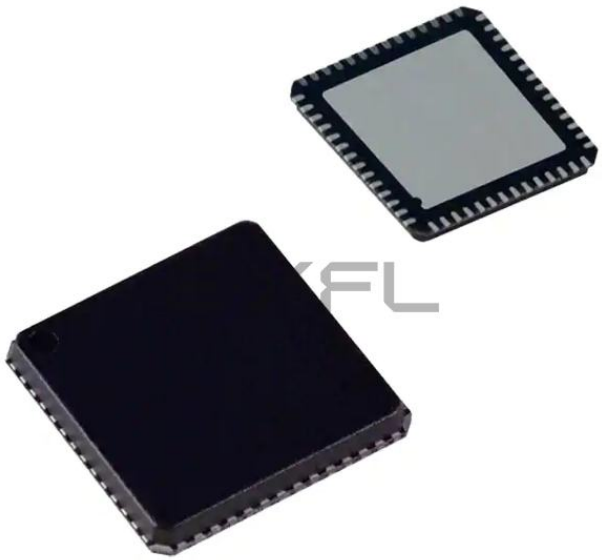

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8052
Core Size	8-Bit
Speed	8.38MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, PSM, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	62KB (62K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	56-VFQFN Exposed Pad, CSP
Supplier Device Package	56-LFCSP-VQ (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc842bcpz62-3

SPECIFICATIONS[1]

Table 1. AV_{DD} = DV_{DD} = 2.7 V to 3.6 V or 4.75 V to 5.25 V; V_{REF} = 2.5 V internal reference, f_{CORE} = 16.78 MHz @ 5 V 8.38 MHz @ 3 V; all specifications T_A = T_{MIN} to T_{MAX}, unless otherwise noted

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
ADC CHANNEL SPECIFICATIONS				
DC ACCURACY[2, 3]				f_{SAMPLE} = 120 kHz, see the Typical Performance Characteristics for typical performance at other values of f_{SAMPLE}
Resolution	12	12	Bits	
Integral Nonlinearity	±1	±1	LSB max	2.5 V internal reference
	±0.3	±0.3	LSB typ	
Differential Nonlinearity	+1/−0.9	+1/−0.9	LSB max	2.5 V internal reference
	±0.3	±0.3	LSB typ	
Integral Nonlinearity[4]	±2	±1.5	LSB max	1 V external reference
Differential Nonlinearity[4]	+1.5/−0.9	+1.5/−0.9	LSB max	1 V external reference
Code Distribution	1	1	LSB typ	ADC input is a dc voltage
CALIBRATED ENDPOINT ERRORS[5, 6]				
Offset Error	±3	±2	LSB max	
Offset Error Match	±1	±1	LSB typ	
Gain Error	±3	±2	LSB max	
Gain Error Match	±1	±1	LSB typ	
DYNAMIC PERFORMANCE				f_{IN} = 10 kHz sine wave f_{SAMPLE} = 120 kHz
Signal-to-Noise Ratio (SNR)[7]	71	71	dB typ	
Total Harmonic Distortion (THD)	−85	−85	dB typ	
Peak Harmonic or Spurious Noise	−85	−85	dB typ	
Channel-to-Channel Crosstalk[8]	−80	−80	dB typ	
ANALOG INPUT				
Input Voltage Range	0 to V_{REF}	0 to V_{REF}	V	
Leakage Current	±1	±1	µA max	
Input Capacitance	32	32	pF typ	
TEMPERATURE SENSOR[9]				
Voltage Output at 25°C	700	700	mV typ	
Voltage TC	−1.4	−1.4	mV/°C typ	
Accuracy	±1.5	±1.5	°C typ	Internal/External 2.5 V V_{REF}
DAC CHANNEL SPECIFICATIONS **Internal Buffer Enabled** ADuC841/ADuC842 **Only**				DAC load to AGND R_L = 10 kΩ, C_L = 100 pF
DC ACCURACY[10]				
Resolution	12	12	Bits	
Relative Accuracy	±3	±3	LSB typ	
Differential Nonlinearity[11]	−1	−1	LSB max	Guaranteed 12-bit monotonic
	±1/2	±1/2	LSB typ	
Offset Error	±50	±50	mV max	V_{REF} range
Gain Error	±1	±1	% max	AV_{DD} range
	±1	±1	% typ	V_{REF} range
Gain Error Mismatch	0.5	0.5	% typ	% of full-scale on DAC1
ANALOG OUTPUTS				
Voltage Range_0	0 to V_{REF}	0 to V_{REF}	V typ	DAC V_{REF} = 2.5 V
Voltage Range_1	0 to V_{DD}	0 to V_{DD}	V typ	DAC V_{REF} = V_{DD}
Output Impedance	0.5	0.5	Ω typ	

[1] Temperature Range –40°C to +85°C.
[2] ADC linearity is guaranteed during normal MicroConverter core operation.
[3] ADC LSB size = $V_{REF}/2^{12}$, that is, for internal V_{REF} = 2.5 V, 1 LSB = 610 µV, and for external V_{REF} = 1 V, 1 LSB = 244 µV.
[4] These numbers are not production tested but are supported by design and/or characterization data on production release.
[5] Offset and gain error and offset and gain error match are measured after factory calibration.
[6] Based on external ADC system components, the user may need to execute a system calibration to remove additional external channel errors to achieve these specifications.
[7] SNR calculation includes distortion and noise components.
[8] Channel-to-channel crosstalk is measured on adjacent channels.
[9] The temperature monitor gives a measure of the die temperature directly; air temperature can be inferred from this result.
[10] DAC linearity is calculated using:
Reduced code range of 100 to 4095, 0 V to V_{REF} range.
Reduced code range of 100 to 3945, 0 V to V_{DD} range.
DAC output load = 10 kΩ and 100 pF.
[11] DAC differential nonlinearity specified on 0 V to V_{REF} and 0 V to V_{DD} ranges.
[12] DAC specification for output impedance in the unbuffered case depends on DAC code.
[13] DAC specifications for I_{SINK}, voltage output settling time, and digital-to-analog glitch energy depend on external buffer implementation in unbuffered mode. DAC in unbuffered mode tested with OP270 external buffer, which has a low input leakage current.
[14] Measured with C_{REF} pin decoupled with 0.47 µF capacitor to ground. Power-up time for the internal reference is determined by the value of the decoupling capacitor chosen for the C_{REF} pin.
[15] When using an external reference device, the internal band gap reference input can be bypassed by setting the ADCCON1.6 bit.
[16] Flash/EE memory reliability characteristics apply to both the Flash/EE program memory and the Flash/EE data memory.
[17] Endurance is qualified to 100,000 cycles as per JEDEC Std. 22 method A117 and measured at –40°C, +25°C, and +85°C. Typical endurance at 25°C is 700,000 cycles.
[18] Retention lifetime equivalent at junction temperature (T_J) = 55°C as per JEDEC Std. 22 method A117. Retention lifetime based on an activation energy of 0.6 eV derates with junction temperature as shown in Figure 38 in the Flash/EE Memory Reliability section.
[19] Power supply current consumption is measured in normal, idle, and power-down modes under the following conditions:

Normal Mode:	Reset = 0.4 V, digital I/O pins = open circuit, Core Clk changed via CD bits in PLLCON (ADuC842/ADuC843), core executing internal software loop.
Idle Mode:	Reset = 0.4 V, digital I/O pins = open circuit, Core Clk changed via CD bits in PLLCON (ADuC842/ADuC843), PCON.0 = 1, core execution suspended in idle mode.
Power-Down Mode:	Reset = 0.4 V, all Port 0 pins = 0.4 V, All other digital I/O and Port 1 pins are open circuit, Core Clk changed via CD bits in PLLCON (ADuC842/ADuC843), PCON.0 = 1, core execution suspended in power-down mode, OSC turned on or off via OSC_PD bit (PLLCON.7) in PLLCON SFR (ADuC842/ADuC843).

[20] DV_{DD} power supply current increases typically by 3 mA (3 V operation) and 10 mA (5 V operation) during a Flash/EE memory program or erase cycle.
[21] Power supply currents are production tested at 5.25 V and 3.3 V for a 5 V and 3 V part, respectively.

Pin No.	Mnemonic	Type[1]	Description
33	P2.3/A11/A19	I/O	Input/Output Port 2 (P2.3). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A11). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A19). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
34	XTAL1	I	Input to the Inverting Oscillator Amplifier.
35	XTAL2	O	Output of the Inverting Oscillator Amplifier.
39	P2.4/A12/A20	I/O	Input/Output Port 2 (P2.4). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A12). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A20). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
40	P2.5/A13/A21	I/O	Input/Output Port 2 (P2.5). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A13). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A21). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
41	P2.6/A14/A22	I/O	Input/Output Port 2 (P2.6). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A14). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A22). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
42	P2.7/A15/A23	I/O	Input/Output Port 2 (P2.7). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A15). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A23). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
43	$\overline{EA}$	I	External Access Enable, Logic Input. When held high, this input enables the device to fetch code from internal program memory locations. The devices do not support external code memory. Do not leave this pin floating.
44	$\overline{PSEN}$	O	Program Store Enable, Logic Output. This pin remains low during internal program execution. $\overline{PSEN}$ enables serial download mode when pulled low through a resistor on power-up or reset. On reset, this pin momentarily becomes an input and the status of the pin is sampled. If there is no pull-down resistor in place, the pin goes momentarily high and then user code executes. If a pull-down resistor is in place, the embedded serial download/debug kernel executes.
45	ALE	O	Address Latch Enable, Logic Output. This output latches the low byte and page byte for 24-bit address space accesses of the address into external data memory.

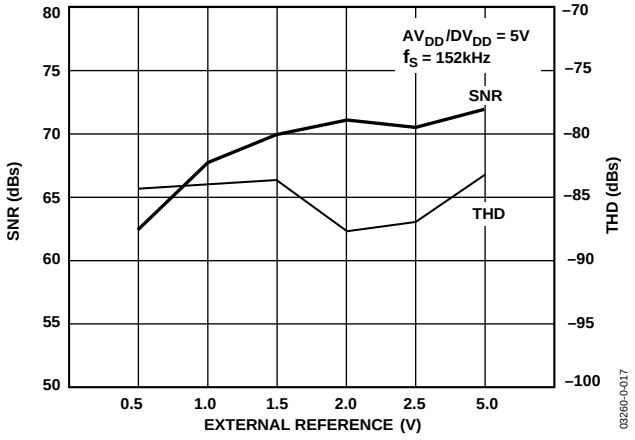

Figure 19. Typical Dynamic Performance vs. V_{REF}, V_{DD} = 5 V

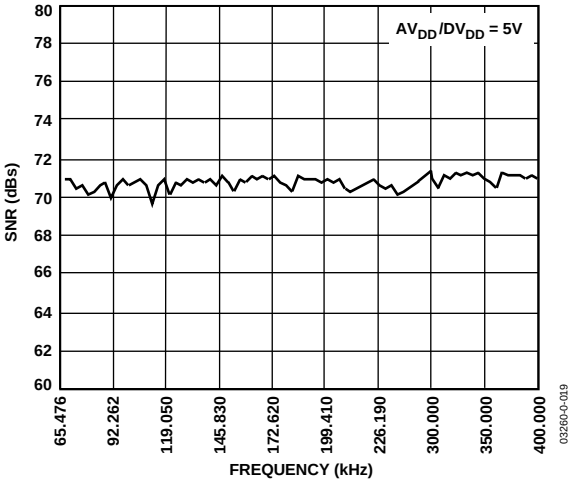

Figure 21. Typical Dynamic Performance vs. Sampling Frequency

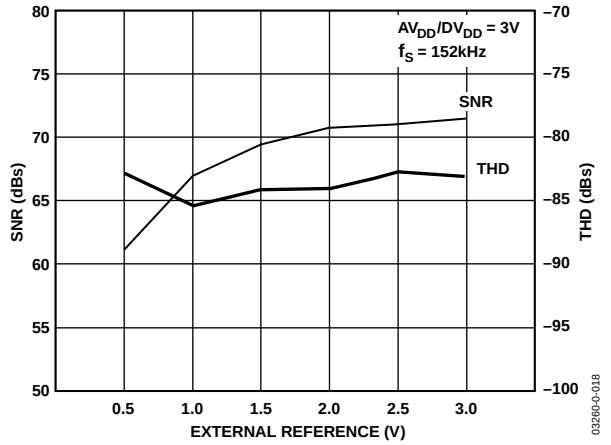

Figure 20. Typical Dynamic Performance vs. V_{REF}, V_{DD} = 3 V

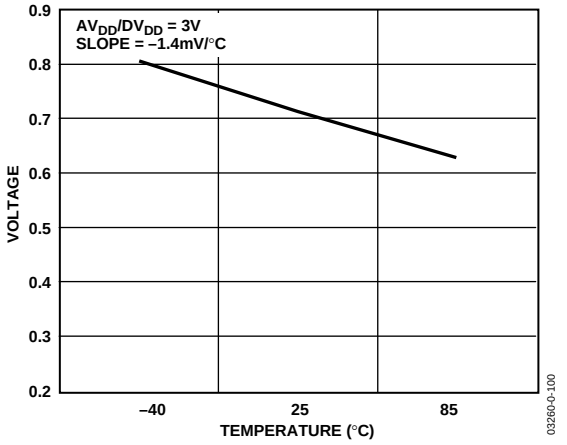

Figure 22. Typical Temperature Sensor Output vs. Temperature

GENERAL DESCRIPTION (*continued*)

The parts also incorporate additional analog functionality with two 12-bit DACs, power supply monitor, and a band gap reference. On-chip digital peripherals include two 16-bit Σ-Δ. DACs, a dual output 16-bit PWM, a watchdog timer, a time interval counter, three timers/counters, and three serial I/O ports (SPI, I²C, and UART).

On the ADuC812 and the ADuC832, the I²C and SPI interfaces share some of the same pins. For backwards compatibility, this is also the case for the ADuC841/ADuC842/ADuC843.

However, there is also the option to allow SPI operate separately on P3.3, P3.4, and P3.5, while I²C uses the standard pins. The I²C interface has also been enhanced to offer repeated start, general call, and quad addressing.

On-chip factory firmware supports in-circuit serial download and debug modes (via UART) as well as single-pin emulation mode via the $\overline{EA}$ pin. A functional block diagram of the parts is shown on the first page.

Mnemonic	Description	Bytes	Cycles
Branching			
JMP @A+DPTR	Jump indirect relative to DPTR	1	3
RET	Return from subroutine	1	4
RETI	Return from interrupt	1	4
ACALL addr11	Absolute jump to subroutine	2	3
AJMP addr11	Absolute jump unconditional	2	3
SJMP rel	Short jump (relative address)	2	3
JC rel	Jump on carry equal to 1	2	3
JNC rel	Jump on carry equal to 0	2	3
JZ rel	Jump on accumulator = 0	2	3
JNZ rel	Jump on accumulator not equal to 0	2	3
DJNZ Rn,rel	Decrement register, JNZ relative	2	3
LJMP	Long jump unconditional	3	4
LCALL addr16	Long jump to subroutine	3	4
JB bit,rel	Jump on direct bit = 1	3	4
JNB bit,rel	Jump on direct bit = 0	3	4
JBC bit,rel	Jump on direct bit = 1 and clear	3	4
CJNE A,dir,rel	Compare A, direct JNE relative	3	4
CJNE A,#data,rel	Compare A, immediate JNE relative	3	4
CJNE Rn,#data,rel	Compare register, immediate JNE relative	3	4
CJNE @Ri,#data,rel	Compare indirect, immediate JNE relative	3	4
DJNZ dir,rel	Decrement direct byte, JNZ relative	3	4
Miscellaneous			
NOP	No operation	1	1

1. One cycle is one clock.
2. Cycles of MOVX instructions are four cycles when they have 0 wait state. Cycles of MOVX instructions are 4 + n cycles when they have n wait states.
3. Cycles of LCALL instruction are three cycles when the LCALL instruction comes from interrupt.

OTHER SINGLE-CYCLE CORE FEATURES

Timer Operation

Timers on a standard 8052 increment by 1 with each machine cycle. On the ADuC841/ADuC842/ADuC843, one machine cycle is equal to one clock cycle; therefore the timers increment at the same rate as the core clock.

ALE

The output on the ALE pin on a standard 8052 part is a clock at 1/6th of the core operating frequency. On the ADuC841/ADuC842/ADuC843 the ALE pin operates as follows. For a single machine cycle instruction, ALE is high for the first half of the machine cycle and low for the second half. The ALE output is at the core operating frequency. For a two or more machine cycle instruction, ALE is high for the first half of the first machine cycle and low for the rest of the machine cycles.

External Memory Access

There is no support for external program memory access on the parts. When accessing external RAM, the EWAIT register may need to be programmed to give extra machine cycles to MOVX commands. This is to account for differing external RAM access speeds.

EWAIT SFR

SFR Address	9FH
Power-On Default	00H
Bit Addressable	No

This special function register (SFR) is programmed with the number of wait states for a MOVX instruction. This value can range from 0H to 7H.

Table 12. Some Single-Supply Op Amps

Op Amp Model	Characteristics
OP281/OP481	Micropower
OP191/OP291/OP491	I/O Good up to V_{DD}, Low Cost
OP196/OP296/OP496	I/O to V_{DD}, Micropower, Low Cost
OP183/OP283	High Gain-Bandwidth Product
OP162/OP262/OP462	High GBP, Micro Package
AD820/AD822/AD824	FET Input, Low Cost
AD823	FET Input, High GBP

Keep in mind that the ADC transfer function is 0 V to V_{REF}, and that any signal range lost to amplifier saturation near ground impacts dynamic range. Though the op amps in Table 12 are capable of delivering output signals that very closely approach ground, no amplifier can deliver signals all the way to ground when powered by a single supply. Therefore, if a negative supply is available, you might consider using it to power the front end amplifiers. If you do, however, be sure to include the Schottky diodes shown in Figure 31 (or at least the lower of the two diodes) to protect the analog input from undervoltage conditions. To summarize this section, use the circuit in Figure 31 to drive the analog input pins of the parts.

Voltage Reference Connections

The on-chip 2.5 V band gap voltage reference can be used as the reference source for the ADC and DACs. To ensure the accuracy of the voltage reference, you must decouple the C_{REF} pin to ground with a 0.47 µF capacitor, as shown in Figure 32. Note that this is different from the ADuC812/ADuC831/ADuC832.

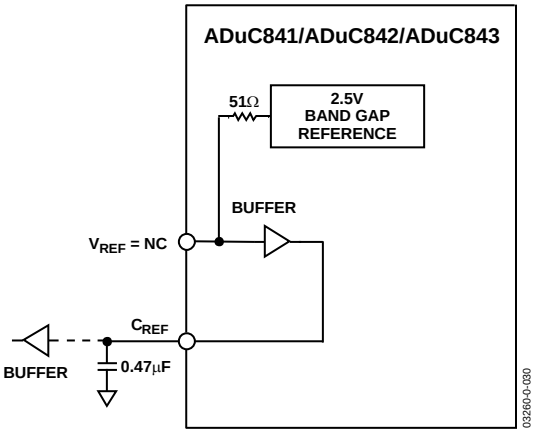

Figure 32. Decoupling V_{REF} and C_{REF}

If the internal voltage reference is to be used as a reference for external circuitry, the C_{REF} output should be used. However, a buffer must be used in this case to ensure that no current is drawn from the C_{REF} pin itself. The voltage on the C_{REF} pin is that of an internal node within the buffer block, and its voltage is critical for ADC and DAC accuracy. The parts power up with their internal voltage reference in the off state.

If an external voltage reference is preferred, it should be connected to the C_{REF} pin as shown in Figure 33. Bit 6 of the ADCCON1 SFR must be set to 1 to switch in the external reference voltage.

To ensure accurate ADC operation, the voltage applied to C_{REF} must be between 1 V and AV_{DD}. In situations where analog input signals are proportional to the power supply (such as in some strain gage applications), it may be desirable to connect the C_{REF} pin directly to AV_{DD}. Operation of the ADC or DACs with a reference voltage below 1 V, however, may incur loss of accuracy, eventually resulting in missing codes or non-monotonicity. For that reason, do not use a reference voltage lower than 1 V.

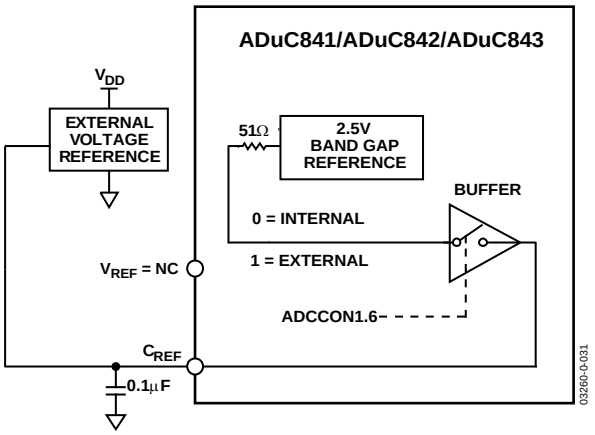

Figure 33. Using an External Voltage Reference

Configuring the ADC

The parts' successive approximation ADC is driven by a divided down version of the master clock. To ensure adequate ADC operation, this ADC clock must be between 400 kHz and 8.38 MHz. Frequencies within this range can be achieved easily with master clock frequencies from 400 kHz to well above 16 MHz, with the four ADC clock divide ratios to choose from. For example, set the ADC clock divide ratio to 8 (that is, ADCCLK = 16.777216 MHz/8 = 2 MHz) by setting the appropriate bits in ADCCON1 (ADCCON1.5 = 1, ADCCON1.4 = 0). The total ADC conversion time is 15 ADC clocks, plus 1 ADC clock for synchronization, plus the selected acquisition time (1, 2, 3, or 4 ADC clocks). For the preceding example, with a 3-clock acquisition time, total conversion time is 19 ADC clocks (or 9.05 µs for a 2 MHz ADC clock).

In continuous conversion mode, a new conversion begins each time the previous one finishes. The sample rate is then simply the inverse of the total conversion time described previously. In the preceding example, the continuous conversion mode sample rate is 110.3 kHz.

The DMA logic operates from the ADC clock and uses pipelining to perform the ADC conversions and to access the external memory at the same time. The time it takes to perform one ADC conversion is called a DMA cycle. The actions performed by the logic during a typical DMA cycle are shown in Figure 36.

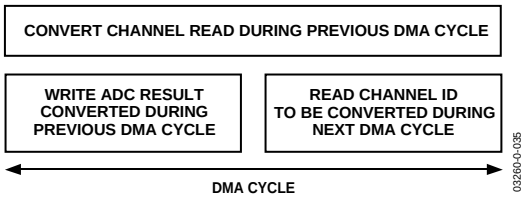

Figure 36. DMA Cycle

Figure 36 shows that during one DMA cycle, the following actions are performed by the DMA logic:

1. An ADC conversion is performed on the channel whose ID was read during the previous cycle.
2. The 12-bit result and the channel ID of the conversion performed in the previous cycle is written to the external memory.
3. The ID of the next channel to be converted is read from external memory.

For the previous example, the complete flow of events is shown in Figure 36. Because the DMA logic uses pipelining, it takes three cycles before the first correct result is written out.

Micro Operation during ADC DMA Mode

During ADC DMA mode, the MicroConverter core is free to continue code execution, including general housekeeping and communication tasks. However, note that MCU core accesses to Ports 0 and 2 (which of course are being used by the DMA controller) are gated off during the ADC DMA mode of operation. This means that even though the instruction that accesses the external Ports 0 or 2 appears to execute, no data is seen at these external ports as a result. Note that during DMA to the internally contained XRAM, Ports 0 and 2 are available for use.

The only case in which the MCU can access XRAM during DMA is when the internal XRAM is enabled and the section of RAM to which the DMA ADC results are being written to lies in an external XRAM. Then the MCU can access the internal XRAM only. This is also the case for use of the extended stack pointer.

The MicroConverter core can be configured with an interrupt to be triggered by the DMA controller when it has finished filling the requested block of RAM with ADC results, allowing the service routine for this interrupt to postprocess data without any real-time timing constraints.

ADC Offset and Gain Calibration Coefficients

The ADuC841/ADuC842/ADuC843 have two ADC calibration coefficients, one for offset calibration and one for gain calibration. Both the offset and gain calibration coefficients are 14-bit words, and are each stored in two registers located in the special function register (SFR) area. The offset calibration coefficient is divided into ADCOFSH (six bits) and ADCOFSL (8 bits), and the gain calibration coefficient is divided into ADCGAINH (6 bits) and ADCGAINL (8 bits).

The offset calibration coefficient compensates for dc offset errors in both the ADC and the input signal. Increasing the offset coefficient compensates for positive offset, and effectively pushes the ADC transfer function down. Decreasing the offset coefficient compensates for negative offset, and effectively pushes the ADC transfer function up. The maximum offset that can be compensated is typically ±5% of V_{REF}, which equates to typically ±125 mV with a 2.5 V reference.

Similarly, the gain calibration coefficient compensates for dc gain errors in both the ADC and the input signal. Increasing the gain coefficient compensates for a smaller analog input signal range and scales the ADC transfer function up, effectively increasing the slope of the transfer function. Decreasing the gain coefficient compensates for a larger analog input signal range and scales the ADC transfer function down, effectively decreasing the slope of the transfer function. The maximum analog input signal range for which the gain coefficient can compensate is $1.025 \times V_{REF}$, and the minimum input range is $0.975 \times V_{REF}$, which equates to typically ±2.5% of the reference voltage.

CALIBRATING THE ADC

Two hardware calibration modes are provided, which can be easily initiated by user software. The ADCCON3 SFR is used to calibrate the ADC. Bit 1 (typical) and CS3 to CS0 (ADCCON2) set up the calibration modes.

Device calibration can be initiated to compensate for significant changes in operating condition frequency, analog input range, reference voltage, and supply voltages. In this calibration mode, offset calibration uses internal AGND selected via ADCCON2 register Bits CS3 to CS0 (1011), and gain calibration uses internal V_{REF} selected by Bits CS3 to CS0 (1100). Offset calibration should be executed first, followed by gain calibration. System calibration can be initiated to compensate for both internal and external system errors. To perform system calibration by using an external reference, tie the system ground and reference to any two of the six selectable inputs. Enable external reference mode (ADCCON1.6). Select the channel connected to AGND via Bits CS3 to CS0 and perform system offset calibration. Select the channel connected to V_{REF} via Bits CS3 to CS0 and perform system gain calibration.

Initiating the Calibration in Code

When calibrating the ADC using ADCCON1, the ADC must be set up into the configuration in which it is used. The ADCCON3 register can then be used to set up the device and to calibrate the ADC offset and gain.

```
MOV ADCCON1,#08CH ;   ADC on; ADCCLK set
                      ;to divide by 32,4
                      ;acquisition clock
```

To calibrate device offset:

```
MOV ADCCON2,#0BH      ;select internal AGND
MOV ADCCON3,#25H      ;select offset calibration,
                      ;31 averages per bit,
                      ;offset calibration
```

To calibrate device gain:

```
MOV ADCCON2,#0CH      ;select internal VREF
MOV ADCCON3,#27H      ;select offset calibration,
                      ;31 averages per bit,
                      ;offset calibration
```

To calibrate system offset, connect system AGND to an ADC channel input (0).

```
MOV ADCCON2,#00H      ;select external AGND
MOV ADCCON3,#25H      ;select offset calibration,
                      ;31 averages per bit
```

To calibrate system gain, connect system V_{REF} to an ADC channel input (1).

```
MOV ADCCON2,#01H      ;select external VREF
MOV ADCCON3,#27H      ;select offset calibration,
                      ;31 averages per bit,
                      ;offset calibration
```

The calibration cycle time T_{CAL} is calculated by the following equation:

$$T_{CAL} = 14 \times ADCCLK \times NUMAV \times (16 + T_{ACQ})$$

For an ADCCLK/FCORE divide ratio of 32, T_{ACQ} = 4 ADCCLK, and NUMAV = 15, the calibration cycle time is

$$T_{CAL} = 14 \times (1/524288) \times 15 \times (16 + 4)$$

$$T_{CAL} = 8\ ms$$

In a calibration cycle, the ADC busy flag (Bit 7), instead of framing an individual ADC conversion as in normal mode, goes high at the start of calibration and returns to zero only at the end of the calibration cycle. It can therefore be monitored in code to indicate when the calibration cycle is completed. The following code can be used to monitor the BUSY signal during a calibration cycle:

```
WAIT:
MOV A, ADCCON3          ;move ADCCON3 to A
JB ACC.7, WAIT          ;If Bit 7 is set jump to
                        WAIT else continue
```

NONVOLATILE FLASH/EE MEMORY

The ADuC841/ADuC842/ADuC843 incorporate Flash/EE memory technology on-chip to provide the user with nonvolatile, in-circuit, reprogrammable code and data memory space. Flash/EE memory is a relatively recent type of nonvolatile memory technology, which is based on a single transistor cell architecture. Flash/EE memory combines the flexible in-circuit reprogrammable features of EEPROM with the space efficient/density features of EPROM as shown in Figure 37.

Because Flash/EE technology is based on a single transistor cell architecture, a flash memory array, such as EPROM, can be implemented to achieve the space efficiencies or memory densities required by a given design. Like EEPROM, flash memory can be programmed in-system at a byte level; it must first be erased, the erase being performed in page blocks. Thus, flash memory is often and more correctly referred to as Flash/EE memory.

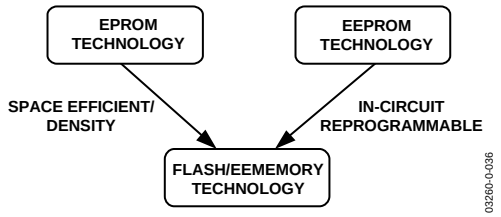

Figure 37. Flash/EE Memory Development

Overall, Flash/EE memory represents a step closer to the ideal memory device that includes nonvolatility, in-circuit programmability, high density, and low cost. Incorporated in the parts, Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one-time programmable (OTP) devices at remote operating nodes.

Flash/EE Memory and the ADuC841/ADuC842/ADuC843

The parts provide two arrays of Flash/EE memory for user applications. Up to 62 kBytes of Flash/EE program space are provided on-chip to facilitate code execution without any external discrete ROM device requirements. The program memory can be programmed in-circuit by using the serial download mode provided, by using conventional third party memory programmers, or via a user defined protocol that can configure it as data if required.

Note that the following sections use the 62 kByte program space as an example when referring to ULOAD mode. For the other memory models (32 kByte and 8 kByte), the ULOAD space moves to the top 8 kBytes of the on-chip program memory, that is, for 32 kBytes, the ULOAD space is from 24 kBytes to 32 kBytes, the kernel still resides in a protected space from 60 kBytes to 62 kBytes. There is no ULOAD space present on the 8 kBtye part.

Example: Programming the Flash/EE Data Memory

A user wants to program F3H into the second byte on Page 03H of the Flash/EE data memory space while preserving the other 3 bytes already in this page. A typical program of the Flash/EE data array involves

1. Setting EADRH/L with the page address.
2. Writing the data to be programmed to the EDATA1–4.
3. Writing the ECON SFR with the appropriate command.

Step 1: Set Up the Page Address

Address registers EADRH and EADRL hold the high byte address and the low byte address of the page to be addressed. The assembly language to set up the address may appear as

```
MOV EADRH,#0            ; Set Page Address Pointer
MOV EADRL,#03H
```

Step 2: Set Up the EDATA Registers

Write the four values to be written into the page into the four SFRs, EDATA1–4. Unfortunately, the user does not know three of them. Thus, the user must read the current page and overwrite the second byte.

```
MOV ECON,#1             ; Read Page into EDATA1-4
MOV EDATA2,#0F3H        ; Overwrite byte 2
```

Step 3: Program Page

A byte in the Flash/EE array can be programmed only if it has previously been erased. To be more specific, a byte can be programmed only if it already holds the value FFH. Because of the Flash/EE architecture, this erase must happen at a page level; therefore, a minimum of 4 bytes (1 page) are erased when an erase command is initiated. Once the page is erase, the user can program the 4 bytes in-page and then perform a verification of the data.

```
MOV ECON,#5             ; ERASE Page
MOV ECON,#2             ; WRITE Page
MOV ECON,#4             ; VERIFY Page
MOV A,ECON              ; Check if ECON=0 (OK!)
JNZ ERROR
```

Although the 4 kBytes of Flash/EE data memory are shipped from the factory pre-erased, that is, byte locations set to FFH, it is nonetheless good programming practice to include an ERASEALL routine as part of any configuration/setup code running on the parts. An ERASEALL command consists of writing 06H to the ECON SFR, which initiates an erase of the 4-kByte Flash/EE array. This command coded in 8051 assembly would appear as

```
MOV ECON,#06H           ; Erase all Command
                        ; 2 ms Duration
```

Flash/EE Memory Timing

Typical program and erase times for the parts are as follows:

Normal Mode (operating on Flash/EE data memory)

READPAGE (4 bytes)	22 machine cycles
WRITEPAGE (4 bytes)	380 µs
VERIFYPAGE (4 bytes)	22 machine cycles
ERASEPAGE (4 bytes)	2 ms
ERASEALL (4 kBytes)	2 ms
READBYTE (1 byte)	9 machine cycles
WRITEBYTE (1 byte)	200 µs

ULOAD Mode (operating on Flash/EE program memory)

WRITEPAGE (256 bytes)	16.5 ms
ERASEPAGE (64 bytes)	2 ms
ERASEALL (56 kBytes)	2 ms
WRITEBYTE (1 byte)	200 µs

Note that a given mode of operation is initiated as soon as the command word is written to the ECON SFR. The core microcontroller operation on the parts is idled until the requested program/read or erase mode is completed. In practice, this means that even though the Flash/EE memory mode of operation is typically initiated with a two machine cycle MOV instruction (to write to the ECON SFR), the next instruction is not executed until the Flash/EE operation is complete. This means that the core cannot respond to interrupt requests until the Flash/EE operation is complete, although the core peripheral functions like counter/timers continue to count and time as configured throughout this period.

The endpoint nonlinearities illustrated in Figure 43 become worse as a function of output loading. Most of the part's specifications assume a 10 kΩ resistive load to ground at the DAC output. As the output is forced to source or sink more current, the nonlinear regions at the top or bottom (respectively) of Figure 43 become larger. Larger current demands can significantly limit output voltage swing. Figure 44 and Figure 45 illustrate this behavior. Note that the upper trace in each of these figures is valid only for an output range selection of 0 V-to-AV_{DD}. In 0 V-to-V_{REF} mode, DAC loading does not cause high-side voltage drops as long as the reference voltage remains below the upper trace in the corresponding figure. For example, if AV_{DD} = 3 V and V_{REF} = 2.5 V, the high-side voltage is not be affected by loads less than 5 mA. But somewhere around 7 mA, the upper curve in Figure 45 drops below 2.5 V (V_{REF}), indicating that at these higher currents the output is not capable of reaching V_{REF}.

To reduce the effects of the saturation of the output amplifier at values close to ground and to give reduced offset and gain errors, the internal buffer can be bypassed. This is done by setting the DBUF bit in the CFG841/CFG842 register. This allows a full rail-to-rail output from the DAC, which should then be buffered externally using a dual-supply op amp in order to get a rail-to-rail output. This external buffer should be located as close as physically possible to the DAC output pin on the PCB. Note that the unbuffered mode works only in the 0 V to V_{REF} range.

To drive significant loads with the DAC outputs, external buffering may be required (even with the internal buffer enabled), as illustrated in Figure 46. Table 12 lists some recommended op amps.

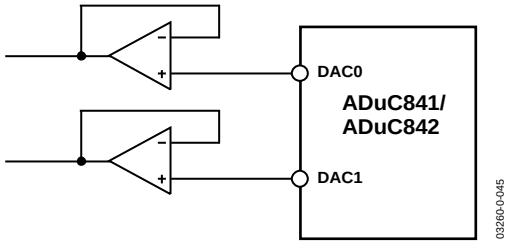

Figure 46. Buffering the DAC Outputs

The DAC output buffer also features a high impedance disable function. In the chip's default power-on state, both DACs are disabled, and their outputs are in a high impedance state (or three-state) where they remain inactive until enabled in software. This means that if a zero output is desired during power-up or power-down transient conditions, then a pull-down resistor must be added to each DAC output. Assuming this resistor is in place, the DAC outputs remain at ground potential whenever the DAC is disabled.

SERIAL PERIPHERAL INTERFACE (SPI)

The ADuC841/ADuC842/ADuC843 integrate a complete hardware serial peripheral interface on-chip. SPI is an industry-standard synchronous serial interface that allows 8 bits of data to be synchronously transmitted and received simultaneously, that is, full duplex. Note that the SPI pins are shared with the I^2C pins. Therefore, the user can enable only one interface or the other on these pins at any given time (see SPE in Table 19). SPI can be operated at the same time as the I^2C interface if the MSPI bit in CFG841/CFG8842 is set. This moves the SPI outputs (MISO, MOSI, and SCLOCK) to P3.3, P3.4, and P3.5, respectively). The SPI port can be configured for master or slave operation and typically consists of four pins, described in the following sections.

MISO (Master In, Slave Out Data I/O Pin)

The MISO pin is configured as an input line in master mode and as an output line in slave mode. The MISO line on the master (data in) should be connected to the MISO line in the slave device (data out). The data is transferred as byte-wide (8-bit) serial data, MSB first.

MOSI (Master Out, Slave In Pin)

The MOSI pin is configured as an output line in master mode and as an input line in slave mode. The MOSI line on the master (data out) should be connected to the MOSI line in the slave device (data in). The data is transferred as byte-wide (8-bit) serial data, MSB first.

SCLOCK (Serial Clock I/O Pin)

The master serial clock (SCLOCK) is used to synchronize the data being transmitted and received through the MOSI and MISO data lines. A single data bit is transmitted and received in each SCLOCK period. Therefore, a byte is transmitted/received after eight SCLOCK periods. The SCLOCK pin is configured as an output in master mode and as an input in slave mode. In master mode, the bit rate, polarity, and phase of the clock are controlled by the CPOL, CPHA, SPR0, and SPR1 bits in the SPICON SFR (see Table 19). In slave mode, the SPICON register must be configured with the phase and polarity (CPHA and CPOL) of the expected input clock. In both master and slave modes, the data is transmitted on one edge of the SCLOCK signal and sampled on the other. It is important, therefore, that CPHA and CPOL are configured the same for the master and slave devices.

$\overline{SS}$ (Slave Select Input Pin)

The $\overline{SS}$ pin is shared with the ADC5 input. To configure this pin as a digital input, the bit must be cleared, for example, CLR P1.5. This line is active low. Data is received or transmitted in slave mode only when the $\overline{SS}$ pin is low, allowing the parts to be used in single-master, multislave SPI configurations. If CPHA = 1, the $\overline{SS}$ input may be permanently pulled low. If CPHA = 0, the $\overline{SS}$ input must be driven low before the first bit in a byte-wide transmission or reception and return high again after the last bit in that byte-wide transmission or reception. In SPI slave mode, the logic level on the external $\overline{SS}$ pin can be read via the SPR0 bit in the SPICON SFR. The SFR registers, described in the following tables, are used to control the SPI interface.

Using the SPI Interface

Depending on the configuration of the bits in the SPICON SFR shown in Table 19, the ADuC841/ADuC842/ADuC843 SPI interface transmits or receives data in a number of possible modes. Figure 54 shows all possible SPI configurations for the parts, and the timing relationships and synchronization between the signals involved. Also shown in this figure is the SPI interrupt bit (ISPI) and how it is triggered at the end of each byte-wide communication.

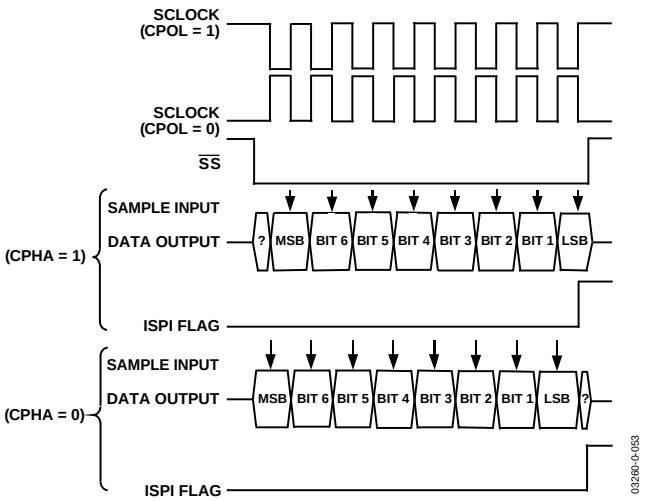

Figure 54. SPI Timing, All Modes

SPI Interface—Master Mode

In master mode, the SCLOCK pin is always an output and generates a burst of eight clocks whenever user code writes to the SPIDAT register. The SCLOCK bit rate is determined by SPR0 and SPR1 in SPICON. Also note that the $\overline{SS}$ pin is not used in master mode. If the parts need to assert the $\overline{SS}$ pin on an external slave device, a port digital output pin should be used.

In master mode, a byte transmission or reception is initiated by a write to SPIDAT. Eight clock periods are generated via the SCLOCK pin and the SPIDAT byte being transmitted via MOSI. With each SCLOCK period, a data bit is also sampled via MISO. After eight clocks, the transmitted byte is completely transmitted, and the input byte waits in the input shift register. The ISPI flag is set automatically, and an interrupt occurs if enabled. The value in the shift register is latched into SPIDAT.

SPI Interface—Slave Mode

In slave mode, SCLOCK is an input. The $\overline{SS}$ pin must also be driven low externally during the byte communication. Transmission is also initiated by a write to SPIDAT. In slave mode, a data bit is transmitted via MISO, and a data bit is received via MOSI through each input SCLOCK period. After eight clocks, the transmitted byte is completely transmitted, and the input byte waits in the input shift register. The ISPI flag is set automatically, and an interrupt occurs if enabled. The value in the shift register is latched into SPIDAT only when the transmission/reception of a byte has been completed. The end of transmission occurs after the eighth clock has been received if CPHA = 1, or when $\overline{SS}$ returns high if CPHA = 0.

TIME INTERVAL COUNTER (TIC)

A TIC is provided on-chip for counting longer intervals than the standard 8051 compatible timers are capable of. The TIC is capable of timeout intervals ranging from 1/128 second to 255 hours. Furthermore, this counter is clocked by the external 32.768 kHz crystal rather than by the core clock, and it has the ability to remain active in power-down mode and time long power-down intervals. This has obvious applications for remote battery-powered sensors where regular widely spaced readings are required.

Six SFRs are associated with the time interval counter, TIMECON being its control register. Depending on the configuration of the IT0 and IT1 bits in TIMECON, the selected time counter register overflow clocks the interval counter. When this counter is equal to the time interval value loaded in the INTVAL SFR, the TII bit (TIMECON.2) is set and generates an interrupt if enabled. If the part is in power-down mode, again with TIC interrupt enabled, the TII bit wakes up the device and resumes code execution by vectoring directly to the TIC interrupt service vector address at 0053H. The TIC-related SFRs are described in Table 25. Note also that the time based SFRs can be written initially with the current time; the TIC can then be controlled and accessed by user software. In effect, this facilitates the implementation of a real-time clock. A block diagram of the TIC is shown in Figure 56.

The TIC is clocked directly from a 32 kHz external crystal on the ADuC842/ADuC843 and by the internal 32 kHz ±10% R/C oscillator on the ADuC841. Due to this, instructions that access the TIC registers are also clocked at this speed. The user should ensure that there is sufficient time between instructions to these registers to allow them to execute correctly.

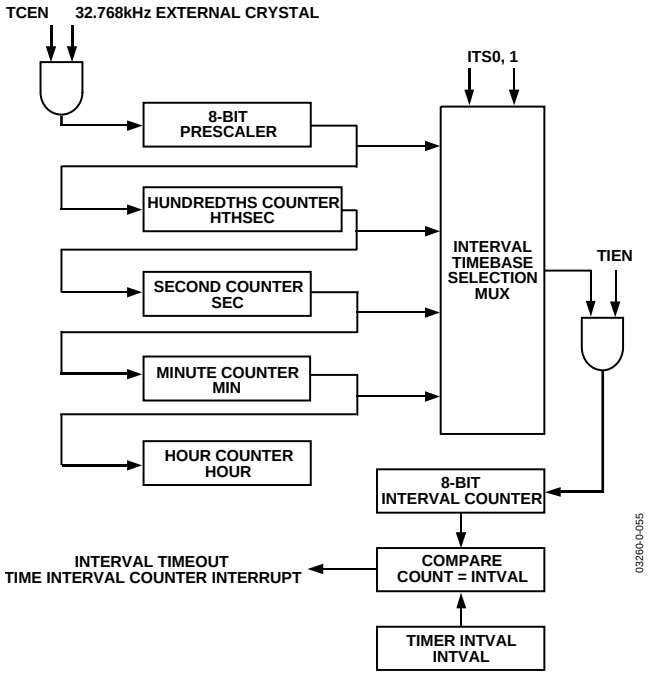

Figure 56. TIC, Simplified Block Diagram

INTVAL	**User Time Interval Select Register**
Function	User code writes the required time interval to this register. When the 8-bit interval counter is equal to the time interval value loaded in the INTVAL SFR, the TII bit (TIMECON.2) is set and generates an interrupt if enabled.
SFR Address	A6H
Power-On Default	00H
Bit Addressable	No
Valid Value	0 to 255 decimal

HTHSEC	**Hundredths Seconds Time Register**
Function	This register is incremented in 1/128 second intervals once TCEN in TIMECON is active. The HTHSEC SFR counts from 0 to 127 before rolling over to increment the SEC time register.
SFR Address	A2H
Power-On Default	00H
Bit Addressable	No
Valid Value	0 to 127 decimal

SEC	**Seconds Time Register**
Function	This register is incremented in 1-second intervals once TCEN in TIMECON is active. The SEC SFR counts from 0 to 59 before rolling over to increment the MIN time register.
SFR Address	A3H
Power-On Default	00H
Bit Addressable	No
Valid Value	0 to 59 decimal

MIN	**Minutes Time Register**
Function	This register is incremented in 1-minute intervals once TCEN in TIMECON is active. The MIN SFR counts from 0 to 59 before rolling over to increment the HOUR time register.
SFR Address	A4H
Power-On Default	00H
Bit Addressable	No
Valid Value	0 to 59 decimal

HOUR	**Hours Time Register**
Function	This register is incremented in 1-hour intervals once TCEN in TIMECON is active. The HOUR SFR counts from 0 to 23 before rolling over to 0.
SFR Address	A5H
Power-On Default	00H
Bit Addressable	No
Valid Value	0 to 23 decimal

8052 COMPATIBLE ON-CHIP PERIPHERALS

This section gives a brief overview of the various secondary peripheral circuits that are also available to the user on-chip. These remaining functions are mostly 8052 compatible (with a few additional features) and are controlled via standard 8052 SFR bit definitions.

Parallel I/O

The ADuC841/ADuC842/ADuC843 use four input/output ports to exchange data with external devices. In addition to performing general-purpose I/O, some ports are capable of external memory operations while others are multiplexed with alternate functions for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general-purpose I/O pin.

Port 0

Port 0 is an 8-bit open-drain bidirectional I/O port that is directly controlled via the Port 0 SFR. Port 0 is also the multiplexed low order address and data bus during accesses to external program or data memory.

Figure 57 shows a typical bit latch and I/O buffer for a Port 0 port pin. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which clocks in a value from the internal bus in response to a write to latch signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a read latch signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a read pin signal from the CPU. Some instructions that read a port activate the read latch signal, and others activate the read pin signal. See the Read-Modify-Write Instructions section for details.

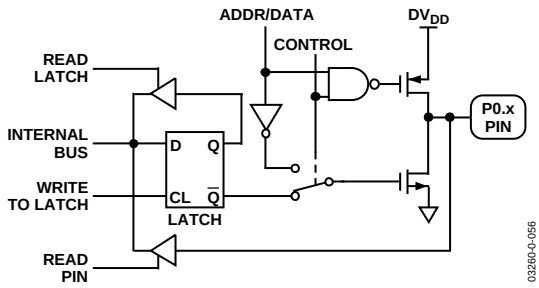

Figure 57. Port 0 Bit Latch and I/O Buffer

As shown in Figure 57, the output drivers of Port 0 pins are switchable to an internal ADDR and ADDR/DATA bus by an internal control signal for use in external memory accesses. During external memory accesses, the P0 SFR has 1s written to it, that is, all of its bit latches become 1. When accessing external memory, the control signal in Figure 57 goes high, enabling push-pull operation of the output pin from the internal address or data bus (ADDR/DATA line). Therefore, no external pull-ups are required on Port 0 for it to access external memory.

In general-purpose I/O port mode, Port 0 pins that have 1s written to them via the Port 0 SFR are configured as open-drain and therefore float. In this state, Port 0 pins can be used as high impedance inputs. This is represented in Figure 57 by the NAND gate whose output remains high as long as the control signal is low, thereby disabling the top FET. External pull-up resistors are therefore required when Port 0 pins are used as general-purpose outputs. Port 0 pins with 0s written to them drive a logic low output voltage (V_{OL}) and are capable of sinking 1.6 mA.

Port 1

Port 1 is also an 8-bit port directly controlled via the P1 SFR. Port 1 digital output capability is not supported on this device. Port 1 pins can be configured as digital inputs or analog inputs. By (power-on) default, these pins are configured as analog inputs, that is, 1 written in the corresponding Port 1 register bit. To configure any of these pins as digital inputs, the user should write a 0 to these port bits to configure the corresponding pin as a high impedance digital input. These pins also have various secondary functions as described in Table 26.

Table 26. Port 1 Alternate Pin Functions

Pin No.	Alternate Function
P1.0	T2 (Timer/Counter 2 External Input)
P1.1	T2EX (Timer/Counter 2 Capture/Reload Trigger)
P1.5	$\overline{SS}$ (Slave Select for the SPI Interface)

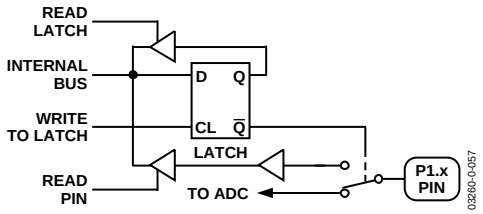

Figure 58. Port 1 Bit Latch and I/O Buffer

Port 2

Port 2 is a bidirectional port with internal pull-up resistors directly controlled via the P2 SFR. Port 2 also emits the high-order address bytes during fetches from external program memory, and middle and high order address bytes during accesses to the 24-bit external data memory space.

As shown in Figure 59, the output drivers of Port 2 are switchable to an internal ADDR and ADDR/DATA bus by an internal control signal for use in external memory accesses (as for Port 0). In external memory addressing mode (CONTROL = 1), the port pins feature push-pull operation controlled by the internal address bus (ADDR line). However, unlike the P0 SFR during external memory accesses, the P2 SFR remains unchanged.

5 V Part

For DV_{DD} below 4.5 V, the internal POR holds the part in reset. As DV_{DD} rises above 4.5 V, an internal timer times out for approximately 128 ms before the part is released from reset. The user must ensure that the power supply has reached a stable 4.75 V minimum level by this time. Likewise on power-down, the internal POR holds the part in reset until the power supply has dropped below 1 V. Figure 83 illustrates the operation of the internal POR in detail.

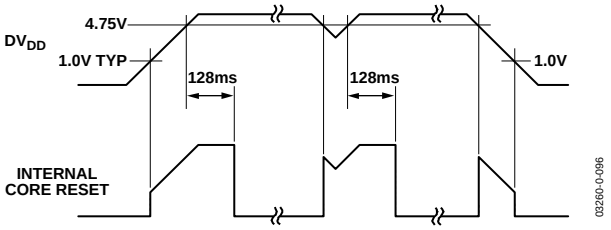

Figure 83. Internal POR Operation

Grounding and Board Layout Recommendations

As with all high resolution data converters, special attention must be paid to grounding and PC board layout of ADuC841/ADuC842/ADuC843 based designs to achieve optimum performance from the ADC and the DACs. Although the parts have separate pins for analog and digital ground (AGND and DGND), the user must not tie these to two separate ground planes unless the two ground planes are connected together very close to the part, as illustrated in the simplified example of Figure 84a. In systems where digital and analog ground planes are connected together somewhere else (for example, at the system's power supply), they cannot be connected again near the part since a ground loop would result. In these cases, tie all the part's AGND and DGND pins to the analog ground plane, as illustrated in Figure 84b. In systems with only one ground plane, ensure that the digital and analog components are physically separated onto separate halves of the board such that digital return currents do not flow near analog circuitry and vice versa. The part can then be placed between the digital and analog sections, as illustrated in Figure 84c.

In all of these scenarios, and in more complicated real-life applications, keep in mind the flow of current from the supplies and back to ground. Make sure the return paths for all currents are as close as possible to the paths that the currents took to reach their destinations. For example, do not power components on the analog side of Figure 84b with DV_{DD} since that would force return currents from DV_{DD} to flow through AGND. Also, try to avoid digital currents flowing under analog circuitry, which could happen if the user places a noisy digital chip on the left half of the board in Figure 84c. Whenever possible, avoid large discontinuities in the ground plane(s) (like those formed by a long trace on the same layer), since they force return signals to travel a longer path. And of course, make all connections to the ground plane directly, with little or no trace separating the pin from its via to ground.

If the user plans to connect fast logic signals (rise/fall time < 5 ns) to any of the part's digital inputs, a series resistor should be added to each relevant line to keep rise and fall times longer than 5 ns at the part's input pins. A value of 100 Ω or 200 Ω is usually sufficient to prevent high speed signals from coupling capacitively into the part and from affecting the accuracy of ADC conversions.

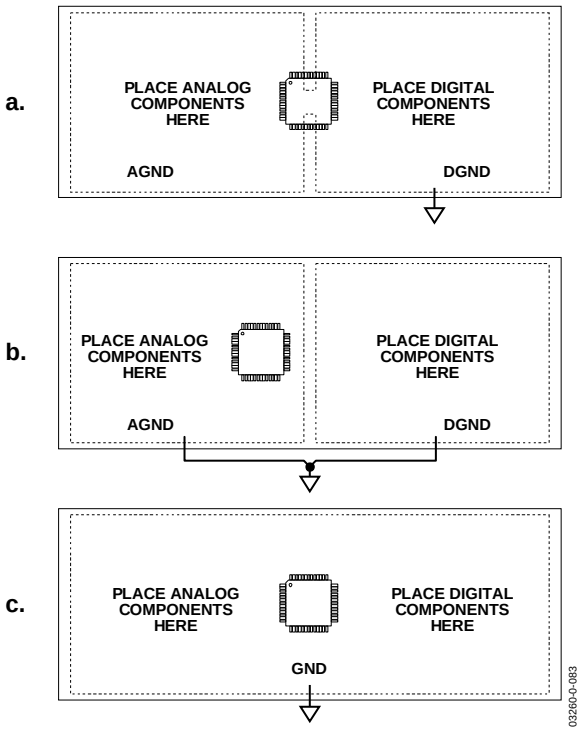

Figure 84. System Grounding Schemes

Parameter		16 MHz Core Clk		8 MHz Core Clock		
EXTERNAL DATA MEMORY READ CYCLE		**Min**	**Max**	**Min**	**Max**	**Unit**
t_{RLRH}	$\overline{RD}$ Pulse Width	60		125		ns
t_{AVLL}	Address Valid after ALE Low	60		120		ns
t_{LLAX}	Address Hold after ALE Low	145		290		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		48		100	Ns
t_{RHDX}	Data and Address Hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float after $\overline{RD}$		150		625	ns
t_{LLDV}	ALE Low to Valid Data In		170		350	ns
t_{AVDV}	Address to Valid Data In		230		470	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	130		255		ns
t_{AVWL}	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	190		375		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		15		35	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	60		120		ns

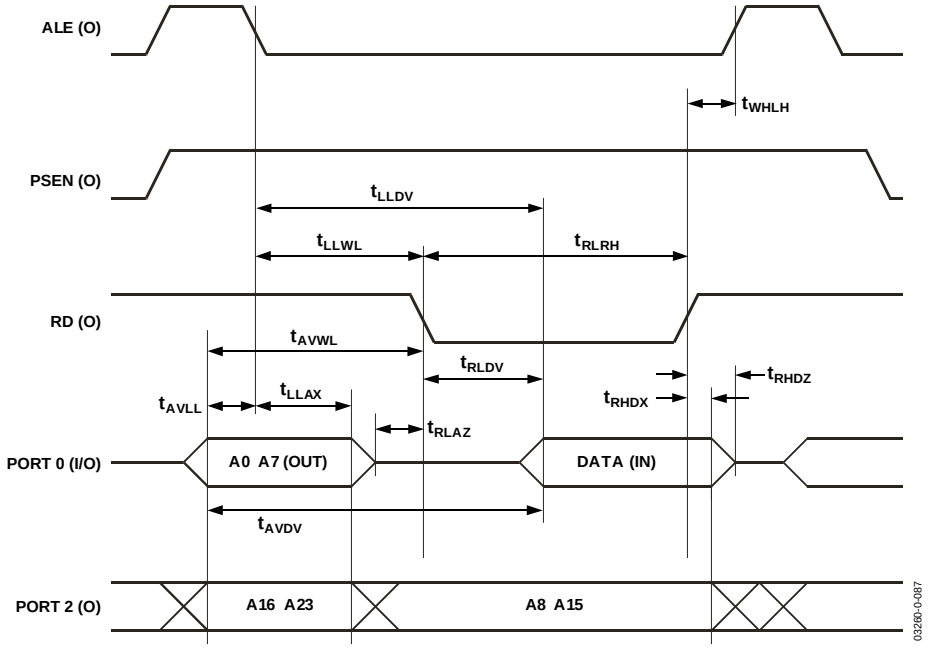

Figure 88. External Data Memory Read Cycle

Parameter		Min	Typ	Max	Unit
SPI SLAVE MODE TIMING (CPHA = 1)					
t_{SS}	$\overline{SS}$ to SCLOCK Edge	0			ns
t_{SL}	SCLOCK Low Pulse Width		330		ns
t_{SH}	SCLOCK High Pulse Width		330		ns
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns
t_{DF}	Data Output Fall Time		10	25	ns
t_{DR}	Data Output Rise Time		10	25	ns
t_{SR}	SCLOCK Rise Time		10	25	ns
t_{SF}	SCLOCK Fall Time		10	25	ns
t_{SFS}	$\overline{SS}$ High after SCLOCK Edge	0			ns

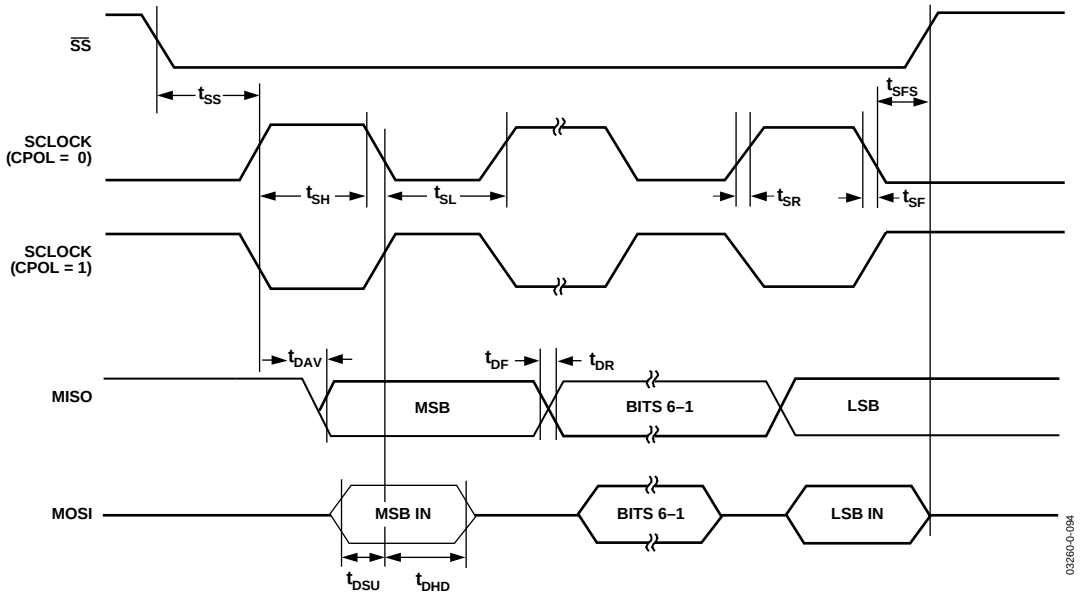

Figure 93. SPI Slave Mode Timing (CPHA = 1)

OUTLINE DIMENSIONS

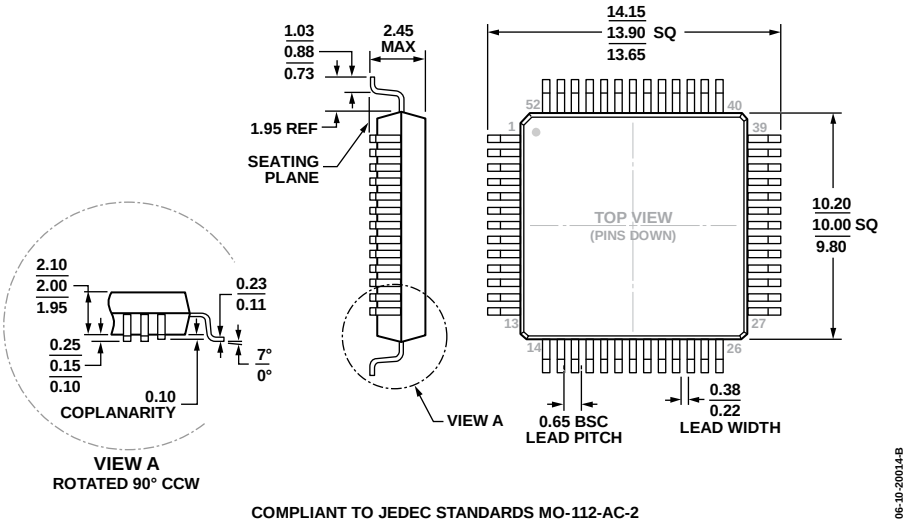

Figure 95. 52-Lead Plastic Quad Flat Package [MQFP]
(S-52-2)
Dimensions shown in millimeters

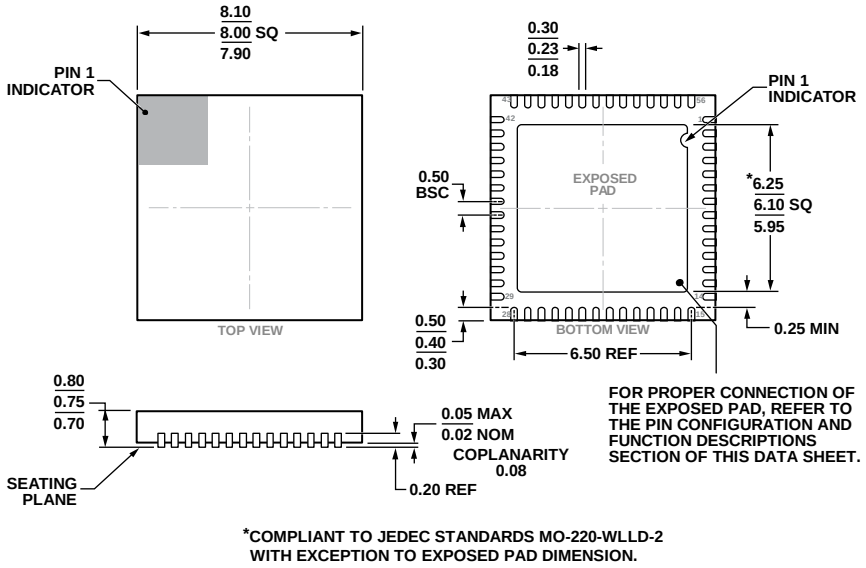

Figure 96. 56-Lead Frame Chip Scale Package [LFCSP]
8 mm × 8 mm and 0.75 mm Package Height
(CP-56-11)
Dimensions shown in millimeters

ORDERING GUIDE

Model[1]	Supply Voltage V_{DD}	User Program Code Space	Temperature Range	Package Description	Package Option
ADuC841BSZ62-5	5	62	–40°C to +85°C	52-Lead Plastic Quad Flat Package [MQFP]	S-52-2
ADuC841BSZ62-3	3	62	–40°C to +85°C	52-Lead Plastic Quad Flat Package [MQFP]	S-52-2
ADuC841BCPZ62-5	5	62	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC841BCPZ62-3	3	62	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC841BCPZ8-5	5	8	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC841BCPZ8-3	3	8	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC842BSZ62-5	5	62	–40°C to +85°C	52-Lead Plastic Quad Flat Package [MQFP]	S-52-2
ADuC842BSZ62-3	3	62	–40°C to +85°C	52-Lead Plastic Quad Flat Package [MQFP]	S-52-2
ADuC842BCPZ62-5	5	62	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC842BCPZ62-3	3	62	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC842BCPZ32-5	5	32	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC842BCPZ32-3	3	32	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC842BCPZ8-5	5	8	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC842BCPZ8-3	3	8	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADUC843BSZ62-5	5	62	–40°C to +85°C	52-Lead Plastic Quad Flat Package [MQFP]	S-52-2
ADuC843BSZ62-3	3	62	–40°C to +85°C	52-Lead Plastic Quad Flat Package [MQFP]	S-52-2
ADuC843BCP62Z-5	5	62	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC843BCPZ62-3	3	62	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC843BCP32Z-5	5	32	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC843BCPZ32-3	3	32	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC843BCPZ8-5	5	8	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
ADuC843BCPZ8-3	3	8	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package [LFCSP]	CP-56-11
EVAL-ADuC841QSZ	5			QuickStart Development System for the ADuC841	
EVAL-ADuC841QSPZ	5			QuickStart Plus Development System	
EVAL-ADuC842QSZ	5			QuickStart Development System for the ADuC842 and ADuC843	
EVAL-ADuC842QSPZ	5			QuickStart Plus Development System	
USB-EA-CONVZ				USB to EA Emulator	

[1] The only difference between the ADuC842 and ADuC843devices is the voltage output DACs on the ADuC842; thus, the evaluation system for the ADuC842 is also suitable for the ADuC843.

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).