Intel 8051

Intel P8051 microcontroller

The **Intel MCS-51** (commonly termed **8051**, typically pronounced *eight-oh-five-one*) is a single chip microcontroller (MCU) series developed by Intel in 1980 for use in embedded systems. The architect of the Intel MCS-51 instruction set was John H. Wharton.[1][2] Intel's original versions were popular in the 1980s and early 1990s, and enhanced binary compatible derivatives remain popular today. It is an example of a complex instruction set computer (but also possessing some of the features of RISC architectures, such as a large register set and register windows) and has separate memory spaces for program instructions and data.

Intel's original MCS-51 family was developed using N-type metal-oxide-semiconductor (NMOS) technology, like its predecessor Intel MCS-48, but later versions, identified by a letter C in their name (e.g., 80C51) use complementary metal–oxide–semiconductor (CMOS) technology and consume less power than their NMOS predecessors. This made them more suitable for battery-powered devices.

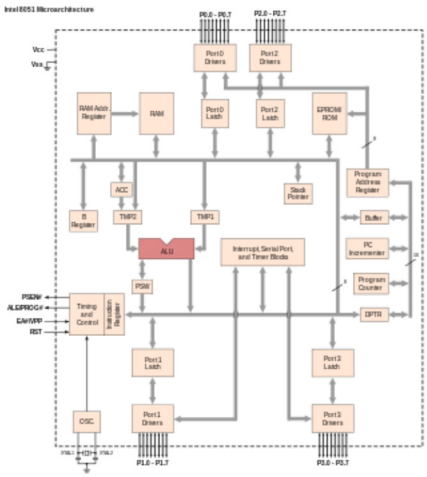
The family was continued in 1996 with the enhanced 8-bit MCS-151 and the 8/16/32-bit MCS-251 family of binary compatible microcontrollers.[3] While Intel no longer manufactures the MCS-51, MCS-151 and MCS-251 family, enhanced binary compatible derivatives made by numerous vendors remain popular today. Some derivatives integrate a digital signal processor (DSP). Beyond these physical devices, several companies also offer MCS-51 derivatives as IP cores for use in field-programmable gate array (FPGA) or application-specific integrated circuit (ASIC) designs.

Contents

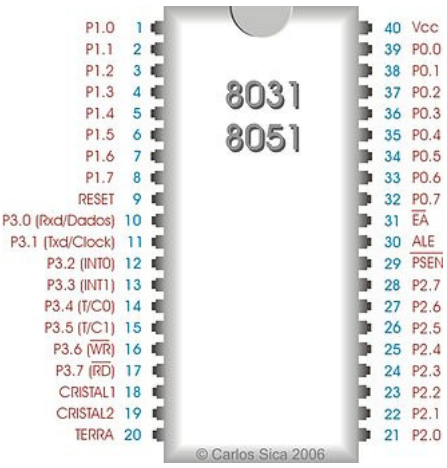
Important features and applications

The 8051 architecture provides many functions (central processing unit (CPU), random-access memory (RAM), read-only memory (ROM), input/output (I/O) ports, serial port, interrupt control, timers) in one package:

- 8-bit arithmetic logic unit (ALU) and accumulator, 8-bit registers (one 16-bit register with special move instructions), 8-bit data bus and 2×16-bit address buses, program counter, data pointer, and related 8/11/16-bit operations; hence it is mainly an 8-bit microcontroller
- Boolean processor with 17 instructions, 1-bit accumulator, 32 registers (4x8-bit, bit-addressable) and up to 144 special 1 bit-addressable RAM variables (18x8-bit)[4]
- Multiply, divide and compare instructions
- Four fast switchable register banks with eight registers each (memory mapped)
- Fast interrupt with optional register bank switching
- Interrupts and threads with selectable priority[5]
- 128 or 256 bytes of on-chip RAM (IRAM)
- Dual 16-bit address bus; it can access 2×2^{16} memory locations: 64 KB (65,536 locations) each of ROM (PMEM) and external RAM (XRAM), using two memory buses in a Harvard architecture.
- On-chip ROM (not included on 803x variants)
- Four 8-bit bi-directional input/output ports, bit addressable
- UART (serial port)
- Two 16-bit counter/timers
- Power saving mode (on some derivatives)



i8051 microarchitecture



i8051 pinout

One 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 select internal registers, ports and select RAM locations. Another feature is the inclusion of four bank selectable working register sets, which greatly reduce the time required to perform the context switches to enter and leave interrupt service routines. With one instruction, the 8051 can switch register banks, avoiding the time-consuming task of transferring the critical registers to RAM.

Once a UART, and a timer if necessary, has been configured, the programmer needs only 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.

Derivative features

As of 2013, new derivatives are still being developed by many major chipmakers, and major compiler suppliers such as IAR Systems, Keil and Altium Tasking[6] continuously release updates.

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. External RAM and ROM share the data and address buses. 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 (denoted "1T"), and have clock frequencies of up to 100 MHz, and are thus capable of an even greater number of instructions per second. All Silicon Labs, some Dallas (now part of Maxim Integrated) and a few Atmel (now part of Microchip) devices have single cycle cores.[7][8][9]

8051 variants may 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, I^2C, SPI, and USB host interfaces, CAN or LIN bus, ZigBee or Bluetooth radio modules, PWM generators, analog comparators, analog-to-digital and digital-to-analog converters, RTCs, extra counters and timers, in-circuit debugging facilities, more interrupt sources, extra power saving modes, more/less parallel ports etc. Intel manufactured a mask programmed version, 8052AH-BASIC, with a BASIC interpreter in ROM, capable of running user programs loaded into RAM.

MCS-51 based microcontrollers have been adapted to extreme environments. Examples for high-temperature variants are the Tekmos TK8H51 family for −40 °C to +250 °C[10] or the Honeywell HT83C51 for −55 °C to +225 °C (with operation for up to 1 year at +300 °C).[11] Radiation-hardenend MCS-51 microcontrollers for use in spacecraft are available; e.g., from Cobham (formerly Aeroflex) as the UT69RH051[12] or from NIIET as the 1830VE32 (Russian: 1830ВЕ32).[13]

In some engineering schools, the 8051 microcontroller is used in introductory microcontroller courses.[14][15][16][17]

Family naming conventions

8051 is the original name by Intel with 4 KB ROM and 128 byte RAM. Variants starting with 87 have a user programmable EPROM, sometimes UV erasable. Variants with a C as the third character are some kind of CMOS. 8031 and 8032 are ROM-less versions, with 128 and 256 bytes RAM. The last digit can indicate memory size, e.g. 8052 with 8 KB ROM, 87C54 16 KB EPROM, and 87C58 with 32 KB EPROM, all with 256 byte RAM.

Memory architecture

The MCS-51 has four distinct types of memory: internal RAM, special function registers, program memory, and external data memory. To access these efficiently, some compilers[18] utilize as many as 7 types of memory definitions: Internal RAM, single-bit access to internal RAM, special function registers, single-bit access to selected (divisible by 8) special function registers, program RAM, external ram accessed using a register indirect access, using one of the standard 8-bit registers, and register indirect external RAM access utilizing the 16-bit indirect access register.

The 8051's instruction set is designed as a Harvard architecture with segregated memory (data and instructions); it can only execute code fetched from program memory, and has no instructions to write to program memory. However, the bus leaving the IC has a single address and data path, and strongly resembles a Von Neumann architecture bus.

Most 8051 systems respect the instruction set, and require customized features to download new executable programs, e.g. in flash memory.

Internal RAM

Internal RAM (IRAM) has an 8-bit address space, using addresses 0 through 0xFF. IRAM from 0x00 to 0x7F contains 128 directly adressible 1-byte registers, which can be accessed using an 8-bit absolute address that is part of the instruction. Alternatively, IRAM can be accessed indirectly: the address is loaded into R0 or R1, and the memory is accessed using the @R0 or @R1 syntax, or as stack memory through the stack pointer SP, with the PUSH and POP operations; and *CALL and RET operations.

The original 8051 has only 128 bytes of IRAM. The 8052 added IRAM from 0x80 to 0xFF, which can *only* be accessed indirectly (e.g. for use as stack space). Most 8051 clones also have a full 256 bytes of IRAM.

Direct accesses to the IRAM addresses 80-FF are, instead, mapped onto the special function registers (SFR), where the accumulators A, B, carry bit C, and other special registers for control, status, etc., are located.

Special function registers

Special function registers (SFR) are located in the same address space as IRAM, at addresses 0x80 to 0xFF, and are accessed directly using the same instructions as for the lower half of IRAM. They cannot be accessed indirectly via @R0 or @R1 or by the stack pointer SP; indirect access to those addresses will access the second half of IRAM, instead.

The special function registers (SFR) include the accumulators A (or ACC, at E0) and B (at F0) and program status word (or PSW, at D0), themselves, as well as the 16-bit data pointer DPTR (at 82, as DPL and 83 as DPH). In addition to these, a small core of other special function registers - including the interrupt enable IE at A8 and interrupt priority IP at B8; the I/O ports

P0 (80), P1 (90), P2 (A0), P3 (B0); the serial I/O control SCON (98) and buffer SBUF (99); the CPU/power control register PCON (87); and the registers for timers 0 and 1 control (TCON at 88) and operation mode (TMOD at 89), the 16-bit timer 0 (TL0 at 8A, TH0 at 8C) and timer 1 (TL1 at 8B, TH1 at 8D) - are present on all versions of the 8051. Other addresses are version-dependent; in particular, the timer 2 registers for the 8052: the control register T2CON (at C8), the 16-bit capture/latch (RCAP2L at CA, RCAP2H at CB) and timer 2 (TL2 at CC and TH2 at CD), are not included with the 8051.

Register windows

The 32 bytes in IRAM from 0x00–0x1F contains space for four eight-byte register windows, which the 8 registers R0–R7 map to. The currently active window is determined by a two-bit address contained in the program status word.

Bit registers

The 16 bytes (128 bits) at IRAM locations 0x20–0x2F contains space for 128 1-bit registers, which are separately addressable as bit registers 00-7F.

The remaining bit registers, addressed as 80-FF, are mapped onto the 16 special function registers 80, 88, 90, 98, ..., F0 and F8 (those whose addresses are multiples of 8), and therefore include the bits comprising the accumulators A, B and program status word PSW. The register window address, being bits 3 and 4 on PSW, is itself addressable as bit registers D3 and D4, respectively; while the carry bit C (or CY), at bit 7 of the PSW, is addressable as bit register D7.

Program memory

Program memory (PMEM, though less common in usage than IRAM and XRAM) is up to 64 KB of read-only memory, starting at address 0 in a separate address space. 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.

In addition to code, it is possible to store read-only data such as lookup tables in program memory, retrieved by the `MOVC A,@A+DPTR` or `MOVC A,@A+PC` instructions. The address is computed as the sum of the 8-bit accumulator and a 16-bit register (PC or DPTR).

Special jump and call instructions (`AJMP` and `ACALL`) slightly reduce the size of code that accesses local (within the same 2 KB) program memory.[19]

When code larger than 64K is required, a common system makes the code bank-switched, with general-purpose I/O selecting the upper address bits. Some 8051 compilers[18] make provisions to automatically access paged code. In these systems the interrupt vectors and paging table are placed in the first 32K of code, and are always resident.

External data memory

External data memory (XRAM) is a third address space, also starting at address 0, and allowing 16 bits of address space. 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 kilobytes of XRAM on the chip.

The first 256 bytes of XRAM may be accessed using the `MOVX A,@R0`, `MOVX A,@R1`, `MOVX @R0,A`, and `MOVX @R1,A` instructions. The full 64KB may be accessed using `MOVX A,@DPTR` and `MOVX @DPTR,A`. The 16-bit address requires the programmer to load the 16-bit index register. For this reason, RAM accesses with 16-bit addresses are substantially slower.

Some CPUs[20] permit the 8-bit indirect address to use any 8-bit general purpose register.

To permit the use of this feature, some 8051-compatible microcontrollers with internal RAM larger than 256 bytes, or an inability to access external RAM[20] access internal RAM as if it were external, and have a special function register (e.g. PDATA) that permits them to set the upper address of the 256-byte page. This emulates the MCS8051 mode that can page the upper byte of a RAM address by setting the general-purpose I/O pins.

When RAM larger than 64K is required, a common system makes the RAM bank-switched, with general-purpose I/O selecting the upper address bits. Some 8051 compilers[18] make provisions to automatically access paged data.

Registers

The only register on an 8051 that is not memory-mapped is the 16-bit program counter (PC). This specifies the address of the next instruction to execute. Relative branch instructions supply an 8-bit signed offset which is added to the PC.

Eight general-purpose registers R0–R7 may be accessed with instructions one byte shorter than others. They are mapped to IRAM between 0x00 and 0x1F. Only eight bytes of that range are used at any given time, determined by the two bank select bits in the PSW.

The following is a partial list of the 8051's registers, which are memory-mapped into the special function register space:

Stack pointer, SP (0x81)
: This is an 8-bit register used by subroutine call and return instructions. The stack grows upward; the SP is incremented before pushing, and decremented after popping a value.

Data pointer, DP (0x82–83)
: This is a 16-bit register that is used for accessing PMEM and XRAM.

Program status word, PSW (0xD0)
: This contains important status flags, by bit number:

0. Parity, P. Gives the parity (XOR of the bits) of the accumulator, A.
1. User defined, UD. May be read and written by software; not otherwise affected by hardware.
2. Overflow flag, OV. Set when addition produces a signed overflow.
3. Register select 0, RS0. The low-order bit of the register bank. Set when banks at 0x08 or 0x18 are in use.
4. Register select 1, RS1. The high-order bit of the register bank. Set when banks at 0x10 or 0x18 are in use.
5. Flag 0, F0. May be read and written by software; not otherwise affected by hardware.
6. Auxiliary carry, AC. Set when addition produces a carry from bit 3 to bit 4.
7. Carry bit, C. Often used as the general register for bit computations, or the "Boolean accumulator".

Accumulator, A (0xE0)
: This register is used by most instructions.

B register (0xF0)
: This is used as an extension to the accumulator for multiply and divide instructions.

256 single bits are directly addressable. These are the 16 IRAM locations from 0x20–0x2F, and the 16 special function registers 0x80, 0x88, 0x90, ..., 0xF8. Any bit of these bytes may be directly accessed by a variety of logical operations and conditional branches.

Note that the PSW does not contain the common negative (N), or zero (Z) flags. For the former, the most significant bit of the accumulator can be addressed directly, as it is a bit-addressable SFR. For the latter, there are explicit instructions to jump on whether or not the accumulator is zero. There is also a two-operand compare and jump operation.

The parity (P) bit is often used to implement serial modes that include parity. To support this, the standard MCS51 UARTs could send 9 bits.

Microarchitecture

The microarchitecture of the Intel MCS8051 is proprietary, but published[21] features suggest how it works. It is a multi-cycle processor. The MCS8051 used 12 clock cycles[21] for most instructions. Many instructions utilize an accumulator.[21] In contrast, most compatible computers execute instructions in one to three cycles, except for the multiply and divide instructions. The much higher speed is a major reason why these have replaced the MCS8051 in most applications.

Each interrupt has four priorities.[21] Within each priority, the interrupts of devices are in a fixed priority.[21]

Instruction set

Instructions are all 1 to 3 bytes long, consisting of an initial opcode byte, followed by up to 2 bytes of operands.

¼ of the opcode bytes, ***x*0–*x*3**, are used for irregular opcodes.

¾ of the opcode bytes, ***x*4–*x*F**, are assigned to 16 basic ALU instructions with 12 possible operands. The least significant nibble of the opcode selects the primary operand as follows:

- ***x*8–*x*F**: Register direct, R0–R7.
- ***x*6–*x*7**: Register indirect, @R0 or @R1.
- ***x*5**: Memory direct, a following byte specifies an IRAM or SFR location.
- ***x*4**: Immediate, a following byte specifies an 8-bit constant. When the operand is a destination (`INC operand`, `DEC operand`) *or* the operation already includes an immediate source (`MOV operand,#data`, `CJNE operand,#data,offset`), this instead specifies that the accumulator is used.

The most significant nibble specifies the operation as follows. Not all support all addressing modes; the immediate mode in particular is unavailable when the primary operand is written to. Instruction mnemonics use *destination*, *source* operand order.

0*y*: `INC operand`

Increment the specified operand. Immediate mode (opcode 0x04) specifies the accumulator, `INC A`.

1*y*: **`DEC operand`**
Decrement the specified operand. Immediate mode (opcode 0x14) specifies the accumulator, `DEC A`.

2*y*: **`ADD A,operand`**
Add the operand to the accumulator, A. Opcode 0x23 (`RL A`, "rotate left" but actually a shift left) may be thought of as `ADD A,A`.

3*y*: **`ADDC A,operand`**
Add the operand, plus the C bit, to the accumulator. Opcode 0x33 (`RLC A`, rotate left through carry) may be thought of as `ADDC A,A`.

4*y*: **`ORL A,operand`**
Logical OR the operand into the accumulator. Two memory-destination forms of this operation, `ORL address,#data` and `ORL address,A`, are specified by opcodes 0x43 and 0x42.

5*y*: **`ANL A,operand`**
Logical AND the operand into the accumulator. Two memory-destination forms of this operation, `ANL address,#data` and `ANL address,A`, are specified by opcodes 0x53 and 0x52.

6*y*: **`XRL A,operand`**
Logical exclusive-OR the operand into the accumulator. Two memory-destination forms of this operation, `XRL address,#data` and `XRL address,A`, are specified by opcodes 0x63 and 0x62.

7*y*: **`MOV operand,#data`**
Move immediate to the operand. Immediate mode (opcode 0x74) specifies the accumulator, `MOV A,#data`.

8*y*: **`MOV address,operand`**
Move value to an IRAM or SFR register. Immediate mode (opcode 0x84) is not used for this operation, as it duplicates opcode 0x75.

9*y*: **`SUBB A,operand`**
Subtract the operand from the accumulator. This operation borrows and there is no subtract *without* borrow.

A*y*: **`MOV operand,address`**
Move value from an IRAM or SFR register. Immediate mode (opcode 0xA4) is not used, as immediates serve only as sources. Memory direct mode (opcode 0xA5) is not used, as it duplicates 0x85.

B*y*: **`CJNE operand,#data,offset`**
Compare *operand* to the immediate `#data`, and jump to PC + *offset* if not equal. Immediate and memory direct modes (opcodes 0xB4 and 0xB5) compare the operand against the accumulator, `CJNE A,operand,offset`. Note that there is no compare and jump if equal instruction, `CJE`.

C*y*: **`XCH A,operand`**
Exchange the accumulator and the operand. Immediate mode (opcode 0xC4) is not used for this operation.

D*y*: **`DJNZ operand,offset`**
Decrement the operand, and jump to PC + *offset* if the result is non-zero. Immediate mode (opcode 0xD4), and register indirect mode (0xD6, 0xD7) are not used.

E*y*: **`MOV A,operand`**
Move operand to the accumulator. Immediate mode is not used for this operation (opcode 0xE4), as it duplicates opcode 0x74.

F*y*: **`MOV operand,A`**
Move accumulator to the operand. Immediate mode (opcode 0xF4) is not used, as it would have no effect.

Only the ADD, ADDC, and SUBB instructions set PSW flags. The INC, DEC, and logical instructions do not. The CJNE instruction modifies the C bit only, to the borrow that results from *operand1* – *operand2*.

The irregular instructions comprise 64 opcodes, having more limited addressing modes, plus several opcodes scavenged from inapplicable modes in the regular instructions.

8051/8052 irregular instructions

Opcode	*x*0	*x*1	*x*2	*x*3	*x*4
0*y*	NOP	AJMP addr11, ACALL addr11	LJMP addr16	RR A (rotate right)	INC A
1*y*	JBC bit,offset (jump if bit set with clear)		LCALL addr16	RRC A (rotate right through carry)	DEC A
2*y*	JB bit,offset (jump if bit set)		RET	RL A (rotate left)	ADD A,#data
3*y*	JNB bit,offset (jump if bit clear)		RETI	RLC A (rotate left through carry)	ADDC A,#data
4*y*	JC offset (jump if carry set)		ORL address,A	ORL address,#data	ORL A,#data
5*y*	JNC offset (jump if carry clear)		ANL address,A	ANL address,#data	ANL A,#data
6*y*	JZ offset (jump if zero)		XRL address,A	XRL address,#data	XRL A,#data
7*y*	JNZ offset (jump if non-zero)		ORL C,bit	JMP @A+DPTR	MOV A,#data
8*y*	SJMP offset (short jump)		ANL C,bit	MOVC A,@A+PC	DIV AB
9*y*	MOV DPTR,#data16		MOV bit,C	MOVC A,@A+DPTR	SUBB A,#data
A*y*	ORL C,/bit		MOV C,bit	INC DPTR	MUL AB
B*y*	ANL C,/bit		CPL bit	CPL C	CJNE A,#data,offset
C*y*	PUSH address		CLR bit	CLR C	SWAP A
D*y*	POP address		SETB bit	SETB C	DA A (decimal adjust)
E*y*	MOVX A,@DPTR		MOVX A,@R0	MOVX A,@R1	CLR A
F*y*	MOVX @DPTR,A		MOVX @R0,A	MOVX @R1,A	CPL A

A5

Unused

B5

CJNE A,address,offset

D6–7

XCHD A,@R0–1 exchange low-order nibble of operands.

The `SJMP` (short jump) opcode takes a signed relative offset byte operand and transfers control there relative to the address of the following instruction. The `AJMP`/`ACALL` opcodes combine the three most significant bits of the opcode byte with the following byte to specify an 11-bit destination that is used to replace 11 bottom bits of the PC register (top 5 bits of PC register remain intact). For larger addresses, the `LJMP` and `LCALL` instructions allow a 16-bit destination.

One of the reasons for the 8051's popularity is its range of operations on single bits. Bits are always specified by absolute addresses; there is no register-indirect or indexed addressing. Instructions that operate on single bits are:

- `SETB bit`, `CLR bit`, `CPL bit`: Set, clear, or complement the specified bit
- `JB bit,offset`: Jump if bit set
- `JNB bit,offset`: Jump if bit clear
- `JBC bit,offset`: Jump if bit set, and clear bit
- `MOV C,bit`, `MOV bit,C`: Move the specified bit to the carry bit, or vice versa
- `ORL C,bit`, `ORL C,/bit`: Or the bit (or its complement) to the carry bit
- `ANL C,bit`, `ANL C,/bit`: And the bit (or its complement) to the carry bit

A bit operand is written in the form `address.number`. Because the carry flag is bit 7 of the bit-addressable program status word, the `SETB C`, `CLR C` and `CPL C` instructions are shorter equivalents to `SETB PSW.7`, `CLR PSW.7` and `CPL PSW.7`.

Although most instructions require that one operand is the accumulator or an immediate constant, opcode 0x85 performs `MOV` directly between two internal RAM locations.

Programming

There are various high-level programming language compilers for the 8051. Several C compilers are available for the 8051, most of which 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.[22] Small Device C Compiler (SDCC) is a popular open source C compiler.[23] Other high level languages such as C++, Forth,[24][25][26][27] BASIC, Object Pascal, Pascal, PL/M and Modula-2 are available for the 8051, but they are less widely used than C and assembly.

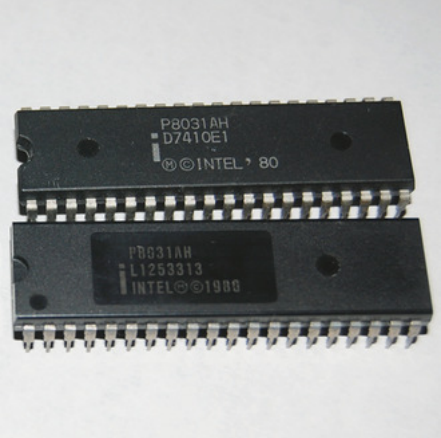
Because IRAM, XRAM, and PMEM (read only) all have an address 0, C compilers for the 8051 architecture provide compiler-specific pragmas or other extensions to indicate where a particular piece of data should be stored (i.e. constants in PMEM or variables needing fast access in IRAM). Since data could 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.

Related processors

Intel discontinued its MCS-51 product line in March 2007;[28][29] however, there are plenty of enhanced 8051 products or silicon intellectual property added regularly from other vendors.

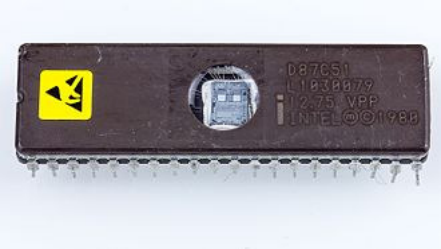
The 8051's predecessor, the 8048, was used in the keyboard of the first IBM PC, where it

converted keypresses into the serial data stream which is sent to the main unit of the computer. An Intel 8049 served a similar role in the Sinclair QL. The 8048 and derivatives are still used today for basic model keyboards.



Intel 8031 microcontrollers

The **8031** was a reduced version of the original 8051 that had no internal program memory (read-only 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 8051s and undersupply of 8031s.



Intel D87C51 microcontroller

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. Most modern 8051-compatible microcontrollers include these features.

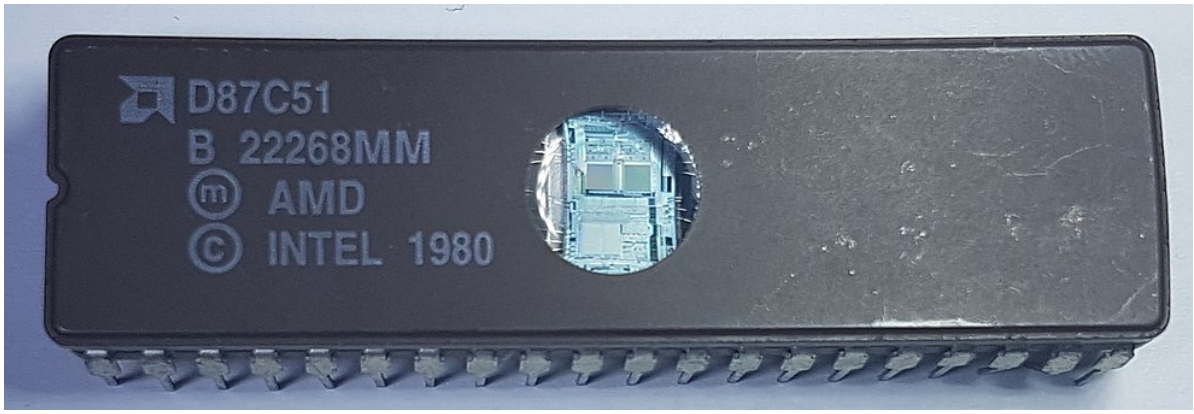
The **8032** had these same features as the 8052 except lacked internal ROM program memory.

The **8751** was an 8051 with 4 KB EPROM instead of 4 KB ROM. They were identical except for the non-volatile memory type. This part was available in a ceramic package with a clear quartz window over the top of the die so UV light could be used to erase the EPROM. Related parts are: 8752 had 8 KB EPROM, 8754 had 16 KB EPROM, 8758 had 32 KB EPROM.

The **80C537** (ROM-less) and **80C517** (8 KB ROM) are CMOS versions, designed for the automotive industry. Enhancements mostly include new and enhanced peripherals. The 80C5x7 has fail-safe mechanisms, analog signal processing facilities, enhanced timer capabilities, and a 32-bit arithmetic peripheral. Other features include:

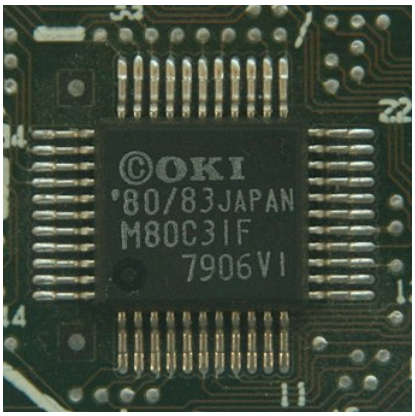
- 256-byte on-chip RAM
- 256 directly addressable bits
- External program and data memory expandable up to 64 KB
- 8-bit A/D converter with 12 multiplexed inputs
- Arithmetic peripheral can perform 16×16→32-bit multiplication, 32/16→16-bit division, 32-bit shift and 32-bit normalize operations
- Eight data pointers instead of one for indirect addressing of program and external data memory
- Extended watchdog facilities
- Nine I/O ports
- Two full-duplex serial interfaces with individual baud rate generators
- Four priority level interrupt systems, 14 interrupt vectors
- Three power saving modes

Intel MCS-51 second sources

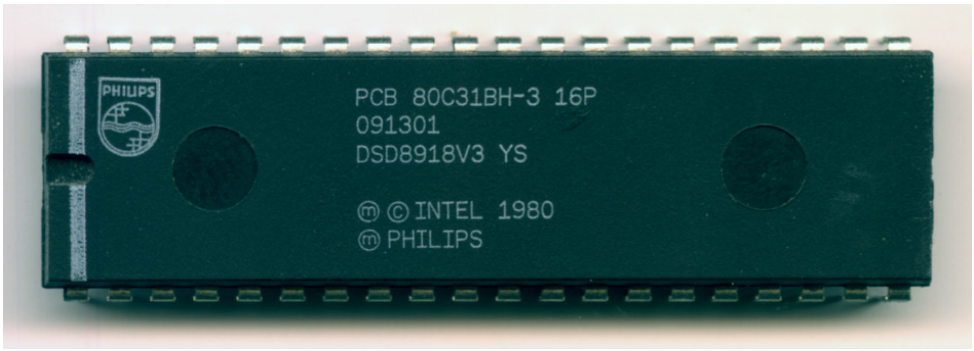


AMD D87C51

MHS S-80C31



OKI M80C31



Philips PCB80C31

Signetics SCN8031

Temic TS80C32

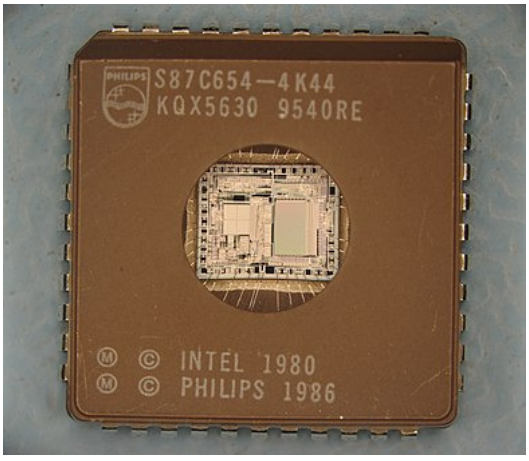
Derivative vendors

More than 20 independent manufacturers produce MCS-51 compatible processors.

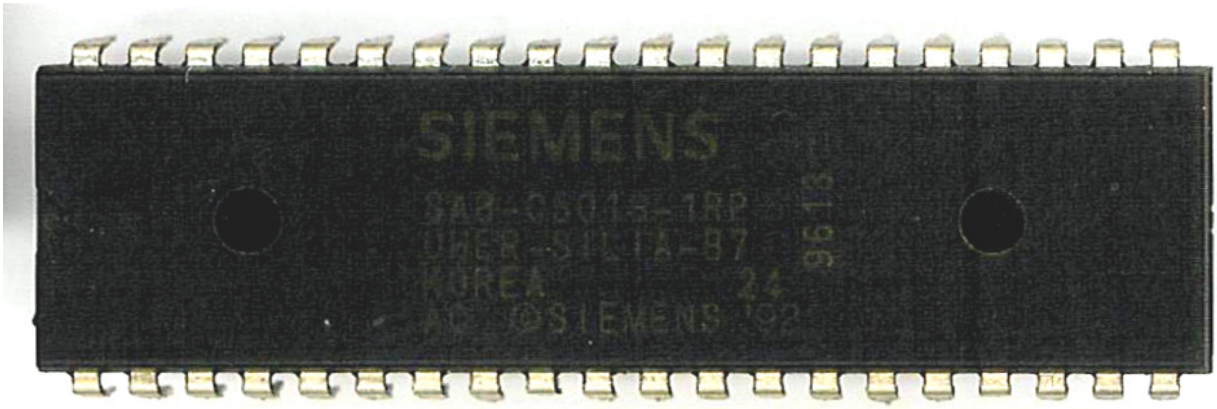
Intel MCS-51 derived microcontrollers

Atmel AT89C2051

Infineon SAB-C515



Philips S87C654



Siemens SAB-C501

STC Micro STC89C52

Other ICs or IPs compatible with the MCS-51 have been developed by Analog Devices,[30] Integral Minsk,[31] Kristall Kyiv,[32] and NIIET Voronesh.[13]

Use as intellectual property

Today, 8051s are still available as discrete parts, but they are mostly used as silicon intellectual property cores.[33] Available in hardware description language source code (such as 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 Cortex-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.

MCUs based on 8051

Silicon Storage Technology 89V54RD2

- ABOV: MC94F, MC95F, MC96F series
- Infineon: XC800
- Maxim Integrated (formerly Dallas): DS80-series etc.[34]
- Mentor Graphics: M8051EW etc. designed for Mentor by SYNTILL8[35]
- Megawin: 74, 82, 84, 86, 87, and 89 series
- Microchip (formerly Atmel): AT89C51, AT89S51, AT83C5134, etc.[9]
- NXP: NXP700 and NXP900 series
- Siemens SAB 80532-N
- Silergy electricity metering SoCs: 71M6511, 71M6513, 71M6531, 71M6533, 71M6534, 71M6542, 71M6543[36] Energy measurement SoCs: 78M6631, 78M6618, 78M6613, 78M6612[37]
- Silicon Labs: C8051 series and EFM8 series[7]
- Silicon Storage Technology: FlashFlex51 MCU (SST89E52RD2, SST89E54RD2, SST89E58RD2, SST89E516RD2SST89V52RD2, SST89V54RD2, SST89V58RD2, SST89V516RD2)[38]
- STC Micro: STC89C51RC, STC90C51RC, STC90C58AD, STC10F08XE, STC11F60XE, STC12C5410AD, STC12C5202AD, STC12C5A60S2, STC12C5628AD, STC15F100, STC15F204EA, STC15F2K60S2, STC15F4K60S2, STC15F101W, STC15F408AD, STC15W104, STC15W408S, STC15W201S, STC15W408AS, STC15W1K16S and STC15W4K56S4 series[39]
- Texas Instruments CC111x, CC24xx and CC25xx families of RF SoCs
- WCH (Nanjing Qinheng Microelectronics): CH551, CH552, CH554, CH546, CH547, CH548, CH558, CH559[40]

Digital signal processor (DSP) variants

Several variants with an additional 16-bit digital signal processor (DSP) (for example for MP3 or Vorbis coding/decoding) with up to 675 million instructions per second (MIPS)[41] and integrated USB 2.0 interface[42] or as intellectual property[43] exist.

Enhanced 8-bit binary compatible microcontroller: MCS-151 family

In 1996 Intel announced the MCS-151 family, an up to 6 times faster variant,[3] that's fully binary and instruction set compatible with 8051. Unlike their 8051 MCS-151 is a pipelined CPU, with 16-bit internal code bus and is 6x the speed. The MCS-151 family was also discontinued by Intel, but is widely available in binary compatible and partly enhanced variants.

8/16/32-bit binary compatible microcontroller: MCS-251 family

The 80251 8/16/32-bit microcontroller with 16 MB (24-bit) address-space and 6 times faster instruction cycle was introduced by Intel in 1996.[3][44] It can perform as an 8-bit 8051, has 24-bit linear addressing, an 8-bit ALU, 8-bit instructions, 16-bit instructions, a limited set of 32-bit instructions, 16 8-bit registers, 16 16-bit registers (8 16-bit registers which do not share space with any 8-bit registers, and 8 16-bit registers which contain 2 8-bit registers per 16-bit register), and 10 32-bit registers (2 dedicated 32-bit registers, and 8 32-bit registers which contain 2 16-bit registers per 32-bit register).[45]

It features extended instructions[46] – see also the programmer's guide[47] – and later variants with higher performance,[48] also available as intellectual property (IP).[49] It is 3-stage pipelined. The MCS-251 family was also discontinued by Intel, but is widely available in binary compatible and partly enhanced variants from many manufacturers.

See also

- DS80C390
- Hitachi HD44780 - LCD controller with XRAM-compatible interface
- Intel PL/M-51
- SDK-51 System Design Kit

References

1. John Wharton: *An Introduction to the Intel MCS-51 Single-Chip Microcomputer Family (https://drive.google.com/uc?export=download&id=0B9rh9tVI0J5mZTFmZjRjZTItNDQ0Yy00MDFlLTgzZTgtM2I3MzVkMTliNTFl)*, Application Note AP-69, May 1980, Intel Corporation.
2. *Intel 8051 Microprocessor Oral History Panel* (https://web.archive.org/web/20120225062256/http://archive.computerhistory.org/resources/text/Oral_History/Intel_8051/102658339.05.01.acc.pdf) (PDF), Computer History Museum, September 16, 2008, archived from the original (http://archive.computerhistory.org/resources/text/Oral_History/Intel_8051/102658339.05.01.acc.pdf) (PDF) on February 25, 2012, retrieved November 17, 2018
3. "Intel MCS 151 and MCS 251 Microcontrollers" (http://datasheets.chipdb.org/Intel/MCS51/151BACK.HTM). *datasheets.chipdb.org*.
4. John Wharton: *Using the Intel MCS-51 Boolean Processing Capabilities* (http://www2.elo.utfsm.cl/~lsb/elo311/aplicaciones/intel/booleanproc.pdf) Archived (https://web.archive.org/web/20160303234549/http://www2.elo.utfsm.cl/~lsb/elo311/aplicaciones/intel/booleanproc.pdf) 2016-03-03 at the Wayback Machine Application Note AP-70, May 1980, Intel Corporation.
5. "8051 Tutorial: Interrupts" (https://web.archive.org/web/20121228002749/http://www.8052.com/tutint.phtml#). Archived from the original (http://www.8052.com/tutint.phtml) on 2012-12-28. Retrieved 2012-12-21.
6. "TASKING" (https://www.tasking.com/). *www.tasking.com*.
7. "8-bit Microcontrollers - 8-bit MCUs - EFM8 - Silicon Labs" (https://www.silabs.com/mcu/8-bit). *www.silabs.com*. Retrieved 2021-06-21.
8. "Site Search | Maxim Integrated" (https://www.maximintegrated.com/en/site-search.html#q=DS80&sort=relevancy&f:@common_product_hierarchy=%5BMCU%20-%20Microcontroller%5D). *www.maximintegrated.com*. Retrieved 2021-06-21.
9. "8051 MCUs | Microchip Technology" (https://www.microchip.com/en-us/products/microcontrollers-and-microprocessors/8-bit-mcus/8051-mcus). *www.microchip.com*. Retrieved 2021-06-21.

10. "TK80H51 250ºC Microcontroller" (https://web.archive.org/web/20170820093849/http://www.tekmos.com/products/80c51-microcontrollers/tk80h51-250-c-microcontroller). Tekmos Inc. Archived from the original (http://www.tekmos.com/products/80c51-microcontrollers/tk80h51-250-c-microcontroller) on 20 August 2017. Retrieved 23 August 2017.
11. "HIGH TEMPERATURE 83C51 MICROCONTROLLER" (http://www.keil.com/dd/docs/datashts/honeywell/ht83c51.pdf) (PDF). Honeywell. Retrieved 23 August 2017.
12. "Microcontrollers and Microprocessors" (https://web.archive.org/web/20170823020620/http://ams.aeroflex.com/pagesproduct/prods-hirel-uprocessors.cfm). Cobham Semiconductor Solutions. Archived from the original (http://ams.aeroflex.com/pagesproduct/prods-hirel-uprocessors.cfm) on 23 August 2017. Retrieved 23 August 2017.
13. "Микроконтроллеры" (https://web.archive.org/web/20170822175635/http://niiet.ru/goods/chips/microcont) [Microcontrollers] (in Russian). Voronezh: OAO "NIIET". Archived from the original (http://niiet.ru/goods/chips/microcont) on 22 August 2017. Retrieved 22 August 2017.
14. "Archived copy" (https://web.archive.org/web/20170822215715/http://play.tojsiab.com/WFhtU3d2djNXckUz). Archived from the original (http://play.tojsiab.com/WFhtU3d2djNXckUz) on 2017-08-22. Retrieved 2017-08-22.
15. Archived at Ghostarchive (https://ghostarchive.org/varchive/youtube/20211205/H9sDn89EvD8) and the Wayback Machine (https://web.archive.org/web/20200608025501/https://www.youtube.com/watch?v=H9sDn89EvD8&gl=US&hl=en): "ELEC2700 Assignment 1 2014: 1D Pong" (https://www.youtube.com/watch?v=H9sDn89EvD8). *YouTube*.
16. "ELEC2700 - Computer Engineering 2 - University of Newcastle - Textbooks | Zookal.com" (https://www.zookal.com/). *www.zookal.com*.
17. http://s3.amazonaws.com/f01.justanswer.com/88willy/2012-06-29_065532_assignment_03.pdf
18. *Keil C51 Users' Manual* (https://www.keil.com/c51/man/c51.htm). Keil, a division of ARM Inc. 2021. Retrieved 17 May 2021.
19. ACALL is a 2-byte subroutine calling instruction, it can access locations within the same 2 KB segment of memory. The absolute memory address is formed by the high 5 bits of the PC and the 11 bits defined by the instruction.
20. "Silergy 71M6513 Data sheet" (https://www.silergy.com/download/upload/pdffile/71M6513_DS_Rev4.pdf/2673). *Silergy electricity metering ICs*. Silergy Corp. Retrieved 17 May 2021.
21. *MCS-51 Microcontroller Family User's Manual* (http://datasheets.chipdb.org/Intel/MCS51/MANUALS/27238302.PDF) (PDF). publication number 121517: Intel. 1994. Retrieved 17 May 2021.
22. Han-Way Huang. "Embedded System Design with C8051" (https://books.google.com/books?id=PnXNNGG_jMMC). p. 238.
23. Lewin A. R. W. Edwards. "So, You Wanna be an Embedded Engineer: The Guide to Embedded Engineering, from Consultancy to the Corporate Ladder" (https://archive.org/details/pdfy-MKFMCCNbqENoR3Ft). 2006. p. 51.
24. Bradford J. Rodriguez. "CamelForth/8051" (http://www.camelforth.com/page.php?4).
25. Brad Rodriguez. "Moving Forth Part 7: CamelForth for the 8051" (http://www.bradrodriguez.com/papers/moving7.htm).
26. "8051 SwiftX Forth development" (http://www.forth.com/embedded/swiftx-embedded-systems-development-tools.html?MCU=8051). Archived (https://web.archive.org/web/20150924014750/http://www.forth.com/embedded/swiftx-embedded-systems-development-tools.html?MCU=8051#) 2015-09-24 at the Wayback Machine
27. "MPE VFX Forth 7 cross compilers" (http://www.mpeforth.com/xc7.htm).
28. Ganssle, Jack (2006-05-29). "Intel bows out, discontinues MCS-51" (https://www.embedded.com/electronics-blogs/break-points/4025678/Intel-bows-out).

29. "MCS 51, MCS 251 and MCS 96 Microcontroller Product Lines, the Intel 186, Intel386 and Intel486 Processors Product Lines, and the i960 32 Bit RISC Processor, PCN 106013-01, Product Discontinuance, Reason for Revision: Add Key Milestone information and revise description of change" (http://qdms.intel.com/dm/d.aspx/7C715FEC-C598-444E-9EFB-2750B98F7956/PCN106013-01.pdf) (PDF). Intel. 2006-05-02.
30. http://www.analog.com/static/imported-files/data_sheets/ADUC832.pdf
31. "Микроконтроллеры и супервизоры питания Серии 1880; 1881; 1842; 588; 1345; 5518АП1ТБМ" (http://integral.by/ru/products/mikrokontrollery-i-supervizory-pitaniya-serii-1880-1881-1842-588-1345-5518ap1tbm) [Microcontrollers and Power Supervisors Series 1880; 1881; 1842; 588; 1345; 5518AP1TBM] (in Russian). Minsk: OAO "Integral". Retrieved 6 January 2017.
32. "Однокристальные микро-эвм" (https://web.archive.org/web/20120530003614/http://krystall.net.ua/ru/products/53.html) [Single-chip microcomputers] (in Russian). Kyiv: Kristall. Archived from the original (http://krystall.net.ua/ru/products/53.html) on 30 May 2012. Retrieved 5 January 2017.
33. Hussaini (20 August 2019). "Why do we have to use the 8051? Isn't it too old?" (https://www.technobyte.org/8051-not-old-still-popular-use-why/). *Technobyte*. Retrieved 20 August 2019.
34. "DS80C320 High-Speed/Low-Power Microcontrollers - Maxim Integrated" (https://www.maximintegrated.com/en/products/microcontrollers/DS80C320.html). *www.maximintegrated.com*. Retrieved 2021-06-21.
35. "Syntill8 - Products" (http://www.syntill8.com/products.html). *www.syntill8.com*. Retrieved 2021-06-21.
36. "Silergy Metering ICs" (https://www.silergy.com/products/electricity_metering_ics). Silergy Corp. Retrieved 12 May 2021.
37. "Silergy Energy Measurement ICs" (https://www.silergy.com/products/energy_measurement_ics). Silergy Corp. Retrieved 12 May 2021.
38. datasheetq.com. "89V54RD2 Datasheet PDF Download - Silicon Storage Technology" (https://www.datasheetq.com/datasheet-download/190246/1/SST/89V54RD2). *www.datasheetq.com*. Retrieved 2020-01-18.
39. "STC Microcontroller---STCmicro Technology Co,.Ltd" (http://www.stcmicro.com/stcmcu.html). *www.stcmicro.com*. Retrieved 2017-02-19.
40. "site index - Nanjing Qinheng Microelectronics Co., Ltd" (http://wch-ic.com/). *wch-ic.com*. Retrieved 2021-06-21.
41. "TI Delivers new low-cost, high-performance audio DSP for Home and Car w/ 8051" (https://web.archive.org/web/20161113175939/http://www.8052.com/news?NEWSID=61#). Archived from the original (http://www.8052.com/news?NEWSID=61#) on 2016-11-13. Retrieved 2013-05-06.
42. "Atmel AT85C51SND3 Audio DSP Data Sheet with USB 2.0" (http://www.keil.com/dd/docs/datashts/atmel/at85c51snd3.pdf) (PDF).
43. Salim, A.J.; Othman, M.; Ali, M.A. Mohd (October 5, 2006). "Integration of 8051 With DSP in Xilinx FPGA". *2006 IEEE International Conference on Semiconductor Electronics*. pp. 562–566. doi:10.1109/SMELEC.2006.380694 (https://doi.org/10.1109%2FSMELEC.2006.380694). ISBN 0-7803-9730-4. S2CID 21616742 (https://api.semanticscholar.org/CorpusID:21616742) – via IEEE Xplore.
44. The 8051 microcontroller By Kenneth J Ayala (https://books.google.com/books?id=l6lveWkWqFoC&pg=PA400&lpg=PA400&ots=aX1rfMDJdF&dq=intel+80251&ie=ISO-8859-1&output=html) Google books
45. http://datasheets.chipdb.org/Intel/MCS51/DATASHTS/27262001.PDF
46. "Temic TSC80251 Architecture" (http://pe2bz.philpem.me.uk/pdf%20on%20typenumber/A-C/C251.pdf) (PDF).

47. "Atmel TSC80251 Programmers Guide" (https://web.archive.org/web/20160304051710/http://www.cs.unc.edu/~vicci/comp261/project/mcu/80251_prog_man.pdf) (PDF). Archived from the original (http://www.cs.unc.edu/~vicci/comp261/project/mcu/80251_prog_man.pdf) (PDF) on 2016-03-04. Retrieved 2013-05-06.
48. DQ80251 32bit Microcontroller (http://dcd.pl/workspace/documentation/lat/dq80251_ds.pdf) DCD
49. R80251XC 32bit Microcontroller (http://www.evatronix.pl/products/pdf/r80251xc_datasheet.pdf) Evatronix

Further reading

Books

- Mazidi; McKinlay; Mazidi (2012). *The 8051 Microcontroller: A Systems Approach*. 648 pp. ISBN 978-0-13-508044-3.
- Schultz, Thomas (2008). *C and the 8051* (4th ed.). 464 pp. ISBN 978-0-9783995-0-4.
- Steiner, Craig (2005). *The 8051/8052 Microcontroller: Architecture, Assembly Language, and Hardware Interfacing*. 348 pp. ISBN 978-1-58112-459-0.
- Calcutt; Cowan; Parchizadeh (2000). *8051 Microcontrollers: Hardware, Software and Applications*. 329 pp. ISBN 978-0-340-67707-0.
- Axelson, Jan (1994). *The Microcontroller Idea Book: Circuits, Programs, and Applications featuring the 8052-BASIC Microcontroller*. 277 pp. ISBN 978-0-9650819-0-0.
- Payne, William (December 19, 1990) [1990]. *Embedded Controller FORTH for the 8051 Family* (hardcover). Boston: Academic Press. 528 pp. ISBN 978-0-12-547570-9.

Intel

- *MCS-51 Microcontroller Family User's Manual (http://datasheets.chipdb.org/Intel/MCS51/MANUALS/27238302.PDF)*; Intel; 1994; publication number 121517.
- *MCS-51 Macro Assembler User's Guide*; Intel; publication number 9800937.
- *8-Bit Embedded Controllers*; Intel; 1991; publication number 270645-003.
- *Microcontroller Handbook*; Intel; 1984; publication number 210918-002.
- *8051 Microcontroller Preliminary Architectural Specification and Functional Description (https://archive.org/details/bitsavers_intel80518liminaryArchitecturalSpecificationMay80_6120863/)*; Intel; 44 pages; 1980.

Misc

- Hennig-Roleff, Werner (1993-02-01) [1988]. "HEX.DOC: EASM51 - HEX Format (elektor Assembler)" (http://spot.fho-emden.de/ftp/micro/sim51_04.zip). *SIM51*. 1.04 (in German). Archived (https://web.archive.org/web/20170811161238/http://spot.fho-emden.de/ftp/micro/sim51_04.zip) from the original on 2017-08-11. Retrieved 2021-12-08. [1] (https://web.archive.org/web/20210806202253/https://www.dos4ever.com/8031board/SIM51D_BASIC.zip)[2] (https://web.archive.org/web/20210806202253/https://www.dos4ever.com/8031board/SIM51_06.zip) (The HEX.DOC file discusses various hex file formats by Intel, Motorola, Tektronix, MOS Technology and elektor.)

External links

Media related to MCS-51 at Wikimedia Commons

- Complete tutorial for 8051 microcontrollers (https://web.archive.org/web/20101123154437/http://ikalogic.com/tut_8051_1.php)
- the source website for tutorials and simulator for 8051 (http://www.edsim51.com/)
- Basic 8051 Interfacing Circuits (http://www.dnatechindia.com/Tutorial/8051-Tutorial.html)
- Open source VHDL 8051 implementation (Oregano Systems) (http://www.oreganosystems.at/?page_id=96)
