Intel MCS-51

The Intel MCS-51 (commonly referred to as 8051) is a Harvard architecture, single chip microcontroller (µC) series which was developed by Intel in 1980 for use in embedded systems.[1][2] Intel's original versions were popular in the 1980s and early 1990s. While Intel no longer manufactures the MCS-51, binary compatible derivatives remain popular today. In addition to these physical devices, several companies also offer MCS-51 derivatives as IP cores for use in FPGAs or ASICs designs.

Intel's original MCS-51 family was developed using NMOS technology, but later versions, identified by a letter C in their name (e.g., 80C51) used CMOS technology and consumed less power than their NMOS predecessors. This made them more suitable for battery-powered devices.

Important features and applications

The 8051 architecture provides many functions (CPU, RAM, ROM, I/O, interrupt logic, timer, etc.) in a single package:

• 8-bit ALU, Accumulator and 8-bit Registers; hence it is an 8-bit microcontroller
• 8-bit data bus — It can access 8 bits of data in one operation
• Dual 16-bit address bus — It can access 2 x 2^16 memory locations — 64 KB (65536 locations) each of RAM and ROM
• On-chip RAM — 128 bytes (data memory)
• On-chip ROM — 4 kByte (program memory)
• Four byte bi-directional input/output port
• UART (serial port)
• Two 16-bit Counter/timers
• Two-level interrupt priority
• Power saving mode (on some derivatives)

One particularly useful feature of the 8051 core was the inclusion of a boolean processing engine which allows bit-level boolean logic operations to be carried out directly and efficiently on select internal registers and select RAM locations. This advantageous feature helped cement the 8051's popularity in industrial control applications because it reduced code size by as much as 30%. Another valued feature is the inclusion of four bank selectable working register sets which greatly reduce the amount of time required to complete an interrupt service routine. With
a single instruction the 8051 can switch register banks as opposed to the time consuming task of transferring the critical registers to the stack or designated RAM locations. These registers also allowed the 8051 to quickly perform a context switch which is essential for time sensitive real-time applications.

The MCS-51 UARTs make it simple to use the chip as a serial communications interface. External pins can be configured to connect to internal shift registers in a variety of ways, and the internal timers can also be used, allowing serial communications in a number of modes, both synchronous and asynchronous. Some modes allow communications with no external components. A mode compatible with an RS-485 multi-point communications environment is achievable, but the 8051’s real strength is fitting in with existing ad-hoc protocols (e.g., when controlling serial-controlled devices).

Once a UART, and a timer if necessary, have been configured, the programmer needs only to write a simple interrupt routine to refill the send shift register whenever the last bit is shifted out by the UART and/or empty the full receive shift register (copy the data somewhere else). The main program then performs serial reads and writes simply by reading and writing 8-bit data to stacks.

MCS-51 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 64 kB of internal program memory, and sometimes a quantity of extended data RAM (ERAM) located in the external data 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.

Features of the modern 8051 include built-in reset timers with brown-out detection, on-chip oscillators, self-programmable Flash ROM program memory, built-in external RAM, extra internal program storage, bootloader code in ROM, EEPROM non-volatile data storage, PC, SPI, and USB host interfaces, CAN or LIN bus, 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.

In many engineering schools the 8051 microcontroller is used in introductory microcontroller courses.

**Memory architecture**

The MCS-51 has four distinct types of memory – internal RAM, special function registers, program memory, and external data memory.

Internal RAM (IRAM) is located from address 0 to address 0xFF. IRAM from 0x00 to 0x7F can be accessed directly, and the bytes from 0x20 to 0x2F are also bit-addressable. IRAM from 0x80 to 0xFF must be accessed indirectly, using the @R0 or @R1 syntax, with the address to access loaded in R0 or R1.

Special function registers (SFR) are located from address 0x80 to 0xFF, and are accessed directly using the same instructions as for the lower half of IRAM. Some of the SFR’s are also bit-addressable.

Program memory (PMEM, though less common in usage than IRAM and XRAM) is located starting at address 0. It may be on- or off-chip, depending on the particular model of chip being used. Program memory is read-only, though some variants of the 8051 use on-chip flash memory and provide a method of re-programming the memory in-system or in-application. Aside from storing code, program memory can also store tables of constants that can be accessed by MOVCA, @DPTR, using the 16-bit special function register DPTR.

External data memory (XRAM) also starts at address 0. It can also be on- or off-chip; what makes it "external" is that it must be accessed using the MOVX (Move eXternal) instruction. Many variants of the 8051 include the standard 256 bytes of IRAM plus a few KB of XRAM on the chip.
Programming

There are various high-level programming language compilers for the 8051. 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. There are many commercial C compilers. SDCC is a popular open source C compiler. Other high level languages such as Forth, BASIC, Pascal/Object Pascal, PL/M and Modula-2 are available for the 8051, but they are less widely used than C and assembly.

Because IRAM, XRAM, and PMEM (read only) all have an address 0, C compilers for the 8051 architecture provide compiler-specific pragmas or other extensions to indicate where a particular piece of data should be stored (i.e. constants in PMEM or variables needing fast access in IRAM). Since data could be in one of three memory spaces, a mechanism is usually provided to allow determining to which memory a pointer refers, either by constraining the pointer type to include the memory space, or by storing metadata with the pointer.

Instruction set

The MCS-51 instruction set offers several addressing modes, including

- direct register, using ACC (the accumulator) and R0-R7
- direct memory, which access the internal RAM or the SFR's, depending on the address
- indirect memory, using R0, R1, or DPTR to hold the memory address. The instruction used may vary to access internal RAM, external RAM, or program memory.
- individual bits of a range of IRAM and some of the SFR's

And some of the common instructions are:

- **mov** | e.g., mov a,#30h ; a = 30h
  
  | syntax : | mov r2,39h ; r2 = |
  | mov destination, source | mov a,r3 |
  | | mov p2,a |
  
  description: *Loads the destination with the value at source and source data remains the same*

- **add** | e.g., add a,r3 ; a=a+r3
  
  description: *Adds the destination with the source and keeps it in the destination*, and source data remains the same

- **subb** | e.g., subb a,r3 ; a=a-r3
  
  description: *Subtracts the destination with the source and keeps it in the destination*,

- **clr** bit | e.g., clr p1.1
  
  description: *clears the bit p1.1*

- **setb** bit | e.g., setb p1.1
  
  description: *sets the bit p1.1 to 1*

- **inc** R0;
  
  description: *increments value at R0*

- **mul** AB;
Many of the operations allow any addressing mode for the source or the destination, for example, MOV 020h, 03fh will copy the value in memory location 0x3f in the internal RAM to the memory location 0x20, also in internal RAM.

Because the 8051 is an accumulator-based architecture, all arithmetic operations must use the accumulator, e.g. ADD A, 020h will add the value in memory location 0x20 in the internal RAM to the accumulator.

One does not need to master these instructions to program the 8051. With the availability of good quality C compilers, including open source SDCC, virtually all programs can be written with high-level language.

### Related processors

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 (ROM). To use this chip, external ROM had to be added containing the program that the 8031 would fetch and execute. An 8051 chip could be sold as a ROM-less 8031, as the 8051’s internal ROM is disabled by the normal state of the EA pin in an 8031-based design. A vendor might sell an 8051 as an 8031 for any number of reasons, such as faulty code in the 8051’s ROM, or simply an oversupply of 8051’s and undersupply of 8031’s.

The 8052 was an enhanced version of the original 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.

Intel discontinued its MCS-51 product line in March 2007,[3] however there are plenty of enhanced 8051 products or silicon intellectual property added regularly from other vendors. Current vendors of MCS-51 compatible processors include more than 20 independent manufacturers including Atmel, Infineon Technologies (formerly Siemens AG), Maxim Integrated Products (via its Dallas Semiconductor subsidiary), NXP (formerly Philips Semiconductor), Microchip Technology, Nuvoton (formerly Winbond), ST Microelectronics, Silicon Laboratories (formerly Cygnal), Texas Instruments, Ramtron International, Silicon Storage Technology, Cypress Semiconductor and Analog Devices.[4]

The 80C537 and 80C517 are CMOS versions, designed for the automotive industry. Enhancements mostly include new peripheral features and expanded arithmetic instructions. The 80C517 has fail-safe mechanisms, analog signal processing facilities and timer capabilities and 8 KB on-chip program memory. Other features include:

- 256 byte on-chip RAM
- 256 directly addressable bits
- External program and data memory expandable up to 64 KB
• 8-bit A/D converter with 12 multiplexed inputs
• Arithmetic unit can make division, multiplication, shift and normalize operations
• Eight data pointers instead of one for indirect addressing of program and external data memory
• Extended watchdog facilities
• Nine ports
• Two full-duplex serial interfaces with own baud rate generators
• Four priority level interrupt systems, 14 interrupt vectors
• Three power saving modes

Use as intellectual property

Today, 8051s are still available as discrete parts, but they are mostly used as silicon intellectual property cores. Available in high-level language source code ( VHDL or Verilog) or FPGA netlist forms, these cores are typically integrated within embedded systems, in products ranging from USB flash drives to washing machines to complex wireless communication systems on a chip. Designers use 8051 silicon IP cores, because of the smaller size, and lower power, compared to 32 bit processors like ARM M series, MIPS and BA22. Modern 8051 cores are faster than earlier packaged versions. Design improvements have increased 8051 performance while retaining compatibility with the original MCS 51 instruction set. The original Intel 8051 ran at 12 clock cycles per machine cycle, and most instructions executed in one or two machine cycles. A typical maximum clock frequency of 12 MHz meant these old 8051s could execute one million single-cycle instructions, or 500,000 two-cycle instructions, per second. In contrast, enhanced 8051 silicon IP cores now run at one clock cycle per machine cycle, and have clock frequencies of up to 450 MHz. That means an 8051-compatible processor can now execute 450 million instructions per second.

News MCU base on 8051

• STC: STC->89C51->STC15F2K60S2
• atmel: AT89C51->AT89S51->AT83C5134

References


Literature

• 8-Bit Embedded Controllers. Intel 1991, Order number 270645-003.
External links

- Complete tutorial for 8051 microcontrollers (http://www.ikalogic.com/tut_8051_1.php)
- Instruction set of 8051 microcontroller (http://www.uc8051.com/instruction-set/all-instructions)
- the source website for tutorials and simulator for 8051 (http://www.edsim51.com/)
- [STC http://www.stcmcu.com/]
- [INTEL http://www.intel.com/design/embcontrol/index.htm?iid=ipp_embed+micro&]
- [WINBOND http://www.winbond.com.tw/c-winbondhtm/partner/b_2_f_2.htm]
- [DALLAS http://www.maxim-ic.com/Microcontrollers.cfm]
- [SST http://www.sst.com/products.xhtml/embedded_controllers/]

This article is based on material taken from the Free On-line Dictionary of Computing prior to 1 November 2008 and incorporated under the "relicensing" terms of the GFDL, version 1.3 or later.
Article Sources and Contributors

Intel MCS-51
Contributors: Abhi3385, Alansohn, Ambiguous Furry Rocking Thing, Amollshah, AndroidX1, Antzerzos, Anmistej, Armstrong1113149, Artelike, Asaco, Bohle, CSWarren, CactusWriter, Calltech, CanisRufus, Chanetti, Choppingmall, Chowbok, Corwin, Crispunmarcher, DBigXray, David Biddulph, Decinovic, Dead Horsey, Dewiotech, Dholin, Dicklyon, Dou44h, Dragonofhavk, Dzlinker, Edhim51, Elinist, Empensobma, Envid, Flex, Flipmode fly, Frappucino, Fugmarsh, GRAHAMUK, Glider87, Henk88, Isnow, JzWed, JamesBWatson, Jerseyjoe, Joel Saks, Kgn, Kinjoare, Kshagunalaikar, Lazar Elena, LeaveShavets, Lightmouse, Ljusa, Ljedina, Macwhys, Makkwong, MarkusHagemlocher, Matanya (renamed), MauriceS, Mcdel, MichaelFrey, Miha Ulanov, Minghong, Mirror Vax, Modal, Mokihishil999, Molla, Mustafa1702, Naha, Naevras, Nihonjoe, Nishiki64, Nyeomark, Oldlapinop221, Outback the koala, Pfaguberg, Pflimemann, Pointillist, R. S. Shaw, Rait, Raj7991, RastaKiss, Rhondle, Riedlauner, Rjwilinski, Roger Irwin, Roguet880, Rrous, RunDolez, Rushsklesloshender, Sqi, Sajapa, Sam Hocvar, Sarplug0y, Shelfskwed, Sheryagoshi, Simon South, Somnya1221, Songisp77, Spacetruck723, Stan Shebs, Surjansh, Tailor-tinker, Thunderbird2, Tobias Bergemann, Toniroked, Tuta, Veinor, Wmahan, Wragge, Wsmarz, Wzlymansk3, 252 anonymous edits

Image Sources, Licenses and Contributors

File:KL Intel P8051.jpg
License: GNU Free Documentation License
Contributors: Konstantin Lanzet (with permission)

File:SAB-C515-LN.jpg
License: Creative Commons Attribution-ShareAlike 3.0 Unported
Contributors: MichaelFrey

File:Intel 8051 arch.svg
License: Creative Commons Attribution-Sharealike 3.0
Contributors: Appaloosa

License

Creative Commons Attribution-Share Alike 3.0 Unported
creativecommons.org/licenses/by-sa/3.0/