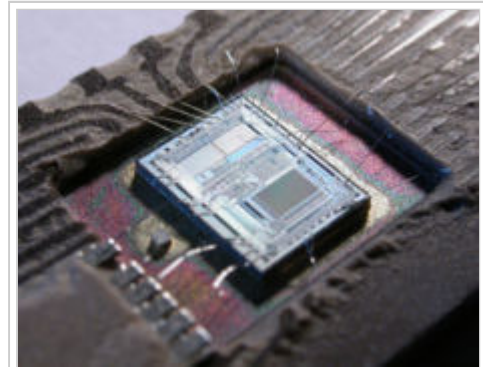


Microcontroller

From Wikipedia, the free encyclopedia

A **microcontroller** (also **MCU** or **μC**) is a computer-on-a-chip. It is a type of microprocessor emphasizing high integration, low power consumption, self-sufficiency and cost-effectiveness, in contrast to a general-purpose microprocessor (the kind used in a PC). In addition to the usual arithmetic and logic elements of a general purpose microprocessor, the microcontroller typically integrates additional elements such as read-write memory for data storage, read-only memory, such as flash for code storage, EEPROM for permanent data storage, peripheral devices, and input/output interfaces. At clock speeds of as little as a few MHz or even lower, microcontrollers often operate at very low speed compared to modern day microprocessors, but this is adequate for typical applications. They consume relatively little power (milliwatts), and will generally have the ability to sleep while waiting for an interesting peripheral event such as a button press to wake them up again to do something. Power consumption while sleeping may be just nano watts, making them ideal for low power and long lasting battery applications.



The integrated circuit from an Intel 8742, an 8-bit microcontroller that includes a CPU running at 12 MHz, 128 bytes of RAM, 2048 bytes of EPROM, and I/O in the same chip.

Microcontrollers are frequently used in automatically controlled products and devices, such as automobile engine control systems, remote controls, office machines, appliances, power tools, and toys. By reducing the size, cost, and power consumption compared to a design using a separate microprocessor, memory, and input/output devices, microcontrollers make it economical to electronically control many more processes.

Contents

- 1 Embedded design
- 2 Higher Integration
- 3 Large Volumes
- 4 Programming Environments
- 5 Interrupt Latency
- 6 History
- 7 Development platforms for hobbyists
 - 7.1 Arduino
 - 7.2 Platforms from Parallax, Inc.

- 7.3 PICAXE
- 7.4 A-WIT Technologies, Inc.
- 7.5 Comfile Technology Inc.
- 7.6 Coridium ARMexpress
- 7.7 ZX-24, ZX-40, ZX-44
- 8 See also
- 9 Notes
- 10 External links

Embedded design

The majority of computer systems in use today are embedded in other machinery, such as telephones, clocks, appliances, and vehicles. An embedded system may have minimal requirements for memory and program length. Input and output devices may be discrete switches, relays, or solenoids. An embedded controller may lack any human-readable interface devices at all. For example, embedded systems usually don't have keyboards, screens, disks, printers, or other recognizable I/O devices of a personal computer. Microcontrollers may control electric motors, relays or voltages, and may read switches, variable resistors or other electronic devices.

Higher Integration

In contrast to general-purpose CPUs, microcontrollers may not implement an external address or data bus as they integrate RAM and non-volatile memory on the same chip as the CPU. Using fewer pins, the chip can be placed in a much smaller, cheaper package.

Integrating the memory and other peripherals on a single chip and testing them as a unit increases the cost of that chip, but often results in decreased net cost of the embedded system as a whole. Even if the cost of a CPU that has integrated peripherals is slightly more than the cost of a CPU + external peripherals, having fewer chips typically allows a smaller and cheaper circuit board, and reduces the labor required to assemble and test the circuit board.

A microcontroller is a single integrated circuit, commonly with the following features:

- central processing unit - ranging from small and simple 4-bit processors to complex 32- or 64-bit processors
- discrete input and output bits, allowing control or detection of the logic state of an individual package pin
- serial input/output such as serial ports (UARTs)
- other serial communications interfaces like I²C, Serial Peripheral Interface and Controller Area Network for system interconnect
- peripherals such as timers, event counters, PWM generators, and watchdog
- volatile memory (RAM) for data storage

- ROM, EPROM, [EEPROM] or Flash memory for program and operating parameter storage
- clock generator - often an oscillator for a quartz timing crystal, resonator or RC circuit
- many include analog-to-digital converters
- in-circuit programming and debugging support

This integration drastically reduces the number of chips and the amount of wiring and PCB space that would be needed to produce equivalent systems using separate chips. Furthermore, and on low pin count devices in particular, each pin may interface to several internal peripherals, with the pin function selected by software. This allows a part to be used in a wider variety of applications than if pins had dedicated functions. Microcontrollers have proved to be highly popular in embedded systems since their introduction in the 1970s.

Some microcontrollers use a Harvard architecture: separate memory buses for instructions and data, allowing accesses to take place concurrently. Where a Harvard architecture is used, instruction words for the processor may be a different bit size than the length of internal memory and registers; for example: 12-bit instructions used with 8-bit data registers.

The decision of which peripheral to integrate is often difficult. The microcontroller vendors often trade operating frequencies and system design flexibility against time-to-market requirements from their customers and overall lower system cost. Manufacturers have to balance the need to minimize the chip size against additional functionality.

Microcontroller architectures vary widely. Some designs include general-purpose microprocessor cores, with one or more ROM, RAM, or I/O functions integrated onto the package. Other designs are purpose built for control applications. A microcontroller instruction set usually has many instructions intended for bit-wise operations to make control programs more compact. For example, a general purpose processor might require several instructions to test a bit in a register and branch if the bit is set, where a microcontroller could have a single instruction that would provide that commonly-required function.

Large Volumes

Microcontrollers take the largest share of sales in the wider microprocessor market. Over 50% are "simple" controllers, and another 20% are more specialized digital signal processors (DSPs). A typical home in a developed country is likely to have only one or two general-purpose microprocessors but somewhere between one and two dozen microcontrollers. A typical mid range automobile has as many as 50 or more microcontrollers. They can also be found in almost any electrical device: washing machines, microwave ovens, telephones etc.

Manufacturers have often produced special versions of their microcontrollers to help the hardware and software development of the target system. These include EPROM versions that have a "window" on the top of the package through which program memory can be erased by ultra violet light, read after a programming ("burn") and test cycle.

An economical option for intermediate levels of production (up to a few thousand parts) is a one-time programmable (OTP) microcontroller. It has the same die as the UV EPROM version of the part, and is programmed using the same equipment, but the package does not include the expensive quartz window required to admit UV light on to the chip.



A PIC18F8720 microcontroller in an 80-pin TQFP package.

Other versions may be available where the ROM is accessed as an external device rather than as internal memory. A simple EPROM programmer, rather than a more complex and expensive microcontroller programmer, may then be used, however there is a potential loss of functionality through pin outs being tied up with external memory addressing rather than for general input/output. These kind of devices usually carry a higher cost but if the target production quantities are small, certainly in the case of a hobbyist, they can be the most economical option compared with the set up charges involved in mask programmed devices.

A more rarely encountered development microcontroller is the "piggy back" version. This device has no internal ROM memory; instead pin outs on the top of the microcontroller form a socket into which a standard EPROM program memory device may be installed. The benefit of this approach is the release of microcontroller pins for input and output use rather than program memory. These kinds of devices are normally expensive and are impractical for anything but the development phase of a project or very small production quantities.

The use of field-programmable devices on a microcontroller may allow field update of the firmware or permit late factory revisions to products that have been assembled but not yet shipped. Programmable memory also reduces the lead time required for deployment of a new product.

Where a large number of systems will be made (say, several thousand), the cost of a mask-programmed memory is amortized over all products sold. A simpler integrated circuit process is used, and the contents of the read-only memory are set in the last step of chip manufacture instead of after assembly and test. However, mask-programmed parts cannot be updated in the field. If product firmware updates are still contemplated, a socket may be used to hold the controller which can then be replaced by a service technician, if required.

Programming Environments

Microcontrollers were originally programmed only in assembly language, but various high-level programming languages are now also in common use to target microcontrollers. These languages are either designed specially for the purpose, or versions of general purpose languages such as the C programming language. Compilers for general purpose languages will typically have some restrictions as well as enhancements to better support the unique characteristics of microcontrollers.

Interpreter firmware is also available for some microcontrollers. The Intel 8052 and Zilog Z8 were available with BASIC very early on, and BASIC is more recently used in the BASIC Stamp MCUs.

Some microcontrollers have environments to aid developing certain types of applications, e.g. Analog Device's Blackfin processors with the LabVIEW environment and its programming language "G".

Simulators are available for some microcontrollers, such as in Microchip's MPLAB environment. These allow a developer to analyse what the behaviour of the microcontroller and their program should be if they were using the actual part. A simulator will show the internal processor state and also that of the outputs, as well as allowing input signals to be generated. While on the one hand most simulators will be limited from being unable to simulate much other hardware in a system, they can exercise conditions that may otherwise be hard to reproduce at will in the physical implementation, and can be the quickest way to debug and analyse problems.

Recent microcontrollers integrated with on-chip debug circuitry accessed by In-circuit emulator via JTAG enables a programmer to debug the software of an embedded system with a debugger.

Interrupt Latency

In contrast to general-purpose computers, microcontrollers used in embedded systems often seek to minimize interrupt latency over instruction throughput.

When an electronic device causes an interrupt, the intermediate results, the registers, have to be saved before the software responsible for handling the interrupt can run, and then must be put back after it is finished. If there are more registers, this saving and restoring process takes more time, increasing the latency.

Low-latency MCUs generally have relatively few registers in their central processing units, or they have "shadow registers", a duplicate register set that is only used by the interrupt software.

History

First microcontroller was Intel 8048 released in 1976.

Development platforms for hobbyists

For almost every manufacturer of bare microcontrollers, there are various companies repacking its products into more hobbyist-friendly packages. Their product is often an MCU preloaded with a BASIC or similar interpreter, soldered onto a board with the same footprint as a Dual Inline Pin package for convenient prototyping, and possibly a few external components such as a power regulator and clock source. PICmicros seem to be very popular here, possibly due to good static protection. More powerful examples (e.g. faster execution, more RAM and code space) are based on Atmel AVR or Hitachi chips and now ARM.

Arduino

Arduino is an open-source physical computing platform based on a simple input/output board and a development environment that implements the Processing/Wiring language. Arduino can be used to develop stand-alone interactive objects or can be connected to software on your computer (e.g. Flash, Processing, MaxMSP). The boards can be assembled by hand or purchased preassembled; the open-source IDE can be downloaded for free. Arduino uses an ATmega8 or ATmega168 microcontroller from Atmel's Atmel AVR series.

Platforms from Parallax, Inc.

Parallax produce the BASIC Stamp. These are Microchip PIC microcontrollers programmed with an interpreter that processes a program stored in an external EEPROM. Several different modules are available of varying processing speeds, RAM, and EEPROM sizes. The BASIC Stamp is used by Parallax as a platform for introductory programming and robotic kits.

SX-Key is Parallax's development tool for the SX line of microcontrollers, supporting every SX chip commercially available. Using free SX-Key software (Assembly language), or the SX/B Compiler (BASIC-style language) from Parallax, the SX-Key programming tool can program SX chips in-system and perform in-circuit source-level debugging.

Propeller is a multi-core microcontroller developed by Parallax, Inc. The currently released version features eight 32bit cores, each operating independently at 80MHz, and 32 I/O pins. It is programmed in a language named SPIN(tm) that was developed by Parallax to support this unique micro.

PICAXE

This PICAXE range of controllers from Revolution Education Limited^[1] are also based upon Microchip PICs and programmed with a BASIC interpreter. Using internal EEPROM or Flash to store the user's program, they deliver a single-chip solution and are quite inexpensive. A PICAXE programmer is simply a serial plug plus two resistors, and complete development software, comprehensive documentation and application notes are all available free of charge.

The BASIC-like programming language is almost identical to that used by Parallax's Basic Stamp 1 (BS1) but has been enhanced to support on-chip hardware and additional functionality. In common with the BS1 programming language, the PICAXE has support only for a limited number of variables, but allows access to internal RAM for storage which helps overcome that limitation.

The 5.0.X versions of the Visual IDE (the Programming Editor) introduced 'enhanced compilers' that support block-structured programming constructs plus conditional compilation and other directives.

Initially targeted at the UK educational sector, use of the PICAXE has spread to hobbyists and semi-professionals, and it can also be found inside some commercial products.

A-WIT Technologies, Inc.

A-WIT Technologies, Inc.^[2] has a microcontroller module named the C STAMP, along with support boards, kits, and software tools and infrastructure. The C STAMP is designed around a PIC microcontroller, and is programmed in a very user friendly subset of the standard C language called WC that is easy and powerful, because it relies on A-WIT's supplied software infrastructure. This microcontroller module is very affordable, and it has 48 pins, 35 KiB of memory, and runs at 40 MHz. The C STAMP also has a vast array of accessories and components, which are supported by A-WIT's software interfaces that enables seamless connectivity. This, in turn, enhances the ease of complete system development.

Comfile Technology Inc.

Comfile Technology Inc.^[3] produces a series of microcontrollers branded as CUBLOC and CuTOUCH, using the Atmel ATmega128 processor. They are very price competitive, being aimed at industrial applications, and include features such as Ladder Logic in addition to BASIC, a 80 kB program memory, and hardware pulse width modulation. Their focus is on developing industrial controllers which are fast, easy-to-use, and versatile. Comfile Technology's CuTOUCH is a visual Touch-screen controller that can be programmed in BASIC and Ladder Logic.

Coridium ARMexpress

ARMexpress^[4] is the first of a new family of DIP-24 (stamp-sized) controllers that combine a 60 MHz ARM CPU with a builtin BASIC compiler to achieve new levels of performance in this form factor. This combination makes this simple to use but very fast controller a good choice for the prototype builder or system integrator. 40K of code and 40K of data are available to the user, and code speed rivals that of programs written in C. The dialect of BASIC conforms more to Visual BASIC, but has hardware extensions like PBASIC.

ZX-24, ZX-40, ZX-44

The ZX series^[5] MCUs are based on the Atmel ATmega32 and ATmega644 processors. The devices run a field-upgradable Virtual Machine that features built-in multi-tasking, 32-bit floating point math and 1.5K to 3.5K of RAM for user's programs. Multi-tasking facilitates a more structured approach to coding for interface devices that require prompt service, e.g. serial devices, infrared remotes, etc.

The programming language for the ZX series is ZBasic, a modern dialect of Basic modeled after Microsoft's Visual Basic. Enhancements over the typical MCU Basic dialect include the availability of parameterized subroutines/functions that support local variables, strong type checking, user-defined types (structures), based variables, sub-byte data types (Bit and Nibble).

See also

- Embedded system
- Microarchitecture
- In-circuit emulator (ICE)
- List of common microcontrollers
- Robotics
- Contiki

Notes

1. ^ Revolution Education Limited (<http://www.rev-ed.co.uk/picaxe>)
2. ^ A-WIT Technologies, Inc. (<http://c-stamp.com/>)
3. ^ Comfile Technology Inc. (<http://cubloc.com/>)
4. ^ ARMexpress (<http://www.armmodule.com/>)
5. ^ Official ZBasic and ZX website (<http://www.zbasic.net/>)

External links

- Microcontroller (http://www.dmoz.org/Business/Electronics_and_Electrical/Control_Systems/Microcontroller at the Open Directory Project)
- Introduction to the MAXQ Architecture ([http://www.maxim-](http://www.maxim-ic.com/techdocs/intro_maxq_architecture.html)

- ic.com/appnotes.cfm?appnote_number=3222&CMP=WP-16)
- Understanding DC Electrical Characteristics of Microcontrollers (http://www.maxim-ic.com/appnotes.cfm?appnote_number=1087&CMP=WP-15)
- Cornell ECE476 final project designs (http://instruct1.cit.cornell.edu/courses/ee476/FinalProjects/)
- Microcontroller Projects Every Day (http://www.embedds.com/)
- Embedded Systems Design (http://www.embedded.com/mag.htm) magazine

Retrieved from "http://en.wikipedia.org/wiki/Microcontroller"

Categories: All articles with unsourced statements | Articles with unsourced statements since February 2007 | Articles with sections needing expansion | Wikipedia articles needing style editing from December 2007 | Digital electronics | Embedded systems | Microcontrollers | Central processing unit

- This page was last modified 18:41, 12 January 2008.
- All text is available under the terms of the GNU Free Documentation License. (See **Copyrights** for details.) Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a U.S. registered 501(c)(3) tax-deductible nonprofit charity.