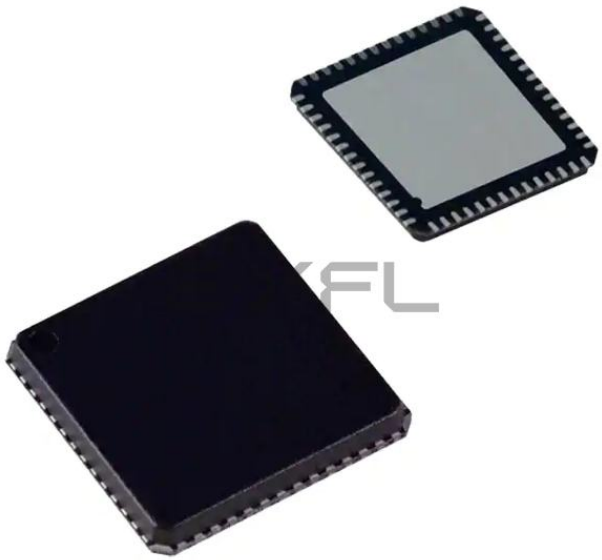

Welcome to E-XFL.COM

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	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, PSM, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
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/aduc841bcpz8-5

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
LOGIC INPUTS[4]				
INPUT VOLTAGES				
All Inputs Except SCLOCK, SDATA, RESET, and XTAL1				
VINL, Input Low Voltage	0.8	0.4	V max	
VINH, Input High Voltage	2.0	2.0	V min	
SDATA				
VINL, Input Low Voltage	0.8	0.8	V max	
VINH, Input High Voltage	2.0	2.0	V min	
SCLOCK and RESET ONLY[4]				
(Schmitt-Triggered Inputs)				
V_{T+}	1.3	0.95	V min	
	3.0	0.25	V max	
V_{T-}	0.8	0.4	V min	
	1.4	1.1	V max	
$V_{T+} - V_{T-}$	0.3	0.3	V min	
	0.85	0.85	V max	
CRYSTAL OSCILLATOR				
Logic Inputs, XTAL1 Only				
V_{INL}, Input Low Voltage	0.8	0.4	V typ	
V_{INH}, Input High Voltage	3.5	2.5	V typ	
XTAL1 Input Capacitance	18	18	pF typ	
XTAL2 Output Capacitance	18	18	pF typ	
MCU CLOCK RATE	16.78	8.38	MHz max	ADuC842/ADuC843 Only
	20	8.38	MHz max	ADuC841 Only
DIGITAL OUTPUTS				
Output High Voltage (V_{OH})	2.4		V min	V_{DD} = 4.5 V to 5.5 V
	4		V typ	I_{SOURCE} = 80 µA
		2.4	V min	V_{DD} = 2.7 V to 3.3 V
		2.6	V typ	I_{SOURCE} = 20 µA
Output Low Voltage (V_{OL})				
ALE, Ports 0 and 2	0.4	0.4	V max	I_{SINK} = 1.6 mA
	0.2	0.2	V typ	I_{SINK} = 1.6 mA
Port 3	0.4	0.4	V max	I_{SINK} = 4 mA
SCLOCK/SDATA	0.4	0.4	V max	I_{SINK} = 8 mA, I^2C Enabled
Floating State Leakage Current[4]	±10	±10	µA max	
	±1	±1	µA typ	
STARTUP TIME				At any core CLK
At Power-On	500	500	ms typ	
From Idle Mode	100	100	µs typ	
From Power-Down Mode				
Wake-up with $\overline{INT0}$ Interrupt	150	400	µs typ	
Wake-up with SPI/I^2C Interrupt	150	400	µs typ	
Wake-up with External RESET	150	400	µs typ	
After External RESET in Normal Mode	30	30	ms typ	
After WDT Reset in Normal Mode	3	3	ms typ	Controlled via WDCON SFR

PIN CONFIGURATIONS AND FUNCTION DESCRIPTIONS

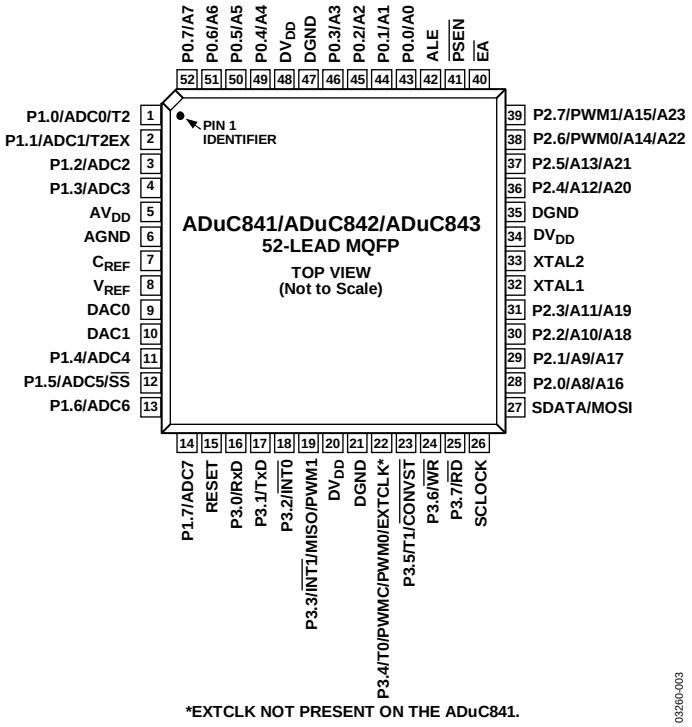

Figure 3. 52-Lead MQFP Pin Configuration

Table 3. 52-Lead MQFP Pin Function Descriptions

Pin No.	Mnemonic	Type[1]	Description
1	P1.0/ADC0/T2	I	Input Port 1 (P1.0). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit. Single-Ended Analog Input (ADC0). Channel selection is via ADCCON2 SFR. Timer 2 Digital Input (T2). Input to Timer/Counter 2. When enabled, Counter 2 is incremented in response to a 1 to 0 transition of the T2 input.
2	P1.1/ADC1/T2EX	I	Input Port 1 (P1.1). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit. Single-Ended Analog Input 1 (ADC1). Channel selection is via ADCCON2 SFR. Capture/Reload Trigger for Counter 2 (T2EX). T2EX is a digital input. This pin also functions as an up/down control input for Counter 2.
3	P1.2/ADC2	I	Input Port 1 (P1.2). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit. Single-Ended Analog Input (ADC2). Channel selection is via ADCCON2 SFR.
4	P1.3/ADC3	I	Input Port 1 (P1.3). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit. Single-Ended Analog Input (ADC3). Channel selection is via ADCCON2 SFR.
5	AV_{DD}	P	Analog Positive Supply Voltage. 3 V or 5 V nominal.
6	AGND	G	Analog Ground. AGND is the ground reference point for the analog circuitry.
7	C_{REF}	I/O	Decoupling Input for On-Chip Reference. Connect a 0.47 µF capacitor between this pin and AGND.
8	V_{REF}	NC	Not Connected. This was a reference output on the ADuC812; use the C_{REF} pin instead.
9	DAC0	O	Voltage Output from DAC0. This pin is a no connect on the ADuC843.
10	DAC1	O	Voltage Output from DAC1. This pin is a no connect on the ADuC843.

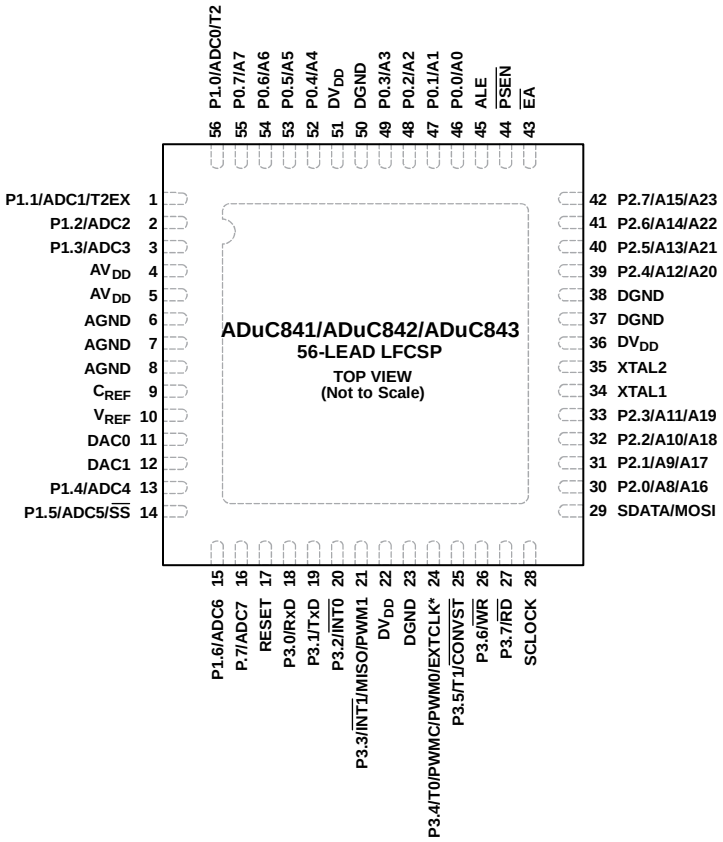

Figure 4. 56-Lead LFCSP Pin Configuration

Table 4. 56-Lead LFCSP Pin Function Descriptions

Pin No.	Mnemonic	Type[1]	Description
1	P1.1/ADC1/T2EX	I	Input Port 1 (P1.1). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 1 (ADC1). Channel selection is via ADCCON2 SFR.
			Capture/Reload Trigger for Counter 2 (T2EX). Digital Input. This pin also functions as an up/down control input for Counter 2.
2	P1.2/ADC2	I	Input Port 1 (P1.2). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input (ADC2). Channel selection is via ADCCON2 SFR.
3	P1.3/ADC3	I	Input Port 1 (P1.3). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input (ADC3). Channel selection is via ADCCON2 SFR.
4, 5	AV_{DD}	P	Analog Positive Supply Voltage. 3 V or 5 V nominal.
6, 7, 8	AGND	G	Analog Ground. AGND is the ground reference point for the analog circuitry.
9	C_{REF}	I/O	Decoupling Input for On-Chip Reference. Connect a 0.47 µF capacitor between this pin and AGND.
10	V_{REF}	NC	Not Connected. This was a reference output on the ADuC812; use the C_{REF} pin instead.
11	DAC0	O	Voltage Output from DAC0. This pin is a no connect on the ADuC843.

TYPICAL PERFORMANCE CHARACTERISTICS

The typical performance plots presented in this section illustrate typical performance of the ADuC841/ADuC842/ADuC843 under various operating conditions.

Figure 5 and Figure 6 show typical ADC integral nonlinearity (INL) errors from ADC Code 0 to Code 4095 at 5 V and 3 V supplies, respectively. The ADC is using its internal reference (2.5 V) and is operating at a sampling rate of 152 kHz; the typical worst-case errors in both plots are just less than 0.3 LSB. Figure 7 and Figure 8 also show ADC INL at a higher sampling rate of 400 kHz. Figure 9 and Figure 10 show the variation in worst-case positive (WCP) INL and worst-case negative (WCN) INL versus external reference input voltage.

Figure 11 and Figure 12 show typical ADC differential nonlinearity (DNL) errors from ADC Code 0 to Code 4095 at 5 V and 3 V supplies, respectively. The ADC is using its internal reference (2.5 V) and is operating at a sampling rate of 152 kHz; the typical worst-case errors in both plots are just less than 0.2 LSB. Figure 13 and Figure 14 show the variation in worst-case positive (WCP) DNL and worst-case negative (WCN) DNL versus external reference input voltage.

Figure 15 shows a histogram plot of 10,000 ADC conversion results on a dc input with V_{DD} = 5 V. The plot illustrates an excellent code distribution pointing to the low noise performance of the on-chip precision ADC.

Figure 16 shows a histogram plot of 10,000 ADC conversion results on a dc input for V_{DD} = 3 V. The plot again illustrates a very tight code distribution of 1 LSB with the majority of codes appearing in one output pin.

Figure 17 and Figure 18 show typical FFT plots for the parts. These plots were generated using an external clock input. The ADC is using its internal reference (2.5 V), sampling a full-scale, 10 kHz sine wave test tone input at a sampling rate of 149.79 kHz. The resulting FFTs shown at 5 V and 3 V supplies illustrate an excellent 100 dB noise floor, 71 dB signal-to-noise ratio (SNR), and THD greater than –80 dB.

Figure 19 and Figure 20 show typical dynamic performance versus external reference voltages. Again, excellent ac performance can be observed in both plots with some roll-off being observed as V_{REF} falls below 1 V.

Figure 21 shows typical dynamic performance versus sampling frequency. SNR levels of 71 dB are obtained across the sampling range of the parts.

Figure 22 shows the voltage output of the on-chip temperature sensor versus temperature. Although the initial voltage output at 25°C can vary from part to part, the resulting slope of −1. 4 mV/°C is constant across all parts.

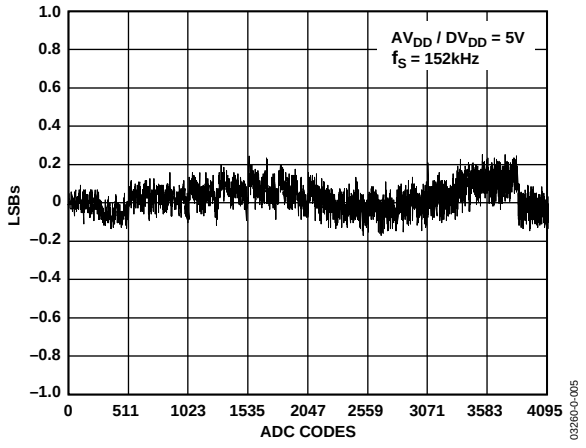

Figure 5. Typical INL Error, V_{DD} = 5 V, f_s = 152 kHz

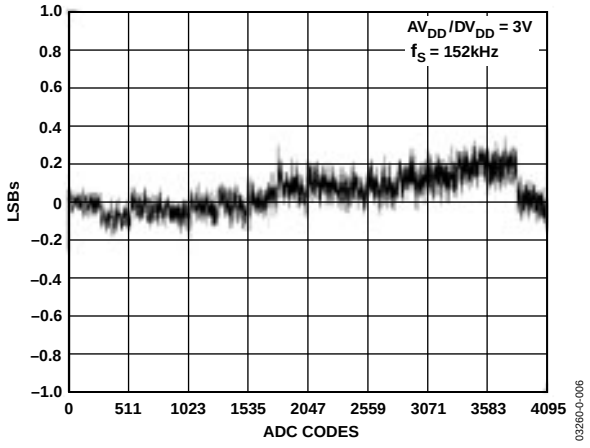

Figure 6. Typical INL Error, V_{DD} = 3 V, f_s = 152 kHz

MEMORY ORGANIZATION

The ADuC841/ADuC842/ADuC843 each contain four different memory blocks:

- Up to 62 kBytes of on-chip Flash/EE program memory
- 4 kBytes of on-chip Flash/EE data memory
- 256 bytes of general-purpose RAM
- 2 kBytes of internal XRAM

Flash/EE Program Memory

The parts provide up to 62 kBytes of Flash/EE program memory to run user code. The user can run code from this internal memory only. Unlike the ADuC812, where code execution can overflow from the internal code space to external code space once the PC becomes greater than 1FFFH, the parts do not support the roll-over from F7FFH in internal code space to F800H in external code space. Instead, the 2048 bytes between F800H and FFFFH appear as NOP instructions to user code.

This internal code space can be downloaded via the UART serial port while the device is in-circuit. 56 kBytes of the program memory can be reprogrammed during run time; thus the code space can be upgraded in the field by using a user defined protocol, or it can be used as a data memory. This is discussed in more detail in the Flash/EE Memory section.

For the 32 kBytes memory model, the top 8 kBytes function as the ULOAD space; this is explained in the Flash/EE Memory section.

Flash/EE Data Memory

4 kBytes of Flash/EE data memory are available to the user and can be accessed indirectly via a group of control registers mapped into the special function register (SFR) area. Access to the Flash/EE data memory is discussed in detail in the Flash/EE Memory section.

General-Purpose RAM

The general-purpose RAM is divided into two separate memories: the upper and the lower 128 bytes of RAM. The lower 128 bytes of RAM can be accessed through direct or indirect addressing. The upper 128 bytes of RAM can be accessed only through indirect addressing because it shares the same address space as the SFR space, which can be accessed only through direct addressing.

The lower 128 bytes of internal data memory are mapped as shown in Figure 23. The lowest 32 bytes are grouped into four banks of eight registers addressed as R0 to R7. The next 16 bytes (128 bits), locations 20H to 2FH above the register banks, form a block of directly addressable bit locations at Bit Addresses 00H to 7FH. The stack can be located anywhere in the internal memory address space, and the stack depth can be expanded up to 2048 bytes.

Reset initializes the stack pointer to location 07H and increments it once before loading the stack to start from location 08H, which is also the first register (R0) of register bank 1. Thus, if the user needs to use more than one register bank, the stack pointer should be initialized to an area of RAM not used for data storage.

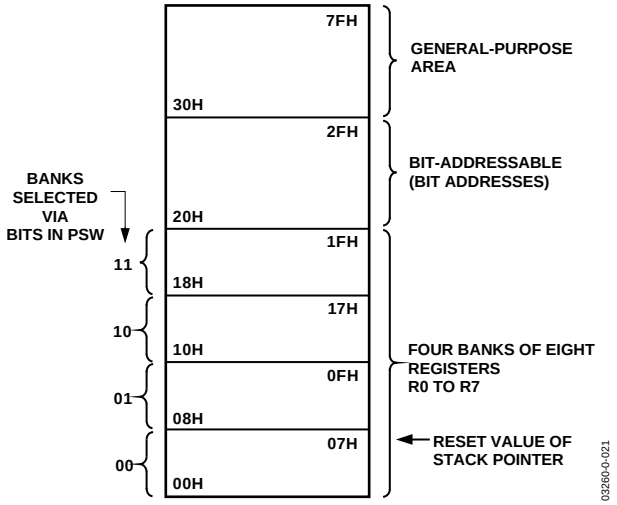

Figure 23. Lower 128 Bytes of Internal Data Memory

The parts contain 2048 bytes of internal XRAM, 1792 bytes of which can be configured to an extended 11-bit stack pointer.

By default, the stack operates exactly like an 8052 in that it rolls over from FFH to 00H in the general-purpose RAM. On the parts, however, it is possible (by setting CFG841.7 or CFG842.7) to enable the 11-bit extended stack pointer. In this case, the stack rolls over from FFH in RAM to 0100H in XRAM.

The 11-bit stack pointer is visible in the SP and SPH SFRs. The SP SFR is located at 81H as with a standard 8052. The SPH SFR is located at B7H. The 3 LSBs of this SFR contain the 3 extra bits necessary to extend the 8-bit stack pointer into an 11-bit stack pointer.

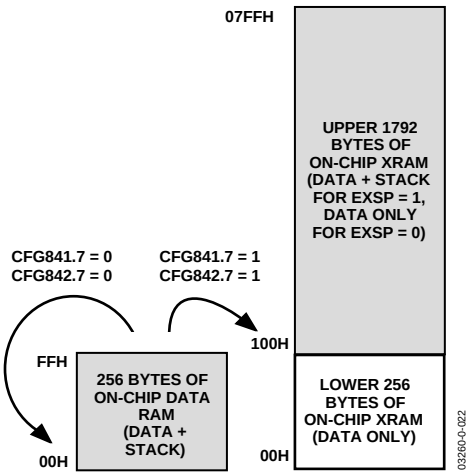

Figure 24. Extended Stack Pointer Operation

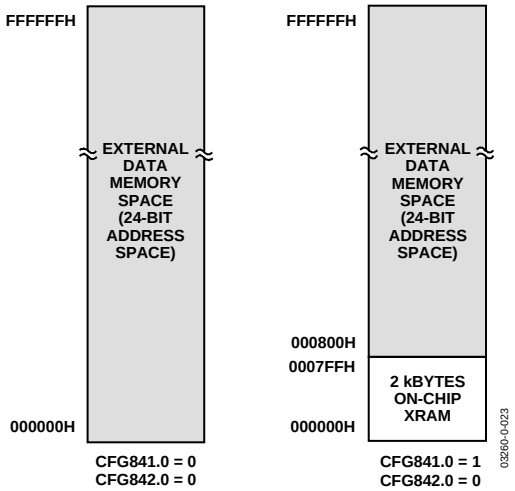

Figure 25. Internal and External XRAM

External Data Memory (External XRAM)

Just like a standard 8051 compatible core, the ADuC841/ADuC842/ADuC843 can access external data memory by using a MOVX instruction. The MOVX instruction automatically outputs the various control strobes required to access the data memory.

The parts, however, can access up to 16 MBytes of external data memory. This is an enhancement of the 64 kBytes of external data memory space available on a standard 8051 compatible core. The external data memory is discussed in more detail in the Hardware Design Considerations section.

Internal XRAM

The parts contain 2 kBytes of on-chip data memory. This memory, although on-chip, is also accessed via the MOVX instruction. The 2 kBytes of internal XRAM are mapped into the bottom 2 kBytes of the external address space if the CFG841/CFG842 bit is set. Otherwise, access to the external data memory occurs just like a standard 8051. When using the internal XRAM, Ports 0 and 2 are free to be used as general-purpose I/O.

SPECIAL FUNCTION REGISTERS (SFRS)

The SFR space is mapped into the upper 128 bytes of internal data memory space and is accessed by direct addressing only. It provides an interface between the CPU and all on-chip peripherals. A block diagram showing the programming model of the parts via the SFR area is shown in Figure 26.

All registers, except the program counter (PC) and the four general-purpose register banks, reside in the SFR area. The SFR registers include control, configuration, and data registers, which provide an interface between the CPU and all on-chip peripherals.

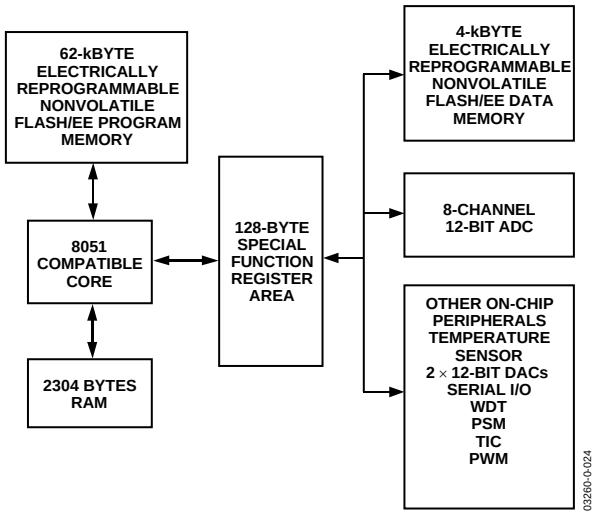

Figure 26. Programming Model

User Download Mode (ULOAD)

Figure 39 shows that it is possible to use the 62 kBytes of Flash/EE program memory available to the user as a single block of memory. In this mode, all of the Flash/EE memory is read-only to user code.

However, the Flash/EE program memory can also be written to during runtime simply by entering ULOAD mode. In ULOAD mode, the lower 56 kBytes of program memory can be erased and reprogrammed by user software as shown in Figure 39. ULOAD mode can be used to upgrade your code in the field via any user defined download protocol. By configuring the SPI port on the part as a slave, it is possible to completely reprogram the 56 kBytes of Flash/EE program memory in only 5 seconds (refer to Application Note uC007).

Alternatively, ULOAD mode can be used to save data to the 56 kBytes of Flash/EE memory. This can be extremely useful in data logging applications where the part can provide up to 60 kBytes of NV data memory on chip (4 kBytes of dedicated Flash/EE data memory also exist).

The upper 6 kBytes of the 62 kBytes of Flash/EE program memory are programmable only via serial download or parallel programming. This means that this space appears as read-only to user code. Therefore, it cannot be accidentally erased or reprogrammed by erroneous code execution, which makes it very suitable to use the 6 kBytes as a bootloader.

A bootload enable option exists in the serial downloader to "always run from E000H after reset." If using a bootloader, this option is recommended to ensure that the bootloader always executes correct code after reset. Programming the Flash/EE program memory via ULOAD mode is described in more detail in the description of ECON and in Application Note uC007.

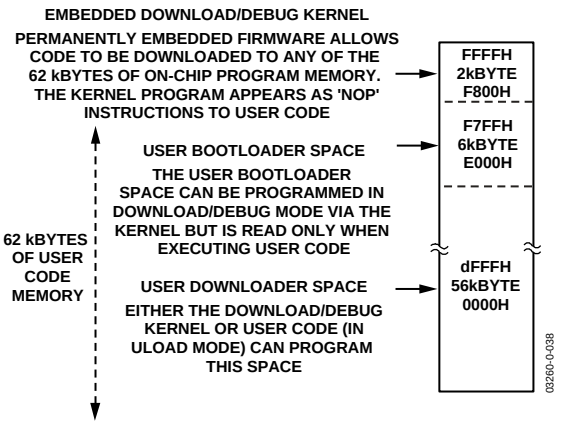

Figure 39. Flash/EE Program Memory Map in ULOAD Mode (62 kByte Part)

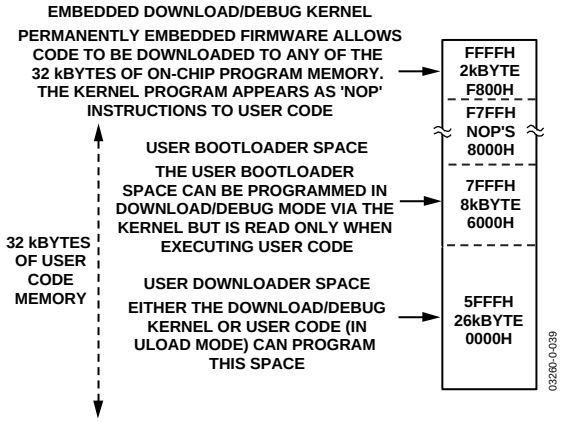

Figure 40. Flash/EE Program Memory Map in ULOAD Mode (32 kByte Part)

Flash/EE Program Memory Security

The ADuC841/ADuC842/ADuC843 facilitate three modes of Flash/EE program memory security. These modes can be independently activated, restricting access to the internal code space. These security modes can be enabled as part of serial download protocol as described in Application Note uC004 or via parallel programming. The security modes available on the parts are as follows:

Lock Mode

This mode locks the code memory, disabling parallel programming of the program memory. However, reading the memory in parallel mode and reading the memory via a MOVC command from external memory is still allowed. This mode is deactivated by initiating a code-erase command in serial download or parallel programming modes.

Secure Mode

This mode locks code in memory, disabling parallel programming (program and verify/read commands) as well as disabling the execution of a MOVC instruction from external memory, which is attempting to read the op codes from internal memory. Read/write of internal data Flash/EE from external memory is also disabled. This mode is deactivated by initiating a code-erase command in serial download or parallel programming modes.

Serial Safe Mode

This mode disables serial download capability on the device. If serial safe mode is activated and an attempt is made to reset the part into serial download mode, that is, $\overline{\text{RESET}}$ asserted and de-asserted with $\overline{\text{PSEN}}$ low, the part interprets the serial download reset as a normal reset only. It therefore cannot enter serial download mode but can only execute as a normal reset sequence. Serial safe mode can be disabled only by initiating a code-erase command in parallel programming mode.

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.

ADuC842/ADuC843 Configuration SFR (CFG842)

The CFG842 SFR contains the necessary bits to configure the internal XRAM, external clock select, PWM output selection, DAC buffer, and the extended SP for both the ADuC842 and the ADuC843. By default, it configures the user into 8051 mode, that is, extended SP is disabled and internal XRAM is disabled. On the ADuC841, this register is the CFG841 register and is described on the next page.

CFG842	ADuC842/ADuC843 Config SFR
SFR Address	AFH
Power-On Default	00H
Bit Addressable	No

Table 14. CFG842 SFR Bit Designations

Bit No.	Name	Description
7	EXSP	Extended SP Enable. When set to 1 by the user, the stack rolls over from SPH/SP = 00FFH to 0100H. When set to 0 by the user, the stack rolls over from SP = FFH to SP = 00H.
6	PWPO	PWM Pin Out Selection. Set to 1 by the user to select P3.4 and P3.3 as the PWM output pins. Set to 0 by the user to select P2.6 and P2.7 as the PWM output pins.
5	DBUF	DAC Output Buffer. Set to 1 by the user to bypass the DAC output buffer. Set to 0 by the user to enable the DAC output buffer.
4	EXTCLK	Set by the user to 1 to select an external clock input on P3.4. Set by the user to 0 to use the internal PLL clock.
3	RSVD	Reserved. This bit should always contain 0.
2	RSVD	Reserved. This bit should always contain 0.
1	MSPI	Set to 1 by the user to move the SPI functionality of MISO, MOSI, and SCLOCK to P3.3, P3.4, and P3.5, respectively. Set to 0 by the user to leave the SPI functionality as usual on MISO, MOSI, and SCLOCK pins.
0	XRAMEN	XRAM Enable Bit. When set to 1 by the user, the internal XRAM is mapped into the lower 2 kBytes of the external address space. When set to 0 by the user, the internal XRAM is not accessible, and the external data memory is mapped into the lower 2 kBytes of external data memory.

Using the DAC

The on-chip DAC architecture consists of a resistor string DAC followed by an output buffer amplifier, the functional equivalent of which is illustrated in Figure 42. Features of this architecture include inherent guaranteed monotonicity and excellent differential linearity.

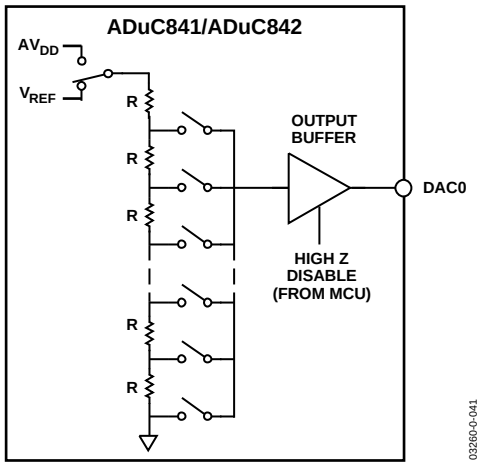

Figure 42. Resistor String DAC Functional Equivalent

As shown in Figure 42, the reference source for each DAC is user selectable in software. It can be either AV_{DD} or V_{REF}. In 0 V-to-AV_{DD} mode, the DAC output transfer function spans from 0 V to the voltage at the AV_{DD} pin. In 0 V-to-V_{REF} mode, the DAC output transfer function spans from 0 V to the internal V_{REF} or, if an external reference is applied, the voltage at the C_{REF} pin. The DAC output buffer amplifier features a true rail-to-rail output stage implementation. This means that unloaded, each output is capable of swinging to within less than 100 mV of both AVDD and ground. Moreover, the DAC's linearity specification (when driving a 10 kΩ resistive load to ground) is guaranteed through the full transfer function except Codes 0 to 100, and, in 0 V-to-AVDD mode only, Codes 3995 to 4095. Linearity degradation near ground and V_{DD} is caused by saturation of the output amplifier, and a general representation of its effects (neglecting offset and gain error) is illustrated in Figure 43. The dotted line in Figure 43 indicates the ideal transfer function, and the solid line represents what the transfer function might look like with endpoint nonlinearities due to saturation of the output amplifier. Note that Figure 43 represents a transfer function in 0 V-to-V_{DD} mode only. In 0 V-to-V_{REF} mode (with $V_{REF} < V_{DD}$), the lower nonlinearity would be similar, but the upper portion of the transfer function would follow the ideal line right to the end (V_{REF} in this case, not V_{DD}), showing no signs of endpoint linearity errors.

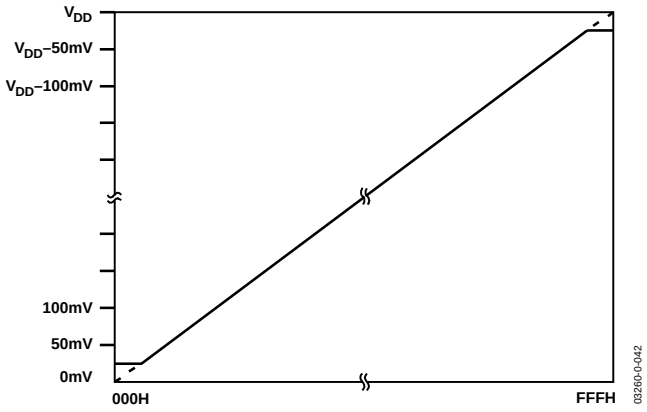

Figure 43. Endpoint Nonlinearities Due to Amplifier Saturation

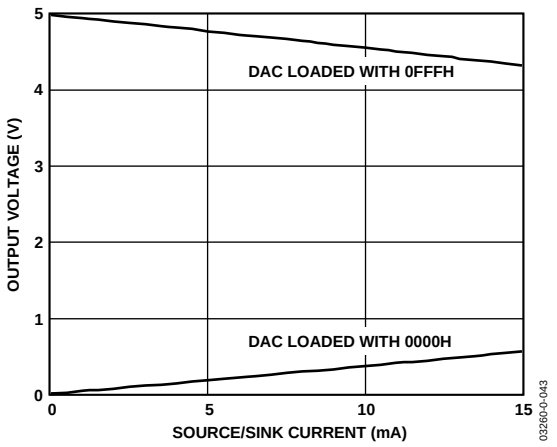

Figure 44. Source and Sink Current Capability with $V_{REF} = V_{DD} = 5$ V

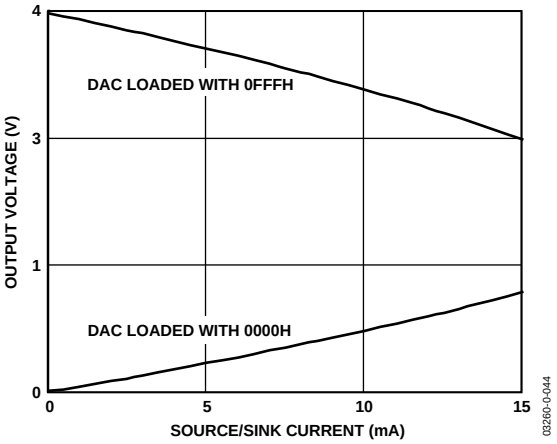

Figure 45. Source and Sink Current Capability with $V_{REF} = V_{DD} = 3$ V

SPICON SPI Control Register

SFR Address	F8H
Power-On Default	04H
Bit Addressable	Yes

Table 19. SPICON SFR Bit Designations

Bit No.	Name	Description
7	ISPI	SPI Interrupt Bit. Set by the MicroConverter at the end of each SPI transfer. Cleared directly by user code or indirectly by reading the SPIDAT SFR.
6	WCOL	Write Collision Error Bit. Set by the MicroConverter if SPIDAT is written to while an SPI transfer is in progress. Cleared by user code.
5	SPE	SPI Interface Enable Bit. Set by the user to enable the SPI interface. Cleared by the user to enable the I^2C pins, this is not requiredto enable the I^2C interface if the MSPI bit is set in CFG841/CFG842. In this case, the I^2C interface is automatically enabled.
4	SPIM	SPI Master/Slave Mode Select Bit. Set by the user to enable master mode operation (SCLOCK is an output). Cleared by the user to enable slave mode operation (SCLOCK is an input).
3	CPOL[1]	Clock Polarity Select Bit. Set by the user if SCLOCK idles high. Cleared by the user if SCLOCK idles low.
2	CPHA[1]	Clock Phase Select Bit. Set by the user if leading SCLOCK edge is to transmit data. Cleared by the user if trailing SCLOCK edge is to transmit data.
1	SPR1	SPI Bit Rate Select Bits.
0	SPR0	These bits select the SCLOCK rate (bit rate) in master mode as follows: SPR1 SPR0 Selected Bit Rate 0 0 $f_{osc}/2$ 0 1 $f_{osc}/4$ 1 0 $f_{osc}/8$ 1 1 $f_{osc}/16$ In SPI slave mode, that is, SPIM = 0, the logic level on the external $\overline{SS}$ pin can be read via the SPR0 bit.

[1]The CPOL and CPHA bits should both contain the same values for master and slave devices.

SPIDAT	**SPI Data Register**
Function	SPIDAT SFR is written by the user to transmit data over the SPI interface or read by user code to read data just received by the SPI interface.
SFR Address	F7H
Power-On Default	00H
Bit Addressable	No

Bit No.	Name	Description
2	I2CRS	I^2C Reset Bit (Slave Mode Only). Set by the user to reset the I^2C interface. Cleared by the user code for normal I^2C operation.
1	I2CTX	I^2C Direction Transfer Bit (Slave Mode Only). Set by the MicroConverter if the interface is transmitting. Cleared by the MicroConverter if the interface is receiving.
0	I2CI	I^2C Interrupt Bit (Slave Mode Only). Set by the MicroConverter after a byte has been transmitted or received. Cleared automatically when user code reads the I2CDAT SFR (see I2CDAT below).

I2CADD	**I^2C Address Register**
Function	Holds the first I^2C peripheral address for the part. It may be overwritten by user code. Application Note uC001 at www.analog.com/microconverter describes the format of the I^2C standard 7-bit address in detail.
SFR Address	9BH
Power-On Default	55H
Bit Addressable	No
I2CADD1	**I^2C Address Register**
Function	Holds the second I^2C peripheral address for the part. It may be overwritten by user code.
SFR Address	91H
Power-On Default	7FH
Bit Addressable	No
I2CADD2	**I^2C Address Register**
Function	Holds the third I^2C peripheral address for the part. It may be overwritten by user code.
SFR Address	92H
Power-On Default	7FH
Bit Addressable	No
I2CADD3	**I^2C Address Register**
Function	Holds the fourth I^2C peripheral address for the part. It may be overwritten by user code.
SFR Address	93H
Power-On Default	7FH
Bit Addressable	No
I2CDAT	**I^2C Data Register**
Function	Written by the user to transmit data over the I^2C interface or read by user code to read data just received by the I^2C interface. Accessing I2CDAT automatically clears any pending I^2C interrupt and the I2CI bit in the I2CCON SFR. User software should access I2CDAT only once per interrupt cycle.
SFR Address	9AH
Power-On Default	00H
Bit Addressable	No

The main features of the MicroConverter I^2C interface are

- Only two bus lines are required: a serial data line (SDATA) and a serial clock line (SCLOCK).
- An I^2C master can communicate with multiple slave devices. Because each slave device has a unique 7-bit address, single master/slave relationships can exist at all times even in a multislave environment.
- Ability to respond to four separate addresses when operating in slave mode.

TIMECON	**TIC Control Register**
SFR Address	A1H
Power-On Default	00H
Bit Addressable	No

Table 25. TIMECON SFR Bit Designations

Bit No.	Name	Description
7	----	Reserved.
6	TFH	Twenty-Four Hour Select Bit. Set by the user to enable the hour counter to count from 0 to 23. Cleared by the user to enable the hour counter to count from 0 to 255.
5	ITS1	Interval Timebase Selection Bits.
4	ITS0	Written by user to determine the interval counter update rate. ITS1 ITS0 Interval Timebase 0 0 1/128 Second 0 1 Seconds 1 0 Minutes 1 1 Hours
3	STI	Single Time Interval Bit. Set by the user to generate a single interval timeout. If set, a timeout clears the TIEN bit. Cleared by the user to allow the interval counter to be automatically reloaded and start counting again at each interval timeout.
2	TII	TIC Interrupt Bit. Set when the 8-bit interval counter matches the value in the INTVAL SFR. Cleared by user software.
1	TIEN	Time Interval Enable Bit. Set by the user to enable the 8-bit time interval counter. Cleared by the user to disable the interval counter.
0	TCEN	Time Clock Enable Bit. Set by the user to enable the time clock to the time interval counters. Cleared by the user to disable the clock to the time interval counters and reset the time interval SFRs to the last value written to them by the user. The time registers (HTHSEC, SEC, MIN, and HOUR) can be written while TCEN is low.

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

In general-purpose I/O port mode, Port 2 pins that have 1s written to them are pulled high by the internal pull-ups (Figure 60) 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. Port 2 pins with 0s written to them drive a logic low output voltage (V_{OL}) and are capable of sinking 1.6 mA.

P2.6 and P2.7 can also be used as PWM outputs. When they are selected as the PWM outputs via the CFG841/CFG842 SFR, the PWM outputs overwrite anything written to P2.6 or P2.7.

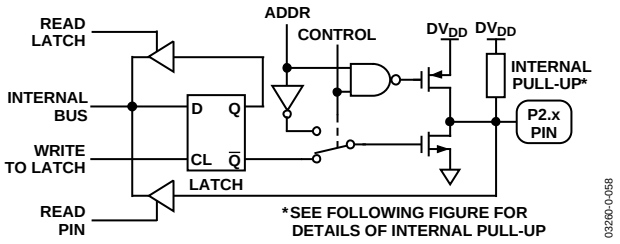

Figure 59. Port 2 Bit Latch and I/O Buffer

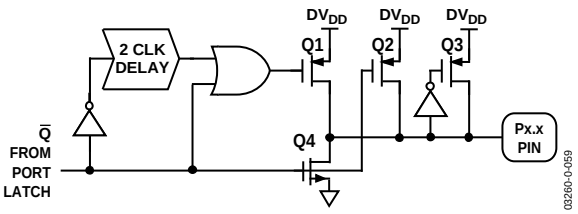

Figure 60. Internal Pull-Up Configuration

Port 3

Port 3 is a bidirectional port with internal pull-ups directly controlled via the P3 SFR. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and, in that state, can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-ups.

Port 3 pins with 0s written to them drive a logic low output voltage (V_{OL}) and are capable of sinking 4 mA. Port 3 pins also have various secondary functions as described in Table 27. The alternate functions of Port 3 pins can be activated only if the corresponding bit latch in the P3 SFR contains a 1. Otherwise, the port pin is stuck at 0.

Table 27. Port 3 Alternate Pin Functions

Pin No.	Alternate Function
P3.0	RxD (UART Input Pin) (or Serial Data I/O in Mode 0)
P3.1	TxD (UART Output Pin) (or Serial Clock Output in Mode 0)
P3.2	$\overline{INT0}$ (External Interrupt 0)
P3.3	$\overline{INT1}$ (External Interrupt 1)/PWM 1/MISO
P3.4	T0 (Timer/Counter 0 External Input) PWM External Clock/PWM 0
P3.5	T1 (Timer/Counter 1 External Input)
P3.6	$\overline{WR}$ (External Data Memory Write Strobe)
P3.7	$\overline{RD}$ (External Data Memory Read Strobe)

P3.3 and P3.4 can also be used as PWM outputs. When they are selected as the PWM outputs via the CFG841/CFG842 SFR, the PWM outputs overwrite anything written to P3.4 or P3.3.

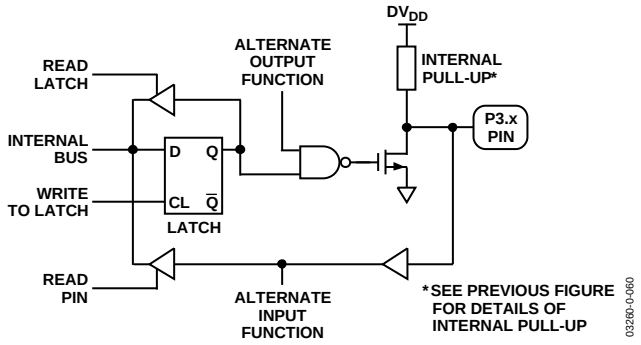

Figure 61. Port 3 Bit Latch and I/O Buffer

Additional Digital I/O

In addition to the port pins, the dedicated SPI/I²C pins (SCLOCK and SDATA/MOSI) also feature both input and output functions. Their equivalent I/O architectures are illustrated in Figure 62 and Figure 64, respectively, for SPI operation and in Figure 63 and Figure 65 for I²C operation. Notice that in I²C mode (SPE = 0), the strong pull-up FET (Q1) is disabled, leaving only a weak pull-up (Q2) present. By contrast, in SPI mode (SPE = 1) the strong pull-up FET (Q1) is controlled directly by SPI hardware, giving the pin push-pull capability.

In I²C mode (SPE = 0), two pull-down FETs (Q3 and Q4) operate in parallel to provide an extra 60% or 70% of current sinking capability. In SPI mode (SPE = 1), however, only one of the pull-down FETs (Q3) operates on each pin, resulting in sink capabilities identical to that of Port 0 and Port 2 pins. On the input path of SCLOCK, notice that a Schmitt trigger conditions the signal going to the SPI hardware to prevent false triggers (double triggers) on slow incoming edges. For incoming signals from the SCLOCK and SDATA pins going to I²C hardware, a filter conditions the signals to reject glitches of up to 50 ns in duration.

Notice also that direct access to the SCLOCK and SDATA/MOSI pins is afforded through the SFR interface in I²C master mode. Therefore, if you are not using the SPI or I²C functions, you can use these two pins to give additional high current digital outputs.

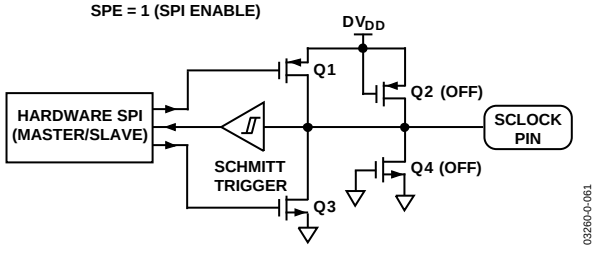

Figure 62. SCLOCK Pin I/O Functional Equivalent in SPI Mode

T2CON	Timer/Counter 2 Control Register
SFR Address	C8H
Power-On Default	00H
Bit Addressable	Yes

Table 31. T2CON SFR Bit Designations

Bit No.	Name	Description
7	TF2	Timer 2 Overflow Flag. Set by hardware on a Timer 2 overflow. TF2 cannot be set when either RCLK = 1 or TCLK = 1. Cleared by user software.
6	EXF2	Timer 2 External Flag. Set by hardware when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. Cleared by user software.
5	RCLK	Receive Clock Enable Bit. Set by the user to enable the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. Cleared by the user to enable Timer 1 overflow to be used for the receive clock.
4	TCLK	Transmit Clock Enable Bit. Set by the user to enable the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. Cleared by the user to enable Timer 1 overflow to be used for the transmit clock.
3	EXEN2	Timer 2 External Enable Flag. Set by the user to enable a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. Cleared by the user for Timer 2 to ignore events at T2EX.
2	TR2	Timer 2 Start/Stop Control Bit. Set by the user to start Timer 2. Cleared by the user to stop Timer 2.
1	CNT2	Timer 2 Timer or Counter Function Select Bit. Set by the user to select counter function (input from external T2 pin). Cleared by the user to select timer function (input from on-chip core clock).
0	CAP2	Timer 2 Capture/Reload Select Bit. Set by the user to enable captures on negative transitions at T2EX if EXEN2 = 1. Cleared by the user to enable autoreloads with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to autoreload on Timer 2 overflow.

Timer/Counter 2 Data Registers

Timer/Counter 2 also has two pairs of 8-bit data registers associated with it. These are used as both timer data registers and as timer capture/reload registers.

TH2 and TL2

Timer 2, data high byte and low byte.
SFR Address = CDH, CCH, respectively.

RCAP2H and RCAP2L

Timer 2, capture/reload byte and low byte.
SFR Address = CBH, CAH, respectively.

IEIP2	Secondary Interrupt Enable Register
SFR Address	A9H
Power-On Default	A0H
Bit Addressable	No

Table 38. IEIP2 SFR Bit Designations

Bit No.	Name	Description
7	----	Reserved.
6	PTI	Priority for time interval interrupt.
5	PPSM	Priority for power supply monitor interrupt.
4	PSI	Priority for SPI/I²C interrupt.
3	----	This bit must contain zero.
2	ETI	Set by the user to enable, or cleared to disable time interval counter interrupts.
1	EPSMI	Set by the user to enable, or cleared to disable power supply monitor interrupts.
0	ESI	Set by the user to enable, or cleared to disable SPI or I²C serial port interrupts.

Interrupt Priority

The interrupt enable registers are written by the user to enable individual interrupt sources, while the interrupt priority registers allow the user to select one of two priority levels for each interrupt. An interrupt of a high priority may interrupt the service routine of a low priority interrupt, and if two interrupts of different priority occur at the same time, the higher level interrupt is serviced first. An interrupt cannot be interrupted by another interrupt of the same priority level. If two interrupts of the same priority level occur simultaneously, a polling sequence is observed as shown in Table 39.

Table 39. Priority within an Interrupt Level

Source	Priority	Description
PSMI	1 (Highest)	Power Supply Monitor Interrupt.
WDS	2	Watchdog Timer Interrupt.
IE0	2	External Interrupt 0.
ADCI	3	ADC Interrupt.
TF0	4	Timer/Counter 0 Interrupt.
IE1	5	External Interrupt 1.
TF1	6	Timer/Counter 1 Interrupt.
ISPI/I2CI	7	SPI Interrupt/I²C Interrupt.
RI + TI	8	Serial Interrupt.
TF2 + EXF2	9	Timer/Counter 2 Interrupt.
TII	11(Lowest)	Time Interval Counter Interrupt.

Interrupt Vectors

When an interrupt occurs, the program counter is pushed onto the stack, and the corresponding interrupt vector address is loaded into the program counter. The interrupt vector addresses are shown in Table 40.

Table 40. Interrupt Vector Addresses

Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI + TI	0023H
TF2 + EXF2	002BH
ADCI	0033H
ISPI/I2CI	003BH
PSMI	0043H
TII	0053H
WDS	005BH

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.

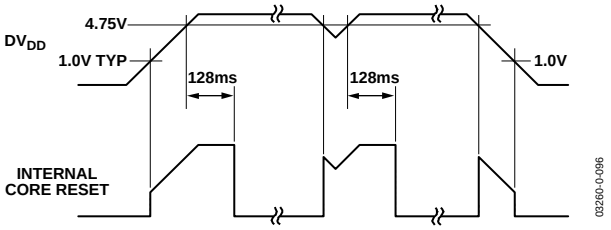

Figure 83. Internal POR Operation

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.

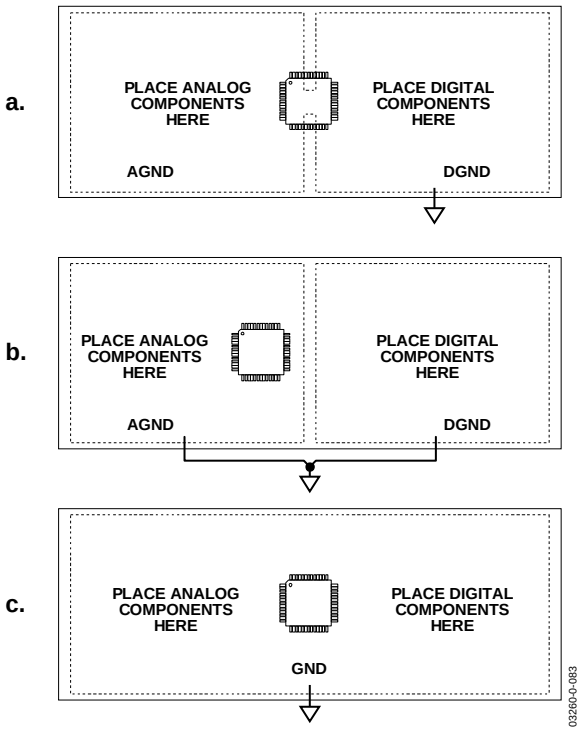

Figure 84. System Grounding Schemes

Note that $\overline{\text{PSEN}}$ is normally an output (as described in the External Memory Interface section) and is sampled as an input only on the falling edge of RESET, that is, at power-up or upon an external manual reset. Note also that if any external circuitry unintentionally pulls $\overline{\text{PSEN}}$ low during power-up or reset events, it could cause the chip to enter download mode and therefore fail to begin user code execution as it should. To prevent this, ensure that no external signals are capable of pulling the $\overline{\text{PSEN}}$ pin low, except for the external $\overline{\text{PSEN}}$ jumper itself.

Embedded Serial Port Debugger

From a hardware perspective, entry into serial port debug mode is identical to the serial download entry sequence described in the preceding section. In fact, both serial download and serial port debug modes can be thought of as essentially one mode of operation used in two different ways. Note that the serial port debugger is fully contained on the part (unlike ROM monitor type debuggers), and therefore no external memory is needed to enable in-system debug sessions.

Single-Pin Emulation Mode

Also built into the part is a dedicated controller for single-pin in-circuit emulation (ICE) using standard production ADuC841/ADuC842/ADuC843 devices. In this mode, emulation access is gained by connection to a single pin, the $\overline{\text{EA}}$ pin. Normally, this pin is hardwired either high or low to select execution from internal or external program memory space, as described earlier. To enable single-pin emulation mode, however, users need to pull the $\overline{\text{EA}}$ pin high through a 1 kΩ resistor, as shown in Figure 85. The emulator then connects to the 2-pin header also shown in Figure 85. To be compatible with the standard connector that comes with the single-pin emulator available from Accutron Limited (www.accutron.com), use a 2-pin 0.1 inch pitch friction lock header from Molex (www.molex.com) such as their part number 22-27-2021. Be sure to observe the polarity of this header. As represented in Figure 85, when the friction lock tab is at the right, the ground pin should be the lower of the two pins (when viewed from the top).

Typical System Configuration

The typical configuration shown in Figure 85 summarizes some of the hardware considerations that were discussed in previous sections.

DEVELOPMENT TOOLS

There are two models of development tools available for the ADuC841/ADuC842/ADuC843:

- QuickStart™—Entry-level development system
- QuickStart Plus—Comprehensive development system

These systems are described briefly in the following sections.

QUICKSTART DEVELOPMENT SYSTEM

The QuickStart Development System is an entry-level, low cost development tool suite supporting the parts. The system consists of the following PC based (Windows® compatible) hardware and software development tools.

Hardware	Evaluation board and serial port programming cable.
Software	Serial download software.
Miscellaneous	CD-ROM documentation and prototype device.

A brief description of some of the software tools and components in the QuickStart Development System follows.

Download—In-Circuit Serial Downloader

The serial downloader is a Windows application that allows the user to serially download an assembled program (Intel® hexadecimal format file) to the on-chip program flash memory via the serial COM1 port on a standard PC. Application Note uC004 details this serial download protocol and is available from www.analog.com/microconverter.

ASPIRE—IDE

The ASPIRE integrated development environment is a Windows application that allows the user to compile, edit, and debug code in the same environment. The ASPIRE software allows users to debug code execution on silicon using the MicroConverter UART serial port. The debugger provides access to all on-chip peripherals during a typical debug session as well as single step, animate, and break-point code execution control.

Note that the ASPIRE IDE is also included as part of the QuickStart Plus System. As part of the QuickStart Plus System, the ASPIRE IDE also supports mixed level and C source debug. This is not available in the QuickStart System, but there is an example project that demonstrates this capability.

QuickStart Plus Development System

The QuickStart Plus Development System offers users enhanced nonintrusive debug and emulation tools. The system consists of the following PC based (Windows compatible) hardware and software development tools.

Hardware	Prototype Board. Accutron Nonintrusive Single-Pin Emulator.
Software	ASPIRE Integrated Development Environment. Features full C and assembly emulation using the Accutron single pin emulator.
Miscellaneous	CD-ROM documentation.

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

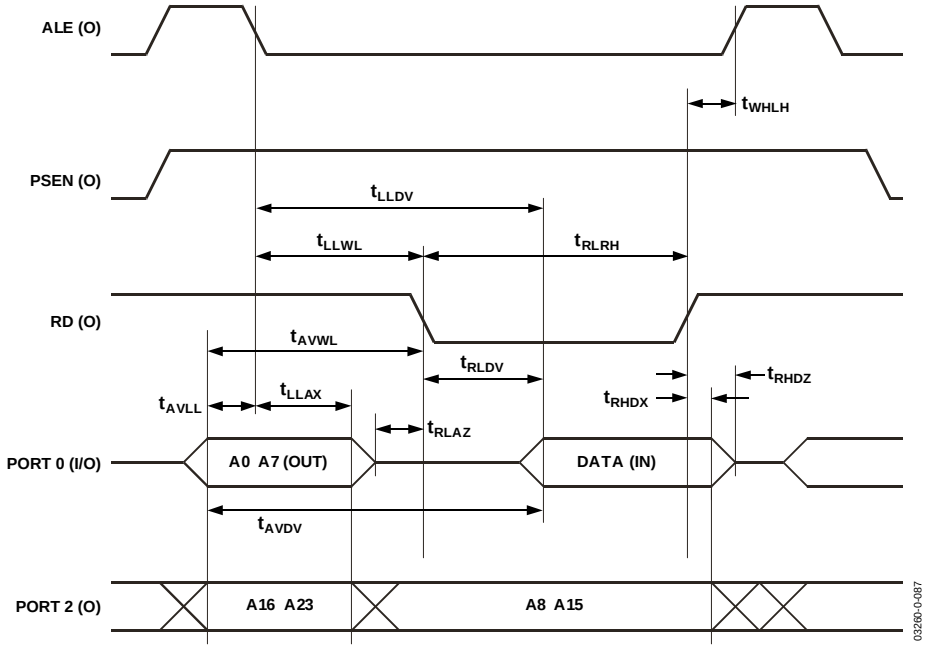

Figure 88. External Data Memory Read Cycle