



Keywords: 8051 instruction set, high-speed microcontroller, high-performance microprocessor, high speed micros, 4 clock per machine cycle, 12 clock per machine cycle, reduced power consumption, bootstrap loader, automatic bank-switching, expanded memory address

May 14, 2003

APPLICATION NOTE 2035

The Fast 8051 Microcontroller: Leading the Way in Growth and Innovation

Abstract: This article describes how Dallas Semiconductor has improved the traditional 8051 microcontroller, creating high-performance drop-in replacements which execute up to 33MIPs. Additional features such as multiple data pointers, expanded memory addressing (up to 16MB), and flash memory increase the speed and usefulness of the device.

System designers know that the microcontroller is the heart of any embedded system—that's where the action takes place. For over 18 years, Dallas Semiconductor, wholly owned subsidiary of Maxim Integrated Products, has been redefining the ubiquitous 8051 microcontroller. Perhaps the biggest improvements in the last 10 years have been made in the speed of instruction execution. Our 1 clock-per-machine-cycle processors reached a remarkable performance goal—1 clock-per-machine-cycle, currently at 33 million instructions per second (MIPS). Using this core, our secure, networked, and mixed-signal 8051-microcontroller families continue to set the standard for feature integration and innovation.

Why base a family of innovative microcontrollers on the venerable 8051 instruction set? Because, quite simply, it is one of the most popular 8-bit microcontroller architectures in the world. The instruction set is simple to understand, making it a favorite of embedded system designers. Many of the instructions directly address I/O pins, allowing quick manipulation (bit-banging) of external peripherals. A tremendous variety of on-chip peripherals is available in an almost limitless number of combinations. In addition, development tools for the 8051-microcontroller family are widely available, so it is easy and inexpensive to start developing an application.

Safe and Secure

In 1987, Dallas Semiconductor/Maxim introduced the DS5000T, an independently developed microcontroller based on the 8051 instruction and feature set. To offer new features and benefits, our engineers based the design on NV SRAM technology rather than EPROM. Leveraging its leadership in low-power technology, the memory partitioning and battery-backup circuitry was integrated directly onto the microcontroller die. The chief advantage of this system was speed. Writing to most nonvolatile memories is slow, but NV SRAM can read or write in a single cycle at high speed. This makes it ideal for high-speed, nonvolatile data-logging applications where data must be captured in real-time. When combined with an external SRAM and battery, the result is a complete microcontroller system with up to 64kB of nonvolatile program and data memory.

NV SRAM technology enables both data and program memory to be in-system reprogrammed, byte-by-byte, on the fly. In a standard microcontroller system, program memory needs either to be physically removed from the system (EPROM) or block erased, prohibiting memory access during the erasure (flash). NV SRAM-based microcontrollers can be quickly and easily programmed from a PC or device programmer by its serial port. A ROM-resident bootstrap loader downloads program and data directly to the microcontroller, allowing fast debugging or field upgrades.

The distinctive advantages of NV SRAM provide a new perspective for firmware security. Because the bootstrap loader completely controls the loading of the program into NV SRAM, we encrypted the address and data bus with a 40-bit or 80-bit encryption key. Any program or data loaded into the microcontroller is automatically encrypted before it is stored in the SRAM. This encryption thwarts hackers from stealing the program or data in the microcontroller. During the execution of an instruction, the microcontroller fetches an encrypted op code, decrypts and executes it in a single machine cycle. The use of NV SRAM allows read/write access at full speed,

with no delay for instruction decoding.

These security features culminated in the DS5250, a secure 8051 microcontroller used in financial terminals and payment systems worldwide. These tamper-reactive microcontrollers incorporate a 4 clock-per-machine cycle 8051 core as well as enhanced triple-DES encryption of their program memory. Security is further enhanced by the addition of intrusion-detection inputs and on-chip tamper sensors that automatically erase the memory as a tamper response. An integral microprobe shield prevents die tampering. Again, NV SRAM is best for high-security applications. Its high-speed write timing allows the microprocessor to erase confidential or sensitive data faster than any other type of memory.

The DS5250 is the only microcontroller capable of providing the highest level of security while executing every instruction at full speed.

More Speed, Less Power

Although the 8051 processing core remained static from its conception in the late 1970s through the 1980s, embedded systems did not. System designers continued to improve and upgrade their 8051-based applications by adding new software features and peripherals. This "feature creep" pushed the limits of available 8051 performance. Unfortunately, improvements to the 8051 core failed to keep pace, and it appeared that system designers would have to switch to another processor and perform a costly redesign to upgrade their systems.

The performance bottleneck was the 1970s vintage processing core of the 8051 microcontroller. Although external crystal speeds approached 40MHz, the traditional 8051 still required 12 clocks to execute a single machine cycle. Each instruction required from 1 to 4 machine cycles, meaning an instruction could take as little as 12 or as many as 48 oscillator clocks. Throughput was therefore limited to just over 3 MIPS, even while executing something as basic as a string of 1-cycle NOP instructions (**Figure 1**).

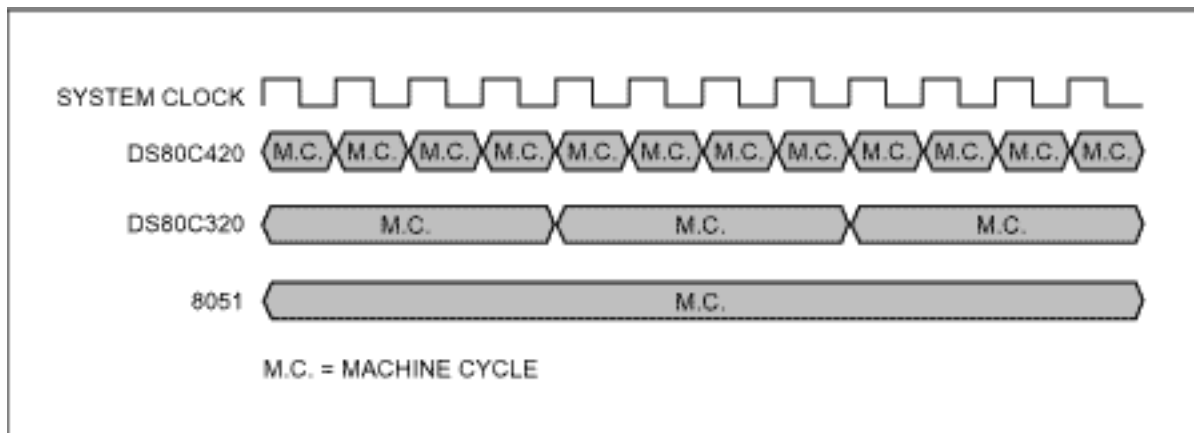


Figure 1. Reducing the number of clocks per machine cycle allows 3x the performance with the same instruction set (12 vs. 4 vs. 1 clock-per-machine cycle).

In 1991 we set out to reengineer the 8051 microcontroller for performance. The engineering team started by analyzing the traditional 8051 design. The original 12 clock-per-machine cycle architecture was terribly wasteful; most instructions were forced to execute dummy cycles. Engineers rebuilt the CPU from the ground up so that it only required 4 clocks-per-machine cycle instead of 12. A second internal data bus eliminated architectural bottlenecks that might hamper performance. High-powered I/O drivers increased switching speed during external memory operations. All the internal peripherals such as timers and serial ports ran at the faster clock speeds. But every step of the way, there was one absolute necessity—the instruction set had to remain op-code compatible with the 8051 instruction set.

The result? A new 8051-based microcontroller delivered triple the efficiency of the original 8051 core, with the majority of instructions operating three times faster for the same oscillator frequency. In addition to the increased efficiency of the core, the maximum external oscillator frequency of most devices increased to 33MHz or 40MHz. System designers previously held back by the older, much slower 8051 were able to upgrade their system to a maximum speed of 10 MIPS without software changes.

In addition to the speed improvement, the core redesign yielded another benefit: reduced power consumption. The laws of physics decree that power consumed by a digital circuit is proportional to the number of transistors

switched and the switching rate (frequency). Because the new core used fewer oscillator clocks per machine cycle, it consumed significantly less power per instruction per second than a traditional 8051.

Power management modes temporarily reduced the power consumption of the microcontroller through the use of a software-configurable internal clock divider. By reducing the machine cycle rate from 4 clocks-per-machine cycle to 64 or 256 clocks-per-machine cycle, power consumption was further reduced. A switchback feature let the device return to divide-by-4 mode upon receipt of an external interrupt or detection of a serial-port start bit. This allowed the device to remain in a low-power state but quickly resumed full-speed operation when needed.

Figure 2 shows the relative power consumption in different modes.

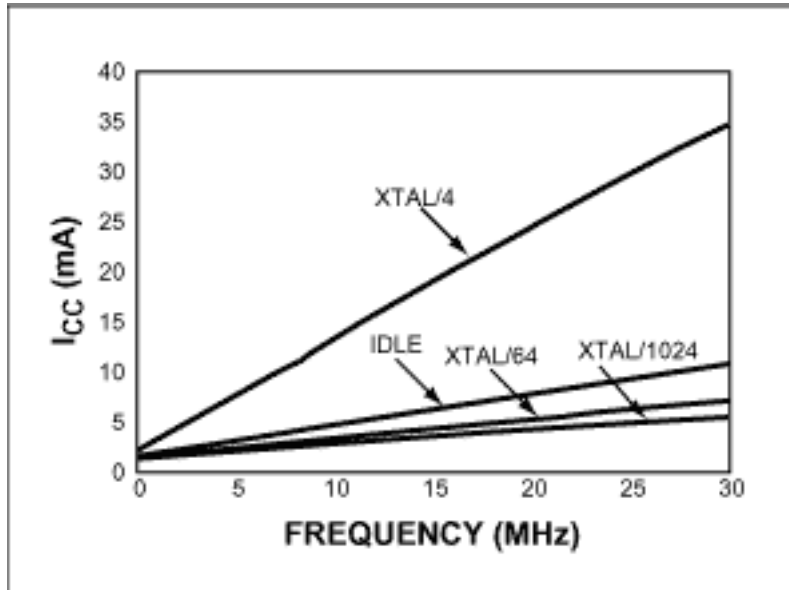


Figure 2. During periods of reduced activity, our power management modes consume less current than Idle mode but still allow CPU operation.

33 MIPS and Beyond

In 1997 Dallas Semiconductor/Maxim began designing a core for ultimate performance. Applications based on the 8051 were continuing to evolve, and customers clamored for even more performance. The engineering team set their sights on the performance peak: a microcontroller that executed the 8051 instruction set but used just 1 clock-per-machine cycle. Using a highly parallel architecture and a new fabrication process, a pin-for-pin, drop-in replacement 8051 was designed.

The result is the new DS89C430/DS89C450, ultra-high-performance 1 clock-per-machine cycle microcontrollers capable of executing up to 33 MIPS (**Figure 3**). These devices break previous performance barriers, providing 16-bit microcontroller performance with an 8-bit price. A variety of bus addressing modes allow users to fine-tune processor operation to the needs of the specific application design. But most importantly, they remain 100% 8051 instruction-set compatible and still execute existing 8051 applications faster than any other 8051-based microcontroller.

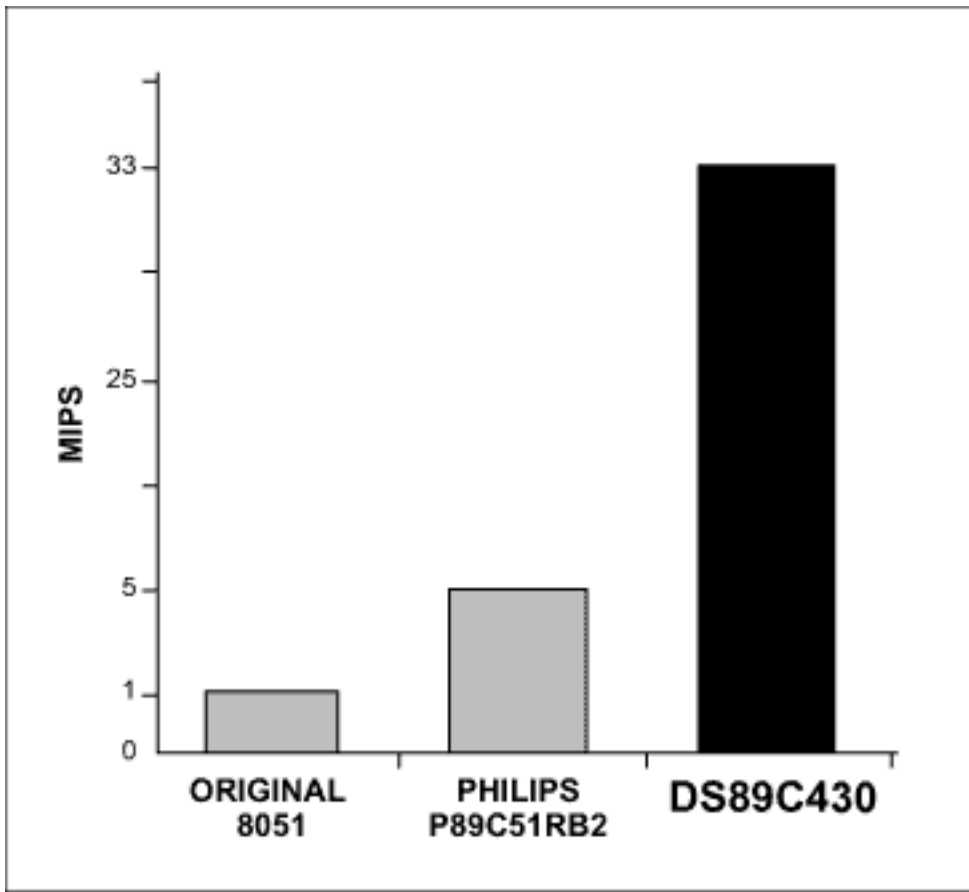


Figure 3. The DS89C430 outstrips the competition by clocking 33 MIPS.

In addition to the lightning-fast core, the DS89C430/DS89C450 incorporate up to 64kB of in-system programmable flash memory. The ROM-based bootstrap loader allows modification of the microcontroller code before, during, or after final assembly, offering maximum flexibility. Unlike other microcontrollers that use proprietary or nonstandard interfaces, the DS89C430 bootstrap loader is accessed by its serial port from a standard PC, using any terminal emulator software.

Fast Execution Times Beg for Bigger Program Sizes

The advantage of speed is lost if programmers cannot have sufficient memory address space for their expressions. The traditional 8051 used a 16-bit memory bus, restricting the memory range to 64kB. For some applications this limited memory range was sufficient. But as applications increased in code size and complexity, we realized applications needed a solution that maintained as much 8051 compatibility as possible.

Some designers found it possible to expand the addressing range by using bank-switching techniques. I/O lines double as address lines, expanding memory at the sacrifice of peripheral I/O. But this has two major shortcomings. Firstly, code must be segmented into 64kB or smaller chunks, which is a time-consuming task that must be redone each time the code is modified. Secondly, software routines must be written to manually switch the I/O lines to their appropriate state each time the code transitions between segments. The software overhead associated with these efforts degrades performance.

A better solution would implement a device with a larger address bus that addresses more memory. One such device, the DS80C400, has a 24-bit address bus that directly addresses 16MB of program memory and 16MB of data memory. This is done without requiring any new op codes in the 8051 instruction set. Two modes are provided. The first is a paged addressing mode, which incorporates advanced automatic bank switching, greatly speeding expanding memory access while remaining binary compliant with traditional 8051 compilers. The second contiguous mode allows transparent addressing of the entire 16MB memory map, and requires a compiler that provides the extra operand required for the 24-bit addresses. The larger address space allows faster access to larger programs, opening new possibilities such as large libraries of math functions, lookup tables, or even the Java™ virtual machine, supported by networked microcontrollers including the DS80C390 and DS80C400 that execute the Tiny Internet Interfaces (TINI™) runtime environment.

Every step of the way there was one absolute necessity—the instruction set had to remain op- code compatible with the 8051 instruction set.

Data Pointers Double Performance

Far-reaching improvements to all facets of the chip were necessary to avoid creating performance bottlenecks. A most important improvement involved accessing MOVX data memory. Manipulation of data memory on the original 8051 was a cumbersome affair. Accessing a single byte of MOVX memory required multiple cycles to load the single 16-bit data pointer before reading or writing the target address.

The DS89C430 remains 100% 8051 instruction-set compatible so it still executes existing 8051 applications faster than any other 8051-based microcontroller.

The inefficiencies multiplied if software needed to perform a block copy operation, which involved moving data from one MOVX memory location to another. The single data-pointer limitation forced it to double as both the source and destination address in a block copy operation. The operation on a traditional 8051 microcontroller has been a complicated, multistep procedure:

1. Load the source address into the data pointer.
2. Increment or modify the data pointer to the next datum.
3. Fetch the data from MOVX memory into the accumulator.
4. Save the modified source address to a storage register.
5. Load the destination address into the data pointer.
6. Increment or modify the data pointer to the next datum.
7. Write the data from the accumulator to MOVX memory.
8. Save the modified destination address to a storage register.

The larger address space allows faster access to larger programs such as the Java virtual machine, supported by the network microcontrollers.

One quickly notes that almost half the steps in the above procedure are dedicated to juggling the source and destination addresses in and out of the single data pointer, which impedes overall performance. The solution adds a second data pointer, creating dedicated registers for the source and destination. With the second data pointer, much of the data manipulation can be handled in hardware, reducing software overhead. The dual data pointers are individually addressable, and a dedicated data pointer select bit indicates which data pointer is the active data pointer during MOVX instructions. The same block copy operation performed with dual data pointers takes many fewer steps.

Perform initialization only once:

1. Initialize the source address into the first data pointer.
2. Initialize the destination address into the second data pointer.

Main loop:

1. Fetch the data into the accumulator.
2. Increment or modify the first data pointer to the next source datum.
3. Switch data pointer selector to second data pointer.
4. Write the data from the accumulator to MOVX memory.
5. Increment or modify the data pointer to the next datum.

Figure 4 shows how a 1000-byte block copy routine on a 33MHz DS89C430 takes 33% less execution time when dual data pointers eliminate the overhead associated with juggling a single data pointer. Some members of the high-speed and ultra-high-speed microcontroller families also have additional optional data pointer enhancements. The auto increment/decrement feature (denoted as AID in Figure 4) automatically increments or

decrements the data pointer following a MOVX-related instruction, eliminating the need for the INC DPTR instruction. The auto-toggle feature (denoted as TSL in Figure 4) automatically toggles the active data pointer following a MOVX-related instruction, eliminating the instruction that switches between data pointers. Figure 4 shows the relative execution times when all these features are considered together. Note that with all features enabled, the DS89C430 performs a 1000-byte block copy routine 103% faster than the original 8051 microprocessor.

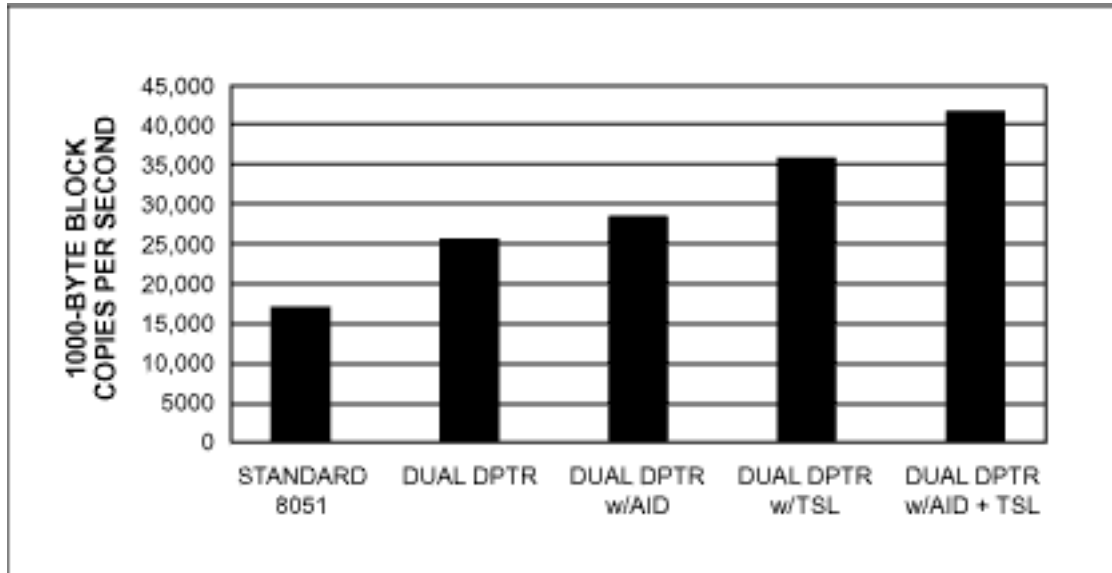


Figure 4. Dual data pointer enhancements improve the speed of block copy operations.

Looking Ahead

As applications demand more and more speed, Maxim works harder to exceed previous performance designs. Whether it is faster stack accesses, expanded addressing, or just raw processing speed, our microcontroller designs continue to meet the needs of embedded system designers.

But competitive designs demand more than just speed. More sophisticated applications require larger programs, so we are expanding our line of drop-in 8051 microcontrollers to include 64kB of flash memory. Our new-product pipeline has peripherals in design to increase the capabilities of their embedded systems while simultaneously reducing board space. The networked microcontrollers have advanced features including CAN, Ethernet, and 1-Wire® net connectivity for multitier networking. Secure microcontrollers have hardware-based math accelerators for public-key cryptography and support rapid zeroization of the keys as a tamper response. Mixed-signal microcontrollers perform real-world signal processing necessary to make better end equipment.

Our commitment to microcontroller performance extends beyond the 8051 core. Our new family of MAXQ® 16-bit RISC microcontrollers achieves a high performance-to-power ratio. The fundamental way this is achieved is through single-cycle instruction execution. Single-cycle instruction execution benefits the end user by increasing instruction bandwidth that leads to higher performance, and/or reduced power consumption made possible by the ability to reduce clock frequency. All MAXQ instructions execute in a single clock cycle except long jump/long call and certain extended register accesses. While many RISC microcontrollers claim to support single-cycle execution, this often applies to a small subset of instructions or addressing modes. With the MAXQ, single-cycle execution is the norm.

Additionally, the MAXQ architecture achieves increased clock-cycle utilization because it does not require an instruction pipeline (common to many RISC microcontrollers) to achieve single-cycle operation. The MAXQ instruction decode and execution hardware is so simple (and timing so fast) that these operations are moved into the same clock cycle as the program fetch itself, with minimal impact to the maximum operating frequency. To illustrate the benefit of eliminating the instruction pipeline, consider the generic RISC CPU that executes from a pipeline. When a program branch occurs, the CPU uses one or more clock cycles (depending upon pipeline depth) to divert program fetching to the target branch address and discards the instruction(s) already fetched. Clearly, using clock cycles to discard instructions, versus executing them, is wasteful and undesirable as it reduces performance and increases power consumption. While the operation is undesirable to the user, the clocks stolen by the CPU to reload the pipeline are an artifact of the architecture and are unavoidable. The MAXQ

architecture distinguishes itself from other 8-bit and 16-bit RISC microcontrollers by offering single-cycle execution without an instruction pipeline (and the wasted clock cycles that accompany it).

For more information about maximizing performance while minimizing power consumption, refer to the white paper "[Teaching Old Dogs New Tricks: Improving the Power Efficiency of 8051-Based Designs](#)".

Java is a trademark of Sun Microsystems, Inc.

TINI is a registered trademark of Dallas Semiconductor Corp.

1-Wire is a registered trademark of Dallas Semiconductor Corp.

MAXQ is a registered trademark of Maxim Integrated Products, Inc.

Dallas Semiconductor is a wholly owned subsidiary of Maxim Integrated Products, Inc.

Application Note 2035: www.maxim-ic.com/an2035

More Information

For technical questions and support: www.maxim-ic.com/support

For samples: www.maxim-ic.com/samples

Other questions and comments: www.maxim-ic.com/contact

Keep Me Informed

Preview new application notes in your areas of interest as soon as they are published. Subscribe to [EE-Mail - Application Notes](#) for weekly updates.

Related Parts

- DS2252T: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)
- DS5000T: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)
- DS5001FP: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)
- DS5001FP: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)
- DS5002FP: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
- DS5250: [QuickView](#) -- [Full \(PDF\) Data Sheet](#)
- DS80C310: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
- DS80C320: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
- DS80C323: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
- DS80C390: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
- DS80C400: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
- DS87C520: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
- DS87C530: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
- DS89C430: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)
- DS89C450: [QuickView](#) -- [Full \(PDF\) Data Sheet](#) -- [Free Samples](#)

AN2035, AN 2035, APP2035, Appnote2035, Appnote 2035

Copyright © by Maxim Integrated Products

Additional legal notices: www.maxim-ic.com/legal