: It is up to the user to include, in the error value, the initial error of the crystal: drift due to temperature and drift due to crystal aging.

16.4 Alarm

- Configurable from half second to one year
- Enabled using the ALRMEN bit (ALCFGRPT<15>)
- One-time alarm and repeat alarm options are available

16.4.1 CONFIGURING THE ALARM

The alarm feature is enabled using the ALRMEN bit. This bit is cleared when an alarm is issued. Writes to ALRMVAL should only take place when ALRMEN = `0`.

As shown in Figure 16-2, the interval selection of the alarm is configured through the AMASKx bits (ALCFGRPT<13:10>). These bits determine which and how many digits of the alarm must match the clock value for the alarm to occur.

The alarm can also be configured to repeat based on a preconfigured interval. The amount of times this occurs, once the alarm is enabled, is stored in the ARPT<7:0> bits (ALCFGRPT<7:0>). When the value of the ARPTx bits equals 00h and the CHIME bit (ALCFGRPT<14>) is cleared, the repeat function is disabled, and only a single alarm will occur. The alarm can be repeated up to 255 times by loading ARPT<7:0> with FFh.

After each alarm is issued, the value of the ARPTx bits is decremented by one. Once the value has reached 00h, the alarm will be issued one last time, after which, the ALRMEN bit will be cleared automatically and the alarm will turn off.

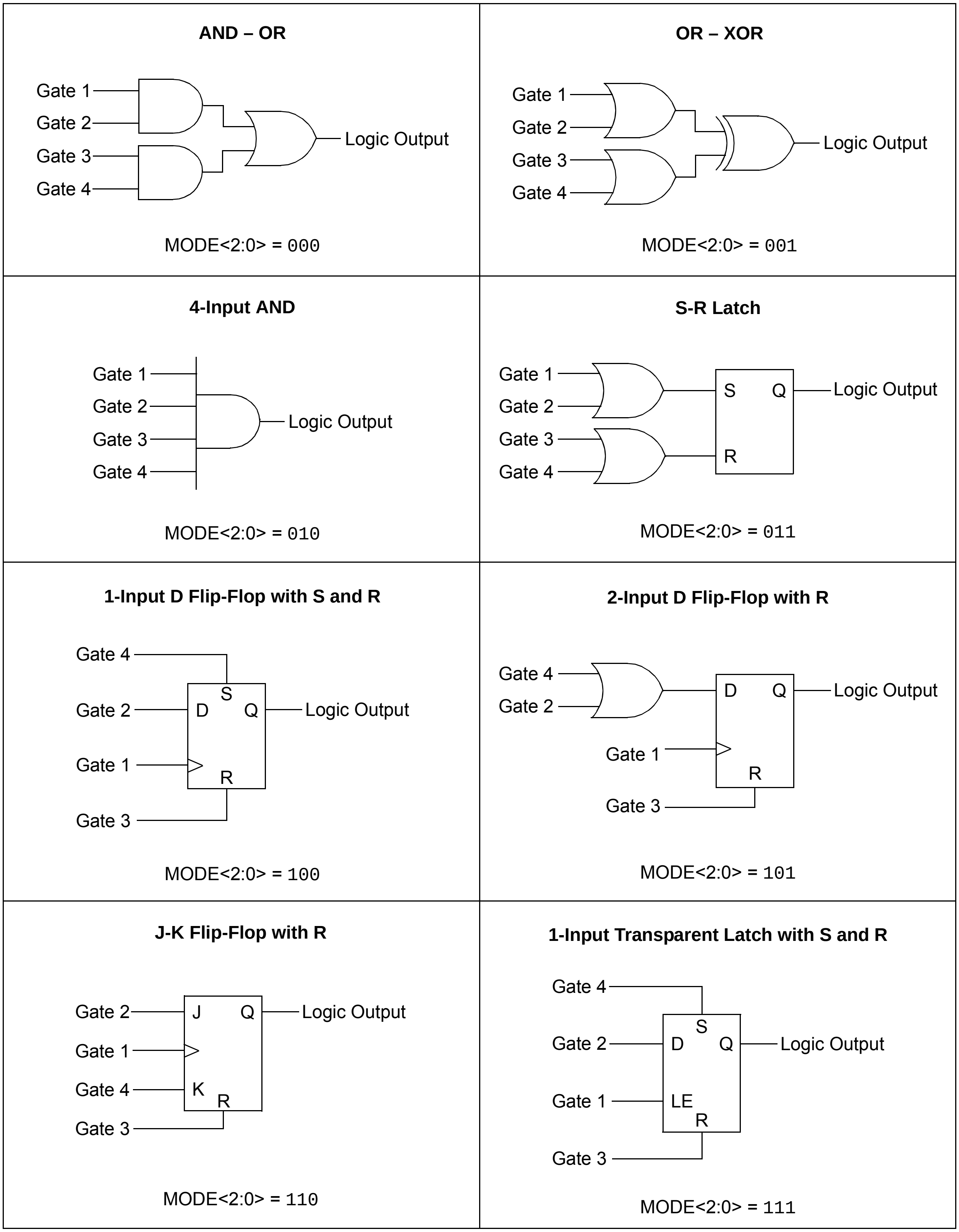
Indefinite repetition of the alarm can occur if the CHIME bit = `1`. Instead of the alarm being disabled when the value of the ARPTx bits reaches 00h, it rolls over to FFh and continues counting indefinitely while CHIME is set.

16.4.2 ALARM INTERRUPT

At every alarm event, an interrupt is generated. In addition, an alarm pulse output is provided that operates at half the frequency of the alarm. This output is completely synchronous to the RTCC clock and can be used as a Trigger clock to other peripherals.

Note: Changing any of the registers, other than the RCFGCAL and ALCFGRPT registers, and the CHIME bit while the alarm is enabled (ALRMEN = `1`), can result in a false alarm event leading to a false alarm interrupt. To avoid a false alarm event, the timer and alarm values should only be changed while the alarm is disabled (ALRMEN = `0`). It is recommended that the ALCFGRPT register and CHIME bit be changed when RTCSYNC = `0`.

FIGURE 17-2: CLCx LOGIC FUNCTION COMBINATORIAL OPTIONS



20.0 8-BIT DIGITAL-TO-ANALOG CONVERTER (DAC)

Note:	This data sheet summarizes the features of this group of PIC24F devices. It is not intended to be a comprehensive reference source. For more information, refer to the *“PIC24F Family Reference Manual”*. Device-specific information in this data sheet supersedes the information in the *“PIC24F Family Reference Manual”*.

PIC24FV16KM204 family devices include two 8-bit Digital-to-Analog Converters (DACs) for generating analog outputs from digital data. A simplified block diagram for a single DAC is shown in Figure 20-1. Both of the DACs are identical.

The DAC generates an analog output voltage based on the digital input code, according to the formula:

$$V_{DAC} = \frac{V_{DACREF} \times DACxDAT}{256}$$

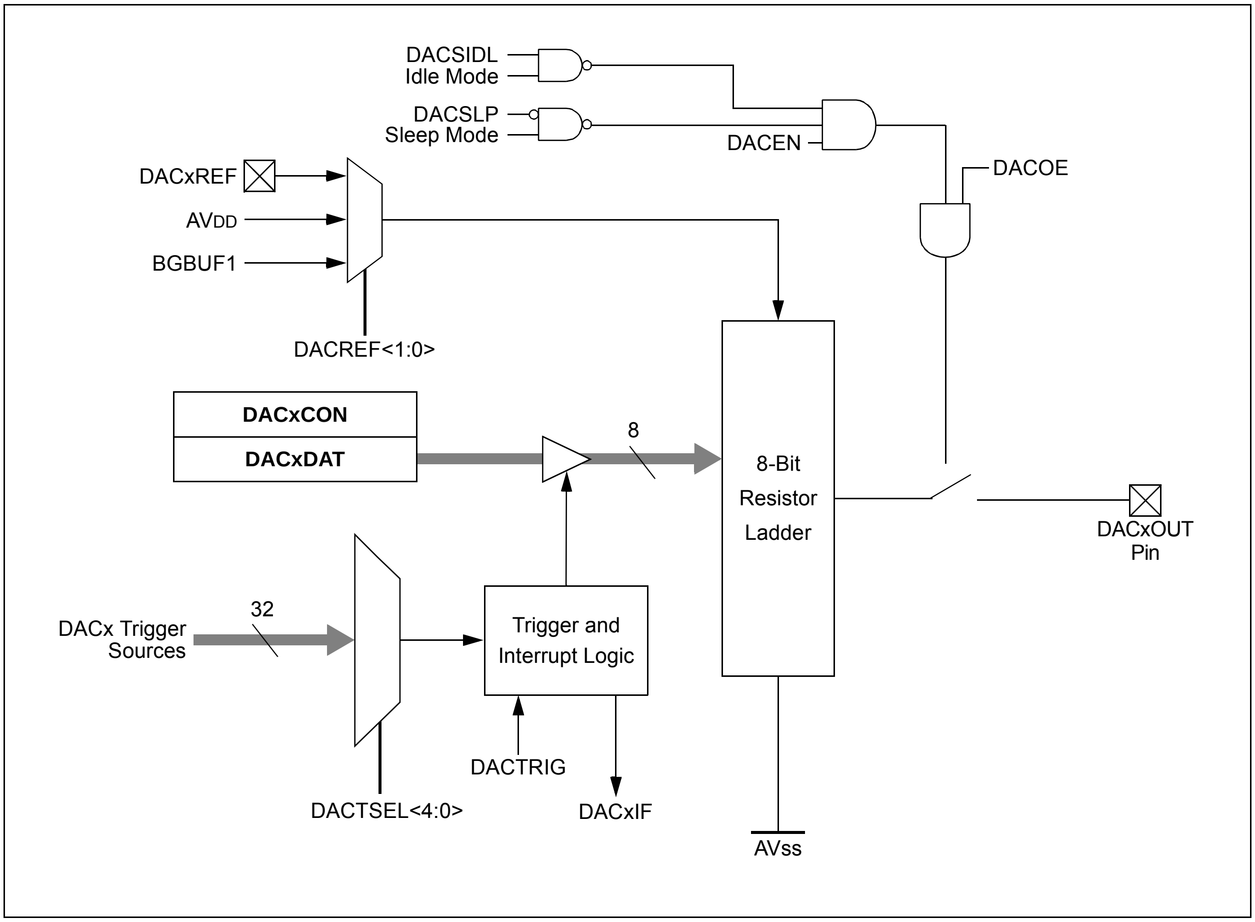
where V_{DAC} is the analog output voltage and V_{DACREF} is the reference voltage selected by DACREF<1:0>.

Each DAC includes these features:

- Precision 8-bit resistor ladder for high accuracy
- Fast settling time, supporting 1 Msps effective sampling rates
- Buffered output voltage
- Three user-selectable voltage reference options
- Multiple conversion Trigger options, plus a manual convert-on-write option
- Left and right justified input data options
- User-selectable Sleep and Idle mode operation

When using the DAC, it is recommended to set the ANSx and TRISx bits for the DACx output pin to configure it as an analog output. See **Section 11.2 “Configuring Analog Port Pins”** for more information.

FIGURE 20-1: SINGLE DACx SIMPLIFIED BLOCK DIAGRAM



REGISTER 21-1: AMPxCON: OP AMP x CONTROL REGISTER(1)

R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	U-0	U-0
AMPEN	—	AMPSIDL	AMPSLP	—	—	—	—
bit 15							bit 8

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPDSEL	—	NINSEL2	NINSEL1	NINSEL0	PINSEL2	PINSEL1	PINSEL0
bit 7							bit 0

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

bit 15 **AMPEN:** Op Amp x Control Module Enable bit
1 = Module is enabled
0 = Module is disabled

bit 14 **Unimplemented:** Read as ‘0’

bit 13 **AMPSIDL:** Op Amp x Peripheral Stop in Idle Mode bit
1 = Discontinues module operation when device enters Idle mode
0 = Continues module operation in Idle mode

bit 12 **AMPSLP:** Op Amp x Peripheral Enabled in Sleep Mode bit
1 = Continues module operation when device enters Sleep mode
0 = Discontinues module operation in Sleep mode

bit 11-8 **Unimplemented:** Read as ‘0’

bit 7 **SPDSEL:** Op Amp x Power/Speed Select bit
1 = Higher power and bandwidth (faster response time)
0 = Lower power and bandwidth (slower response time)

bit 6 **Unimplemented:** Read as ‘0’

bit 5-3 **NINSEL<2:0>:** Negative Op Amp Input Select bits
111 = Reserved; do not use
110 = Reserved; do not use
101 = Op amp negative input is connected to the op amp output (voltage follower)
100 = Reserved; do not use
011 = Reserved; do not use
010 = Op amp negative input is connected to the OAxIND pin
001 = Op amp negative input is connected to the OAxINB pin
000 = Op amp negative input is connected to AVSS

bit 2-0 **PINSEL<2:0>:** Positive Op Amp Input Select bits
111 = Op amp positive input is connected to the output of the A/D input multiplexer
110 = Reserved; do not use
101 = Op amp positive input is connected to the DAC1 output for OA1 (DAC2 output for OA2)
100 = Reserved; do not use
011 = Reserved; do not use
010 = Op amp positive input is connected to the OAxINC pin
001 = Op amp positive input is connected to the OAxINA pin
000 = Op amp positive input is connected to AVSS

Note 1: This register is available only on PIC24F(V)16KM2XX devices.

26.2 MPLAB XC Compilers

The MPLAB XC Compilers are complete ANSI C compilers for all of Microchip's 8, 16 and 32-bit MCU and DSC devices. These compilers provide powerful integration capabilities, superior code optimization and ease of use. MPLAB XC Compilers run on Windows, Linux or MAC OS X.

For easy source level debugging, the compilers provide debug information that is optimized to the MPLAB X IDE.

The free MPLAB XC Compiler editions support all devices and commands, with no time or memory restrictions, and offer sufficient code optimization for most applications.

MPLAB XC Compilers include an assembler, linker and utilities. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. MPLAB XC Compiler uses the assembler to produce its object file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

26.3 MPASM Assembler

The MPASM Assembler is a full-featured, universal macro assembler for PIC10/12/16/18 MCUs.

The MPASM Assembler generates relocatable object files for the MPLINK Object Linker, Intel® standard HEX files, MAP files to detail memory usage and symbol reference, absolute LST files that contain source lines and generated machine code, and COFF files for debugging.

The MPASM Assembler features include:

- Integration into MPLAB X IDE projects
- User-defined macros to streamline assembly code
- Conditional assembly for multipurpose source files
- Directives that allow complete control over the assembly process

26.4 MPLINK Object Linker/ MPLIB Object Librarian

The MPLINK Object Linker combines relocatable objects created by the MPASM Assembler. It can link relocatable objects from precompiled libraries, using directives from a linker script.

The MPLIB Object Librarian manages the creation and modification of library files of precompiled code. When a routine from a library is called from a source file, only the modules that contain that routine will be linked in with the application. This allows large libraries to be used efficiently in many different applications.

The object linker/library features include:

- Efficient linking of single libraries instead of many smaller files
- Enhanced code maintainability by grouping related modules together
- Flexible creation of libraries with easy module listing, replacement, deletion and extraction

26.5 MPLAB Assembler, Linker and Librarian for Various Device Families

MPLAB Assembler produces relocatable machine code from symbolic assembly language for PIC24, PIC32 and dsPIC DSC devices. MPLAB XC Compiler uses the assembler to produce its object file. The assembler generates relocatable object files that can then be archived or linked with other relocatable object files and archives to create an executable file. Notable features of the assembler include:

- Support for the entire device instruction set
- Support for fixed-point and floating-point data
- Command-line interface
- Rich directive set
- Flexible macro language
- MPLAB X IDE compatibility

NOTES: