

**mmike**

Joined: 04 Jun 2006

Posts: 553

**Helped: 19**

▣ 12 Aug 2006 11:11

## Motorola chip pic16f88 18 pin

---

The Intel 8051 was a Harvard architecture single chip microcontroller (IC) developed by Intel in 1980 for use in embedded systems. It was extremely popular in the 1980s and early 1990s, but today it has largely been superseded by a vast range of enhanced devices with 8051-compatible processor cores that are manufactured by more than 20 independent manufacturers including Atmel, Maxim IC (via its Dallas Semiconductor subsidiary), Philips, Winbond, and Silicon Laboratories. Intel's official designation for the 8051 family of ICs is MCS 51.

Intel's original 8051 family was developed using NMOS technology, but later versions, identified by a letter "C" in their name, e.g. 80C51, used CMOS technology and were less power-hungry than their NMOS predecessors - this made them eminently more suitable for battery-powered devices.

Important Features of 8051 :

- \* It contains Processor (CPU), RAM, ROM, Serial Port , Parallel Port, Interrupt logic, Timer etc.
- \* Data bus - 8 bit data bus. Can access 8 bit data in one operation. Hence called 8-bit microprocessor.
- \* Address bus - 16 bit address bus. Can access 2<sup>16</sup> memory locations i.e 64 KB of memory each of RAM and ROM.
- \* On chip RAM - 128 Bytes (Data Memory).
- \* On chip ROM - 4 KB (Program Memory).
- \* Four Byte bi-directional Input Output port.
- \* UART (Serial Port).
- \* Two 16 - bit Up-Counter.
- \* Two level interrupt priority.
- \* Power saving mode.

A particularly useful feature of the 8051 core is the inclusion of a boolean processing engine which allows bit-level boolean logic operations to be carried out directly and efficiently on internal registers and RAM. This feature helped to cement the 8051's popularity in industrial control applications. Another valued feature is that it has four separate register sets, which can be used to greatly reduce interrupt latency compared to the more common method of storing interrupt context on a stack.

The 8051 UART can be configured to use a 9th data bit that can provide addressable

communications in an RS-485 multi-point communications environment.

8051 based microcontrollers typically include one or two UARTs, two or three timers, 128 or 256 bytes of internal data RAM (16 bytes of which are bit-addressable), up to 128 bytes of I/O, 512 bytes to 128kb of internal program memory, and sometimes a quantity of extended data RAM (ERAM) located in the program address space. The original 8051 core ran at 12 clock cycles per machine cycle, with most instructions executing in one or two machine cycles. With a 12 MHz clock frequency, the 8051 could thus execute 1 million one-cycle instructions per second or 500,000 two-cycle instructions per second. Enhanced 8051 cores are now commonly used which run at six, four, two, or even one clock per machine cycle, and have clock frequencies of up to 100 MHz, and are thus capable of an even greater number of instructions per second. All SILabs, some Dallas and a few Atmel devices have single cycle cores.

Even higher speed single cycle 8051 cores, in the range 130 MHz to 150 MHz, are now available in internet downloadable form for use in programmable logic devices such as FPGAs, and at many hundreds of MHz in ASICs, for example the netlist from e8051.com.

Common features included in modern 8051 based microcontrollers include built-in reset timers with brown-out detection, on-chip oscillators, self-programmable Flash ROM program memory, bootloader code in ROM, EEPROM non-volatile data storage, I2C, SPI, and USB host interfaces, PWM generators, analog comparators, A/D and D/A converters, RTCs, extra counters and timers, in-circuit debugging facilities, more interrupt sources, and extra power saving modes.

Several C compilers are available for the 8051, most of which feature extensions that allow the programmer to specify where each variable should be stored in its six types of memory, and provide access to 8051 specific hardware features such as the multiple register banks and bit manipulation instructions. Other high level languages such as Forth, BASIC, PASCAL, PL/M and Modula 2 are available for the 8051, but they are less widely used than C and assembly.

The 8051's predecessor, the 8048, was used in the keyboard of the first IBM PC, where it converted keypresses into the serial data stream which is sent to the main unit of the computer. The 8048 and derivatives are still used today for basic model keyboards.

The 8031 was a cut down version of the original Intel 8051 that did not contain any internal program memory.

The 8052 was an enhanced version of the original Intel 8051 that featured 256 bytes of internal RAM instead of 128 bytes, 8 kB of ROM instead of 4 kB, and a third 16-bit timer. The 8032 had these same features except for the internal ROM program memory. The 8052 and 8032 are largely considered to be obsolete because these features and more are included in nearly all modern 8051 based microcontrollers.

## Code:

PIC is a family of RISC microcontrollers made by Microchip Technology, derived from the PIC1650 originally developed by General Instrument's Microelectronics Division.

Microchip Technology does not use PIC as an acronym; in fact the brand name is PICmicro. It is generally regarded that PIC stands for Peripheral Interface Controller, although General Instruments' original acronym for the PIC1650 was "Programmable Intelligent Computer". The original PIC was built to be used with GI's new 16-bit CPU, the CP1600. While generally a good CPU, the CP1600 had poor I/O performance, and the 8-bit PIC was developed in 1975 to improve performance of the overall system by offloading I/O tasks from the CPU. The PIC used simple microcode stored in ROM to perform its tasks, and although the term wasn't used at the time, it is a RISC design that runs one instruction per cycle (4 oscillator cycles).

In 1985 General Instruments spun off their microelectronics division, and the new ownership cancelled almost everything ♦ which by this time was mostly out-of-date. The PIC, however, was upgraded with EPROM to produce a programmable channel controller, and today a huge variety of PICs are available with various on-board peripherals (serial communication modules, UARTs, motor control kernels, etc.) and program memory from 512 words to 32k words and more (a "word" is one assembly language instruction, varying from 12, 14 or 16 bits depending on the specific PICmicro family).

Various PIC microcontrollers

Enlarge

Various PIC microcontrollers

Contents

[hide]

- \* 1 Coding for PICs
- \* 2 Programming PICs
  - o 2.1 Development tools
    - + 2.1.1 Simple serial port programmers
    - + 2.1.2 Simple parallel port programmers
    - + 2.1.3 Intelligent programmers
- \* 3 Debugging
- \* 4 Word size
- \* 5 Modern PICs
- \* 6 Features
- \* 7 PICs on the Internet
- \* 8 PIC clones
- \* 9 Wireless PICs
- \* 10 dsPICs (Digital Signal PICs)
- \* 11 8/16-bit PIC microcontroller product families
- \* 12 Commonly used PICs
- \* 13 See also
- \* 14 External links
  - o 14.1 Linux PIC Programmer Software
  - o 14.2 Forums and discussion groups

### o 14.3 Other links

[edit]

#### Coding for PICs

PICs use a RISC instruction set, which varies in length from about 35 instructions for the low-end PICs to about 70 instructions for the high-end PICs. The instruction set includes instructions to perform a variety of operations on the accumulator and a constant or the accumulator and a memory location, as well as for conditionally executing code and jumping/calling other parts of the program and returning from them, and specific hardware features like interrupts and one low-power mode called sleep. Microchip provides a freeware IDE package called MPLAB, that also includes a software simulator as well as an assembler.

Third parties make C and BASIC language compilers for PICs. Microchip also sells compilers for the high-end PICs ("C18" for the 18F series and "C30" for the dsPICs). They also make available for download a "student edition/demo" version of C18 or C30 which disables some optimiser features after a timeout period. Open-source compilers for the C, Pascal, JAL, and for the Forth programming language, PicForth, have also been released.

GPUTILS is an Open Source collection of tools, distributed under the GNU General Public License. GPUTILS includes an assembler and linker and works on Linux, Mac OS X, OS/2 and Microsoft Windows. GPSIM is an Open Source simulator for the PIC microcontrollers featuring hardware modules that simulate specific devices that might be connected to them, like LCDs.

[edit]

#### Programming PICs

A development board for low pin-count MCU, from Microchip

Enlarge

A development board for low pin-count MCU, from Microchip

Devices called "programmers" are traditionally used to get program code into the target PIC. Most PICs that Microchip sells nowadays have ICSP (In Circuit Serial Programming) and/or LVP (Low Voltage Programming) capabilities, allowing the PIC to be programmed while it is sitting in the target circuit. ICSP programming is performed using the RB6 and RB7 pins for clock and data, while a high voltage (12V) is present on the Vpp/MCLR pin. Low voltage programming allows for the elimination of the extra voltage rail in the programmer but comes at the cost of an IO pin and can therefore be disabled (once disabled it can only be re-enabled using high voltage programming). There are many programmers for PIC microcontrollers, ranging from the extremely simple designs that rely on the communications software for taking care of all the communication details to complex designs that can verify the device at several supply voltages and can do much of the work in the hardware. Many of these complex programmers use a pre-programmed PIC themselves to send the programming commands to the PIC that is to be programmed.

Many of the higher end flash based PICs can also write to their own program memory. Demo

boards are available with a small bootloader factory programmed that can be used to load user programs over an interface such as RS-232 or USB.

There are many programmers/debuggers available directly from Microchip:

Microchip Programmers (as of 10/2005)

- \* PICStart Plus (RS232 serial interface)
- \* Promate II (RS232 serial interface)
- \* MPLAB PM3 (RS232 serial and USB interface)
- \* ICD2 (RS232 serial and USB interface)
- \* PICKit 1 (USB interface)
- \* PICKit 2 (USB interface)

[edit]

Development tools

Microchip home page

microEngineering Labs, Inc. The creators of PICBASIC also offer USB PIC Programmers, LAB-X Experimenter Boards, PICPROTO Boards, books and other resources.

Proton Development Suite Popular BASIC Compiler, comprehensive and well supported by its users.

KTechlab Open-source development environment for electronic circuits.

There are also programmers available from other sources, ranging from plans as to how to build your own, to kits to put together yourself, to fully tested ready-to-go units. Some are simple designs which require a PC to do the low-level programming signalling (these typically connect to the serial or parallel port and consist of a few simple components), while others have the programming logic built into them (these typically use a serial or USB connection, are usually faster, and are many times built using PICs themselves for control). These are some common programmers:

[edit]

Simple serial port programmers

These generally rely on driving the PIC's Vss line negative to get the necessary voltage differences from programming. Hence they are compact and cheap but great care is needed if using them for in circuit programming.

- \* JDM programmer (serial-port programmer)
- \* Feng's RCD programmer (serial-port programmer)

Compatible software:

- \* IC Prog A freeware programmer for MS Windows

[edit]

Simple parallel port programmers

Simple to understand but often have much higher part counts and generally require external power supplies.

- \* P16PRO40 from <http://kitsrus.com> (parallel port programmer)
- \* Tait-style programmer (parallel port programmer)
- \* MIT's Low Voltage PIC programmer (parallel port programmer)

Compatible Linux software:

- \* PICPrg A fast Linux command line (or ncurses) program, to program any Microchip PIC using the PC parallel port interface. Program/verify PIC data, erase non-protected/protected PIC data or calculate the PIC checksum.

- \* JimPIC Fast & simple program for Linux and Windows to control the MIT Low Voltage PIC programmer using the parallel port.

[edit]

Intelligent programmers

Generally faster and more reliable (especially on laptops which tend to have idiosyncrasies in the way they implement their ports) but far more complex to build (in particular they tend to use a PIC in the programmer which must itself be programmed somehow).

- \* Wisp628 (intelligent serial port in-circuit programmer built around a PIC16F628. Kit available.)

- \* EasyProg (intelligent serial port in-circuit and standalone programmer. Kit and ready-to-go version available)

An exhaustive list of these is not practical, but a query put out on the piclist ([piclist@mit.edu](mailto:piclist@mit.edu) ◆subscription required) would quickly provide many answers from the knowledgeable people who frequent that list and are happy to help.

[edit]

Debugging

MPLAB (which is a free download) includes a software emulator for PICs. However software emulation of a microcontroller will always suffer from limited simulation of the devices interactions with its target circuit.

The MPLAB ICD2 (RS-232 serial and USB interface). Offers fairly cheap (about £100) in circuit debugging. A programming/debug connector (the ICD2 has a RJ12 socket and comes with a cross wired RJ12 lead but there is nothing preventing the use of custom cables to other connector types) is designed into the target circuit. This cheap and simple debugging system comes at a price however, namely limited breakpoint count (1 on older pics 3 on newer PICs), loss of some IO (with the exception of some surface mount 44 pin PICs which have dedicated lines for debugging) and loss of some features of the chip. For small PICs where the loss of IO caused by this method would be unacceptable special headers are made which are fitted with PICs that have extra pins specifically for debugging. Some third party debugging solutions also exist using the on chip facilities provided for the ICD2.

Finally microchip offers two full in circuit emulators. The MPLAB ICE2000 (parallel interface, a USB converter is available) and the newer MPLAB ICE4000 (USB interface).

[edit]

### Word size

The word size of PICs is a source of much confusion. All PICs (except dsPICs and PIC24s) handle data in 8-bit chunks, so they should be called 8-bit microcontrollers. But unlike high-end CPUs, PICs use Harvard architecture, so the size of an instruction can be different from the size of the data. In fact, different PIC families use different instruction sizes, which makes it a challenge to compare the code size of PICs to other microcontrollers. For example, say a microcontroller has 6144 bytes of program memory. For a 12-bit PIC, this works out to 4096 words (or assembly instructions); for a 16-bit PIC, this is 3072 words.

[edit]

### Modern PICs

The old PROM and EPROM PICs are now gradually replaced by chips with Flash memory. Likewise the original 12-bit instruction set of the PIC1650 and its direct descendants has been superseded by 14-bit and 16-bit instruction sets. Microchip still sells OTP (one-time-programmable, or PROM) and UV-erasable (EPROM) versions of most of its PICs for legacy support, or volume orders. It should be noted that the Microchip website lists PICs that are not electrically erasable as OTP despite the fact that UV erasable windowed versions of these chips can be ordered.

[edit]

### Features

Current PICs offer a wide range of built-in hardware, such as:

- \* 8/16 bit Modified Harvard Architecture CPU cores
- \* Flash and ROM Memory options in 256 byte to 256 kilobyte array sizes
- \* I/O Ports (0 to 5.5 volts Typical)
- \* 8/16 Bit Timers

- \* Nanowatt Technology for Power Moding
- \* Synchronous/Asynchronous Serial Peripherals USART, AUSART, EUSARTs
- \* Analog-to-digital converters, 10/12 bit
- \* Voltage Comparators
- \* Capture/Compare/PWM modules
- \* LCD Drivers
- \* MSSP Peripheral for I<sub>2</sub>C and SPI Communications
- \* Internal EEPROM Memory - up to 1M erase/write cycles durability
- \* Motor Control Peripherals
- \* USB interfacing support
- \* Ethernet controller support
- \* CAN controller support
- \* LIN controller support
- \* IrDA controller support
- \* Integrated analog RF front ends (PIC16F639, and rfPIC)
- \* KEELOQ Rolling code encryption peripheral (encode/decode)
- \* DSP digital signal processing capabilities (dsPIC)

[edit]

#### PICs on the Internet

PICs are well-documented on the Internet, mainly for two reasons. First, PICs were commonly used to defeat the security system of popular consumer products (pay-TV, PlayStation), which attracted the attention of crackers. Second, the now obsolete PIC16C84 was the first widely available microcontroller that could easily be re-programmed by hobbyists. There are many forums and lists concerning PIC microcontrollers where the user can post questions and receive free answers.

[edit]

#### PIC clones

Every now and then there are companies that offer cheap and/or enhanced PIC versions. Most seem to disappear quite soon. Uvicom (formerly Scenix) seems to avoid this fate and produces the SX, a PIC clone that runs much faster than the original (as of November 2005 Parallax is the exclusive supplier of the SX). OpenCores has a pic16f84 core written in Verilog.

[edit]

#### Wireless PICs

The rfPIC microcontroller devices integrate the power of Microchip's PICmicro devices with UHF wireless communication capabilities for low power RF applications. The devices offer small package outline and low external component count to fit the most space-constrained applications.

[edit]

## dsPICs (Digital Signal PICs)

dsPICs are Microchip's newest family, which entered mass production in late 2004. They are designed as a PIC with digital signal processing capabilities. These are Microchip's first inherent 16-bit (data) microcontrollers. They build on the PICs existing strengths by offering hardware MAC (multiply-accumulate), barrel shifting, bit reversal, (16x16)-bit multiplication and other digital signal processing operations.

[edit]

## 8/16-bit PIC microcontroller product families

These links take you to product selection matrices at the manufacturer's site.

## 8-bit Microcontrollers

- \* PIC10
- \* PIC12
- \* PIC14
- \* PIC16
- \* PIC17
- \* PIC18

## 16-bit Microcontrollers

- \* PIC24

## 16-bit Digital Signal Controllers

- \* dsPIC30
- \* dsPIC33F

The F in a name generally indicates the PICmicro uses flash memory and can be erased electronically. A C generally means it can only be erased by exposing the die to ultraviolet light (which is only possible if a windowed package style is used).

[edit]

## Commonly used PICs

- \* PIC12C508/509 (Small 8-pin package, internal oscillator, popular in small designs such as the iPod remote)
- \* PIC16F84 (generally considered obsolete, but still popular)
- \* PIC16F84A (upgrade to above, some versions do 20 MHz, 1:1 compatible with PIC16F84)
- \* PIC16F88 (an excellent all-round 18-pin PICmicro)
- \* PIC12F629/675/683
- \* PIC16F628 (replaces the PIC16F84. The 16F628A has more program memory and fully compatible)

- \* The PIC16F87X family (The PIC16F84's "big brothers", with lots of built in hardware similar to the 16F88. Quite common in hobby projects.)
- \* PIC16F877 (RS232, Parallel Port Interface, PWM and much more)
- \* PIC18F452
- \* PIC18F4550 and relatives (Full-speed USB support, all sorts of inbuilt hardware. Very powerful and versatile)