Keywords: Dallas Semiconductor, DS80C320, DS87C520, 8051, high-speed microcontroller, programmable clock rate, clock speed, clock source, stop mode, idle mode, power management mode, burst mode

APPLICATION NOTE 1771

Microcontrollers Improve Power Efficiency of 8051-Based Designs

By: Kevin Self
Oct 21, 2002

Abstract: High-speed microcontrollers are widely used in portable equipment. With the proper use of a power management mode, the designer can optimize the life of batteries used in today's new systems. This application note looks at ways to reduce power consumption by choosing efficient clock source and clock speed, the use of stop mode, idle mode and burst mode. The Maxim High-Speed Microcontroller and Ultra-High-Speed Flash Microcontroller families provide the user with advanced features for battery-backed applications.

Portable products continue to advance in features and capabilities. Customers are demanding more performance out of their products, which requires greater computing power. At the same time, they want products with less power consumption. At the heart of these competing demands is the microcontroller, which is typically one of the largest power consumers in portable instruments.

Low-power processors exist, they are often limited in performance. The high-speed microcontroller family from Maxim is a good compromise of power and performance. It is based on the 8051 architecture, one of the most popular microcontrollers in the world. Designers prize its ease of use, rich I/O structure, and wide acceptance. Its prevalence has carried over into the portable arena, where it has found a home in many applications.

This article addresses approaches to minimizing power consumption using 8051 controllers, with emphasis on new architectural improvements that can extend the battery life of high-performance 8051-based designs. Integrating peripherals on-chip and selecting the proper clock source are discussed as ways to reduce power consumption. Software techniques for power conservation are presented, as well as a method of reducing power consumption in systems that use Stop mode.

Clock Speed

The most important factor in determining power consumption in any microcontroller design is the system clock speed. The power consumption of complementary metal oxide semiconductor (CMOS) devices is directly proportional to clock speed. It follows, then, that it is beneficial from a power standpoint to run a processor at the slowest speed possible.

Figure 1 shows a typical power curve for a generic 8051 microcontroller, a relationship known to all portable system designers. In general, the current vs. frequency characteristic is linear, with a DC offset.

This quiescent current is caused by static circuitry on-chip, such as comparators, operational amplifiers, etc. While this number is typically small (< 1mA), it is a constant drain that needs to be considered.

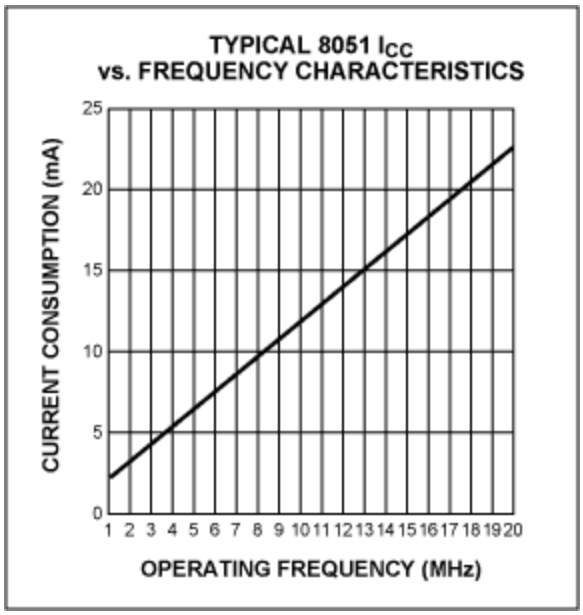

Figure 1. Typical power curve for generic 8051 microcontroller.

Any power-conscious design will attempt to run as slowly as possible. Determination of the minimum system frequency, and hence minimum power consumption, is dependent on a number of factors, including desired performance and interrupt latency. Whatever criteria are used, however, the end goal is the same: match the operating frequency of the device as closely as possible to the requirements of the application.

High-Speed Core

The most direct approach to decreasing power consumption of an 8051-based design is to improve the efficiency of the microcontroller. The original design of the 8051 was based on a 12-clock, 2-fetch-per-machine cycle architecture. The high-speed microcontroller family, however, uses a 4- or 1-clock per machine cycle core. It is more computationally efficient and requires fewer clock cycles to execute an instruction, resulting in faster execution times and increased maximum clock rates.

Although the advantages of a high-speed core are usually considered in terms of performance, they have important implications for power consumption as well. When the instruction execution of the processor is optimized, it takes less time to accomplish the same task. Many portable products operate in a burst mode, characterized by brief periods of activity, such as recording environmental data or scanning a bar code, followed by long period of inactivity. Reducing the time that the processor must be active achieves a corresponding reduction in energy consumption.

Another consequence of this improved efficiency is that comparable performance can be achieved while reducing clock speed. If a redesigned core uses 4 clocks per cycle rather than 12, this means that the same work can be accomplished at a reduced crystal speed. Because power consumption is directly proportional to crystal speed, power consumption can be reduced without sacrificing performance.

Figure 2 shows the power consumption of three microcontrollers performing the same task with the same level of performance. Two microcontrollers are standard 80C3x derivatives operating at 12 external

clocks per machine cycle, while the second is a DS80C320 microcontroller operating at 4 clocks per machine cycle. Current consumption was measured for all devices and then compared, assuming a conservative performance improvement of 250% (2.5x) for the DS80C320. As is evident from the figure, the reduced clock per cycle core exhibits a significant current reduction at the same throughput, most notably at high performance levels.

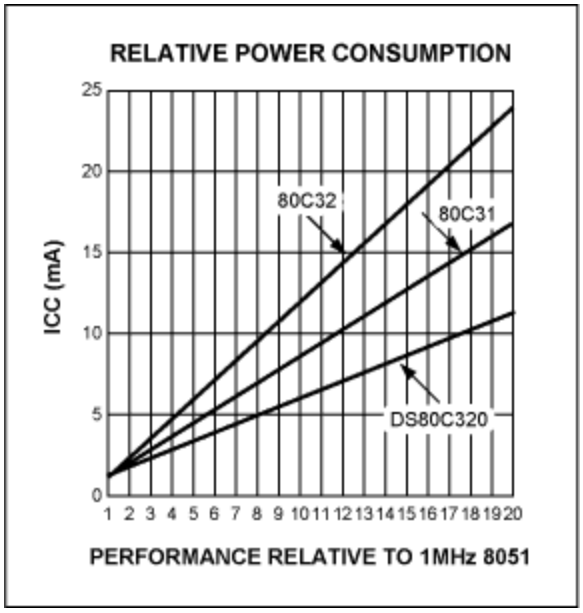

Figure 2. Reduced clock cycle core uses less current for same throughout.

Integration

Integrating peripherals on-chip is a method of power conservation. When driving a signal off-chip, the generating device must contend with the switching power required to drive the external loads and any DC losses. Switching power, PSW, is the power consumed when a digital signal changes. The switching power can be approximated as follows:

$$P_{SW} \alpha\ CV^2/T \qquad (1)$$

where C is the lumped capacitance of the receiving gate input buffer and the interconnection between the two gates, and T is the clock period of the signal. A typical input capacitance for a CMOS input is 10pF. Although it is difficult to calculate an exact value of the switching power for a system, it is obvious that each additional external load or pin that the microcontroller must drive consumes additional power.

Microcontroller-based systems typically use a number of peripherals. These range from external UARTs and power-on reset circuitry to watchdog timers. One of the strengths of the 8051 product family is the large number of peripheral functions that are available on-chip. In addition to simplifying a design by eliminating components, integrated peripherals also can reduce power consumption. One can assume that the core functionality of any peripheral consumes the same amount of power whether located internal or external to the processor. Locating a peripheral on-chip, however, will eliminate the switching power losses associated with driving an external bus.

Internal Program Memory

Another 8051 feature that is not commonly perceived as a peripheral is program memory. All 8051 derivatives incorporate various amounts of on-chip program memory. This is desired by many system designers as a method of reducing the component count and board area, but it also improves battery life in portable systems. As mentioned previously, this will reduce power consumption by eliminating the need to drive an external bus. There is an additional power savings when using on-chip memory. The 8051 architecture requires the use of a 74373-type latch to demultiplex the lower byte of address. **Figure 3** compares the use of internal vs. external program memory. The first uses a DS87C520 High-Speed Microcontroller with a 74AC573 latch and a 27C256 EPROM with an access time of 70ns. The second system uses the same microcontroller, but operating from internal memory. Both systems are operating at 11.0592MHz, executing a short, generic program. From the figure, it is apparent that as much as 49mA can be saved at high frequencies by eliminating the external EPROM and latch from the system.

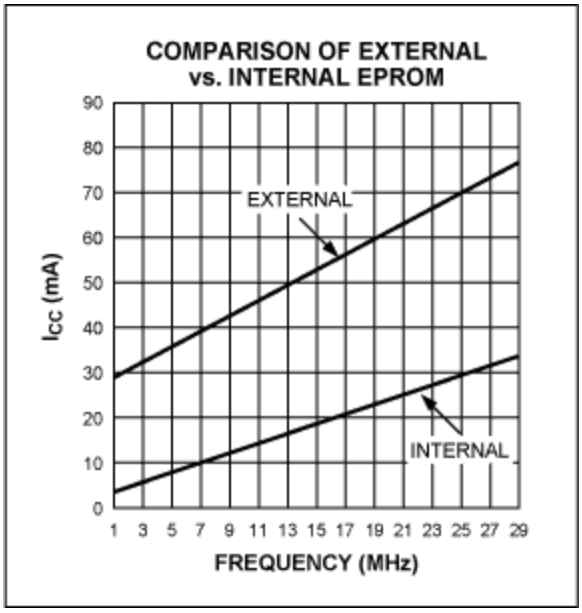

Figure 3. Using internal memory significantly reduces current consumption.

Internal Data Memory

As previously mentioned, the use of on-chip memory instead of external RAM will save power. The enlarged scratchpad of the 80C32 derivatives (256 bytes) is sufficient for stack operations and some data storage in small programs, eliminating the need for external RAM.

For designs requiring more data memory or needing to implement an external stack, however, additional SRAM may be required. Although low-power SRAMs are available, their power consumption must also include that associated with a 74373 series latch as well as capacitive losses driving the external bus. This can be mitigated by using devices with expanded on-chip RAM. **Figure 4** shows the power consumption of two systems using SRAM mapped into the 8051 MOVX data space. The first uses a DS87C520 high-speed microcontroller with a 74AC573 latch and a DS2064 SRAM. The second system uses the same microcontroller but uses 1Kbyte of internal MOVX data memory. Both microcontrollers are operating at 11.0592MHz, executing a short, generic program that reads and writes to MOVX data memory. From the figure, it is apparent that as much as 9mA can be saved at high frequencies by eliminating the external SRAM and latch from the system.

Clock Source

Another critical system component from a power standpoint is the clock source. Standard 8051 designs typically either excite an external quartz crystal with an internal oscillator amplifier or use an external crystal oscillator. If an external crystal oscillator is used, the waveform of the clock can affect power consumption. The input stage of the XTAL1 pin, used to drive external clock signals into an 8051, typically employs complementary drivers. As the input clock transitions between high and low, the drivers will momentarily both be on, causing a significant current rush. With a square wave, the transition between high and low states is almost instantaneous, and the time in which both devices are on is minimized. A waveform with a slower rise and fall time, such as a sine or triangle wave, will take longer to complete the transition and will spend more time with both drivers on. This will increase current and power consumption.

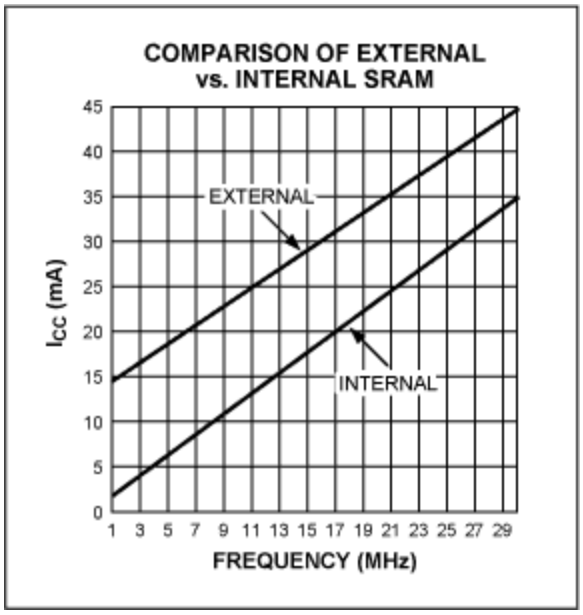

Figure 4. Eliminating external SRAM and latch saves power.

Figure 5 shows the relationship of current consumption to the waveform shape. The clock source was a programmable waveform generator, with the ability to output sine, triangle, or square waves. The current consumption shown is an average of four devices, both traditional and high-speed core. The comparison shows that current consumption is directly proportional to the rise (and fall) time of the clock waveform. The triangle wave has the lowest slope and the square wave the highest slope. The square wave averages 0.75mA less than the triangle wave. This implies that current consumption in external clock oscillator designs can be reduced by using oscillators with fast rise and fall times. This becomes even more important at lower frequencies, when the device spends more time in transition.

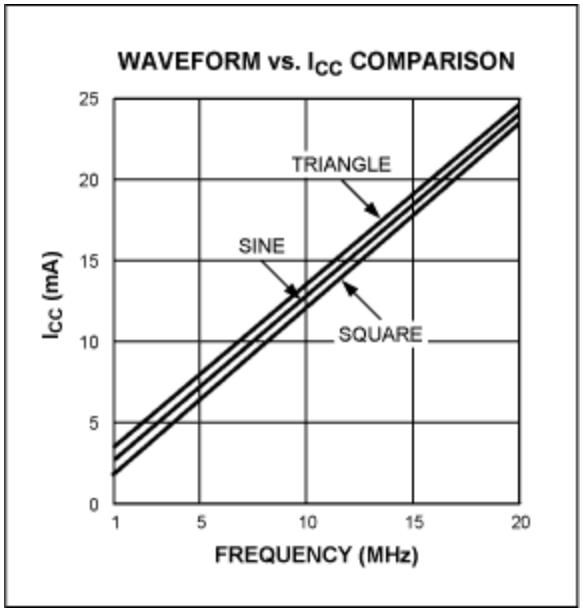

Figure 5. Oscillator waveforms with sharper edges reduce power consumption.

Some 8051 derivatives incorporate an on-chip internal ring oscillator. This is typically a chain of inverters that propagates a pulse around it. This provides an internal clock source of approximately 2MHz to 4MHz, capable of operating the device. Because it does not require the use of a crystal, it is a very-low-power clock source. Characterization of a DS87C520 high-speed microcontroller shows that operation from the ring oscillator can deliver performance comparable to a 7MHz 8051 at approximately 3.6mA. Although ring oscillators do not exhibit the stability of piezoelectric crystals, their low power and negligible power-on delay make them a significant factor in a power management scheme.

Clock Management

As mentioned previously, the operating frequency of the microcontroller is the single largest factor affecting the power consumption of the device. Although the system clock frequency is primarily a hardware function, the 8051 has the ability to exercise limited control over it. These methods rely on slowing or halting the internal operating frequency of all or part of the device. Traditional 8051 architecture has used two clock control modes: Idle and Stop.

Improving Stop Mode

Stop mode is the lowest power state available to 8051 designers. In this mode, the internal crystal amplifier is stopped, halting operation of the device. Exiting from Stop mode is typically initiated by an external reset. Some variants also support exiting from Stop mode using external interrupts.

One of the disadvantages associated with Stop mode is the power consumed during the "dead time" while the crystal is resuming operation. A crystal oscillator relies on the motion of a quartz crystal for its operation. Physical limitations require a finite amount of time for the crystal oscillation to achieve sufficient amplitude for device operation. This warm-up period is encountered regardless of whether the clock source is an external crystal and internal crystal amplifier, or whether an external crystal oscillator is used. This time can be on the order of 3ms to 12ms, depending on the characteristics of the crystal and associated amplifier.

The effect of the warm-up period on power consumption is that while the device is not performing any

useful work during this period, it is still consuming power. This can become significant if the device is entering and exiting Stop mode frequently, or is exiting Stop mode to perform short tasks. In fact, if the task is very short (< 5ms), the crystal restart period can consume more power than the task itself. If a ring oscillator is used to perform a "quick start" from Stop mode, this delay can be avoided. This will greatly reduce the amount of power when out of Stop mode.

Figure 6 shows the operation of two systems exiting from Stop mode and performing a short task. One device incorporates an internal ring oscillator, and the other uses a traditional external crystal. The device without the ring oscillator must endure a crystal warm-up period. During this time the device continues to consume power, but no useful work is done. The second device is a DS87C520 high-speed microcontroller that incorporates an internal ring oscillator. This allows the device to resume operation immediately when exiting Stop mode. In this example, the routine to be executed is less than 4ms in duration at approximately 2MHz. As can be seen in the figure, energy consumption can be greatly reduced by using a ring oscillator to perform short tasks when exiting Stop mode.

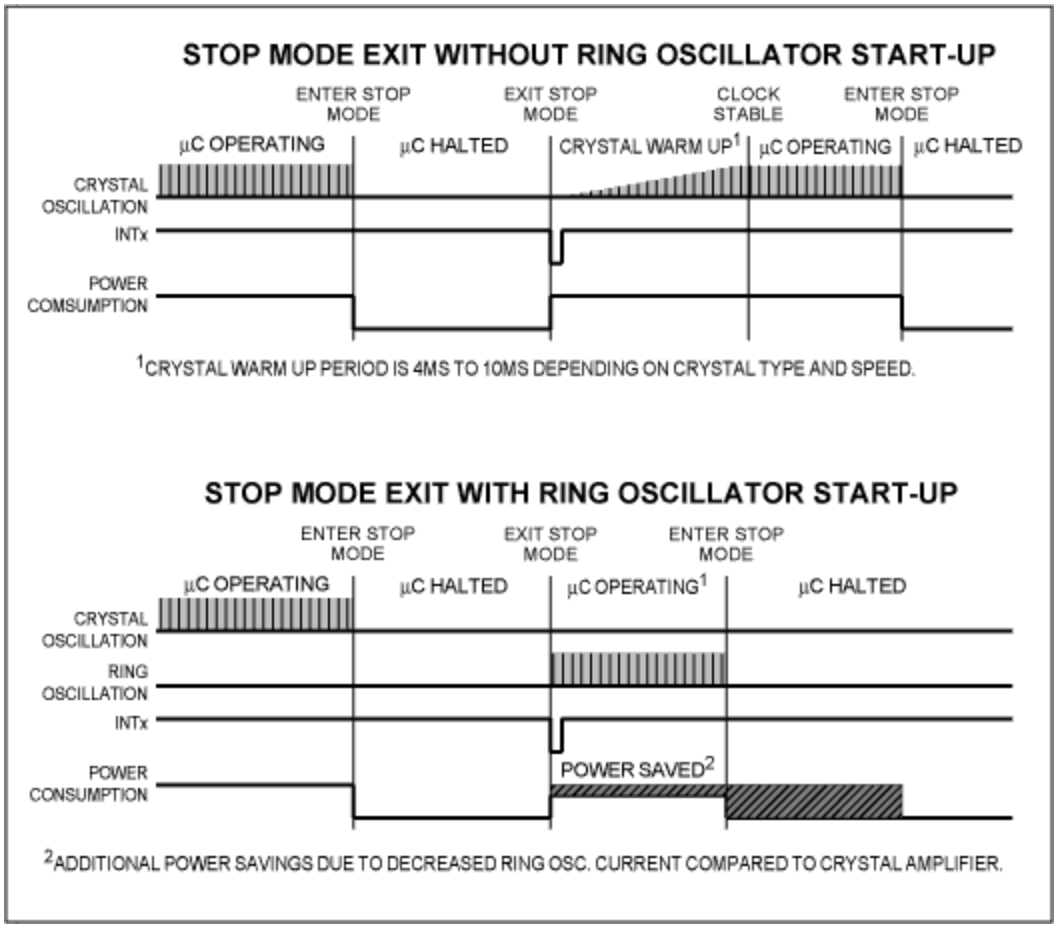

Figure 6. Comparison of stop mode exit with and without ring.

In some applications, the stability of a crystal oscillator may be required shortly after exiting Stop mode. In this case, the ring oscillator can still be advantageous. Immediately upon exiting Stop mode, the device should restart the crystal oscillator. The device can then initialize any data or registers necessary while the crystal is still warming up. Most high-speed microcontrollers incorporate a status bit that indicates whether the crystal oscillator has stabilized or not. Once the initialization routine for the crystal oscillator code is complete, the software can poll the status bit to determine when the high-precision

timing operation can commence.

Another method of improving Stop mode efficiency is to use an interrupt to exit instead of reset. This allows the processor to resume operation immediately with the instruction following the setting of the STOP bit, instead of having to restart from the reset vector. This eliminates the need to determine the cause of the reset and allows the processor to begin performing useful work in less time.

Idle Mode

Idle mode is the second clock management mode used in original 8051 architecture. This mode halts operation of the CPU, but keeps the on-chip, general-purpose timers operational. In a power-sensitive application, these timers are used to periodically wake the device to perform a task or to poll if a task should be performed.

Because standard 8051 timers are limited to 16 bits, this allows a maximum timeout period of 31ms at a clock rate of 16MHz. If longer periods are needed, multiple timer overflows will be required. This will consume additional power because the device must resume full operation occasionally to increment a counter but not perform any useful work.

For longer periods, use an internal timer with a longer period. Some 8051 derivatives incorporate a watchdog timer, which can also be used to awaken the device. Watchdog timers can be programmed for long time-out periods, on the order of 226 clock cycles. This would allow a maximum timeout period of 4.2s at 16MHz. As an example, assume an application wishes to awake from a low power state every 3s to perform a task. If the internal timers are used to time the operation, the device would have to exit Idle mode 96 times without doing useful work. If a watchdog timer with a long timeout is used, the device would only exit Idle mode once, perform the task, and return to the low-power state.

One other option is to use a microcontroller with a real-time clock (RTC). The DS87C530 high-speed microcontroller incorporates an RTC capable of generating an alarm period as long as 24 hours. The internal interrupt generated by this alarm can cause the device to exit Idle or Stop mode. Using an RTC to exit Stop mode is the most efficient way to suspend device operation for long periods of time.

Power Management Modes

Although Idle mode reduces power consumption by halting program execution, the internal timers continue to operate at the external clock frequency. This consumes a considerable amount of power, considering the timers are basically operating in a "standby" capacity.

A better approach is to reduce the clock rate of the entire device. This can be done with an internal clock divisor, which divides the external clock frequency before it enters the CPU. Such a scheme has been implemented in the DS87C520 High-Speed Microcontroller. This device employs two clock divisor functions: Power Management Mode 1, which divides the input clock source by 64, and Power Management Mode 2, which divides the input clock source by 1024. These modes are enabled by setting the appropriate bits in a Special Function Register.

Figure 7 shows a comparison of the clock divisor and clock control modes on the DS87C520 High-Speed Microcontroller. The figure contrasts the current consumption in full speed (divide by 4), Power Management Mode 1 (divide by 64), Power Management Mode 2 (divide by 1024), Idle mode, and Stop Mode. As expected, Stop mode draws the least current, because all internal clocking is halted. One interesting result of this comparison is that the two power management modes draw less current than Idle mode. This not only allows the device to conserve power, but permits it to function continuously at a low level of operation. In the traditional 8051 architecture, performing any type of CPU activity was "all or

nothing." A device was forced to operate constantly at the highest performance level, even if high performance was only required for short periods. This unnecessarily increased power consumption. The use of power management modes (PMM) allows the device (and system) to match its power consumption to the level of performance needed.

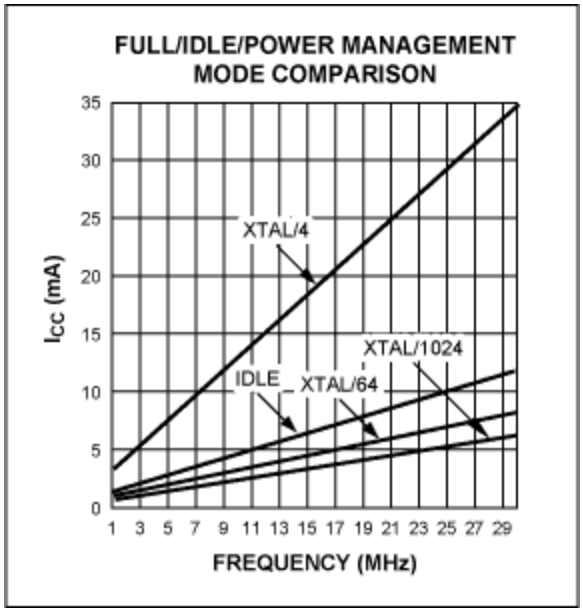

Figure 7. Full/Idle/Power Management Mode Comparison.

Using Interrupts with PMM

One possible consequence of using an internal clock divisor is that interrupt latencies may be greatly increased. In addition, slowing the internal timers would affect the ability of the 8051 serial ports to generate or synchronize with a standard baud rate. This could seriously interfere with the device's ability to respond to external stimuli. One solution is to incorporate a feature that automatically restores the device to full operation when an external interrupt or serial port activity is recognized. Such a mechanism has been implemented in the DS87C520. That device's Switchback feature allows the device to respond quickly to external interrupts. As soon as the interrupt is acknowledged, the device will automatically switch back to full speed (divide by 4) without software intervention.

The serial ports operate in a similar fashion. Upon receipt of a falling edge (start bit) on a serial port reception pin, the device will automatically switch back to full speed (divide by 4). Because this happens immediately at the start of the transmission, the device will be at full speed to correctly receive the rest of the transmission. With a traditional 8051 architecture, the only way to use the serial ports in a low-power configuration was with the Idle mode. The use of Power Management Modes provides a lower power alternative.

Improving Burst Mode Operation

A common mode of operation in power-conscious designs is to have the system awake from Stop mode, perform a burst of activity, and then return to Stop mode. One way to decrease power consumption in such a system is to increase the operating frequency. At first, this may seem counter-intuitive. For the time in which the device is operating, it will consume more power than a system operating at a lower frequency. The quiescent current consumed while the system is operating, however, is not a function of frequency. In the final system design, energy is typically evaluated to determine battery life. This

distinction is important when evaluating a high-performance microcontroller because it combines the concept of time and processing power. If the product of power and time is smaller for a given system, then it will require less energy, regardless of the individual terms. In many instances, it can be shown that a high-speed microcontroller can actually reduce energy consumption by running fast for short periods, as opposed to running more slowly for a longer period of time.

This can be demonstrated by re-examining Figure 7. Assume that upon resuming from Stop mode, a DS87C520 must read an I/O port, perform a mathematical computation, and output the result to another port, requiring 500 machine cycles of CPU time. From the figure, the current consumption is 12.4mA at 10MHz, and 34.6mA at 30MHz. Table 1 summarizes the results of the task at both speeds. As can be seen from the table, operation at 30MHz is the most energy efficient, with a more than 6% reduction in energy consumed.

Hurry Up and Wait

In many applications, the time out of Stop mode is not entirely speed-dependent. Frequently, a device will have to access a peripheral with a fixed response time, such as an A/D converter or thermostat. In such a case, the microcontroller will have a burst of activity, typically to initiate a process, followed by a period of little or no activity. In such a case, a combination of power conservation techniques can be effective.

A practical example can illustrate the advantage of a high-speed microcontroller with PMM in such a system. Suppose that the DS87C520 is interfaced to a DS1620 digital thermometer and thermostat. This device is addressed serially using a standard 8051 serial port operating in mode 0. A host processor will occasionally wake the DS87C520 from Stop mode using an external interrupt and request that it read the temperature from the DS1620. After the data has been retrieved, the DS87C520 will store it in internal memory to be transmitted later. The DS1620 functions similarly to many A/D converters: a command is issued to start a conversion, then there is a delay while the conversion is completed, then the data is shifted out. In the case of the DS1620, conversion time is approximately 1 second. The device is polled to determine when the conversion is complete. The DS87C520 is well suited to such a task because it can perform the initialization and computation functions quickly. The device can then place itself in PMM while waiting for the conversion to complete. In a conventional 8051, Idle mode would be used to place the conventional 8051 in a low-power state once the conversion was started. The use of this mode allows an internal 16-bit timer to measure the conversion period. Operating at 16MHz, the conventional 8051 could require exiting Idle mode as many as 32 times before the conversion was complete.

Table 1. Energy Consumed vs. Processor Speed for a 500 Machine Cycle Task

Clock Frequency	Machine Cycle Period	Machine Cycles Required	Total Time	I_{CC}	Current Time Product
10MHz	400ns	500	200s	12.41mA	2.48As
30MHz	133ns	500	66.5s	34.66mA	2.30As

One further improvement can be made in this example. Because the DS1620 is addressed as a synchronous device, high precision timing operations are not required. As a result, the microcontroller can operate from the ring oscillator while initiating and when reading the results of the conversion. This results in further power savings by eliminating the dead time needed to stabilize an external crystal.

Figure 8 illustrates the operation of two 8051 systems implementing the "hurry up and wait" schemes mentioned above. As can be seen from the figure, there is a significant power savings during program execution following the exit from Stop Mode. In addition to the power saved by using PMM2 instead of Idle mode, the elimination of the crystal warm-up period means that the routine can return to Stop mode

more quickly. Running from the ring oscillator during the 1s conversion delay slows the processor speed even more, allowing for greater power savings.

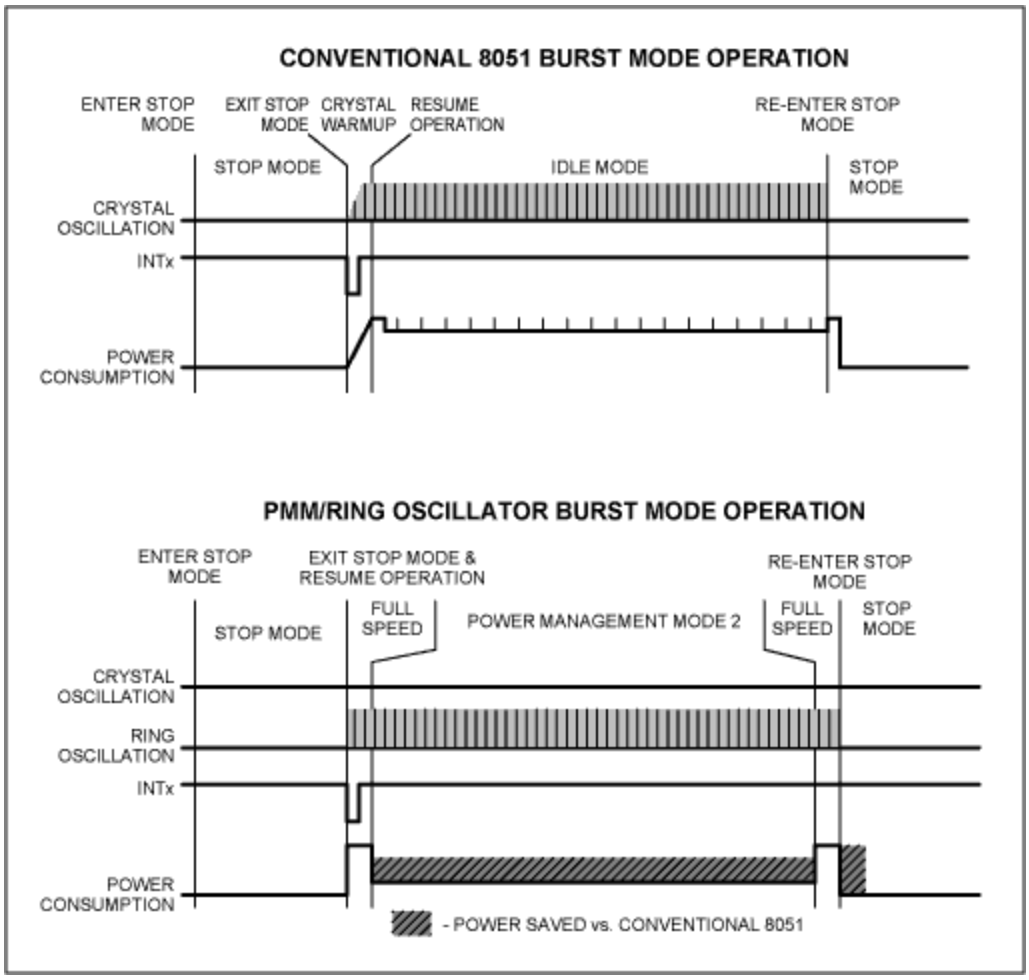

Figure 8. Implementing an 8051 "Hurry Up and Wait" scheme.

Summary

The 8051 microcontroller family remains one of the most popular processors in the world. Its ease of use and relatively high performance make it ideal for many applications, including portable and handheld products. The introduction of Maxim high-speed microcontrollers allows a way for existing 8051 designs to improve their power efficiency without a costly redesign.

The benefits of the high-speed microcontrollers that reduce power consumption can be summarized as follows:

- A high performance CPU allows the processor clock to be slowed, resulting in the same level of performance at less power. Alternatively, the performance of an existing system can be increased without increasing power consumption.

- The high-speed microcontroller incorporates features such as watchdog timers, additional UARTs, and precision reset circuits. External components consume more power.

The introduction of two new low-power modes provides a low-power alternative to the Idle mode. In addition to reducing current consumption, power management modes such as those used in the DS87C520 allow the processor to perform tasks such as polling while in a low state. Conventional 8051 architectures require the processor to operate at the maximum clock rate, even if only minimal processing power is required.

- The benefits of a programmable clock rate and high-performance core can be combined with the Stop mode to greatly reduce power consumption. Examples have been presented that show how energy consumption can be reduced by matching the clock rate of the device to the desired performance level.

Related Parts		
DS1620	Digital Thermometer and Thermostat	Free Samples
DS80C320	High-Speed/Low-Power Microcontrollers	Free Samples
DS87C520	EPROM/ROM High-Speed Microcontrollers	Free Samples

More Information
For Technical Support: http://www.maximintegrated.com/support
For Samples: http://www.maximintegrated.com/samples
Other Questions and Comments: http://www.maximintegrated.com/contact

Application Note 1771: http://www.maximintegrated.com/an1771
APPLICATION NOTE 1771, AN1771, AN 1771, APP1771, Appnote1771, Appnote 1771