MicroConverter® 12-Bit ADCs and DACs with Embedded High Speed 62-kB Flash MCU

ADuC841/ADuC842/ADuC843

FEATURES

Pin compatable ugrade of ADuC812/ADuC831/ADuC832
- **Increased performance**
  - **Single-cycle 20 MIPS 8052 core**
  - **High speed 420 kSPS 12-bit ADC**
- **Increased memory**
  - **Up to 62 kBytes on-chip Flash/EE program memory**
  - **4 kBytes on-chip Flash/EE data memory**
- **In-circuit reprogrammable**
  - **Flash/EE, 100 year retention, 100 kCycle endurance**
  - **2304 bytes on-chip data RAM**
- **Smaller package**
  - **8 mm × 8 mm chip scale package**
  - **52-lead PQFP—pin compatable upgrade**

Analog I/O
- **8-channel, 420 kSPS high accuracy, 12-bit ADC**
- **On-chip, 15 ppm/°C voltage reference**
- **DMA controller, high speed ADC-to-RAM capture**
- **Two 12-bit voltage output DACs[1]**
- **Dual output PWM Σ-Δ DACs**
- **On-chip temperature monitor function**

8052 based core
- **8051 compatible instruction set (20 MHz max)**

High performance single-cycle core
- **32 kHz external crystal, on-chip programmable PLL**
- **12 interrupt sources, 2 priority levels**
- **Dual data pointers, extended 11-bit stack pointer**

On-chip peripherals
- **Time interval counter (TIC)**
- **UART, I²C®, and SPI® Serial I/O**
- **Watchdog timer (WDT)**
- **Power supply monitor (PSM)**

Power
- **Normal: 4.5 mA @ 3 V (core CLK = 2.098 MHz)**
- **Power-down: 10 µA @ 3 V[2]**

Development tools
- **Low cost, comprehensive development system incorporating nonintrusive single-pin emulation, IDE based assembly and C source debugging**

APPLICATIONS

Optical networking—laser power control
Base station systems
Precision instrumentation, smart sensors
Transient capture systems
DAS and communications systems

[1] ADuC841/ADuC842 only.
[2] ADuC842/ADuC843 only, ADuC841 driven directly by external crystal.

Rev. 0
Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

FUNCTIONAL BLOCK DIAGRAM

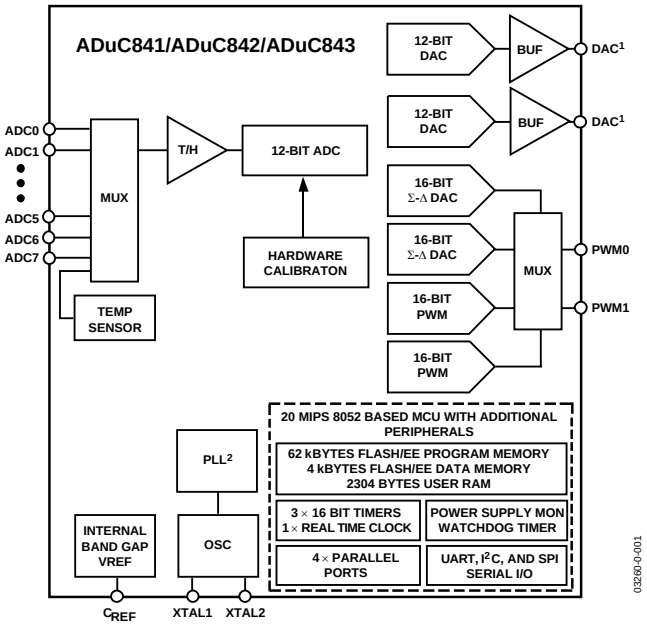


Figure 1.

GENERAL DESCRIPTION

The ADuC841/ADuC842/ADuC843 are complete smart transducer front ends, that integrates a high performance self-calibrating multichannel ADC, a dual DAC, and an optimized single-cycle 20 MHz 8-bit MCU (8051 instruction set compatible) on a single chip.

The ADuC841 and ADuC842 are identical with the exception of the clock oscillator circuit; the ADuC841 is clocked directly from an external crystal up to 20 MHz whereas the ADuC842 uses a 32 kHz crystal with an on-chip PLL generating a programmable core clock up to 16.78 MHz.

The ADuC843 is identical to the ADuC842 except that the ADuC843 has no analog DAC outputs.

The microcontroller is an optimized 8052 core offering up to 20 MIPS peak performance. Three different memory options are available offering up to 62 kBytes of nonvolatile Flash/EE program memory. Four kBytes of nonvolatile Flash/EE data memory, 256 bytes RAM, and 2 kBytes of extended RAM are also integrated on-chip.

(*continued on page 15*)

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 **www.analog.com**
Fax: 781.326.8703 **© 2003 Analog Devices, Inc. All rights reserved.**

TABLE OF CONTENTS

Specifications ... 3

Absolute Maximum Ratings ... 8

ESD Caution ... 8

Pin Configurations and Functional Descriptions ... 9

Terminology ... 11

ADC Specifications ... 11

DAC Specifications ... 11

Typical Performance Characteristics ... 12

Functional Description ... 16

8052 Instruction Set ... 16

Other Single-Cycle Core Features ... 18

Memory Organization ... 19

Special Function Registers (SFRs) ... 20

Accumulator SFR (ACC) ... 21

Special Function Register Banks ... 22

ADC Circuit Information ... 23

Calibrating the ADC ... 30

Nonvolatile Flash/EE Memory ... 31

Using Flash/EE Data Memory ... 34

User Interface to On-Chip Peripherals ... 38

On-Chip PLL ... 41

Pulse-Width Modulator (PWM) ... 42

Serial Peripheral Interface (SPI) ... 45

I^2C Compatible Interface ... 48

Dual Data Pointer ... 51

Power Supply Monitor ... 52

Watchdog Timer ... 53

Time Interval Counter (TIC) ... 54

8052 Compatible On-Chip Peripherals ... 57

Timer/Counter 0 and 1 Operating Modes ... 62

Timer/Counter Operating Modes ... 64

UART Serial Interface ... 65

SBUF ... 65

Interrupt System ... 70

Hardware Design Considerations ... 72

Other Hardware Considerations ... 76

Development Tools ... 77

QuickStart Development System ... 77

Timing Specifications ... 78

Outline Dimensions ... 86

Ordering Guides ... 87

REVISION HISTORY

Revision 0: Initial Version

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	

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
DAC AC CHARACTERISTICS				
Voltage Output Settling Time	15	15	µs typ	Full-scale settling time to within ½ LSB of final value
Digital-to-Analog Glitch Energy	10	10	nV-sec typ	1 LSB change at major carry
DAC CHANNEL SPECIFICATIONS[12, 13]				
Internal Buffer Disabled ADuC841/ADuC842 Only				
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	±5	±5	mV max	V_{REF} range
Gain Error	±0.5	±0.5	% typ	V_{REF} range
Gain Error Mismatch[4]	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
REFERENCE INPUT/OUTPUT REFERENCE OUTPUT[14]				
Output Voltage (V_{REF})	2.5	2.5	V	
Accuracy	±10	±10	mV Max	Of V_{REF} measured at the C_{REF} pin T_A = 25°C
Power Supply Rejection	65	67	dB typ	
Reference Temperature Coefficient	±15	±15	ppm/°C typ	
Internal V_{REF} Power-On Time	2	2	ms typ	
EXTERNAL REFERENCE INPUT[15]				
Voltage Range (V_{REF})[4]	1	1	V min	
	V_{DD}	V_{DD}	V max	
Input Impedance	20	20	kΩ typ	
Input Leakage	1	1	µA max	Internal band gap deselected via ADCCON1.6
POWER SUPPLY MONITOR (PSM)				
DV_{DD} Trip Point Selection Range		2.93	V min	Two trip points selectable in this range programmed via TPD1–0 in PSMCON, 3 V part only
		3.08	V max	
DV_{DD} Power Supply Trip Point Accuracy		±2.5	% max	
WATCHDOG TIMER (WDT)[4]				
Timeout Period	0	0	ms min	Nine timeout periods selectable in this range
	2000	2000	ms max	
FLASH/EE MEMORY RELIABILITY CHARACTERISTICS[16]				
Endurance[17]	100,000	100,000	Cycles min	
Data Retention[18]	100	100	Years min	
DIGITAL INPUTS				
Input Leakage Current (Port 0, $\overline{EA}$)	±10	±10	µA max	V_{IN} = 0 V or V_{DD}
	±1	±1	µA typ	V_{IN} = 0 V or V_{DD}
Logic 1 Input Current				
(All Digital Inputs), SDATA, SCLOCK	±10	±10	µA max	V_{IN} = V_{DD}
	±1	±1	µA typ	V_{IN} = V_{DD}
Logic 0 Input Current (Ports 1, 2, 3) SDATA, SCLOCK	–75	–25	µA max	
	–40	–15	µA typ	V_{IL} = 450 mV
Logic 1 to Logic 0 Transition Current (Ports 2 and 3)	–660	–250	µA max	V_{IL} = 2 V
	–400	–140	µA typ	V_{IL} = 2 V
RESET	±10	±10	µA max	V_{IN} = 0 V
	10	5	µA min	V_{IN} = 5 V, 3 V Internal Pull Down
	105	35	µA max	V_{IN} = 5 V, 3 V Internal Pull Down

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

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
POWER REQUIREMENTS[19, 20]				
Power Supply Voltages				
AV_{DD}/DV_{DD} – AGND		2.7	V min	AV_{DD}/DV_{DD} = 3 V nom
		3.6	V max	
	4.75		V min	AV_{DD}/DV_{DD} = 5 V nom
	5.25		V max	
Power Supply Currents **Normal** Mode[21]				
DV_{DD} Current[4]	10	4.5	mA typ	Core CLK = 2.097 MHz
AV_{DD} Current	1.7	1.7	mA max	Core CLK = 2.097 MHz
DV_{DD} Current	38	12	mA max	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
	33	10	mA typ	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
AV_{DD} Current	1.7	1.7	mA max	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
DV_{DD} Current[4]	45	N/A	mA max	Core CLK = 20MHz **ADuC841 Only**
Power Supply Currents **Idle** Mode[21]				
DV_{DD} Current	4.5	2.2	mA typ	Core CLK = 2.097 MHz
AV_{DD} Current	3	2	µA typ	Core CLK = 2.097 MHz
DV_{DD} Current[4]	12	5	mA max	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
	10	3.5	mA typ	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
AV_{DD} Current	3	2	µA typ	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
Power Supply Currents **Power-Down** Mode[21]				Core CLK = any frequency
DV_{DD} Current	28	18	µA max	Oscillator Off / TIMECON.1 = 0
	20	10	µA typ	
AV_{DD} Current	2	1	µA typ	Core CLK = any frequency
				ADuC841 Only
DV_{DD} Current[4]	3	1	mA max	TIMECON.1 = 1
DV_{DD} Current[4]	50	22	µA max	Core CLK = any frequency
	40	15	µA typ	**ADuC842/ADuC843 Only**
				Oscillator On
Typical Additional Power Supply Currents				
PSM Peripheral	15	10	µA typ	$AV_{DD} = DV_{DD}$
ADC[4]	1.0	1.0	mA min	MCLK Divider = 32
	2.8	1.8	mA max	MCLK Divider = 2
DAC	150	130	µA typ	

See footnotes on the next page.

[1] Temperature Range –40°C to +85°C.
[2] ADC linearity is guaranteed during normal MicroConverter core operation.
[3] ADC LSB size = $V_{REF}/2^{12}$, i.e., 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.

ABSOLUTE MAXIMUM RATINGS

Table 2. T_A = 25°C, unless otherwise noted

Parameter	Rating
AV_{DD} to DV_{DD}	−0.3 V to +0.3 V
AGND to DGND	−0.3 V to +0.3 V
DV_{DD} to DGND, AV_{DD} to AGND	−0.3 V to +7 V
Digital Input Voltage to DGND	−0.3 V to DV_{DD} + 0.3 V
Digital Output Voltage to DGND	−0.3 V to DV_{DD} + 0.3 V
V_{REF} to AGND	−0.3 V to AV_{DD} + 0.3 V
Analog Inputs to AGND	−0.3 V to AV_{DD} + 0.3 V
Operating Temperature Range, Industrial ADuC841BS,ADuC842BS,ADuC843BS ADuC841BCP, ADuC842BCP, ADuC843BCP	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Junction Temperature	150°C
θ_{JA} Thermal Impedance (ADuC84xBS)	90°C/W
θ_{JA} Thermal Impedance (ADuC84xBCP)	52°C/W
Lead Temperature, Soldering	
Vapor Phase (60 sec)	215°C
Infrared (15 sec)	220°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although this product features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

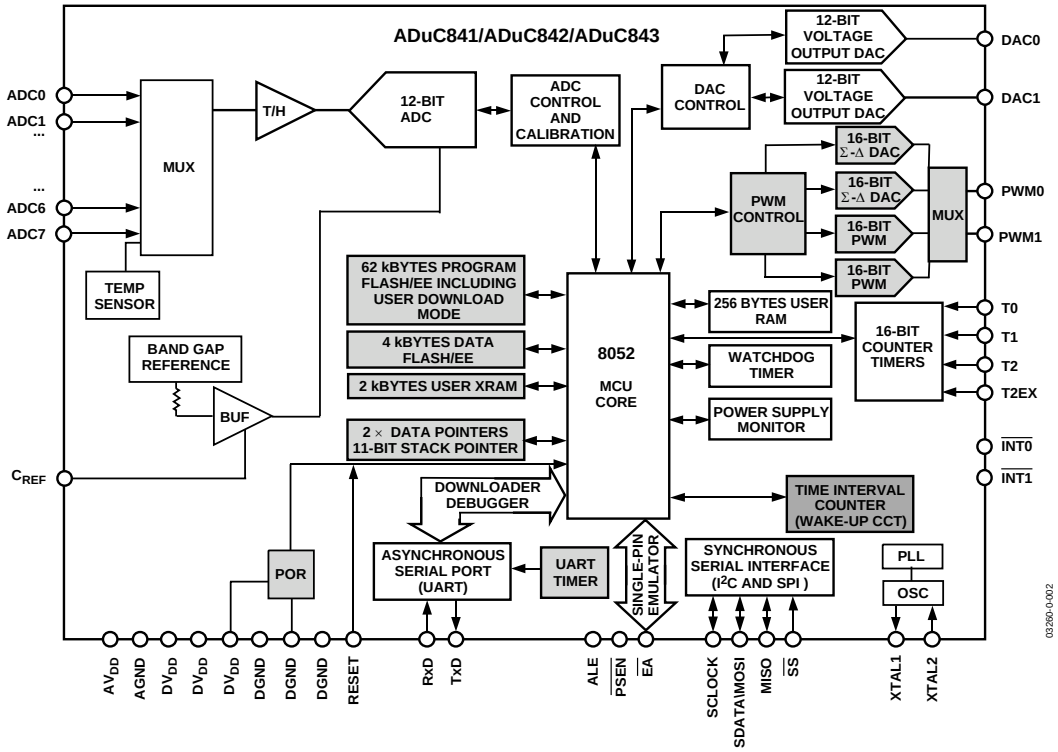


Figure 2. ADuC Block Diagram (Shaded Areas are Features Not Present on the ADuC812), No DACs on ADuC843, PLL on ADuC842/ADuC843 Only.

PIN CONFIGURATIONS AND FUNCTIONAL DESCRIPTIONS

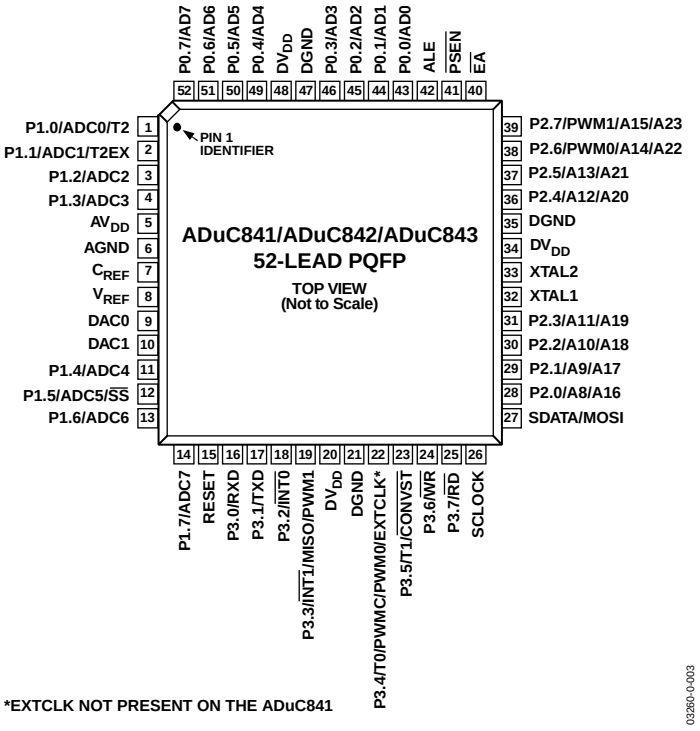


Figure 3. 52-Lead PQFP

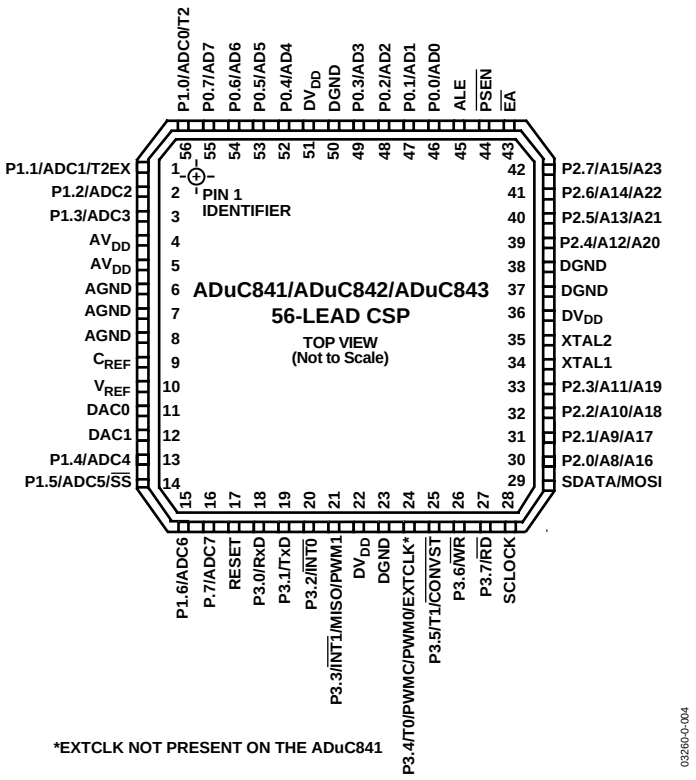


Figure 4. 56-Lead CSP

Table 3. Pin Function Descriptions

Mnemonic	Type	Function
DV_{DD}	P	Digital Positive Supply Voltage. 3 V or 5 V nominal.
AV_{DD}	P	Analog Positive Supply Voltage. 3 V or 5 V nominal.
C_{REF}	I/O	Decoupling Input for On-Chip Reference. Connect a 0.47 µF capacitor between this pin and AGND.
V_{REF}	NC	Not connected. This was reference out on the ADuC812; the C_{REF} pin should be used instead.
AGND	G	Analog Ground. Ground reference point for the analog circuitry.
P1.0–P1.7	I	Port 1 is an 8-bit input port only. Unlike other ports, Port 1 defaults to analog input mode. To configure any of these port pins as a digital input, write a 0 to the port bit.
ADC0–ADC7	I	Analog Inputs. Eight single-ended analog inputs. Channel selection is via ADCCON2 SFR.
T2	I	Timer 2 Digital Input. Input to Timer/Counter 2. When enabled, Counter 2 is incremented in response to a 1-to-0 transition of the T2 input.
T2EX	I	Digital Input. Capture/reload trigger for Counter 2; also functions as an up/down control input for Counter 2.
$\overline{SS}$	I	Slave Select Input for the SPI Interface.
SDATA	I/O	User Selectable, I^2C Compatible, or SPI Data Input/Output Pin.
SCLOCK	I/O	Serial Clock Pin for I^2C Compatible or for SPI Serial Interface Clock.
MOSI	I/O	SPI Master Output/Slave Input Data I/O Pin for SPI Interface.
MISO	I/O	SPI Master Input/Slave Output Data I/O Pin for SPI Serial Interface.
DAC0	O	Voltage Output from DAC0. This pin is a no connect on the ADuC843.
DAC1	O	Voltage Output from DAC1. This pin is a no connect on the ADuC843.
RESET	I	Digital Input. A high level on this pin for 24 master clock cycles while the oscillator is running resets the device.

Mnemonic	Type	Function
P3.0–P3.7	I/O	Port 3 is a bidirectional port with internal pull-up resistors. Port 3 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 3 pins being pulled externally low source current because of the internal pull-up resistors. Port 3 pins also contain various secondary functions, which are described below.
PWMC	I	PWM Clock Input.
PWM0	O	PWM0 Voltage Output. PWM outputs can be configured to use Ports 2.6 and 2.7 or Ports 3.4 and 3.3.
PWM1	O	PWM1 Voltage Output. See the CFG841/CFG842 register for further information.
RxD	I/O	Receiver Data Input (Asynchronous) or Data Input/Output (Synchronous) of the Serial (UART) Port.
TxD	O	Transmitter Data Output (Asynchronous) or Clock Output (Synchronous) of the Serial (UART) Port.
$\overline{\text{INT0}}$	I	Interrupt 0. Programmable edge or level triggered interrupt input; can be programmed to one of two priority levels. This pin can also be used as a gate control input to Timer 0.
$\overline{\text{INT1}}$	I	Interrupt 1. Programmable edge or level triggered interrupt input; can be programmed to one of two priority levels. This pin can also be used as a gate control input to Timer 1.
T0	I	Timer/Counter 0 Input.
T1	I	Timer/Counter 1 Input.
$\overline{\text{CONVST}}$	I	Active Low Convert Start Logic Input for the ADC Block when the External Convert Start Function is Enabled. A low-to-high transition on this input puts the track-and-hold into hold mode and starts the conversion.
EXTCLK	I	Input for External Clock Signal. Has to be enabled via the CFG842 register.
$\overline{\text{WR}}$	O	Write Control Signal, Logic Output. Latches the data byte from Port 0 into the external data memory.
$\overline{\text{RD}}$	O	Read Control Signal, Logic Output. Enables the external data memory to Port 0.
XTAL2	O	Output of the Inverting Oscillator Amplifier.
XTAL1	I	Input to the Inverting Oscillator Amplifier.
DGND	G	Digital Ground. Ground reference point for the digital circuitry.
P2.0–P2.7 (A8–A15) (A16–A23)	I/O	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. Port 2 emits the middle and high-order address bytes during accesses to the external 24-bit external data memory space.
$\overline{\text{PSEN}}$	O	Program Store Enable, Logic Output. This pin remains low during internal program execution. $\overline{\text{PSEN}}$ is used to enable serial download mode when pulled low through a resistor on power-up or reset. On reset this pin will momentarily become an input and the status of the pin is sampled. If there is no pulldown resistor in place the pin will go momentarilly high and then user code will execute. If a pull-down resistor is in place, the embedded serial download/debug kernel will execute.
ALE	O	Address Latch Enable, Logic Output. This output is used to latch the low byte and page byte for 24-bit address space accesses of the address into external data memory.
$\overline{\text{EA}}$	I	External Access Enable, Logic Input. When held high, this input enables the device to fetch code from internal program memory locations. The parts do not support external code memory. This pin should not be left floating.
P0.7–P0.0 (A0-A7)	I/O	Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high impedance inputs. Port 0 is also the multiplexed low-order address and data bus during accesses to external data memory. In this application, it uses strong internal pull-ups when emitting 1s.

Types: P = Power, G = Ground, I= Input, O = Output., NC = No Connect

TERMINOLOGY

ADC SPECIFICATIONS

Integral Nonlinearity

The maximum deviation of any code from a straight line passing through the endpoints of the ADC transfer function. The endpoints of the transfer function are zero scale, a point ½ LSB below the first code transition, and full scale, a point ½ LSB above the last code transition.

Differential Nonlinearity

The difference between the measured and the ideal 1 LSB change between any two adjacent codes in the ADC.

Offset Error

The deviation of the first code transition (0000 . . . 000) to (0000 . . . 001) from the ideal, i.e., +½ LSB.

Gain Error

The deviation of the last code transition from the ideal AIN voltage (Full Scale – ½ LSB) after the offset error has been adjusted out.

Signal-to-(Noise + Distortion) Ratio

The measured ratio of signal to (noise + distortion) at the output of the ADC. The signal is the rms amplitude of the fundamental. Noise is the rms sum of all nonfundamental signals up to half the sampling frequency ($f_S/2$), excluding dc. The ratio depends on the number of quantization levels in the digitization process; the more levels, the smaller the quantization noise. The theoretical signal-to-(noise + distortion) ratio for an ideal N-bit converter with a sine wave input is given by

$$\textit{Signal-to-(Noise + Distortion)} = (6.02N + 1.76)\ dB$$

Thus for a 12-bit converter, this is 74 dB.

Total Harmonic Distortion (THD)

The ratio of the rms sum of the harmonics to the fundamental.

DAC SPECIFICATIONS

Relative Accuracy

Relative accuracy or endpoint linearity is a measure of the maximum deviation from a straight line passing through the endpoints of the DAC transfer function. It is measured after adjusting for zero error and full-scale error.

Voltage Output Settling Time

The amount of time it takes for the output to settle to a specified level for a full-scale input change.

Digital-to-Analog Glitch Impulse

The amount of charge injected into the analog output when the inputs change state. It is specified as the area of the glitch in nV-sec.

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.

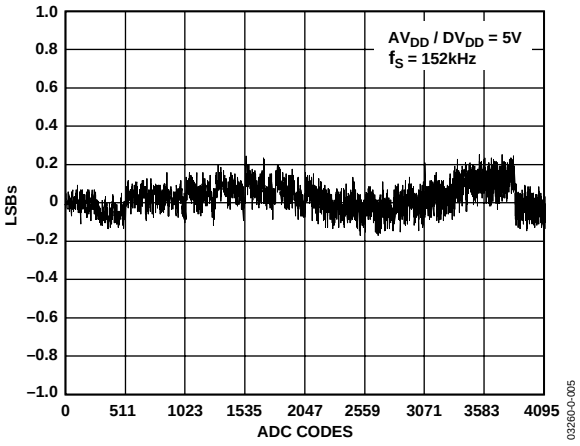


Figure 5. Typical INL Error, V_{DD} = 5 V, f_s = 152 kHz

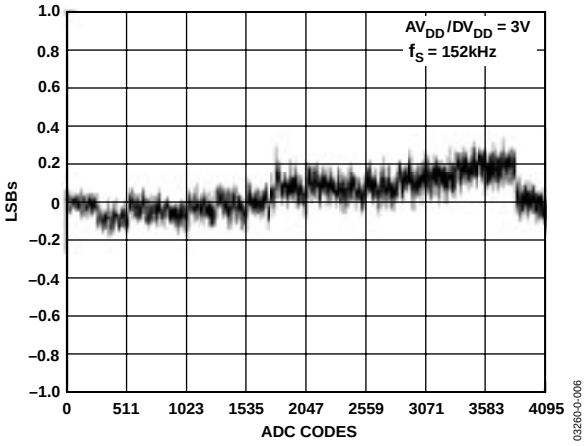


Figure 6. Typical INL Error, V_{DD} = 3 V, f_s = 152 kHz

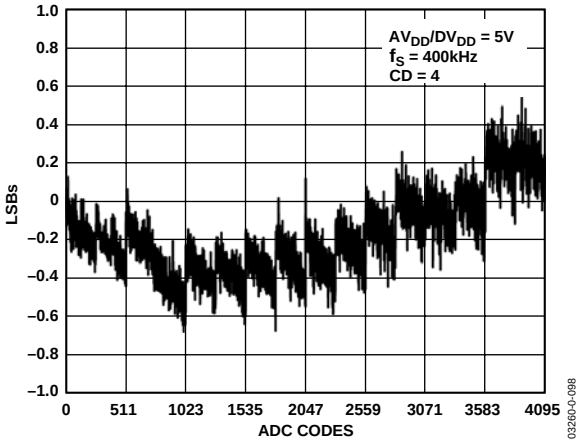


Figure 7. Typical INL Error, V_{DD} = 5 V, f_S = 400 kHz

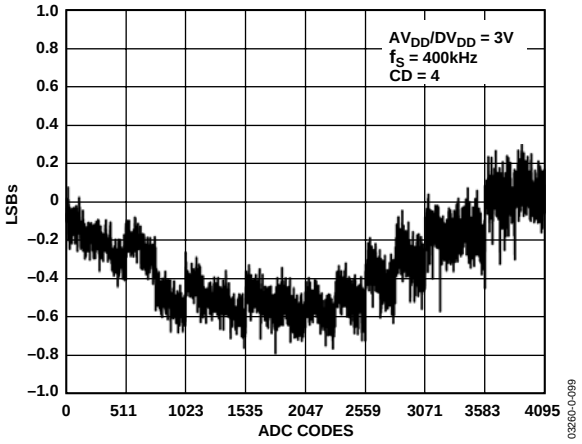


Figure 8. Typical INL Error, V_{DD} = 3 V, f_S = 400 kHz

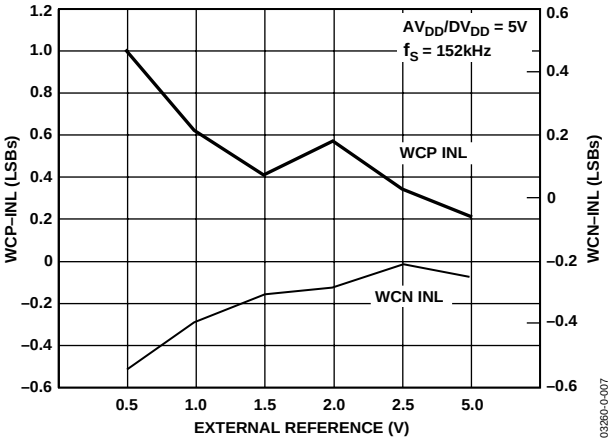


Figure 9. Typical Worst-Case INL Error vs. V_{REF}, V_{DD} = 5 V

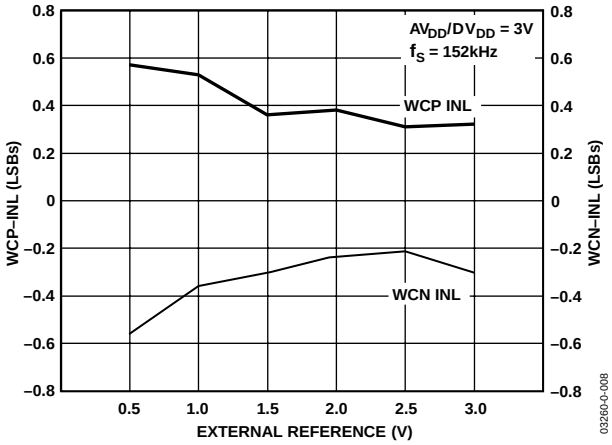


Figure 10. Typical Worst-Case INL Error vs. V_{REF}, V_{DD} = 3 V

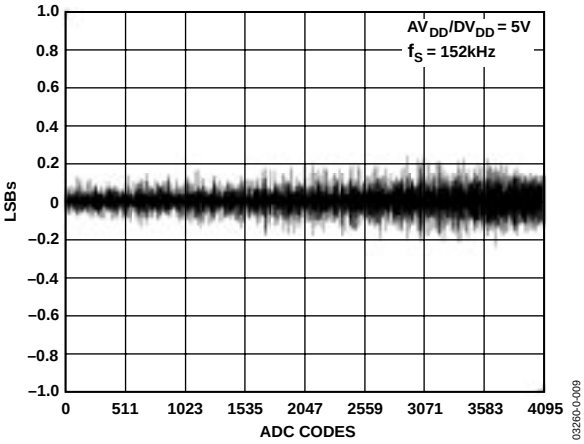


Figure 11. Typical DNL Error, V_{DD} = 5 V

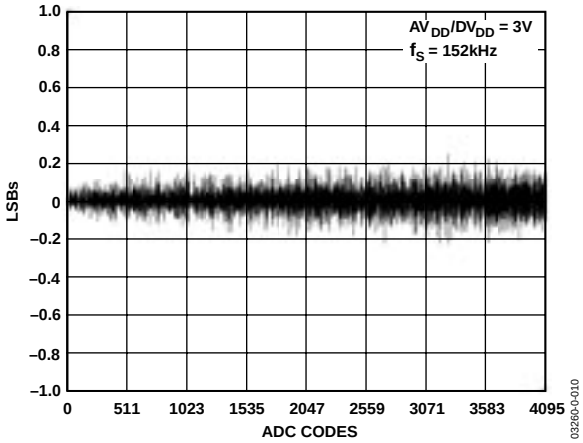


Figure 12. Typical DNL Error, V_{DD} = 3 V

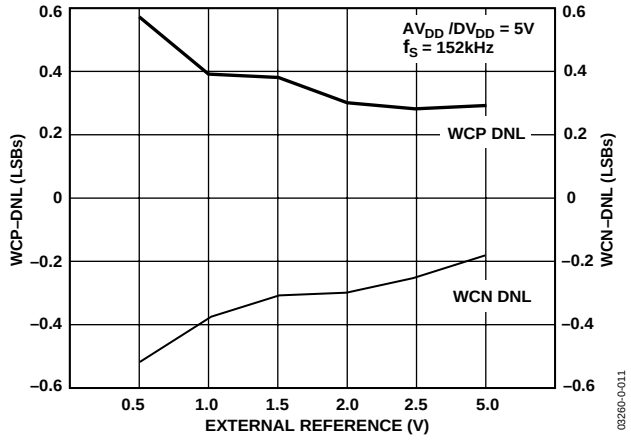


Figure 13. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 5 V

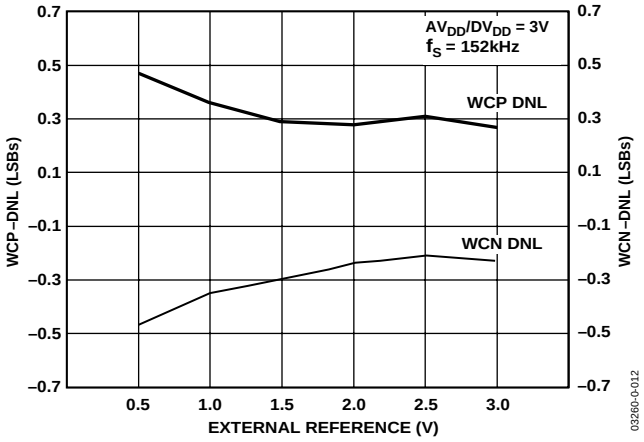


Figure 14. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 3 V

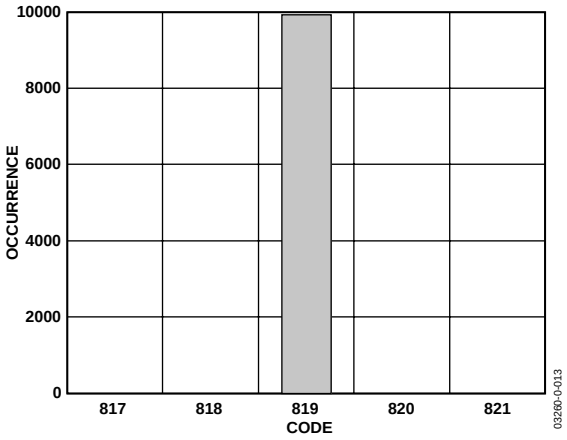


Figure 15. Code Histogram Plot, V_{DD} = 5 V

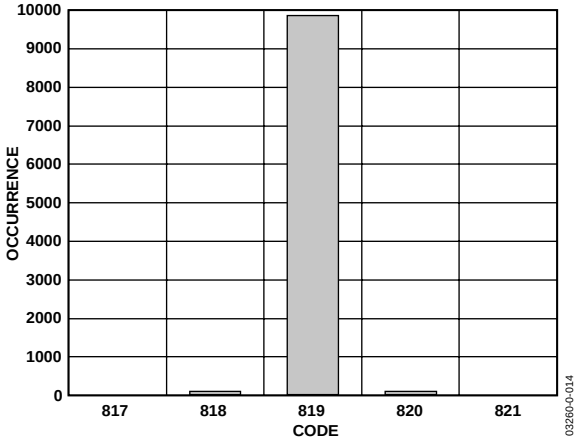


Figure 16. Code Histogram Plot, V_{DD} = 3 V

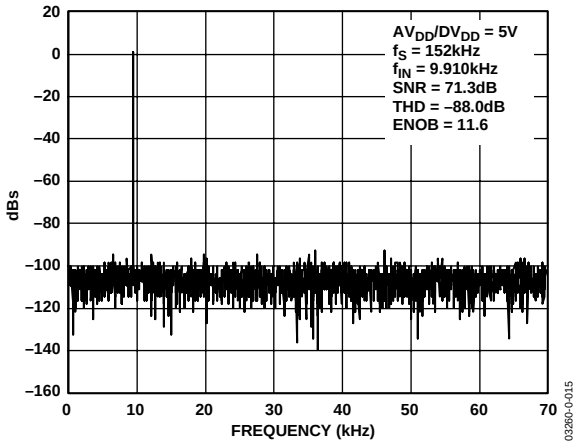


Figure 17. Dynamic Performance at V_{DD} = 5 V

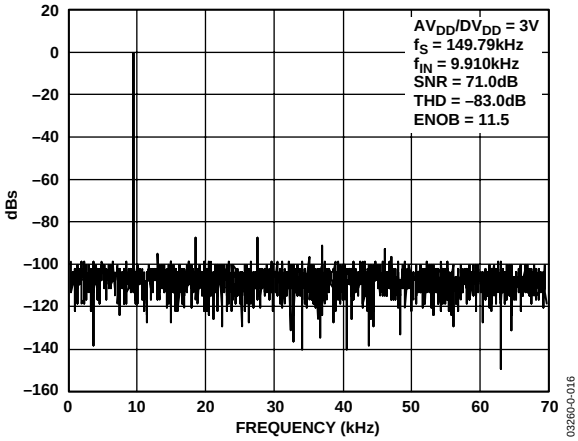


Figure 18. Dynamic Performance at V_{DD} = 3 V

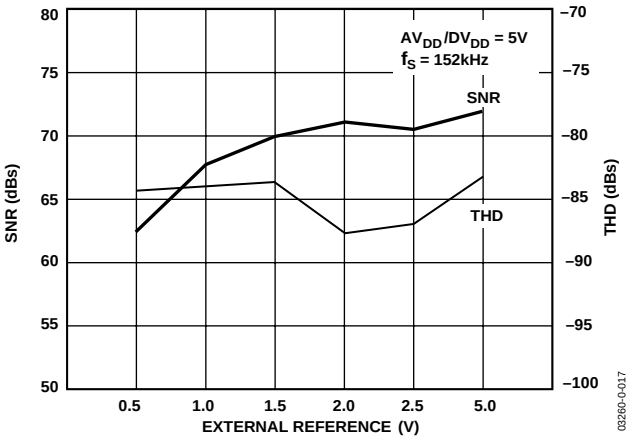


Figure 19. Typical Dynamic Performance vs. V_{REF}, V_{DD} = 5 V

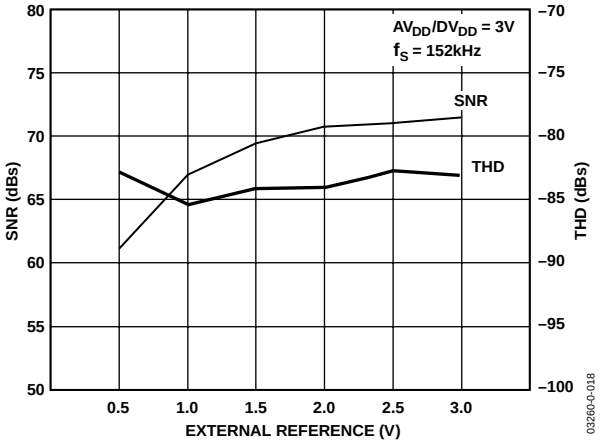


Figure 20. Typical Dynamic Performance vs. V_{REF}, V_{DD} = 3 V

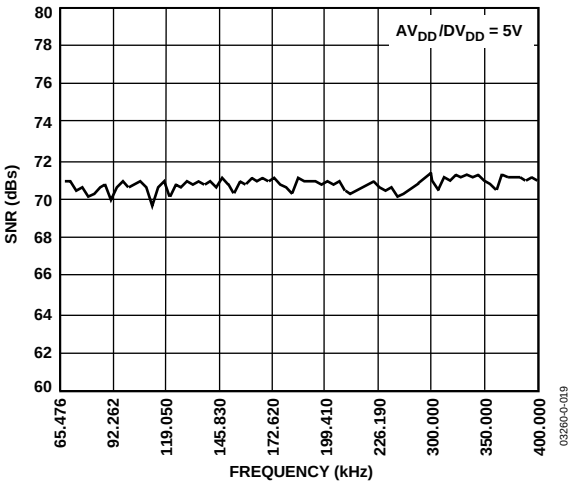


Figure 21. Typical Dynamic Performance vs. Sampling Frequency

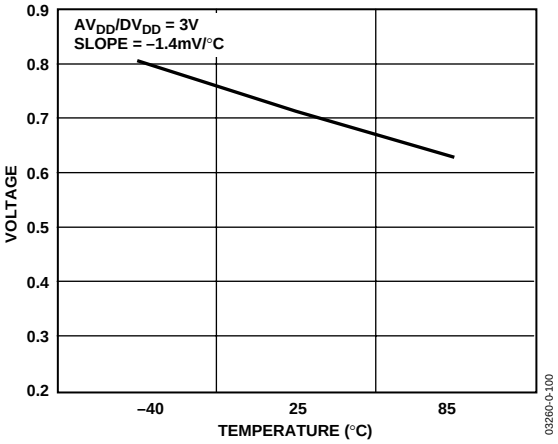


Figure 22. Typical Temperature Sensor Output vs. Temperature

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 ∑-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.

FUNCTIONAL DESCRIPTION

8052 INSTRUCTION SET

Table 4 documents the number of clock cycles required for each instruction. Most instructions are executed in one or two clock cycles, resulting in a 16 MIPS peak performance when operating at PLLCON = 00H on the ADuC842/ADuC843. On the ADuC841, 20 MIPS peak performance is possible with a 20 MHz external crystal.

Table 4. Instructions

Mnemonic	Description	Bytes	Cycles
Arithmetic			
ADD A,Rn	Add register to A	1	1
ADD A,@Ri	Add indirect memory to A	1	2
ADD A,dir	Add direct byte to A	2	2
ADD A,#data	Add immediate to A	2	2
ADDC A,Rn	Add register to A with carry	1	1
ADDC A,@Ri	Add indirect memory to A with carry	1	2
ADDC A,dir	Add direct byte to A with carry	2	2
ADD A,#data	Add immediate to A with carry	2	2
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	2
SUBB A,dir	Subtract direct from A with borrow	2	2
SUBB A,#data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC @Ri	Increment indirect memory	1	2
INC dir	Increment direct byte	2	2
INC DPTR	Increment data pointer	1	3
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC @Ri	Decrement indirect memory	1	2
DEC dir	Decrement direct byte	2	2
MUL AB	Multiply A by B	1	9
DIV AB	Divide A by B	1	9
DA A	Decimal adjust A	1	2
Logic			
ANL A,Rn	AND register to A	1	1
ANL A,@Ri	AND indirect memory to A	1	2
ANL A,dir	AND direct byte to A	2	2
ANL A,#data	AND immediate to A	2	2
ANL dir,A	AND A to direct byte	2	2
ANL dir,#data	AND immediate data to direct byte	3	3
ORL A,Rn	OR register to A	1	1
ORL A,@Ri	OR indirect memory to A	1	2
ORL A,dir	OR direct byte to A	2	2
ORL A,#data	OR immediate to A	2	2
ORL dir,A	OR A to direct byte	2	2
ORL dir,#data	OR immediate data to direct byte	3	3
XRL A,Rn	Exclusive-OR register to A	1	1
XRL A,@Ri	Exclusive-OR indirect memory to A	2	2
XRL A,#data	Exclusive-OR immediate to A	2	2
XRL dir,A	Exclusive-OR A to direct byte	2	2

Mnemonic	Description	Bytes	Cycles
XRL A,dir	Exclusive-OR indirect memory to A	2	2
XRL dir,#data	Exclusive-OR immediate data to direct	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
Data Transfer			
MOV A,Rn	Move register to A	1	1
MOV A,@Ri	Move indirect memory to A	1	2
MOV Rn,A	Move A to register	1	1
MOV @Ri,A	Move A to indirect memory	1	2
MOV A,dir	Move direct byte to A	2	2
MOV A,#data	Move immediate to A	2	2
MOV Rn,#data	Move register to immediate	2	2
MOV dir,A	Move A to direct byte	2	2
MOV Rn, dir	Move register to direct byte	2	2
MOV dir, Rn	Move direct to register	2	2
MOV @Ri,#data	Move immediate to indirect memory	2	2
MOV dir,@Ri	Move indirect to direct memory	2	2
MOV @Ri,dir	Move direct to indirect memory	2	2
MOV dir,dir	Move direct byte to direct byte	3	3
MOV dir,#data	Move immediate to direct byte	3	3
MOV DPTR,#data	Move immediate to data pointer	3	3
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	4
MOVC A,@A+PC	Move code byte relative PC to A	1	4
MOVX A,@Ri	Move external (A8) data to A	1	4
MOVX A,@DPTR	Move external (A16) data to A	1	4
MOVX @Ri,A	Move A to external data (A8)	1	4
MOVX @DPTR,A	Move A to external data (A16)	1	4
PUSH dir	Push direct byte onto stack	2	2
POP dir	Pop direct byte from stack	2	2
XCH A,Rn	Exchange A and register	1	1
XCH A,@Ri	Exchange A and indirect memory	1	2
XCHD A,@Ri	Exchange A and indirect memory nibble	1	2
XCH A,dir	Exchange A and direct byte	2	2
Boolean			
CLR C	Clear carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement carry	1	1
CPL bit	Complement direct bit	2	2
ANL C,bit	AND direct bit and carry	2	2
ANL C,/bit	AND direct bit inverse to carry	2	2
ORL C,bit	OR direct bit and carry	2	2
ORL C,/bit	OR direct bit inverse to carry	2	2
MOV C,bit	Move direct bit to carry	2	2
MOV bit,C	Move carry to direct bit	2	2

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.

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.

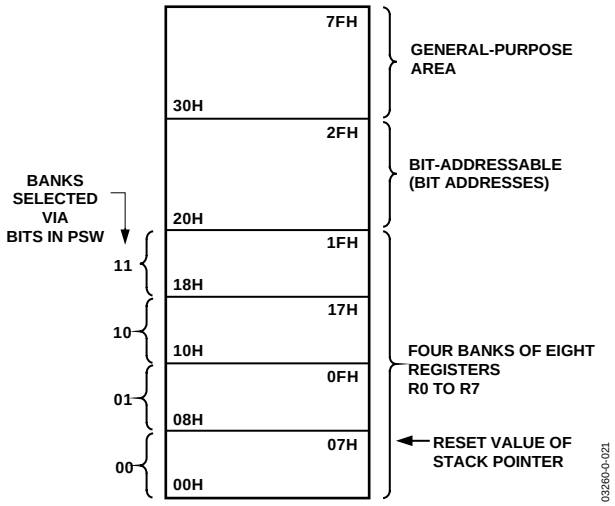


Figure 23. Lower 128 Bytes of Internal Data Memory

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.

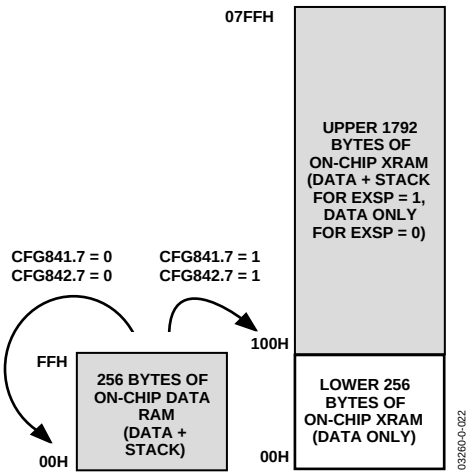


Figure 24. Extended Stack Pointer Operation

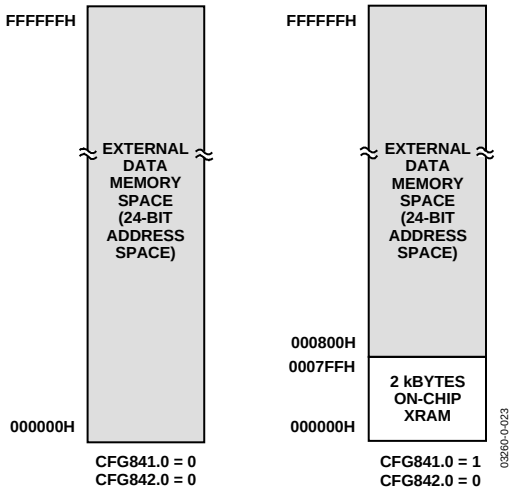


Figure 25. Internal and External XRAM

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.

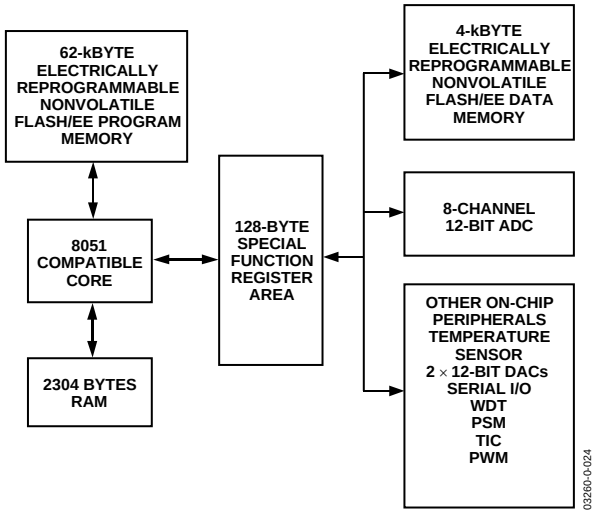


Figure 26. Programming Model

ACCUMULATOR SFR (ACC)

ACC is the accumulator register and is used for math operations including addition, subtraction, integer multiplication and division, and Boolean bit manipulations. The mnemonics for accumulator-specific instructions refer to the accumulator as A.

B SFR (B)

The B register is used with the ACC for multiplication and division operations. For other instructions, it can be treated as a general-purpose scratchpad register.

Stack Pointer (SP and SPH)

The SP SFR is the stack pointer and is used to hold an internal RAM address that is called the top of the stack. The SP register is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the SP register is initialized to 07H after a reset, which causes the stack to begin at location 08H.

As mentioned earlier, the parts offer an extended 11-bit stack pointer. The 3 extra bits used to make up the 11-bit stack pointer are the 3 LSBs of the SPH byte located at B7H.

Data Pointer (DPTR)

The data pointer is made up of three 8-bit registers named DPP (page byte), DPH (high byte), and DPL (low byte). These are used to provide memory addresses for internal and external code access and for external data access. They may be manipulated as a 16-bit register (DPTR = DPH, DPL), although INC DPTR instructions automatically carry over to DPP, or as three independent 8-bit registers (DPP, DPH, DPL). The parts support dual data pointers. Refer to the Dual Data Pointer section.

Program Status Word (PSW)

The PSW SFR contains several bits reflecting the current status of the CPU, as detailed in Table 5.

SFR Address	D0H
Power-On Default	00H
Bit Addressable	Yes

Table 5. PSW SFR Bit Designations

Bit	Name	Description		
7	CY	Carry Flag.		
6	AC	Auxiliary Carry Flag.		
5	F0	General-Purpose Flag.		
4	RS1	Register Bank Select Bits.		
3	RS0	RS1	RS0	Selected Bank
		0	0	0
		0	1	1
		1	0	2
		1	1	3
2	OV	Overflow Flag.		
1	F1	General-Purpose Flag.		
0	P	Parity Bit.		

Power Control SFR (PCON)

The PCON SFR contains bits for power-saving options and general-purpose status flags, as shown in Table 6.

SFR Address	87H
Power-On Default	00H
Bit Addressable	No

Table 6. PCON SFR Bit Designations

Bit No.	Name	Description
7	SMOD	Double UART Baud Rate.
6	SERIPD	I^2C/SPI Power-Down Interrupt Enable.
5	INT0PD	$\overline{INT0}$ Power-Down Interrupt Enable.
4	ALEOFF	Disable ALE Output.
3	GF1	General-Purpose Flag Bit.
2	GF0	General-Purpose Flag Bit.
1	PD	Power-Down Mode Enable.
0	IDL	Idle Mode Enable.

SPECIAL FUNCTION REGISTER BANKS

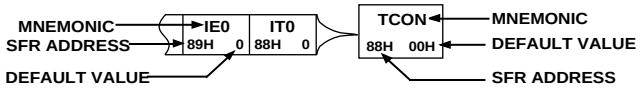
All registers except the program counter and the four general-purpose register banks reside in the special function register (SFR) area. The SFR registers include control, configuration, and data registers, which provide an interface between the CPU and other on-chip peripherals. Figure 27 shows a full SFR memory map and SFR contents on reset. Unoccupied SFR locations are shown dark-shaded in the figure (NOT USED). Unoccupied locations in the SFR address space are not implemented, i.e., no register exists at this location. If an unoccupied location is read, an unspecified value is returned. SFR locations reserved for on-chip testing are shown lighter shaded (RESERVED) and should not be accessed by user software. Sixteen of the SFR locations are also bit addressable and denoted by 1 in Figure 27, i.e., the bit addressable SFRs are those whose address ends in 0H or 8H.

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS
I2CSI/MDO EFH 0	I2CGC/MDE EEH 0	I2C1O1MCO EDH 0	I2C1O0/MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS
ADCI DFH 0	DMA DEH 0	CCONV DDH 0	SCONV DCH 0	CS3 DBH 0	CS2 DAH 0	CS1 D9H 0	CS0 D8H 0	BITS
CY D7H 0	AC D6H 0	F0 D5H 0	RS1 D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS
PSI BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	TI 99H 0	RI 98H 0	BITS
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS

SPICON[1] F8H 04H	DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	RESERVED
B[1] F0H 00H	ADCOFSL[3] F1H 00H	ADCOFSH[3] F2H 20H	ADCGAINL[3] F3H 00H	ADCGAINH[3] F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H
I2CCON[1] E8H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 40H
ACC[1] E0H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
ADCCON2[1] D8H 00H	ADCDATAL D9H 00H	ADCDATAH DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH
PSW[1] D0H 00H	RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	PLLCON D7H 53H
T2CON[1] C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
WDCON[1] C0H 10H	RESERVED	CHIPID C2H XXH	RESERVED	RESERVED	RESERVED	EDARL C6H 00H	EDARH C7H 00H
IP[1] B8H 00H	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
P3[1] B0H FFH	PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	NOT USED	NOT USED	SPH B7H 00H
IE[1] A8H 00H	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG841/ CFG842 AFH 00H
P2[1] A0H FFH	TIMECON A1H 00H	HTHSEC A2H 00H	SEC A3H 00H	MIN A4H 00H	HOUR A5H 00H	INTVAL A6H 00H	DPCON A7H 00H
SCON[1] 98H 00H	SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	NOT USED
P1[1, 2] 90H FFH	I2CADD1 91H 7FH	I2CADD2 92H 7FH	I2CADD3 93H 7FH	NOT USED	NOT USED	NOT USED	NOT USED
TCON[1] 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
P0[1] 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

SFR MAP KEY:

THESE BITS ARE CONTAINED IN THIS BYTE.



NOTES

[1]SFRs WHOSE ADDRESS ENDS IN 0H OR 8H ARE BIT ADDRESSABLE.

[2]THE PRIMARY FUNCTION OF PORT1 IS AS AN ANALOG INPUT PORT; THEREFORE, TO ENABLE THE DIGITAL SECONDARY FUNCTIONS ON THESE PORT PINS, WRITE A 0 TO THE CORRESPONDING PORT 1 SFR BIT.

[3]CALIBRATION COEFFICIENTS ARE PRECONFIGURED ON POWER-UP TO FACTORY CALIBRATED VALUES.

03260-0-025

Figure 27. Special Function Register Locations and Reset Values

ADC CIRCUIT INFORMATION

General Overview

The ADC conversion block incorporates a fast, 8-channel, 12-bit, single-supply ADC. This block provides the user with multichannel mux, track-and-hold, on-chip reference, calibration features, and ADC. All components in this block are easily configured via a 3-register SFR interface.

The ADC converter consists of a conventional successive approximation converter based around a capacitor DAC. The converter accepts an analog input range of 0 V to V_{REF}. A high precision, 15 ppm, low drift, factory calibrated 2.5 V reference is provided on-chip. An external reference can be connected as described in the Voltage Reference Connections section. This external reference can be in the range 1 V to AV_{DD}.

Single-step or continuous conversion modes can be initiated in software or alternatively by applying a convert signal to an external pin. Timer 2 can also be configured to generate a repetitive trigger for ADC conversions. The ADC may be configured to operate in a DMA mode whereby the ADC block continuously converts and captures samples to an external RAM space without any interaction from the MCU core. This automatic capture facility can extend through a 16 MByte external data memory space.

The ADuC841/ADuC842/ADuC843 are shipped with factory programmed calibration coefficients that are automatically downloaded to the ADC on power-up, ensuring optimum ADC performance. The ADC core contains internal offset and gain calibration registers that can be hardware calibrated to minimize system errors.

A voltage output from an on-chip band gap reference proportional to absolute temperature can also be routed through the front end ADC multiplexer (effectively a 9th ADC channel input), facilitating a temperature sensor implementation.

ADC Transfer Function

The analog input range for the ADC is 0 V to V_{REF}. For this range, the designed code transitions occur midway between successive integer LSB values, i.e., 0.5 LSB, 1.5 LSB, 2.5 LSB . . . FS –1.5 LSB. The output coding is straight binary with 1 LSB = FS/4096 or 2.5 V/4096 = 0.61 mV when V_{REF} = 2.5 V. The ideal input/output transfer characteristic for the 0 V to V_{REF} range is shown in Figure 28.

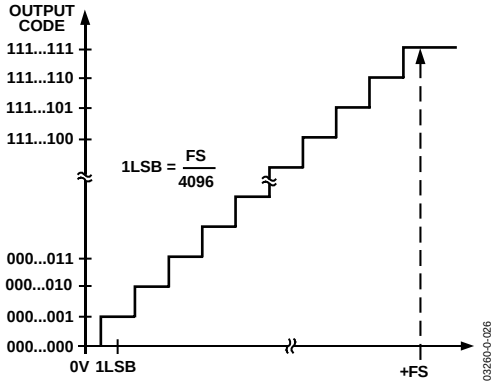


Figure 28. ADC Transfer Function

Typical Operation

Once configured via the ADCCON 1–3 SFRs, the ADC converts the analog input and provides an ADC 12-bit result word in the ADCDATAH/L SFRs. The top 4 bits of the ADCDATAH SFR are written with the channel selection bits to identify the channel result. The format of the ADC 12-bit result word is shown in Figure 29.

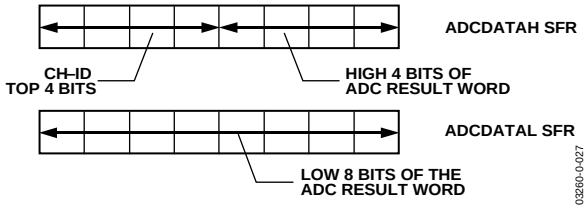


Figure 29. ADC Result Word Format

ADCCON1—(ADC Control SFR 1)

The ADCCON1 register controls conversion and acquisition times, hardware conversion modes, and power-down modes as detailed below.

SFR Address	EFH
SFR Power-On Default	40H
Bit Addressable	No

Table 7. ADCCON1 SFR Bit Designations

Bit No.	Name	Description
7	MD1	The mode bit selects the active operating mode of the ADC. Set by the user to power up the ADC. Cleared by the user to power down the ADC.
6	EXT_REF	Set by the user to select an external reference. Cleared by the user to use the internal reference.
5 4	CK1 CK0	The ADC clock divide bits (CK1, CK0) select the divide ratio for the PLL master clock (ADuC842/ADuC843) or the external crystal (ADuC841) used to generate the ADC clock. To ensure correct ADC operation, the divider ratio must be chosen to reduce the ADC clock to 8.38 MHz or lower. A typical ADC conversion requires 16 ADC clocks plus the selected acquisition time. The divider ratio is selected as follows: CK1 CK0 MCLK Divider 0 0 32 0 1 4 (Do not use with a CD setting of 0) 1 0 8 1 1 2
3 2	AQ1 AQ0	The ADC acquisition select bits (AQ1, AQ0) select the time provided for the input track-and-hold amplifier to acquire the input signal. An acquisition of three or more ADC clocks is recommended; clocks are as follows: AQ1 AQ0 No. ADC Clks 0 0 1 0 1 2 1 0 3 1 1 4
1	T2C	The Timer 2 conversion bit (T2C) is set by the user to enable the Timer 2 overflow bit to be used as the ADC conversion start trigger input.
0	EXC	The external trigger enable bit (EXC) is set by the user to allow the external Pin P3.5 ($\overline{\text{CONVST}}$) to be used as the active low convert start input. This input should be an active low pulse (minimum pulse width >100 ns) at the required sample rate.

ADCCON2—(ADC Control SFR 2)

The ADCCON2 register controls ADC channel selection and conversion modes as detailed below.

SFR Address	D8H
SFR Power-On Default	00H
Bit Addressable	Yes

Table 8. ADCCON2 SFR Bit Designations

Bit No.	Name	Description
7	ADCI	ADC Interrupt Bit. Set by hardware at the end of a single ADC conversion cycle or at the end of a DMA block conversion. Cleared by hardware when the PC vectors to the ADC interrupt service routine. Otherwise, the ADCI bit is cleared by user code.
6	DMA	DMA Mode Enable Bit. Set by the user to enable a preconfigured ADC DMA mode operation. A more detailed description of this mode is given in the ADC DMA Mode section. The DMA bit is automatically set to 0 at the end of a DMA cycle. Setting this bit causes the ALE output to cease; it will start again when DMA is started and will operate correctly after DMA is complete.
5	CCONV	Continuous Conversion Bit. Set by the user to initiate the ADC into a continuous mode of conversion. In this mode, the ADC starts converting based on the timing and channel configuration already set up in the ADCCON SFRs; the ADC automatically starts another conversion once a previous conversion has completed.
4	SCONV	Single Conversion Bit. Set to initiate a single conversion cycle. The SCONV bit is automatically reset to 0 on completion of the single conversion cycle.
3 2 1 0	CS3 CS2 CS1 CS0	Channel Selection Bits. Allow the user to program the ADC channel selection under software control. When a conversion is initiated, the converted channel is the one pointed to by these channel selection bits. In DMA mode, the channel selection is derived from the channel ID written to the external memory.

CS3	CS2	CS1	CS0	CH#	
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	2	
0	0	1	1	3	
0	1	0	0	4	
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0	Temp Monitor	Requires minimum of 1 µs to acquire.
1	0	0	1	DAC0	Only use with internal DAC output buffer on.
1	0	1	0	DAC1	Only use with internal DAC output buffer on.
1	0	1	1	AGND	
1	1	0	0	V_{REF}	
1	1	1	1	DMA STOP	Place in XRAM location to finish DMA sequence; refer to the ADC DMA Mode section.

All other combinations reserved.

ADCCON3—(ADC Control SFR 3)

The ADCCON3 register controls the operation of various calibration modes and also indicates the ADC busy status.

SFR Address	F5H
SFR Power-On Default	00H
Bit Addressable	No

Table 9. ADCCON3 SFR Bit Designations

Bit No.	Name	Description
7	BUSY	ADC Busy Status Bit. A read-only status bit that is set during a valid ADC conversion or during a calibration cycle. Busy is automatically cleared by the core at the end of conversion or calibration.
6	RSVD	Reserved. This bit should always be written as 0.
5	AVGS1	Number of Average Selection Bits.
4	AVGS0	This bit selects the number of ADC readings that are averaged during a calibration cycle. AVGS1 AVGS0 Number of Averages 0 0 15 0 1 1 1 0 31 1 1 63
3	RSVD	Reserved. This bit should always be written as 0.
2	RSVD	This bit should always be written as 1 by the user when performing calibration.
1	TYPICAL	Calibration Type Select Bit. This bit selects between offset (zero-scale) and gain (full-scale) calibration. Set to 0 for offset calibration. Set to 1 for gain calibration.
0	SCAL	Start Calibration Cycle Bit. When set, this bit starts the selected calibration cycle. It is automatically cleared when the calibration cycle is completed.

The ADC incorporates a successive approximation architecture (SAR) involving a charge-sampled input stage. Figure 30 shows the equivalent circuit of the analog input section. Each ADC conversion is divided into two distinct phases, as defined by the position of the switches in Figure 30. During the sampling phase (with SW1 and SW2 in the track position), a charge proportional to the voltage on the analog input is developed across the input sampling capacitor. During the conversion phase (with both switches in the hold position), the capacitor DAC is adjusted via internal SAR logic until the voltage on Node A is 0, indicating that the sampled charge on the input capacitor is balanced out by the charge being output by the capacitor DAC. The final digital value contained in the SAR is then latched out as the result of the ADC conversion. Control of the SAR and timing of acquisition and sampling modes is handled automatically by built-in ADC control logic. Acquisition and conversion times are also fully configurable under user control.

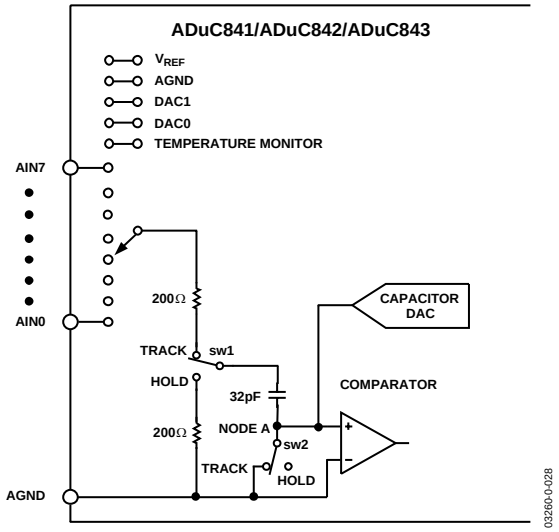


Figure 30. Internal ADC Structure

Note that whenever a new input channel is selected, a residual charge from the 32 pF sampling capacitor places a transient on the newly selected input. The signal source must be capable of recovering from this transient before the sampling switches go into hold mode. Delays can be inserted in software (between channel selection and conversion request) to account for input stage settling, but a hardware solution alleviates this burden from the software design task and ultimately results in a cleaner system implementation. One hardware solution is to choose a very fast settling op amp to drive each analog input. Such an op amp would need to fully settle from a small signal transient in less than 300 ns in order to guarantee adequate settling under all software configurations. A better solution, recommended for use with any amplifier, is shown in Figure 31. Though at first glance the circuit in Figure 31 may look like a simple antialiasing filter, it actually serves no such purpose since its corner frequency is well above the Nyquist frequency, even at a 200 kHz sample rate. Though the R/C does help to reject some incoming high frequency noise, its primary function is to ensure that the transient demands of the ADC input stage are met.

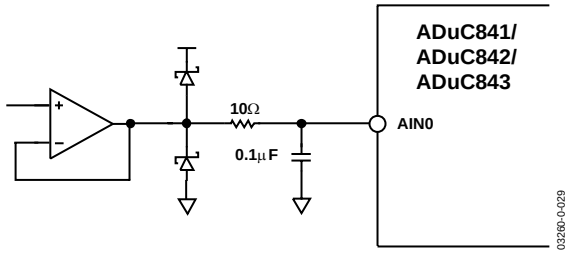


Figure 31. Buffering Analog Inputs

It does so by providing a capacitive bank from which the 32 pF sampling capacitor can draw its charge. Its voltage does not change by more than one count (1/4096) of the 12-bit transfer function when the 32 pF charge from a previous channel is dumped onto it. A larger capacitor can be used if desired, but not a larger resistor (for reasons described below). The Schottky diodes in Figure 31 may be necessary to limit the voltage applied to the analog input pin per the Absolute Maximum Ratings. They are not necessary if the op amp is powered from the same supply as the part since in that case the op amp is unable to generate voltages above V_{DD} or below ground. An op amp of some kind is necessary unless the signal source is very low impedance to begin with. DC leakage currents at the parts' analog inputs can cause measurable dc errors with external source impedances as low as 100 Ω or so. To ensure accurate ADC operation, keep the total source impedance at each analog input less than 61 Ω. The Table 10 illustrates examples of how source impedance can affect dc accuracy.

Table 10. Source Impedance and DC Accuracy

Source Impedance Ω	Error from 1 µA Leakage Current	Error from 10 µA Leakage Current
61	61 µV = 0.1 LSB	610 µV = 1 LSB
610	610 µV = 1 LSB	6.1 mV = 10 LSB

Although Figure 31 shows the op amp operating at a gain of 1, one can, of course, configure it for any gain needed. Also, one can just as easily use an instrumentation amplifier in its place to condition differential signals. Use an amplifier that is capable of delivering the signal (0 V to V_{REF}) with minimal saturation. Some single-supply rail-to-rail op amps that are useful for this purpose are described in Table 11. Check Analog Devices website www.analog.com for details on these and other op amps and instrumentation amps.

Table 11. 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's transfer function is 0 V to V_{REF}, and that any signal range lost to amplifier saturation near ground will impact dynamic range. Though the op amps in Table 11 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.

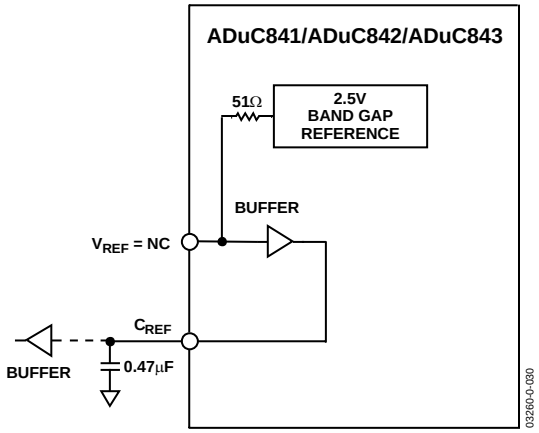


Figure 32. Decoupling V_{REF} and C_{REF}

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.

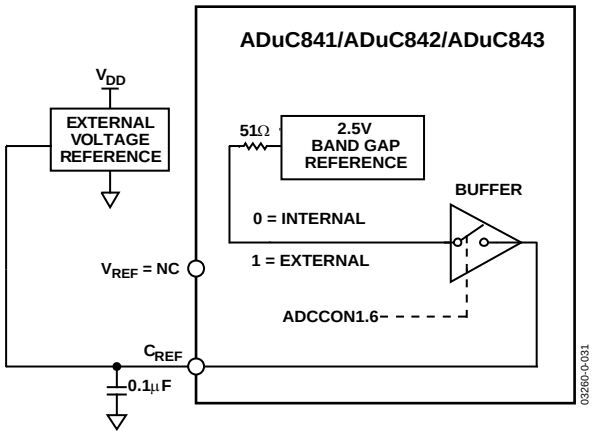


Figure 33. Using an External Voltage Reference

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 (i.e., 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.

If using the temperature sensor as the ADC input, the ADC should be configured to use an ADCCLK of MCLK/32 and four acquisition clocks.

Increasing the conversion time on the temperature monitor channel improves the accuracy of the reading. To further improve the accuracy, an external reference with low temperature drift should also be used.

ADC DMA Mode

The on-chip ADC has been designed to run at a maximum conversion speed of 2.38 µs (420 kHz sampling rate). When converting at this rate, the ADuC841/ADuC842/ADuC843 MicroConverter has 2 µs to read the ADC result and to store the result in memory for further postprocessing; otherwise the next ADC sample could be lost. In an interrupt driven routine, the MicroConverter would also have to jump to the ADC interrupt service routine, which also increases the time required to store the ADC results. In applications where the parts cannot sustain the interrupt rate, an ADC DMA mode is provided.

To enable DMA mode, Bit 6 in ADCCON2 (DMA) must be set, which allows the ADC results to be written directly to a 16 MByte external static memory SRAM (mapped into data memory space) without any interaction from the core of the part. This mode allows the part to capture a contiguous sample stream at full ADC update rates (420 kHz).

Typical DMA Mode Configuration Example

Setting the parts to DMA mode consists of the following steps:

1. The ADC must be powered down. This is done by ensuring that MD1 and MD0 are both set to 0 in ADCCON1.

2. The DMA address pointer must be set to the start address of where the ADC results are to be written. This is done by writing to the DMA mode address pointers DMAL, DMAH, and DMAP. DMAL must be written to first, followed by DMAH, and then by DMAP.

3. The external memory must be preconfigured. This consists of writing the required ADC channel IDs into the top four bits of every second memory location in the external SRAM, starting at the first address specified by the DMA address pointer. Because the ADC DMA mode operates independently from the ADuC841/ADuC842/ADuC843 core, it is necessary to provide it with a stop command. This is done by duplicating the last channel ID to be converted followed by 1111 into the next channel selection field. A typical preconfiguration of external memory is shown in Figure 34.

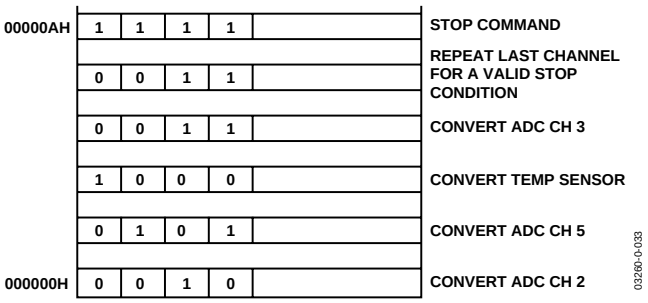


Figure 34. Typical DMA External Memory Preconfiguration

4. The DMA is initiated by writing to the ADC SFRs in the following sequence:

   a. ADCCON2 is written to enable the DMA mode, i.e., MOV ADCCON2, #40H; DMA mode enabled.

   b. ADCCON1 is written to configure the conversion time and power-up of the ADC. It can also enable Timer 2 driven conversions or external triggered conversions if required.

   c. ADC conversions are initiated. This is done by starting single conversions, starting Timer 2, running for Timer 2 conversions, or receiving an external trigger.

When the DMA conversions are complete, the ADC interrupt bit, ADCI, is set by hardware, and the external SRAM contains the new ADC conversion results as shown in Figure 35. Note that no result is written to the last two memory locations.

When the DMA mode logic is active, it takes the responsibility of storing the ADC results away from both the user and the core logic of the part. As the DMA interface writes the results of the ADC conversions to external memory, it takes over the external memory interface from the core. Thus, any core instructions that access the external memory while DMA mode is enabled does not get access to the external memory. The core executes the instructions, and they take the same time to execute, but they cannot access the external memory.

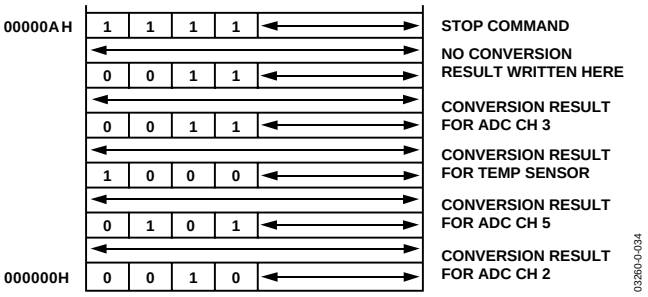


Figure 35. Typical External Memory Configuration Post ADC DMA Operation

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.

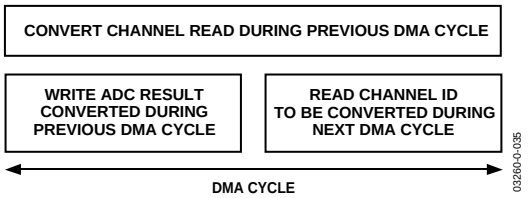


Figure 36. DMA Cycle

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 will be 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.

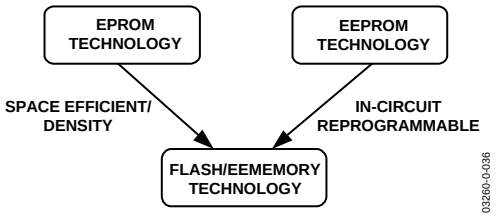


Figure 37. Flash/EE Memory Development

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, i.e., 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.

A 4 kByte Flash/EE data memory space is also provided on-chip. This may be used as a general-purpose nonvolatile scratchpad area. User access to this area is via a group of six SFRs. This space can be programmed at a byte level, although it must first be erased in 4-byte pages.

Flash/EE Memory Reliability

The Flash/EE program and data memory arrays on the parts are fully qualified for two key Flash/EE memory characteristics: Flash/EE memory cycling endurance and Flash/EE memory data retention.

Endurance quantifies the ability of the Flash/EE memory to be cycled through many program, read, and erase cycles. In real terms, a single endurance cycle is composed of four independent, sequential events, defined as

1. Initial page erase sequence.
2. Read/verify sequence a single Flash/EE.
3. Byte program sequence memory.
4. Second read/verify sequence endurance cycle.

In reliability qualification, every byte in both the program and data Flash/EE memory is cycled from 00H to FFH until a first fail is recorded, signifying the endurance limit of the on-chip Flash/EE memory.

As indicated in the Specifications table, the parts' Flash/EE memory endurance qualification has been carried out in accordance with JEDEC Retention Lifetime Specification A117 over the industrial temperature range of –40°C to +25°C and +25°C to +85°C. The results allow the specification of a minimum endurance figure over supply and over temperature of 100,000 cycles, with an endurance figure of 700,000 cycles being typical of operation at 25°C.

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. Again, the parts have been qualified in accordance with the formal JEDEC Retention Lifetime Specification (A117) at a specific junction temperature (T_J = 55°C). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit, described previously, before data retention is characterized. This means that the Flash/EE memory is guaranteed to retain its data for its fully specified retention lifetime every time the Flash/EE memory is reprogrammed. Also note that retention lifetime, based on an activation energy of 0.6 eV, derates with T_J as shown in Figure 38.

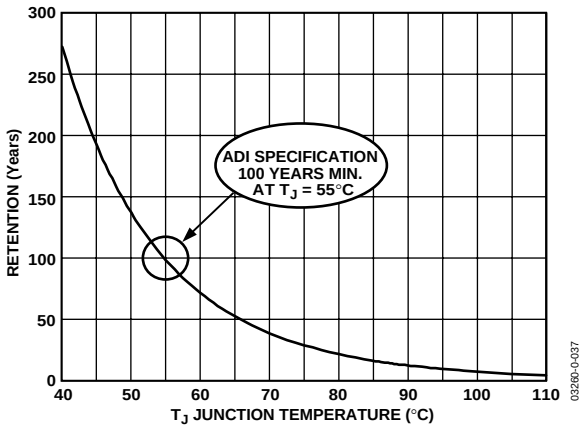


Figure 38. Flash/EE Memory Data Retention

Using the Flash/EE Program Memory

The 62 kByte Flash/EE program memory array is mapped into the lower 62 kBytes of the 64 kByte program space addressable by the parts, and is used to hold user code in typical applications. The program Flash/EE memory array can be programmed in three ways:

Serial Downloading (In-Circuit Programming)

The parts facilitate code download via the standard UART serial port. The parts enter serial download mode after a reset or power cycle if the $\overline{PSEN}$ pin is pulled low through an external 1 kΩ resistor. Once in serial download mode, the user can download code to the full 62 kBytes of Flash/EE program memory while the device is in-circuit in its target application hardware.

A PC serial download executable is provided as part of the ADuC841/ADuC842 QuickStart development system. The serial download protocol is detailed in MicroConverter Application Note uC004.

Parallel Programming

Parallel programming mode is fully compatible with conventional third party flash or EEPROM device programmers. In this mode, Ports P0, P1, and P2 operate as the external data and address bus interface, ALE operates as the write enable strobe, and Port P3 is used as a general configuration port, which configures the device for various program and erase operations during parallel programming. The high voltage (12 V) supply required for flash programming is generated using on-chip charge pumps to supply the high voltage program lines. The complete parallel programming specification is available on the MicroConverter home page at www.analog.com/microconverter.

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.

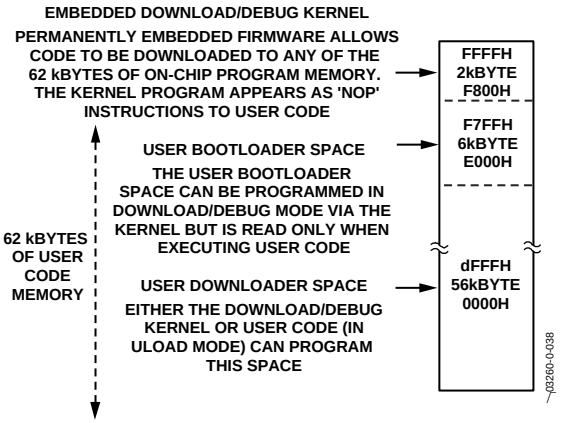


Figure 39. Flash/EE Program Memory Map in ULOAD Mode (62 kByte Part)

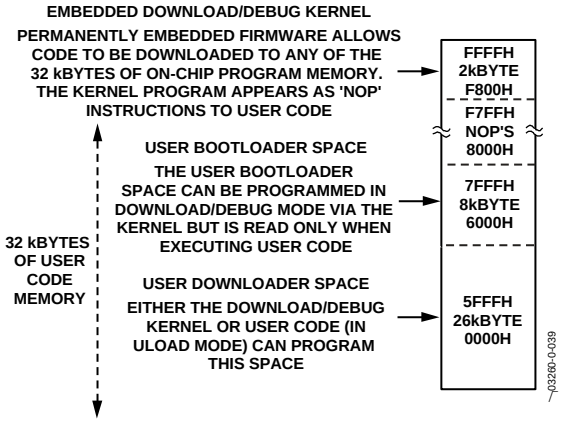


Figure 40. Flash/EE Program Memory Map in ULOAD Mode (32 kByte Part)

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, i.e., $\overline{\text{RESET}}$ 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.

USING FLASH/EE DATA MEMORY

The 4 kBytes of Flash/EE data memory are configured as 1024 pages, each of 4 bytes. As with the other ADuC841/ADuC842/ADuC843 peripherals, the interface to this memory space is via a group of registers mapped in the SFR space. A group of four data registers (EDATA1–4) is used to hold the four bytes of data at each page. The page is addressed via the two registers, EADRH and EADRL. Finally, ECON is an 8-bit control register that may be written with one of nine Flash/EE memory access commands to trigger various read, write, erase, and verify functions. A block diagram of the SFR interface to the Flash/EE data memory array is shown in Figure 41.

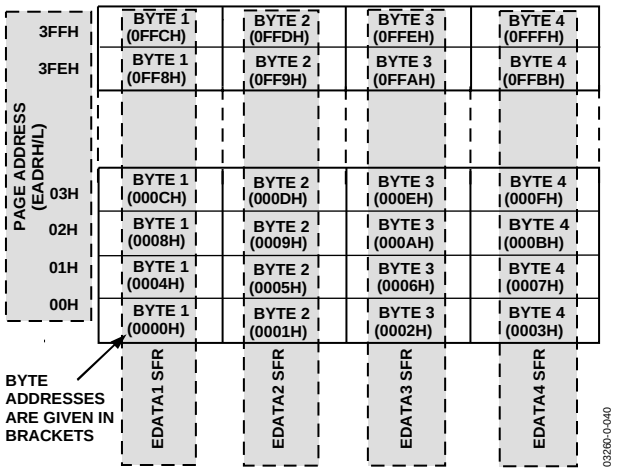


Figure 41. Flash/EE Data Memory Control and Configuration

ECON—Flash/EE Memory Control SFR

Programming of either Flash/EE data memory or Flash/ EE program memory is done through the Flash/EE memory control SFR (ECON). This SFR allows the user to read, write, erase, or verify the 4 kBytes of Flash/EE data memory or the 56 kBytes of Flash/EE program memory.

Table 12. ECON—Flash/EE Memory Commands

ECON VALUE	Command Description (Normal Mode) (Power-On Default)	Command Description (ULOAD Mode)
01H READ	Results in 4 bytes in the Flash/EE data memory, addressed by the page address EADRH/L, being read into EDATA1–4.	Not implemented. Use the MOVC instruction.
02H WRITE	Results in 4 bytes in EDATA1–4 being written to the Flash/EE data memory at the page address given by EADRH/L (0 – EADRH/L < 0400H). Note that the 4 bytes in the page being addressed must be pre-erased.	Results in bytes 0–255 of internal XRAM being written to the 256 bytes of Flash/EE program memory at the page address given by EADRH (0 – EADRH < E0H). Note that the 256 bytes in the page being addressed must be pre-erased.
03H	Reserved.	Reserved.
04H VERIFY	Verifies that the data in EDATA1–4 is contained in the page address given by EADRH/L. A subsequent read of the ECON SFR results in 0 being read if the verification is valid, or a nonzero value being read to indicate an invalid verification.	Not implemented. Use the MOVC and MOVX instructions to verify the write in software.
05H ERASE PAGE	Results in erasing the 4-byte page of Flash/EE data memory addressed by the page address EADRH/L.	Results in the 64 byte page of Flash/EE program memory, addressed by the byte address EADRH/L, being erased. EADRL can equal any of 64 locations within the page. A new page starts whenever EADRL is equal to 00H, 40H, 80H, or C0H.
06H ERASE ALL	Results in erasing the entire 4 kBytes of Flash/EE data memory.	Results in erasing the entire 56 kBytes of ULOAD Flash/EE program memory.
81H READBYTE	Results in the byte in the Flash/EE data memory, addressed by the byte address EADRH/L, being read into EDATA1 (0 – EADRH / L – 0FFFH).	Not implemented. Use the MOVC command.
82H WRITEBYTE	Results in the byte in EDATA1 being written into Flash/EE data memory at the byte address EADRH/L	Results in the byte in EDATA1 being written into Flash/EE program memory at the byte address EADRH/L (0 – EADRH/L – DFFFH).
0FH EXULOAD	Leaves the ECON instructions to operate on the Flash/EE data memory.	Enters normal mode directing subsequent ECON instructions to operate on the Flash/EE data memory.
F0H ULOAD	Enters ULOAD mode, directing subsequent ECON instructions to operate on the Flash/EE program memory.	Leaves the ECON instructions to operate on the Flash/EE program memory.

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, i.e., 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, i.e., 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 13. 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.

CFG841	**ADuC841 Config SFR**
SFR Address	AFH
Power-On Default	10H[1]
Bit Addressable	No

Table 14. CFG841 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	EPM2	Flash/EE Controller and PWM Clock Frequency Configuration Bits. Frequency should be configured such that F_{OSC}/Divide Factor = 32 kHz + 50%.
3	EPM1	EPM2 EPM1 EPM0 Divide Factor
2	EPM0	0 0 0 32
		0 0 1 64
		0 1 0 128
		0 1 1 256
		1 0 0 512
		1 0 1 1024
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 two 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 two kBytes of external data memory.

[1] Note that the Flash/EE controller bits EPM2, EPM1, EPM0 are set to their correct values depending on the crystal frequency at power-up. The user should not modify these bits so all instructions to the CFG841 register should use the ORL, XRL, or ANL instructions. Value of 10H is for 11.0592 MHz crystal.

USER INTERFACE TO ON-CHIP PERIPHERALS

This section gives a brief overview of the various peripherals also available on-chip. A summary of the SFRs used to control and configure these peripherals is also given.

DAC

The ADuC841/ADuC842 incorporate two 12-bit voltage output DACs on-chip. Each has a rail-to-rail voltage output buffer capable of driving 10 kΩ/100 pF. Each has two selectable ranges, 0 V to V_{REF} (the internal band gap 2.5 V reference) and 0 V to AV_{DD}. Each can operate in 12-bit or 8-bit mode.

Both DACs share a control register, DACCON, and four data registers, DAC1H/L, DAC0/L. Note that in 12-bit asynchronous mode, the DAC voltage output is updated as soon as the DACL data SFR has been written; therefore, the DAC data registers should be updated as DACH first, followed by DACL. Note that for correct DAC operation on the 0 V to V_{REF} range, the ADC must be switched on. This results in the DAC using the correct reference value.

DACCON	**DAC Control Register**
SFR Address	FDH
Power-On Default	04H
Bit Addressable	No

Table 15. DACCON SFR Bit Designations

Bit No.	Name	Description
7	MODE	The DAC MODE bit sets the overriding operating mode for both DACs. Set to 1 by the user to select 8-bit mode (write 8 bits to DACxL SFR). Set to 0 by the user to select 12-bit mode.
6	RNG1	DAC1 Range Select Bit. Set to 1 by the user to select the range for DAC1 as 0 V to V_{DD}. Set to 0 by the user to select the range for DAC1 as 0 V to V_{REF}.
5	RNG0	DAC0 Range Select Bit. Set to 1 by the user to select the range for DAC0 as 0 V to V_{DD}. Set to 0 by the user to select the range for DAC0 as 0 V to V_{REF}.
4	CLR1	DAC1 Clear Bit. Set to 1 by the user to leave the output of DAC1 at its normal level. Set to 0 by the user to force the output of DAC1 to 0 V.
3	CLR0	DAC0 Clear Bit. Set to 1 by the user to leave the output of DAC0 at its normal level. Set to 0 by the user to force the output of DAC0 to 0 V.
2	SYNC	DAC0/1 Update Synchronization Bit. When set to 1, the DAC outputs update as soon as DACxL SFRs are written. The user can simultaneously update both DACs by first updating the DACxL/H SFRs while SYNC is 0. Both DACs then update simultaneously when the SYNC bit is set to 1.
1	PD1	DAC1 Power-Down Bit. Set to 1 by the user to power on DAC1. Set to 0 by the user to power off DAC1.
0	PD0	DAC0 Power-Down Bit. Set to 1 by the user to power on DAC0. Set to 0 by the user to power off DAC0.

DACxH/L	**DAC Data Registers**	
Function	DAC data registers, written by the user to update the DAC output.	
SFR Address	DAC0L (DAC0 Data Low Byte) -> F9H; DAC1L (DAC1 Data Low Byte) -> FBH	
	DACH (DAC0 Data High Byte) -> FAH; DAC1H (DAC1 Data High Byte) -> FCH	
Power-On Default	00H	All Four Registers.
Bit Addressable	No	All Four Registers.

The 12-bit DAC data should be written into DACxH/L right-justified such that DACxL contains the lower 8 bits, and the lower nibble of DACxH contains the upper 4 bits.

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. Details of the actual DAC architecture can be found in U.S. Patent Number 5969657 (www.uspto.gov). Features of this architecture include inherent guaranteed monotonicity and excellent differential linearity.

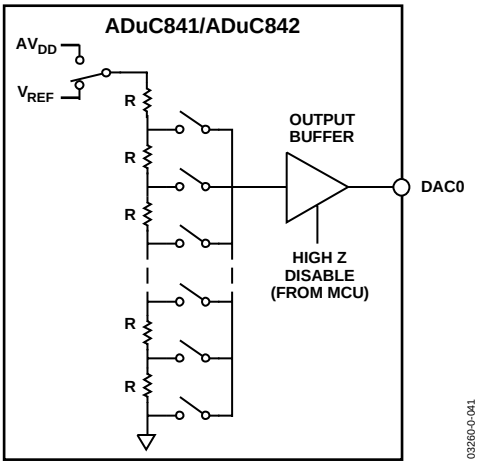


Figure 42. Resistor String DAC Functional Equivalent

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.

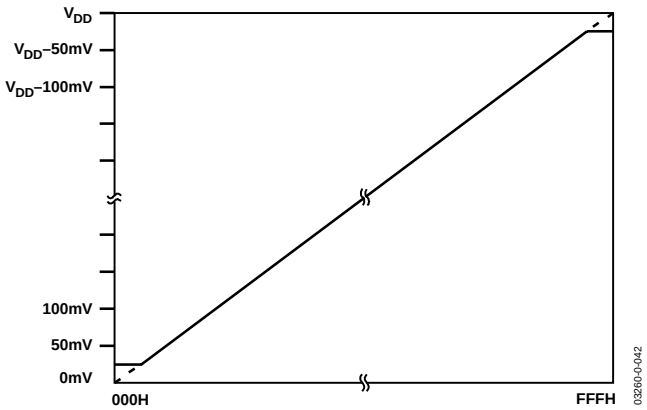


Figure 43. Endpoint Nonlinearities Due to Amplifier Saturation

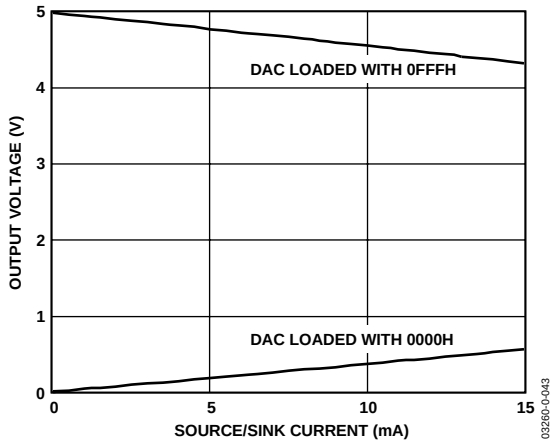


Figure 44. Source and Sink Current Capability with $V_{REF} = V_{DD} = 5$ V

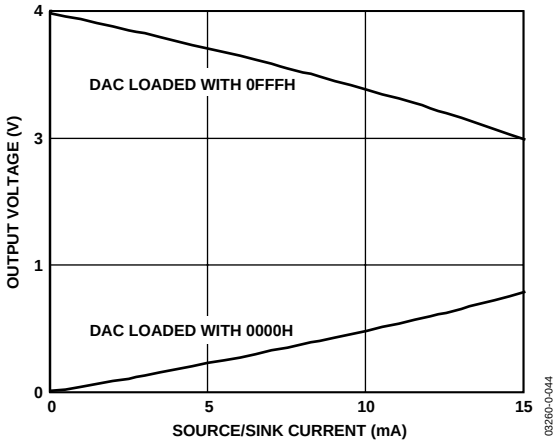


Figure 45. Source and Sink Current Capability with $V_{REF} = V_{DD} = 3$ V

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 11 lists some recommended op amps.

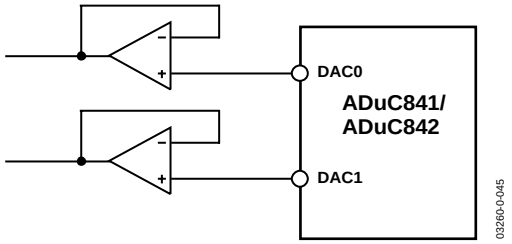


Figure 46. Buffering the DAC Outputs

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.

ON-CHIP PLL

The ADuC842 and ADuC843 are intended for use with a 32.768 kHz watch crystal. A PLL locks onto a multiple (512) of this to provide a stable 16.78 MHz clock for the system. The ADuC841 operates directly from an external crystal. The core can operate at this frequency or at binary submultiples of it to allow power saving in cases where maximum core performance is not required. The default core clock is the PLL clock divided by 8 or 2.097152 MHz. The ADC clocks are also derived from the PLL clock, with the modulator rate being the same as the crystal oscillator frequency. The preceding choice of frequencies ensures that the modulators and the core are synchronous, regardless of the core clock rate. The PLL control register is PLLCON.

At 5 V the core clock can be set to a maximum of 16.78 MHz, while at 3 V the maximum core clock setting is 8.38 MHz. The CD bits should not be set to 0 on a 3 V part.

Note that on the ADuC841, changing the CD bits in PLLCON causes the core speed to change. The core speed is crystal freq/ 2^{CD}. The other bits in PLLCON are reserved in the case of the ADuC841 and should be written with 0.

PLLCON PLL	**Control Register**
SFR Address	D7H
Power-On Default	53H
Bit Addressable	No

Table 16. PLLCON SFR Bit Designations

Bit No.	Name	Description
7	OSC_PD	Oscillator Power-Down Bit. Set by the user to halt the 32 kHz oscillator in power-down mode. Cleared by the user to enable the 32 kHz oscillator in power-down mode. This feature allows the TIC to continue counting even in power-down mode.
6	LOCK	PLL Lock Bit. This is a read-only bit. Set automatically at power-on to indicate that the PLL loop is correctly tracking the crystal clock. If the external crystal subsequently becomes disconnected, the PLL will rail. Cleared automatically at power-on to indicate that the PLL is not correctly tracking the crystal clock. This may be due to the absence of a crystal clock or an external crystal at power-on. In this mode, the PLL output can be 16.78 MHz ±20%.
5	----	Reserved. Should be written with 0.
4	----	Reserved. Should be written with 0.
3	FINT	Fast Interrupt Response Bit. Set by the user enabling the response to any interrupt to be executed at the fastest core clock frequency, regardless of the configuration of the CD2–0 bits (see below). Once user code has returned from an interrupt, the core resumes code execution at the core clock selected by the CD2–0 bits. Cleared by the user to disable the fast interrupt response feature.
2	CD2	CPU (Core Clock) Divider Bits.
1	CD1	This number determines the frequency at which the microcontroller core operates.
0	CD0	

CD2	CD1	CD0	Core Clock Frequency (MHz)
0	0	0	16.777216
0	0	1	8.388608
0	1	0	4.194304
0	1	1	2.097152 (Default Core Clock Frequency)
1	0	0	1.048576
1	0	1	0.524288
1	1	0	0.262144
1	1	1	0.131072

PULSE-WIDTH MODULATOR (PWM)

The PWM on the ADuC841/ADuC842/ADuC843 is a highly flexible PWM offering programmable resolution and an input clock, and can be configured for any one of six different modes of operation. Two of these modes allow the PWM to be configured as a ∑-Δ DAC with up to 16 bits of resolution. A block diagram of the PWM is shown in Figure 47. Note the PWM clock's sources are different for the ADuC841, and are given in Table 17.

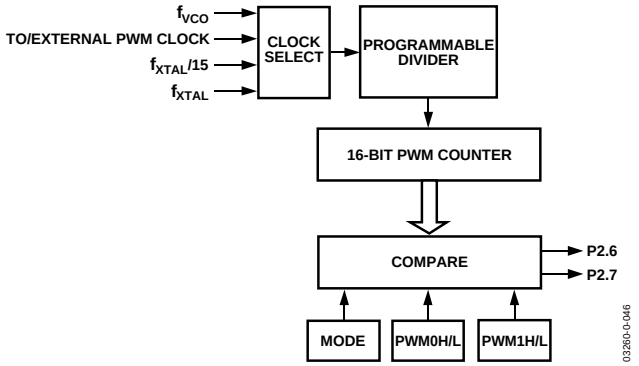


Figure 47. PWM Block Diagram

The PWM uses five SFRs: the control SFR (PWMCON) and four data SFRs (PWM0H, PWM0L, PWM1H, and PWM1L).

PWMCON, as described in the following sections, controls the different modes of operation of the PWM as well as the PWM clock frequency.

PWM0H/L and PWM1H/L are the data registers that determine the duty cycles of the PWM outputs. The output pins that the PWM uses are determined by the CFG841/CFG842 register, and can be either P2.6 and P2.7 or P3.4 and P3.3. In this section of the data sheet, it is assumed that P2.6 and P2.7 are selected as the PWM outputs.

To use the PWM user software, first write to PWMCON to select the PWM mode of operation and the PWM input clock. Writing to PWMCON also resets the PWM counter. In any of the 16-bit modes of operation (Modes 1, 3, 4, 6), user software should write to the PWM0L or PWM1L SFRs first. This value is written to a hidden SFR. Writing to the PWM0H or PWM1H SFRs updates both the PWMxH and the PWMxL SFRs but does not change the outputs until the end of the PWM cycle in progress. The values written to these 16-bit registers are then used in the next PWM cycle.

PWMCON PWM	**Control SFR**
SFR Address	AEH
Power-On Default	00H
Bit Addressable	No

Table 17. PWMCON SFR Bit Designations

Bit No.	Name	Description			
7	SNGL	Turns off PMW output at P2.6 or P3.4, leaving the port pin free for digital I/O.			
6	MD2	PWM Mode Bits.			
5	MD1	The MD2/1/0 bits choose the PWM mode as follows:			
4	MD0	MD2	MD1	MD0	Mode
		0	0	0	Mode 0: PWM Disabled
		0	0	1	Mode 1: Single variable resolution PWM on P2.7 or P3.3
		0	1	0	Mode 2: Twin 8-bit PWM
		0	1	1	Mode 3: Twin 16-bit PWM
		1	0	0	Mode 4: Dual NRZ 16-bit ∑-Δ DAC
		1	0	1	Mode 5: Dual 8-bit PWM
		1	1	0	Mode 6: Dual RZ 16-bit ∑-Δ DAC
		1	1	1	Reserved
3	CDIV1	PWM Clock Divider.			
2	CDIV0	Scale the clock source for the PWM counter as follows:			
		CDIV1	CDIV0	Description	
		0	0	PWM Counter = Selected Clock/1	
		0	1	PWM Counter = Selected Clock/4	
		1	0	PWM Counter = Selected Clock/16	
		1	1	PWM Counter = Selected Clock/64	
1	CSEL1	PWM Clock Divider.			
0	CSEL0	Select the clock source for the PWM as follows:			
		CSEL1	CSEL0	Description	
		0	0	PWM Clock = f_{XTAL}/15, ADuC841 = f_{OCS}/DIVIDE FACTOR /15 (see the CFG841 register)	
		0	1	PWM Clock = f_{XTAL}, ADuC841 = f_{OCS}/DIVIDE FACTOR (see the CFG841 register)	
		1	0	PWM Clock = External input at P3.4/T0	
		1	1	PWM Clock = f_{VCO} = 16.777216 MHz, ADuC841 = f_{OSC}	

PWM Modes of Operation

Mode 0: PWM Disabled

The PWM is disabled allowing P2.6 and P2.7 to be used as normal.

Mode 1: Single Variable Resolution PWM

In Mode 1, both the pulse length and the cycle time (period) are programmable in user code, allowing the resolution of the PWM to be variable.

PWM1H/L sets the period of the output waveform. Reducing PWM1H/L reduces the resolution of the PWM output but increases the maximum output rate of the PWM. For example, setting PWM1H/L to 65536 gives a 16-bit PWM with a maximum output rate of 266 Hz (16.777 MHz/65536). Setting PWM1H/L to 4096 gives a 12-bit PWM with a maximum output rate of 4096 Hz (16.777 MHz/4096).

PWM0H/L sets the duty cycle of the PWM output waveform, as shown in Figure 48.

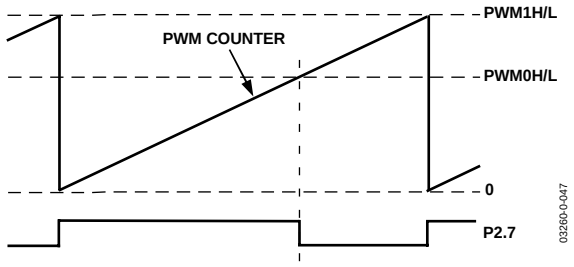


Figure 48. PWM in Mode 1

Mode 2: Twin 8-Bit PWM

In Mode 2, the duty cycle of the PWM outputs and the resolution of the PWM outputs are both programmable. The maximum resolution of the PWM output is 8 bits.

PWM1L sets the period for both PWM outputs. Typically, this is set to 255 (FFH) to give an 8-bit PWM, although it is possible to reduce this as necessary. A value of 100 could be loaded here to give a percentage PWM, i.e., the PWM is accurate to 1%.

The outputs of the PWM at P2.6 and P2.7 are shown in Figure 49. As can be seen, the output of PWM0 (P2.6) goes low when the PWM counter equals PWM0L. The output of PWM1 (P2.7) goes high when the PWM counter equals PWM1H and goes low again when the PWM counter equals PWM0H. Setting PWM1H to 0 ensures that both PWM outputs start simultaneously.

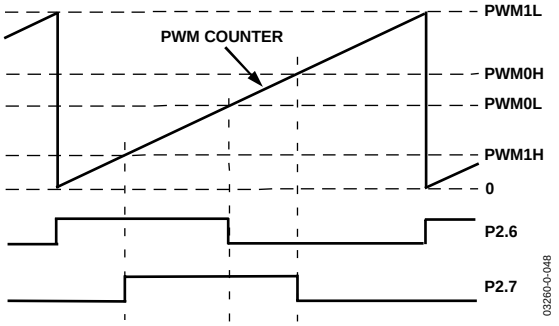


Figure 49. PWM Mode 2

Mode 3: Twin 16-Bit PWM

In Mode 3, the PWM counter is fixed to count from 0 to 65536, giving a fixed 16-bit PWM. Operating from the 16.777 MHz core clock results in a PWM output rate of 256 Hz. The duty cycle of the PWM outputs at P2.6 and P2.7 is independently programmable.

As shown in Figure 50, while the PWM counter is less than PWM0H/L, the output of PWM0 (P2.6) is high. Once the PWM counter equals PWM0H/L, PWM0 (P2.6) goes low and remains low until the PWM counter rolls over.

Similarly, while the PWM counter is less than PWM1H/L, the output of PWM1 (P2.7) is high. Once the PWM counter equals PWM1H/L, PWM1 (P2.7) goes low and remains low until the PWM counter rolls over.

In this mode, both PWM outputs are synchronized, i.e., once the PWM counter rolls over to 0, both PWM0 (P2.6) and PWM1 go high.

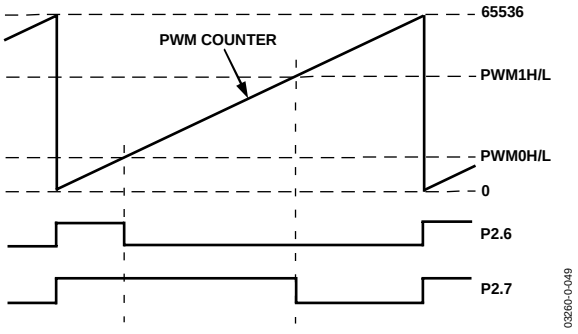


Figure 50. PWM Mode 3

Mode 4: Dual NRZ 16-Bit Σ-Δ DAC

Mode 4 provides a high speed PWM output similar to that of a Σ-Δ DAC. Typically, this mode is used with the PWM clock equal to 16.777216 MHz. In this mode, P2.6 and P2.7 are updated every PWM clock (60 ns in the case of 16 MHz). Over any 65536 cycles (16-bit PWM) PWM0 (P2.6) is high for PWM0H/L cycles and low for (65536 – PWM0H/L) cycles. Similarly, PWM1 (P2.7) is high for PWM1H/L cycles and low for (65536 – PWM1H/L) cycles.

For example, if PWM1H is set to 4010H (slightly above one quarter of FS), then typically P2.7 will be low for three clocks and high for one clock (each clock is approximately 60 ns). Over every 65536 clocks, the PWM compensates for the fact that the output should be slightly above one quarter of full scale by having a high cycle followed by only two low cycles.

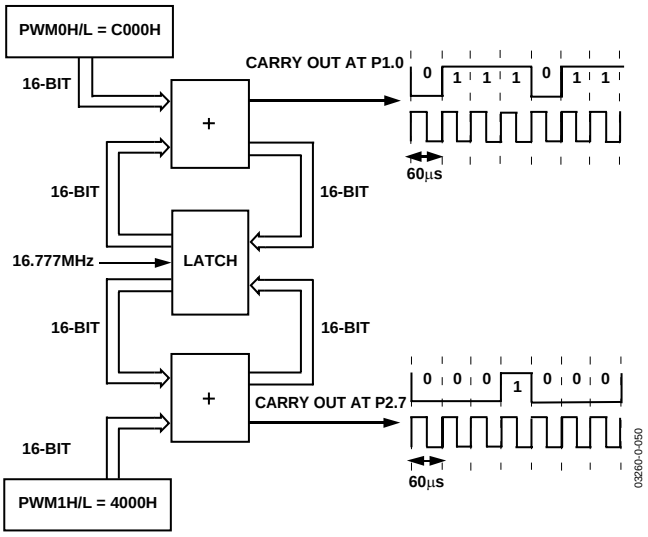


Figure 51. PWM Mode 4

For faster DAC outputs (at lower resolution), write 0s to the LSBs that are not required. If, for example, only 12-bit performance is required, write 0s to the four LSBs. This means that a 12-bit accurate Σ-Δ DAC output can occur at 4.096 kHz. Similarly writing 0s to the 8 LSBs gives an 8-bit accurate Σ-Δ DAC output at 65 kHz.

Mode 5: Dual 8-Bit PWM

In Mode 5, the duty cycle of the PWM outputs and the resolution of the PWM outputs are individually programmable. The maximum resolution of the PWM output is 8 bits. The output resolution is set by the PWM1L and PWM1H SFRs for the P2.6 and P2.7 outputs, respectively. PWM0L and PWM0H sets the duty cycles of the PWM outputs at P2.6 and P2.7, respectively. Both PWMs have the same clock source and clock divider.

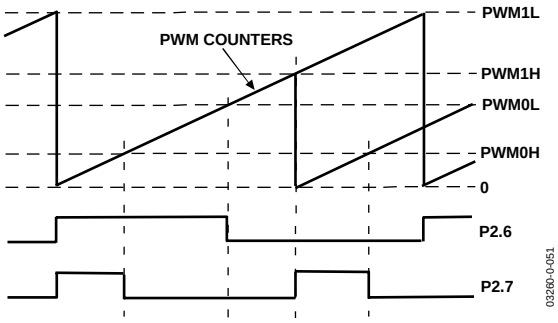


Figure 52. PWM Mode 5

Mode 6: Dual RZ 16-Bit Σ-Δ DAC

Mode 6 provides a high speed PWM output similar to that of a Σ-Δ DAC. Mode 6 operates very similarly to Mode 4. However, the key difference is that Mode 6 provides return-to-zero (RZ) Σ-Δ DAC output. Mode 4 provides non-return-to-zero Σ-Δ DAC outputs. The RZ mode ensures that any difference in the rise and fall times will not affect the Σ-Δ DAC INL. However, the RZ mode halves the dynamic range of the Σ-Δ DAC outputs from 0 V–AV_{DD} down to 0 V–AV_{DD}/2. For best results, this mode should be used with a PWM clock divider of 4.

If PWM1H is set to 4010H (slightly above one quarter of FS), typically P2.7 will be low for three full clocks (3 × 60 ns), high for half a clock (30 ns), and then low again for half a clock (30 ns) before repeating itself. Over every 65536 clocks, the PWM will compensate for the fact that the output should be slightly above one quarter of full scale by leaving the output high for two half clocks in four. The rate at which this happens depends on the value and degree of compensation required.

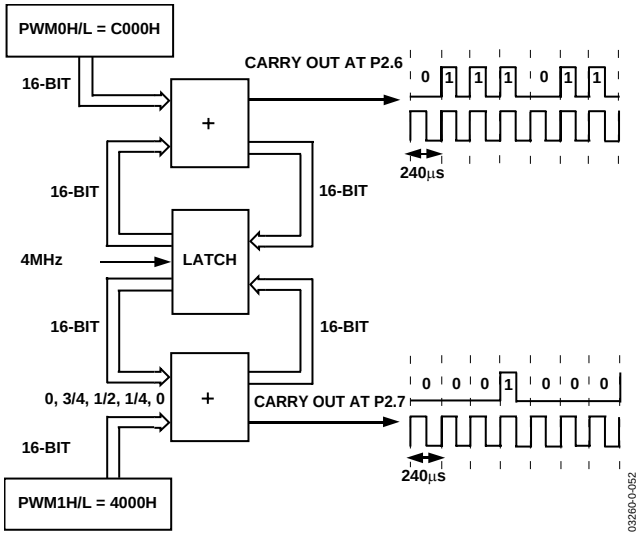


Figure 53. PWM Mode 6

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, i.e., full duplex. Note that the SPI pins are shared with the I²C pins. Therefore, the user can enable only one interface or the other on these pins at any given time (see SPE in Table 18). SPI can be operated at the same time as the I²C 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 18). 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, e.g., 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.

SPICON SPI Control Register

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

Table 18. 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, i.e., 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

Using the SPI Interface

Depending on the configuration of the bits in the SPICON SFR shown in Table 18, 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.

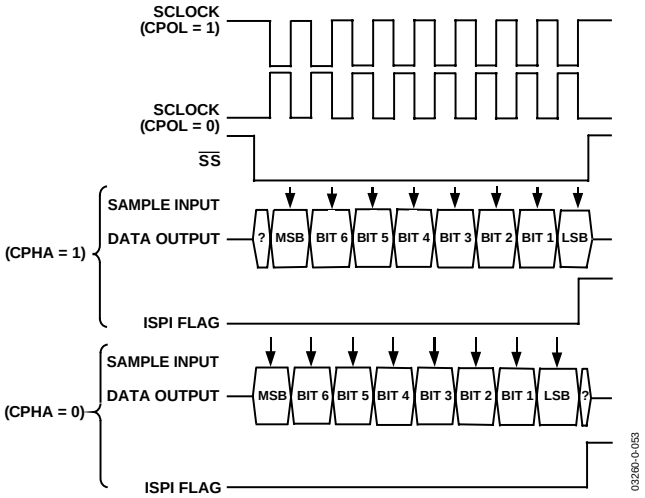


Figure 54. SPI Timing, All Modes

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 will be completely transmitted, and the input byte will be waiting in the input shift register. The ISPI flag will be set automatically, and an interrupt will occur if enabled. The value in the shift register will be 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 will be completely transmitted, and the input byte will be waiting in the input shift register. The ISPI flag will be set automatically, and an interrupt will occur if enabled. The value in the shift register will be 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.

I²C COMPATIBLE INTERFACE

The ADuC841/ADuC842/ADuC843 support a fully licensed I²C serial interface. The I²C interface is implemented as a full hardware slave and software master. SDATA is the data I/O pin, and SCLOCK is the serial clock. These two pins are shared with the MOSI and SCLOCK pins of the on-chip SPI interface. To enable the I²C interface, the SPI interface must be turned off (see SPE in Table 18) or the SPI interface must be moved to P3.3, P3.4, and P3.5 via the CFG841.1/CFG842.1 bit. Application Note uC001 describes the operation of this interface as implemented and is available from the MicroConverter website at www.analog.com/microconverter.

Three SFRs are used to control the I²C interface and are described in the following tables.

I2CCON	**I²C Control Register**
SFR Address	E8H
Power-On Default	00H
Bit Addressable	Yes

Table 19. I2CCON SFR Bit Designations, Master Mode

Bit No.	Name	Description
7	MDO	I²C Software Master Data Output Bit (Master Mode Only). This data bit is used to implement a master I²C transmitter interface in software. Data written to this bit is output on the SDATA pin if the data output enable (MDE) bit is set.
6	MDE	I²C Software Master Data Output Enable Bit (Master Mode Only). Set by the user to enable the SDATA pin as an output (Tx). Cleared by the user to enable the SDATA pin as an input (Rx).
5	MCO	I²C Software Master Clock Output Bit (Master Mode Only). This data bit is used to implement a master I²C transmitter interface in software. Data written to this bit is output on the SCLOCK pin.
4	MDI	I²C Software Master Data Input Bit (Master Mode Only). This data bit is used to implement a master I²C receiver interface in software. Data on the SDATA pin is latched into this bit on SCLOCK if the data output enable (MDE) bit is 0.
3	I2CM	I²C Master/Slave Mode Bit. Set by the user to enable I²C software master mode. Cleared by the user to enable I²C hardware slave mode.
2	----	Reserved.
1	----	Reserved.
0	----	Reserved.

Table 20. I2CCON SFR Bit Designations, Slave Mode

Bit No.	Name	Description
7	I2CSI	I²C Stop Interrupt Enable Bit. Set by the user to enable I²C stop interrupts. If set, a stop bit that follows a valid start condition generates an interrupt. Cleared by the user to disable I²C stop interrupts.
6	I2CGC	I²C General Call Status Bit. Set by hardware after receiving a general call address. Cleared by the user.
5	I2CID1	I²C Interrupt Decode Bits.
4	I2CID0	Set by hardware to indicate the source of an I²C interrupt. 00 Start and Matching Address. 01 Repeated Start and Matching Address. 10 User Data. 11 Stop after a Start and Matching Address.
3	I2CM	I²C Master/Slave Mode Bit. Set by the user to enable I²C software master mode. Cleared by the user to enable I²C hardware slave mode.

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.

- An I²C slave can respond to repeated start conditions without a stop bit in between. This allows a master to change direction of transfer without giving up the bus. Note that the repeated start is detected only when a slave has previously been configured as a receiver.
- On-chip filtering rejects <50 ns spikes on the SDATA and the SCLOCK lines to preserve data integrity.

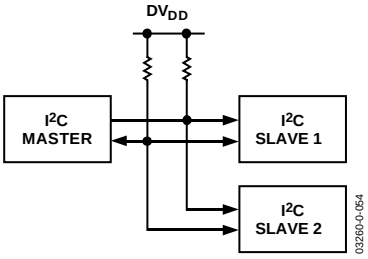


Figure 55. Typical I²C System

Software Master Mode

The ADuC841/ADuC842/ADuC843 can be used as I²C master devices by configuring the I²C peripheral in master mode and writing software to output the data bit by bit. This is referred to as a software master. Master mode is enabled by setting the I2CM bit in the I2CCON register.

To transmit data on the SDATA line, MDE must be set to enable the output driver on the SDATA pin. If MDE is set, the SDATA pin is pulled high or low depending on whether the MDO bit is set or cleared. MCO controls the SCLOCK pin and is always configured as an output in master mode. In master mode, the SCLOCK pin is pulled high or low depending on the whether MCO is set or cleared.

To receive data, MDE must be cleared to disable the output driver on SDATA. Software must provide the clocks by toggling the MCO bit and reading the SDATA pin via the MDI bit. If MDE is cleared, MDI can be used to read the SDATA pin. The value of the SDATA pin is latched into MDI on a rising edge of SCLOCK. MDI is set if the SDATA pin was high on the last rising edge of SCLOCK. MDI is clear if the SDATA pin was low on the last rising edge of SCLOCK.

Software must control MDO, MCO, and MDE appropriately to generate the start condition, slave address, acknowledge bits, data bytes, and stop conditions. These functions are described in Application Note uC001.

Hardware Slave Mode

After reset, the ADuC841/ADuC842/ADuC843 default to hardware slave mode. The I²C interface is enabled by clearing the SPE bit in SPICON (this is not necessary if the MSPI bit is set). Slave mode is enabled by clearing the I2CM bit in I2CCON. The parts have a full hardware slave. In slave mode, the I²C address is stored in the I2CADD register. Data received or to be transmitted is stored in the I2CDAT register.

Once enabled in I²C slave mode, the slave controller waits for a start condition. If the part detects a valid start condition, followed by a valid address, followed by the R/$\overline{W}$ bit, the I2CI interrupt bit is automatically set by hardware. The I²C peripheral generates a core interrupt only if the user has pre-configured the I²C interrupt enable bit in the IEIP2 SFR as well as the global interrupt bit, $\overline{EA}$, in the IE SFR. i.e.,

```
;Enabling I2C Interrupts for the ADuC842
MOV IEIP2,#01h            ; enable I2C interrupt
SETB EA
```

An autoclear of the I2CI bit is implemented on the parts so that this bit is cleared automatically on a read or write access to the I2CDAT SFR.

```
MOV I2CDAT, A             ; I2CI auto-cleared
MOV A, I2CDAT             ; I2CI auto-cleared
```

If for any reason the user tries to clear the interrupt more than once, i.e., access the data SFR more than once per interrupt, then the I²C controller will halt. The interface will then have to be reset using the I2CRS bit.

The user can choose to poll the I2CI bit or to enable the interrupt. In the case of the interrupt, the PC counter vectors to 003BH at the end of each complete byte. For the first byte, when the user gets to the I2CI ISR, the 7-bit address and the R/$\overline{W}$ bit appear in the I2CDAT SFR.

The I2CTX bit contains the R/$\overline{W}$ bit sent from the master. If I2CTX is set, the master is ready to receive a byte. Therefore the slave will transmit data by writing to the I2CDAT register. If I2CTX is cleared, the master is ready to transmit a byte. Therefore the slave will receive a serial byte. Software can interrogate the state of I2CTX to determine whether it should write to or read from I2CDAT.

Once the part has received a valid address, hardware holds SCLOCK low until the I2CI bit is cleared by software. This allows the master to wait for the slave to be ready before transmitting the clocks for the next byte.

The I2CI interrupt bit is set every time a complete data byte is received or transmitted, provided it is followed by a valid ACK. If the byte is followed by a NACK, an interrupt is not generated.

The part continues to issue interrupts for each complete data byte transferred until a stop condition is received or the interface is reset.

When a stop condition is received, the interface resets to a state in which it is waiting to be addressed (idle). Similarly, if the interface receives a NACK at the end of a sequence, it also returns to the default idle state. The I2CRS bit can be used to reset the I²C interface. This bit can be used to force the interface back to the default idle state.

DUAL DATA POINTER

The ADuC841/ADuC842/ADuC843 incorporate two data pointers. The second data pointer is a shadow data pointer and is selected via the data pointer control SFR (DPCON). DPCON also includes some useful features such as automatic hardware post-increment and post-decrement as well as automatic data pointer toggle. DPCON is described in Table 21.

DPCON	**Data Pointer Control SFR**
SFR Address	A7H
Power-On Default	00H
Bit Addressable	No

Table 21. DPCON SFR Bit Designations

Bit No.	Name	Description
7	----	Reserved.
6	DPT	Data Pointer Automatic Toggle Enable. Cleared by the user to disable autoswapping of the DPTR. Set in user software to enable automatic toggling of the DPTR after each each MOVX or MOVC instruction.
5	DP1m1	Shadow Data Pointer Mode.
4	DP1m0	These two bits enable extra modes of the shadow data pointer's operation, allowing for more compact and more efficient code size and execution. m1 m0 Behavior of the shadow data pointer. 0 0 8052 behavior. 0 1 DPTR is post-incremented after a MOVX or a MOVC instruction. 1 0 DPTR is post-decremented after a MOVX or MOVC instruction. 1 1 DPTR LSB is toggled after a MOVX or MOVC instruction. (This instruction can be useful for moving 8-bit blocks to/from 16-bit devices.)
3	DP0m1	Main Data Pointer Mode.
2	DP0m0	These two bits enable extra modes of the main data pointer operation, allowing for more compact and more efficient code size and execution. m1 m0 Behavior of the main data pointer. 0 0 8052 behavior. 0 1 DPTR is post-incremented after a MOVX or a MOVC instruction. 1 0 DPTR is post-decremented after a MOVX or MOVC instruction. 1 1 DPTR LSB is toggled after a MOVX or MOVC instruction. (This instruction can be useful for moving 8-bit blocks to/from 16-bit devices.)
1	----	This bit is not implemented to allow the INC DPCON instruction toggle the data pointer without incrementing the rest of the SFR.
0	DPSEL	Data Pointer Select. Cleared by the user to select the main data pointer. This means that the contents of this 24-bit register are placed into the three SFRs: DPL, DPH, and DPP. Set by the user to select the shadow data pointer. This means that the contents of a separate 24-bit register appears in the three SFRs: DPL, DPH, and DPP.

Note 1: This is the only place where the main and shadow data pointers are distinguished. Everywhere else in this data sheet wherever the DPTR is mentioned, operation on the active DPTR is implied.

Note 2: Only MOVC/MOVX @DPTR instructions are relevant above. MOVC/MOVX PC/@Ri instructions do not cause the DPTR to automatically post increment/decrement, and so on.

To illustrate the operation of DPCON, the following code copies 256 bytes of code memory at address D000H into XRAM starting from Address 0000H.

```
    MOV DPTR,#0         ; Main DPTR = 0
    MOV DPCON,#55H      ; Select shadow DPTR
                        ; DPTR1 increment mode,
                        ; DPTR0 increment mode
                        ; DPTR auto toggling ON
    MOV DPTR,#0D000H    ; Shadow DPTR = D000H
MOVELOOP:
    CLR A
    MOVC A,@A+DPTR      ; Get data
                        ; Post Inc DPTR
                        ; Swap to Main DPTR (Data)
    MOVX @DPTR,A        ; Put ACC in XRAM
                        ; Increment main DPTR
                        ; Swap Shadow DPTR (Code)
    MOV A, DPL
    JNZ MOVELOOP
```

POWER SUPPLY MONITOR

As its name suggests, the power supply monitor, once enabled, monitors the DV_{DD} supply on the ADuC841/ADuC842/ADuC843. It indicates when any of the supply pins drops below one of two user selectable voltage trip points, 2.93 V and 3.08 V. For correct operation of the power supply monitor function, AV_{DD} must be equal to or greater than 2.7 V. Monitor function is controlled via the PSMCON SFR. If enabled via the IEIP2 SFR, the monitor interrupts the core using the PSMI bit in the PSMCON SFR. This bit is not cleared until the failing power supply has returned above the trip point for at least 250 ms. This monitor function allows the user to save working registers to avoid possible data loss due to the low supply condition, and also ensures that normal code execution does not resume until a safe supply level has been well established. The supply monitor is also protected against spurious glitches triggering the interrupt circuit.

Note that the 5 V part has an internal POR trip level of 4.5 V, which means that there are no usable PSM levels on the 5 V part. The 3 V part has a POR trip level of 2.45 V, allowing all PSM trip points to be used.

PSMCON	**Power Supply Monitor Control Register**
SFR Address	DFH
Power-On Default	DEH
Bit Addressable	No

Table 22. PSMCON SFR Bit Designations

Bit No.	Name	Description
7	----	Reserved.
6	CMPD	DV_{DD} Comparator Bit. This is a read-only bit that directly reflects the state of the DV_{DD} comparator. Read 1 indicates that the DV_{DD} supply is above its selected trip point. Read 0 indicates that the DV_{DD} supply is below its selected trip point.
5	PSMI	Power Supply Monitor Interrupt Bit. This bit is set high by the MicroConverter if either CMPA or CMPD is low, indicating low analog or digital supply. The PSMI bit can be used to interrupt the processor. Once CMPD and/or CMPA return (and remain) high, a 250 ms counter is started. When this counter times out, the PSMI interrupt is cleared. PSMI can also be written by the user. However, if either comparator output is low, it is not possible for the user to clear PSMI.
4	TPD1	DV_{DD} Trip Point Selection Bits.
3	TPD0	These bits select the DV_{DD} trip point voltage as follows: TPD1 TPD0 Selected DV_{DD} Trip Point (V) 0 0 Reserved 0 1 3.08 1 0 2.93 1 1 Reserved
2	----	Reserved.
1	----	Reserved.
0	PSMEN	Power Supply Monitor Enable Bit. Set to 1 by the user to enable the power supply monitor circuit. Cleared to 0 by the user to disable the power supply monitor circuit.

WATCHDOG TIMER

The purpose of the watchdog timer is to generate a device reset or interrupt within a reasonable amount of time if the ADuC841/ADuC842/ADuC843 enter an erroneous state, possibly due to a programming error or electrical noise. The watchdog function can be disabled by clearing the WDE (watchdog enable) bit in the watchdog control (WDCON) SFR. When enabled, the watchdog circuit generates a system reset or interrupt (WDS) if the user program fails to set the watchdog (WDE) bit within a predetermined amount of time (see PRE3-0 bits in Table 23. The watchdog timer is clocked directly from the 32 kHz external crystal on the ADuC842/ADuC843. On the ADuC841, the watchdog timer is clocked by an internal R/C oscillator at 32 kHz ±10%. The WDCON SFR can be written only by user software if the double write sequence described in WDWR below is initiated on every write access to the WDCON SFR.

WDCON Watchdog Timer	**Control Register**
SFR Address	C0H
Power-On Default	10H
Bit Addressable	Yes

Table 23. WDCON SFR Bit Designations

Bit No.	Name	Description
7	PRE3	Watchdog Timer Prescale Bits.
6	PRE2	The watchdog timeout period is given by the equation $t_{WD} = (2^{PRE} \times (2^9 / f_{XTAL}))$
5	PRE1	(0 – PRE – 7; f_{XTAL} = 32.768 kHz (ADuC842/ADuC843), or 32kHz ± 10%(ADuC841))
4	PRE0	PRE3 PRE2 PRE1 PRE0 Timeout Period (ms) Action 0 0 0 0 15.6 Reset or Interrupt 0 0 0 1 31.2 Reset or Interrupt 0 0 1 0 62.5 Reset or Interrupt 0 0 1 1 125 Reset or Interrupt 0 1 0 0 250 Reset or Interrupt 0 1 0 1 500 Reset or Interrupt 0 1 1 0 1000 Reset or Interrupt 0 1 1 1 2000 Reset or Interrupt 1 0 0 0 0.0 Immediate Reset PRE3–0 > 1000 Reserved
3	WDIR	Watchdog Interrupt Response Enable Bit. If this bit is set by the user, the watchdog generates an interrupt response instead of a system reset when the watchdog timeout period has expired. This interrupt is not disabled by the CLR $\overline{EA}$ instruction, and it is also a fixed, high priority interrupt. If the watchdog is not being used to monitor the system, it can be used alternatively as a timer. The prescaler is used to set the timeout period in which an interrupt will be generated.
2	WDS	Watchdog Status Bit. Set by the watchdog controller to indicate that a watchdog timeout has occurred. Cleared by writing a 0 or by an external hardware reset. It is not cleared by a watchdog reset.
1	WDE	Watchdog Enable Bit. Set by the user to enable the watchdog and clear its counters. If this bit is not set by the user within the watchdog timeout period, the watchdog generates a reset or interrupt, depending on WDIR. Cleared under the following conditions: user writes 0, watchdog reset (WDIR = 0); hardware reset; PSM interrupt.

0	WDWR	Watchdog Write Enable Bit.
		To write data to the WDCON SFR involves a double instruction sequence. The WDWR bit must be set and the very next instruction must be a write instruction to the WDCON SFR.
		For example:
		`CLR EA ;disable interrupts while writing`
		`;to WDT`
		`SETB WDWR ;allow write to WDCON`
		`MOV WDCON,#72H ;enable WDT for 2.0s timeout`
		`SETB EA ;enable interrupts again (if rqd)`

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 24. 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 will also be clocked at this speed. The user should ensure that there is sufficient time between instructions to these registers to allow them to execute correctly.

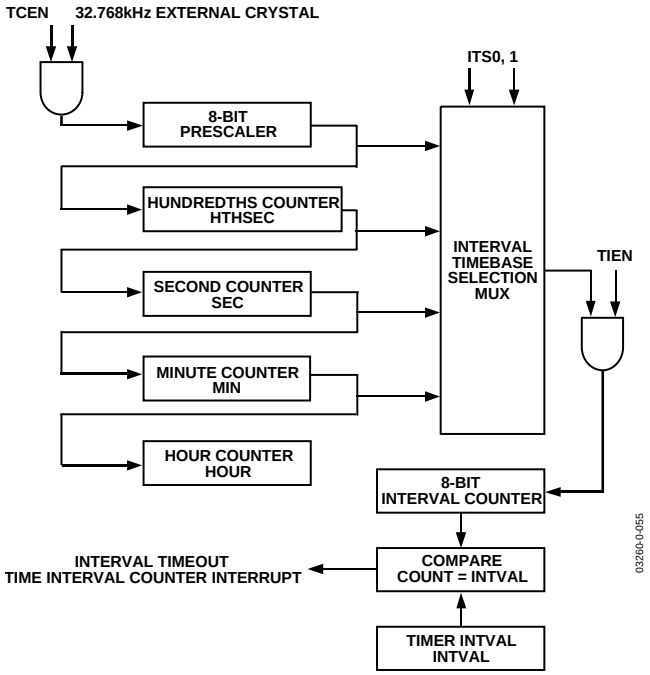


Figure 56. TIC, Simplified Block Diagram

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

Table 24. 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

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.

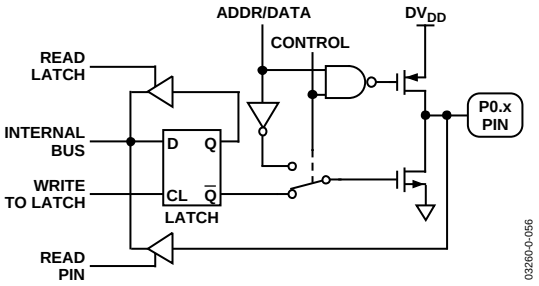


Figure 57. Port 0 Bit Latch and I/O Buffer

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, i.e., 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 will 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, i.e., 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 25.

Table 25. 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)

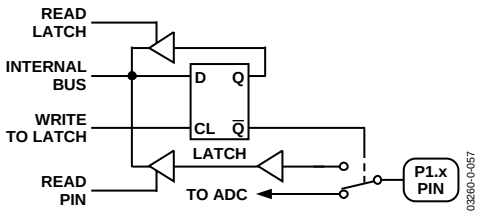


Figure 58. Port 1 Bit Latch and I/O Buffer

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.

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.

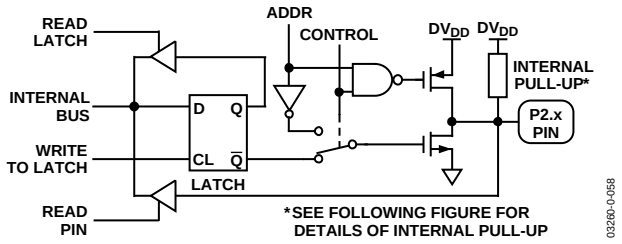


Figure 59. Port 2 Bit Latch and I/O Buffer

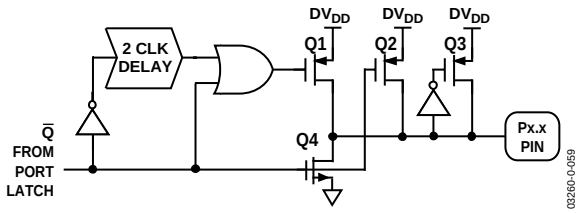


Figure 60. Internal Pull-Up Configuration

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 will 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 26. 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 26. 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	$\overline{\text{INT0}}$ (External Interrupt 0)
P3.3	$\overline{\text{INT1}}$ (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	$\overline{\text{WR}}$ (External Data Memory Write Strobe)
P3.7	$\overline{\text{RD}}$ (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.

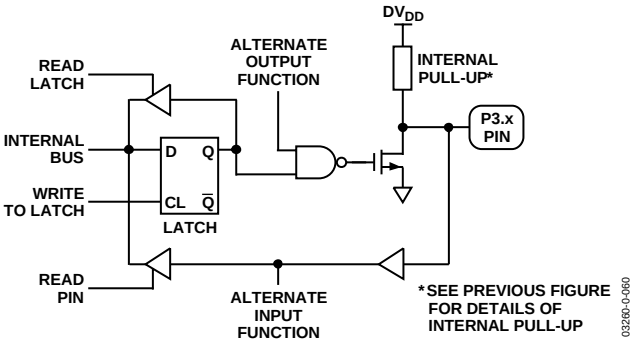


Figure 61. Port 3 Bit Latch and I/O Buffer

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.

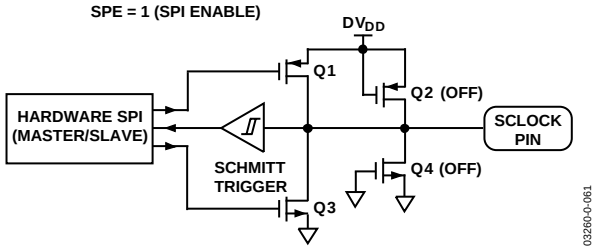


Figure 62. SCLOCK Pin I/O Functional Equivalent in SPI Mode

MOSI is shared with P3.3 and, as such, has the same configuration as the one shown in Figure 61.

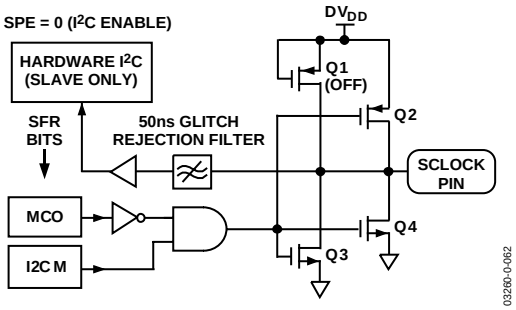


Figure 63. SCLOCK Pin I/O Functional Equivalent in I²C Mode

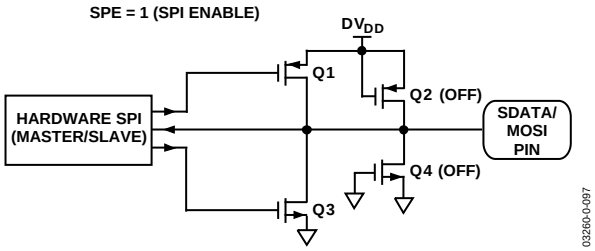


Figure 64. SDATA/MOSI Pin I/O Functional Equivalent in SPI Mode

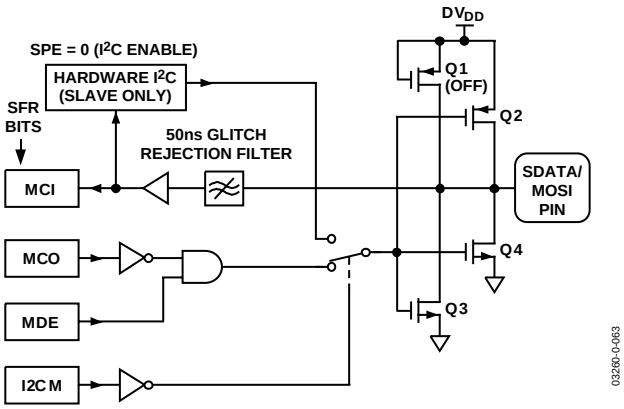


Figure 65. SDATA/MOSI Pin I/O Functional Equivalent in I²C Mode

Read-Modify-Write Instructions

Some 8051 instructions that read a port read the latch while others read the pin. The instructions that read the latch rather than the pins are the ones that read a value, possibly change it, and then rewrite it to the latch. These are called read-modify-write instructions, which are listed below. When the destination operand is a port or a port bit, these instructions read the latch rather than the pin.

Table 27. Read-Write-Modify Instructions

Instruction	Description
ANL	Logical AND, e.g., ANL P1, A
ORL	(Logical OR, e.g., ORL P2, A
XRL	(Logical EX-OR, e.g., XRL P3, A
JBC	Jump if Bit = 1 and clear bit, e.g., JBC P1.1, LABEL
CPL	Complement bit, e.g., CPL P3.0
INC	Increment, e.g., INC P2
DEC	Decrement, e.g., DEC P2
DJNZ	Decrement and Jump if Not Zero, e.g., DJNZ P3, LABEL
MOV PX.Y, C[1]	Move Carry to Bit Y of Port X
CLR PX.Y[1]	Clear Bit Y of Port X
SETB PX.Y[1]	Set Bit Y of Port X

[1] These instructions read the port byte (all 8 bits), modify the addressed bit, and then write the new byte back to the latch.

Read-modify-write instructions are directed to the latch rather than to the pin to avoid a possible misinterpretation of the voltage level of a pin. For example, a port pin might be used to drive the base of a transistor. When 1 is written to the bit, the transistor is turned on. If the CPU then reads the same port bit at the pin rather than the latch, it reads the base voltage of the transistor and interprets it as a Logic 0. Reading the latch rather than the pin returns the correct value of 1.

Timers/Counters

The ADuC841/ADuC842/ADuC843 have three 16-bit timer/counters: Timer 0, Timer 1, and Timer 2. The timer/counter hardware is included on-chip to relieve the processor core of the overhead inherent in implementing timer/counter functionality in software. Each timer/counter consists of two 8-bit registers: THx and TLx (x = 0, 1, and 2). All three can be configured to operate either as timers or as event counters.

In timer function, the TLx register is incremented every machine cycle. Thus, one can think of it as counting machine cycles. Since a machine cycle on a single-cycle core consists of one core clock period, the maximum count rate is the core clock frequency.

In counter function, the TLx register is incremented by a 1-to-0 transition at its corresponding external input pin: T0, T1, or T2. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. Since it takes two machine cycles (two core clock periods) to recognize a 1-to-0 transition, the maximum count rate is half the core clock frequency.

There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for a minimum of one full machine cycle. User configuration and control of all timer operating modes is achieved via three SFRs:

TMOD, TCON	Control and configuration for Timers 0 and 1.
T2CON	Control and configuration for Timer 2.
TMOD	Timer/Counter 0 and 1 Mode Register
SFR Address	89H
Power-On Default	00H
Bit Addressable	No

Table 28. TMOD SFR Bit Designations

Bit No.	Name	Description
7	Gate	Timer 1 Gating Control. Set by software to enable Timer/Counter 1 only while the $\overline{INT1}$ pin is high and the TR1 control bit is set. Cleared by software to enable Timer 1 whenever the TR1 control bit is set.
6	C/T	Timer 1 Timer or Counter Select Bit. Set by software to select counter operation (input from T1 pin). Cleared by software to select timer operation (input from internal system clock).
5	M1	Timer 1 Mode Select Bit 1 (Used with M0 Bit).
4	M0	Timer 1 Mode Select Bit 0. M1 M0 0 0 TH1 operates as an 8-bit timer/counter. TL1 serves as 5-bit prescaler. 0 1 16-Bit Timer/Counter. TH1 and TL1 are cascaded; there is no prescaler. 1 0 8-Bit Autoreload Timer/Counter. TH1 holds a value that is to be reloaded into TL1 each time it overflows. 1 1 Timer/Counter 1 Stopped.
3	Gate	Timer 0 Gating Control. Set by software to enable Timer/Counter 0 only while the $\overline{INT0}$ pin is high and the TR0 control bit is set. Cleared by software to enable Timer 0 whenever the TR0 control bit is set.
2	C/T	Timer 0 Timer or Counter Select Bit. Set by software to select counter operation (input from T0 pin). Cleared by software to select timer operation (input from internal system clock).
1	M1	Timer 0 Mode Select Bit 1.
0	M0	Timer 0 Mode Select Bit 0. M1 M0 0 0 TH0 operates as an 8-bit timer/counter. TL0 serves as a 5-bit prescaler. 0 1 16-Bit Timer/Counter. TH0 and TL0 are cascaded; there is no prescaler. 1 0 8-Bit Autoreload Timer/Counter. TH0 holds a value that is to be reloaded into TL0 each time it overflows. 1 1 TL0 is an 8-bit timer/counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only, controlled by Timer 1 control bits.

TCON	Timer/Counter 0 and 1 Control Register
SFR Address	88H
Power-On Default	00H
Bit Addressable	Yes

Table 29. TCON SFR Bit Designations

Bit No.	Name	Description
7	TF1	Timer 1 Overflow Flag. Set by hardware on a Timer/Counter 1 overflow. Cleared by hardware when the program counter (PC) vectors to the interrupt service routine.
6	TR1	Timer 1 Run Control Bit. Set by the user to turn on Timer/Counter 1. Cleared by the user to turn off Timer/Counter 1.
5	TF0	Timer 0 Overflow Flag. Set by hardware on a Timer/Counter 0 overflow. Cleared by hardware when the PC vectors to the interrupt service routine.
4	TR0	Timer 0 Run Control Bit. Set by the user to turn on Timer/Counter 0. Cleared by the user to turn off Timer/Counter 0.
3	IE1[1]	External Interrupt 1 ($\overline{\text{INT1}}$) Flag. Set by hardware by a falling edge or by a zero level being applied to the external interrupt pin, $\overline{\text{INT1}}$, depending on the state of Bit IT1. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
2	IT1[1]	External Interrupt 1 (IE1) Trigger Type. Set by software to specify edge-sensitive detection, i.e., 1-to-0 transition. Cleared by software to specify level-sensitive detection, i.e., zero level.
1	IE0[1]	External Interrupt 0 ($\overline{\text{INT0}}$) Flag. Set by hardware by a falling edge or by a zero level being applied to external interrupt pin $\overline{\text{INT0}}$, depending on the state of Bit IT0. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
0	IT0[1]	External Interrupt 0 (IE0) Trigger Type. Set by software to specify edge-sensitive detection, i.e.,1-to-0 transition. Cleared by software to specify level-sensitive detection, i.e., zero level.

[1]These bits are not used in the control of Timer/Counter 0 and 1, but are used instead in the control and monitoring of the external $\overline{\text{INT0}}$ and INT1 interrupt pins.

Timer/Counter 0 and 1 Data Registers

Each timer consists of two 8-bit registers. These can be used as independent registers or combined into a single 16-bit register depending on the timer mode configuration.

TH0 and TL0

Timer 0 high byte and low byte.
SFR Address = 8CH 8AH, respectively.

TH1 and TL1

Timer 1 high byte and low byte.
SFR Address = 8DH, 8BH, respectively.

TIMER/COUNTER 0 AND 1 OPERATING MODES

The following sections describe the operating modes for Timer/Counters 0 and 1. Unless otherwise noted, assume that these modes of operation are the same for both Timer 0 and Timer 1.

Mode 0 (13-Bit Timer/Counter)

Mode 0 configures an 8-bit timer/counter. Figure 66 shows Mode 0 operation. Note that the divide-by-12 prescaler is not present on the single-cycle core.

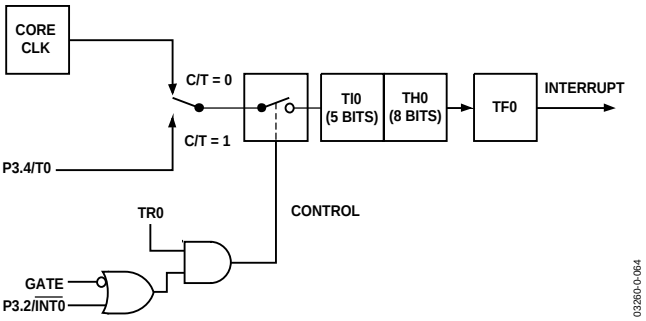


Figure 66. Timer/Counter 0, Mode 0

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the timer overflow flag, TF0. TF0 can then be used to request an interrupt. The counted input is enabled to the timer when TR0 = 1 and either Gate = 0 or $\overline{INT0}$ = 1. Setting Gate = 1 allows the timer to be controlled by external input $\overline{INT0}$ to facilitate pulse-width measurements. TR0 is a control bit in the special function register TCON; Gate is in TMOD. The 13-bit register consists of all 8 bits of TH0 and the lower five bits of TL0. The upper 3 bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers.

Mode 1 (16-Bit Timer/Counter)

Mode 1 is the same as Mode 0, except that the Mode 1 timer register is running with all 16 bits. Mode 1 is shown in Figure 67.

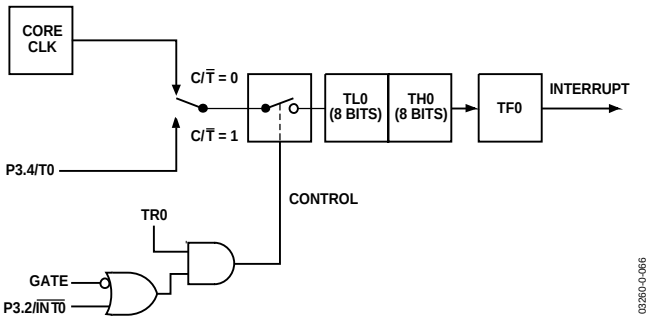


Figure 67. Timer/Counter 0, Mode 1

Mode 2 (8-Bit Timer/Counter with Autoreload)

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in Figure 68. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is preset by software. The reload leaves TH0 unchanged.

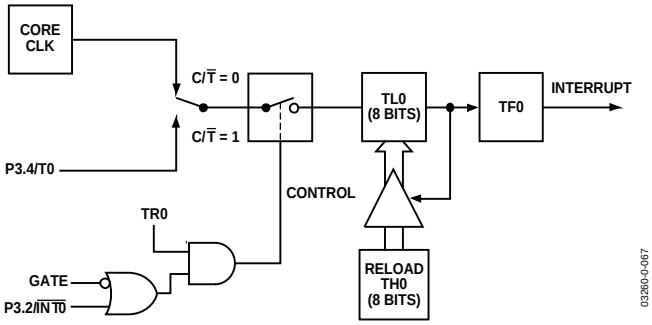


Figure 68. Timer/Counter 0, Mode 2

Mode 3 (Two 8-Bit Timer/Counters)

Mode 3 has different effects on Timer 0 and Timer 1. Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. This configuration is shown in Figure 69. TL0 uses the Timer 0 control bits: C/$\overline{T}$, Gate, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt. Mode 3 is provided for applications requiring an extra 8-bit timer or counter.

When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3, or it can still be used by the serial interface as a baud rate generator. In fact, it can be used in any application not requiring an interrupt from Timer 1 itself.

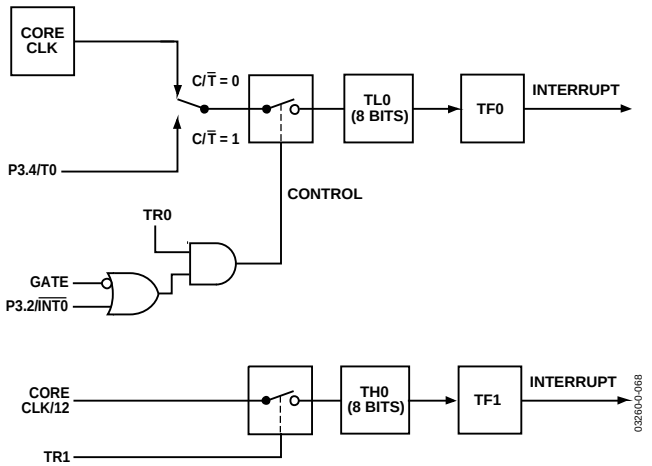


Figure 69. Timer/Counter 0, Mode 3

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

Table 30. 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.

TIMER/COUNTER OPERATING MODES

The following sections describe the operating modes for Timer/Counter 2. The operating modes are selected by bits in the T2CON SFR, as shown in Table 31.

Table 31. T2CON Operating Modes

RCLK (or) TCLK	CAP2	TR2	Mode
0	0	1	16-Bit Autoreload
0	1	1	16-Bit Capture
1	X	1	Baud Rate
X	X	0	OFF

16-Bit Autoreload Mode

Autoreload mode has two options that are selected by Bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over, it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still performs the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2. Autoreload mode is illustrated in Figure 70.

16-Bit Capture Mode

Capture mode also has two options that are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter that, upon overflowing, sets Bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still performs the above, but a l-to-0 transition on external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes Bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt. Capture mode is illustrated in Figure 71. The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1.

In either case, if Timer 2 is being used to generate the baud rate, the TF2 interrupt flag will not occur. Therefore, Timer 2 interrupts will not occur, so they do not have to be disabled. In this mode, the EXF2 flag, however, can still cause interrupts, which can be used as a third external interrupt. Baud rate generation is described as part of the UART serial port operation in the following section.

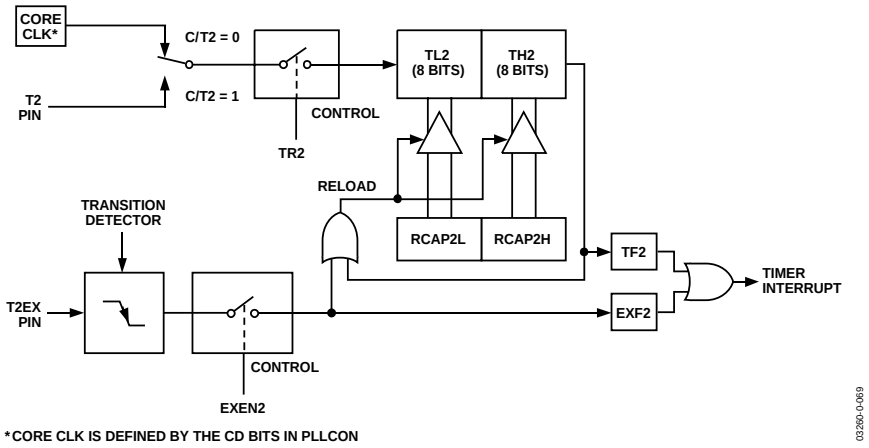


Figure 70. Timer/Counter 2, 16-Bit Autoreload Mode

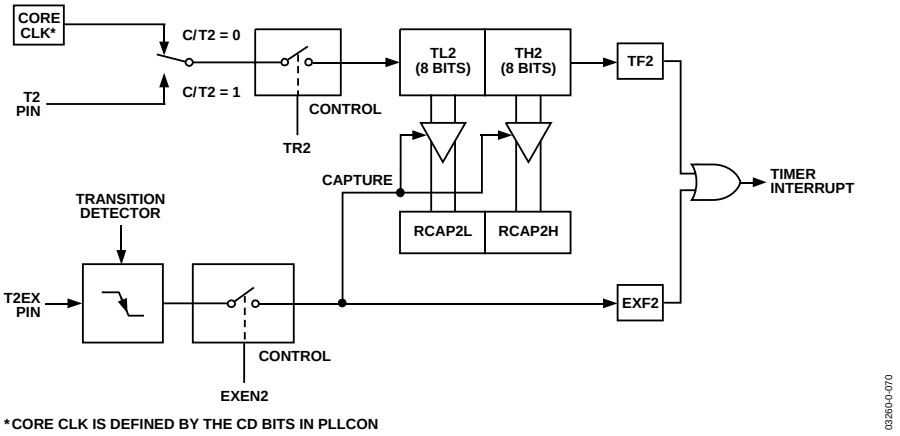


Figure 71. Timer/Counter 2, 16-Bit Capture Mode

UART SERIAL INTERFACE

The serial port is full-duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can begin receiving a second byte before a previously received byte has been read from the receive register. However, if the first byte still has not been read by the time reception of the second byte is complete, the first byte is lost. The physical interface to the serial data network is via Pins RxD(P3.0) and TxD(P3.1), while the SFR interface to the UART is comprised of SBUF and SCON, as described below.

SBUF

Both the serial port receive and transmit registers are accessed through the SBUF SFR (SFR address = 99H). Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

SCON UART	**Serial Port Control Register**
SFR Address	98H
Power-On Default	00H
Bit Addressable	Yes

Table 32. SCON SFR Bit Designations

Bit No.	Name	Description
7	SM0	UART Serial Mode Select Bits.
6	SM1	These bits select the serial port operating mode as follows: SM0 SM1 Selected Operating Mode. 0 0 Mode 0: Shift Register, fixed baud rate (Core_Clk/2). 0 1 Mode 1: 8-bit UART, variable baud rate. 1 0 Mode 2: 9-bit UART, fixed baud rate (Core_Clk/32) or (Core_Clk/16). 1 1 Mode 3: 9-bit UART, variable baud rate.
5	SM2	Multiprocessor Communication Enable Bit. Enables multiprocessor communication in Modes 2 and 3. In Mode 0, SM2 must be cleared. In Mode 1, if SM2 is set, RI is not activated if a valid stop bit was not received. If SM2 is cleared, RI is set as soon as the byte of data has been received. In Modes 2 or 3, if SM2 is set, RI is not activated if the received 9th data bit in RB8 is 0. If SM2 is cleared, RI is set as soon as the byte of data has been received.
4	REN	Serial Port Receive Enable Bit. Set by user software to enable serial port reception. Cleared by user software to disable serial port reception.
3	TB8	Serial Port Transmit (Bit 9). The data loaded into TB8 is the 9th data bit transmitted in Modes 2 and 3.
2	RB8	Serial Port Receiver Bit 9. The 9th data bit received in Modes 2 and 3 is latched into RB8. For Mode 1, the stop bit is latched into RB8.
1	TI	Serial Port Transmit Interrupt Flag. Set by hardware at the end of the 8th bit in Mode 0, or at the beginning of the stop bit in Modes 1, 2, and 3. TI must be cleared by user software.
0	RI	Serial Port Receive Interrupt Flag. Set by hardware at the end of the 8th bit in Mode 0, or halfway through the stop bit in Modes 1, 2, and 3. RI must be cleared by software.

Mode 0: 8-Bit Shift Register Mode

Mode 0 is selected by clearing both the SM0 and SM1 bits in the SFR SCON. Serial data enters and exits through RxD. TxD outputs the shift clock. Eight data bits are transmitted or received. Transmission is initiated by any instruction that writes to SBUF. The data is shifted out of the RxD line. The 8 bits are transmitted with the least significant bit (LSB) first.

Reception is initiated when the receive enable bit (REN) is 1 and the receive interrupt bit (RI) is 0. When RI is cleared, the data is clocked into the RxD line, and the clock pulses are output from the TxD line.

Mode 1: 8-Bit UART, Variable Baud Rate

Mode 1 is selected by clearing SM0 and setting SM1. Each data byte (LSB first) is preceded by a start bit (0) and followed by a stop bit (1). Therefore, 10 bits are transmitted on TxD or are received on RxD. The baud rate is set by the Timer 1 or Timer 2 overflow rate, or a combination of the two (one for transmission and the other for reception).

Transmission is initiated by writing to SBUF. The write to SBUF signal also loads a 1 (stop bit) into the 9th bit position of the transmit shift register. The data is output bit by bit until the stop bit appears on TxD and the transmit interrupt flag (TI) is automatically set, as shown in Figure 72.

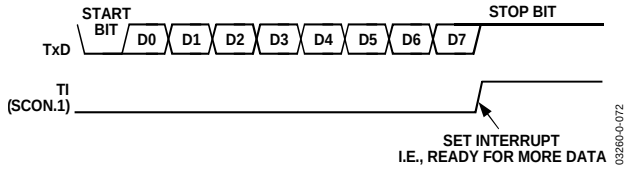


Figure 72. UART Serial Port Transmission, Mode 1

Reception is initiated when a 1-to-0 transition is detected on RxD. Assuming a valid start bit is detected, character reception continues. The start bit is skipped and the 8 data bits are clocked into the serial port shift register. When all 8 bits have been clocked in, the following events occur:

- The 8 bits in the receive shift register are latched into SBUF.
- The 9th bit (stop bit) is clocked into RB8 in SCON.
- The receiver interrupt flag (RI) is set.

This is the case if, and only if, all of the following conditions are met at the time the final shift pulse is generated:

- RI = 0
- Either SM2 = 0 or SM2 = 1
- The received stop bit = 1

If any of these conditions is not met, the received frame is irretrievably lost, and RI is not set.

Mode 2: 9-Bit UART with Fixed Baud Rate

Mode 2 is selected by setting SM0 and clearing SM1. In this mode, the UART operates in 9-bit mode with a fixed baud rate. The baud rate is fixed at Core_Clk/32 by default, although by setting the SMOD bit in PCON, the frequency can be doubled to Core_Clk/16. Eleven bits are transmitted or received: a start bit (0), 8 data bits, a programmable 9th bit, and a stop bit (1). The 9th bit is most often used as a parity bit, although it can be used for anything, including a 9th data bit if required.

To transmit, the 8 data bits must be written into SBUF. The 9th bit must be written to TB8 in SCON. When transmission is initiated, the 8 data bits (from SBUF) are loaded onto the transmit shift register (LSB first). The contents of TB8 are loaded into the 9th bit position of the transmit shift register. The transmission starts at the next valid baud rate clock. The TI flag is set as soon as the stop bit appears on TxD.

Reception for Mode 2 is similar to that of Mode 1. The 8 data bytes are input at RxD (LSB first) and loaded onto the receive shift register. When all 8 bits have been clocked in, the following events occur:

- The 8 bits in the receive shift register are latched into SBUF.
- The 9th data bit is latched into RB8 in SCON.
- The receiver interrupt flag (RI) is set.

This is the case if, and only if, all of the following conditions are met at the time the final shift pulse is generated:

- RI = 0
- Either SM2 = 0 or SM2 = 1
- The received stop bit = 1

If any of these conditions is not met, the received frame is irretrievably lost, and RI is not set.

Mode 3: 9-Bit UART with Variable Baud Rate

Mode 3 is selected by setting both SM0 and SM1. In this mode, the 8051 UART serial port operates in 9-bit mode with a variable baud rate determined by either Timer 1 or Timer 2. The operation of the 9-bit UART is the same as for Mode 2, but the baud rate can be varied as for Mode 1.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

UART Serial Port Baud Rate Generation

Mode 0 Baud Rate Generation

The baud rate in Mode 0 is fixed.

Mode 0 Baud Rate = (*Core Clock Frequency*/12)

Mode 2 Baud Rate Generation

The baud rate in Mode 2 depends on the value of the SMOD bit in the PCON SFR. If SMOD = 0, the baud rate is 1/32 of the core clock. If SMOD = 1, the baud rate is 1/16 of the core clock:

Mode 2 Baud Rate = (2^{SMOD}/32 × [*Core Clock Frequency*])

Modes 1 and 3 Baud Rate Generation

The baud rates in Modes 1 and 3 are determined by the overflow rate in Timer 1 or Timer 2, or in both (one for transmit and the other for receive).

Timer 1 Generated Baud Rates

When Timer 1 is used as the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

Modes 1 and 3 Baud Rate = (2^{SMOD}/32 × (*Timer 1 Overflow Rate*)

The Timer 1 interrupt should be disabled in this application. The timer itself can be configured for either timer or counter operation, and in any of its three running modes. In the most typical application, it is configured for timer operation in the autoreload mode (high nibble of TMOD = 0010 binary). In that case, the baud rate is given by the formula

Modes 1 and 3 Baud Rate =
(2^{SMOD}/32) × (*Core Clock*/ [256 − *TH1*])

Timer 2 Generated Baud Rates

Baud rates can also be generated using Timer 2. Using Timer 2 is similar to using Timer 1 in that the timer must overflow 16 times before a bit is transmitted/received. Because Timer 2 has a 16-bit autoreload mode, a wider range of baud rates is possible using Timer 2.

Modes 1 and 2 Baud Rate = (1/16) × (*Timer 2 Overflow Rate*)

Therefore, when Timer 2 is used to generate baud rates, the timer increments every two clock cycles rather than every core machine cycle as before. Thus, it increments six times faster than Timer 1, and therefore baud rates six times faster are possible. Because Timer 2 has 16-bit autoreload capability, very low baud rates are still possible.

Timer 2 is selected as the baud rate generator by setting the TCLK and/or RCLK in T2CON. The baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode as shown in Figure 73.

In this case, the baud rate is given by the formula

Modes 1 and 3 Baud Rate =
(*Core Clock*)/(16 × [65536 − (*RCAP 2H, RCAP 2L*)])

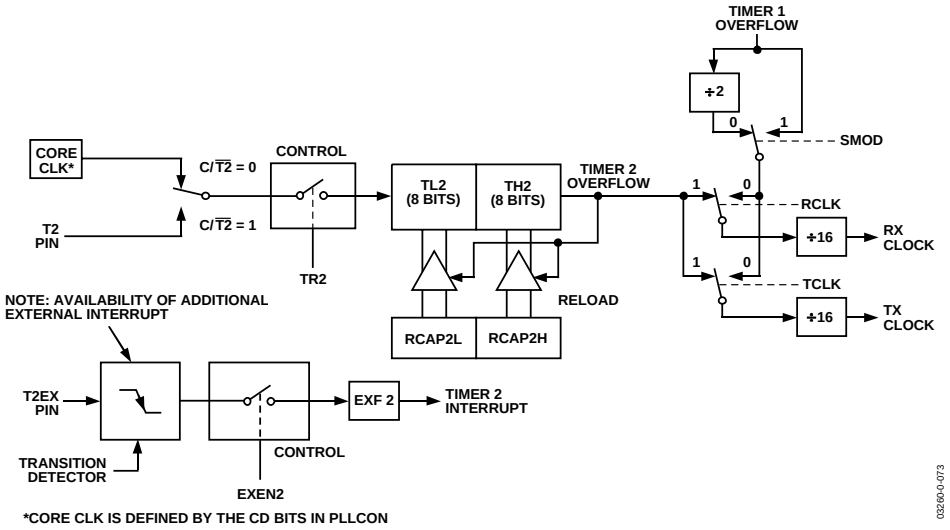


Figure 73. Timer 2, UART Baud Rates

Timer 3 Generated Baud Rates

The high integer dividers in a UART block mean that high speed baud rates are not always possible using some particular crystals. For example, using a 12 MHz crystal, a baud rate of 115200 is not possible. To address this problem, the part has added a dedicated baud rate timer (Timer 3) specifically for generating highly accurate baud rates. Timer 3 can be used instead of Timer 1 or Timer 2 for generating very accurate high speed UART baud rates including 115200 and 230400. Timer 3 also allows a much wider range of baud rates to be obtained. In fact, every desired bit rate from 12 bit/s to 393216 bit/s can be generated to within an error of ±0.8%. Timer 3 also frees up the other three timers, allowing them to be used for different applications. A block diagram of Timer 3 is shown in Figure 74.

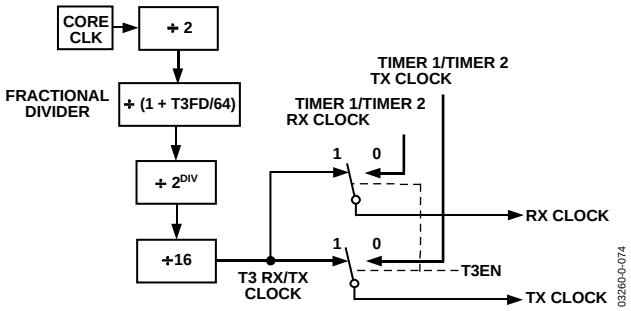


Figure 74. Timer 3, UART Baud Rates

Two SFRs (T3CON and T3FD) are used to control Timer 3. T3CON is the baud rate control SFR, allowing Timer 3 to be used to set up the UART baud rate, and setting up the binary divider (DIV).

The appropriate value to write to the DIV2-1-0 bits can be calculated using the following formula where f_{CORE} is defined in PLLCON SFR. Note that the DIV value must be rounded down.

$$DIV = \frac{\log\left(\frac{f_{CORE}}{16 \times Baud\ Rate}\right)}{\log(2)}$$

T3FD is the fractional divider ratio required to achieve the required baud rate. The appropriate value for T3FD can be calculated with the following formula:

$$T3FD = \frac{2 \times f_{CORE}}{2^{DIV-1} \times Baud\ Rate} - 64$$

Note that *T3FD* should be rounded to the nearest integer. Once the values for *DIV* and *T3FD* are calculated, the actual baud rate can be calculated with the following formula:

$$Actual\ Baud\ Rate = \frac{2 \times f_{CORE}}{2^{DIV-1} \times (T3FD + 64)}$$

For example, to get a baud rate of 115200 while operating at 16.7 MHz, i.e., CD = 0

$$DIV = \log(16777216/(16 \times 115200))/\log 2 = 3.18 = 3$$

$$T3FD = (2 \times 16777216)/(2^2 \times 115200) - 64 = 9 = 09H$$

Therefore, the actual baud rate is 114912 bit/s.

Table 33. T3CON SFR Bit Designations

Bit No.	Name	Description			
7	T3BAUDEN	T3UARTBAUD Enable.			
		Set to enable Timer 3 to generate the baud rate. When set, PCON.7, T2CON.4, and T2CON.5 are ignored.			
		Cleared to let the baud rate be generated as per a standard 8052.			
6		Reserved.			
5		Reserved.			
4		Reserved.			
3		Reserved.			
2	DIV2	Binary Divider Factor.			
1	DIV1	DIV2	DIV1	DIV0	Bin Divider
0	DIV0	0	0	0	1
		0	0	1	1
		0	1	0	1
		0	1	1	1
		1	0	0	1
		1	0	1	1
		1	1	0	1
		1	1	1	1

Table 34. Commonly Used Baud Rates Using Timer 3 with the 16.777216 MHz PLL Clock

Ideal Baud	CD	DIV	T3CON	T3FD	% Error
230400	0	2	82H	09H	0.25
115200	0	3	83H	09H	0.25
115200	1	2	82H	09H	0.25
115200	2	1	81H	09H	0.25
57600	0	4	84H	09H	0.25
57600	1	3	83H	09H	0.25
57600	2	2	82H	09H	0.25
57600	3	1	81H	09H	0.25
38400	0	4	84H	2DH	0.2
38400	1	3	83H	2DH	0.2
38400	2	2	82H	2DH	0.2
38400	3	1	81H	2DH	0.2
19200	0	5	85H	2DH	0.2
19200	1	4	84H	2DH	0.2
19200	2	3	83H	2DH	0.2
19200	3	2	82H	2DH	0.2
19200	4	1	81H	2DH	0.2
9600	0	6	86H	2DH	0.2
9600	1	5	85H	2DH	0.2
9600	2	4	84H	2DH	0.2
9600	3	3	83H	2DH	0.2
9600	4	2	82H	2DH	0.2
9600	5	1	81H	2DH	0.2

INTERRUPT SYSTEM

The ADuC841/ADuC842/ADuC843 provide a total of nine interrupt sources with two priority levels. The control and configuration of the interrupt system is carried out through three interrupt-related SFRs:

IE	Interrupt Enable Register
IP	Interrupt Priority Register
IEIP2	Secondary Interrupt Enable Register

IE	**Interrupt Enable Register**
SFR Address	A8H
Power-On Default	00H
Bit Addressable	Yes

Table 35. IE SFR Bit Designations

Bit No.	Name	Description
7	EA	Set by the user to enable, or cleared to disable all interrupt sources.
6	EADC	Set by the user to enable, or cleared to disable ADC interrupts.
5	ET2	Set by the user to enable, or cleared to disable Timer 2 interrupts.
4	ES	Set by the user to enable, or cleared to disable UART serial port interrupts.
3	ET1	Set by the user to enable, or cleared to disable 0 Timer 1 interrupts.
2	EX1	Set by the user to enable, or cleared to disable External Interrupt 1.
1	ET0	Set by the user to enable, or cleared to disable Timer 0 interrupts.
0	EX0	Set by the user to enable, or cleared to disable External Interrupt 0 .

IP	**Interrupt Priority Register**
SFR Address	B8H
Power-On Default	00H
Bit Addressable	Yes

Table 36. IP SFR Bit Designations

Bit No.	Name	Description
7	----	Reserved.
6	PADC	Written by the user to select the ADC interrupt priority (1 = High; 0 = Low).
5	PT2	Written by the user to select the Timer 2 interrupt priority (1 = High; 0 = Low).
4	PS	Written by the user to select the UART serial port interrupt priority (1 = High; 0 = Low).
3	PT1	Written by the user to select the Timer 1 interrupt priority (1 = High; 0 = Low).
2	PX1	Written by the user to select External Interrupt 1 priority (1 = High; 0 = Low).
1	PT0	Written by the user to select the Timer 0 interrupt priority (1 = High; 0 = Low).
0	PX0	Written by the user to select External Interrupt 0 priority (1 = High; 0 = Low).

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

Table 37. 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 38.

Table 38. 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 39.

Table 39. 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

HARDWARE DESIGN CONSIDERATIONS

This section outlines some of the key hardware design considerations that must be addressed when integrating the ADuC841/ADuC842/ADuC843 into any hardware system.

Clock Oscillator

The clock source for the parts can be generated by the internal PLL or by an external clock input. To use the internal PLL, connect a 32.768 kHz parallel resonant crystal between XTAL1 and XTAL2, and connect a capacitor from each pin to ground as shown in Figure 75. The parts contain an internal capacitance of 18 pF on the XTAL1 and XTAL2 pins, which is sufficient for most watch crystals. This crystal allows the PLL to lock correctly to give an f_{VCO} of 16.777216 MHz. If no crystal is present, the PLL will free run, giving an fVCO of 16.7 MHz ±20%. In this mode, the CD bits are limited to CD = 1, giving a max core clock of 8.38 MHz. This is useful if an external clock input is required. The part powers up and the PLL will free run; the user then writes to the CFG842 SFR in software to enable the external clock input on P3.4. Note that double the required clock must be provided externally since the part runs at CD = 1. A better solution is to use the ADuC841 with the external clock.

For the ADuC841, connect the crystal in the same manner; external capacitors should be connected as per the crystal manufacturer's recommendations. A minimum capacitance of 20 pF is recommended on XTAL1 and XTAL2. The ADuC841 will not operate if no crystal is present.

An external clock may be connected as shown in Figure 76 and Figure 77.

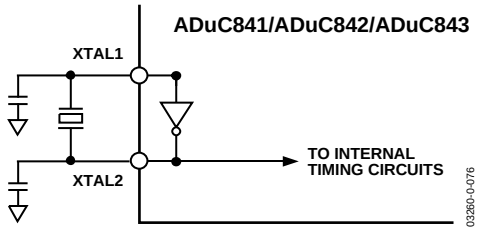


Figure 75. External Parallel Resonant Crystal Connections

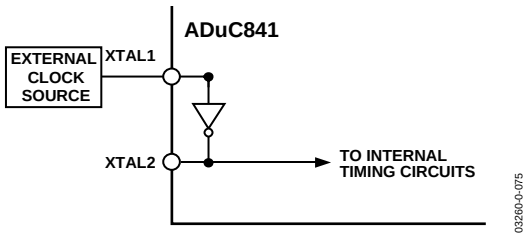


Figure 76. Connecting an External Clock Source (ADuC841)

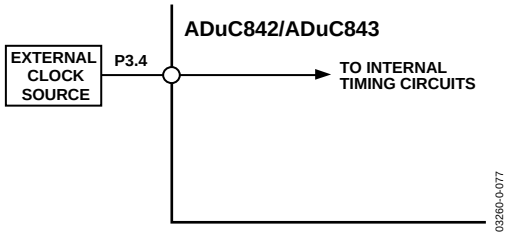


Figure 77. Connecting an External Clock Source (ADuC842/ADuC843)

Whether using the internal PLL or an external clock source, the parts' specified operational clock speed range is 400 kHz to 16.777216 MHz, (20 MHz, ADuC841). The core itself is static, and functions all the way down to dc. But at clock speeds slower that 400 kHz, the ADC can no longer function correctly. Therefore, to ensure specified operation, use a clock frequency of at least 400 kHz and no more than 20 MHz.

External Memory Interface

In addition to its internal program and data memories, the parts can access up to 16 MBytes of external data memory (SRAM). Note that the parts cannot access external program memory.

Figure 78 shows a hardware configuration for accessing up to 64 kBytes of external RAM. This interface is standard to any 8051 compatible MCU.

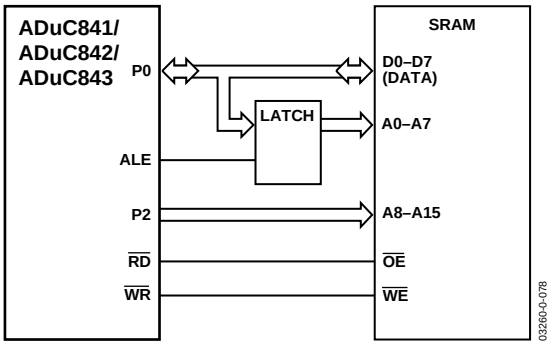


Figure 78. External Data Memory Interface (64 kBytes Address Space)

If access to more than 64 kBytes of RAM is desired, a feature unique to the ADuC841/ADuC842/ADuC843 allows addressing up to 16 MBytes of external RAM simply by adding an additional latch as illustrated in Figure 79.

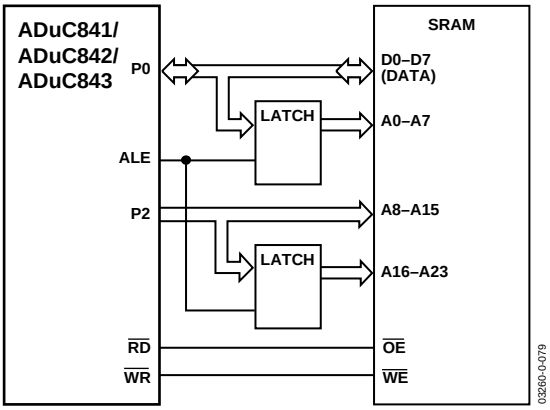


Figure 79. External Data Memory Interface (16 MBytes Address Space)

In either implementation, Port 0 (P0) serves as a multiplexed address/data bus. It emits the low byte of the data pointer (DPL) as an address, which is latched by a pulse of ALE prior to data being placed on the bus by the ADuC841/ADuC842/ADuC843 (write operation) or by the SRAM (read operation). Port 2 (P2) provides the data pointer page byte (DPP) to be latched by ALE, followed by the data pointer high byte (DPH). If no latch is connected to P2, DPP is ignored by the SRAM, and the 8051 standard of 64 kBytes external data memory access is maintained.

Power Supplies

The operational power supply voltage of the parts depends on whether the part is the 3 V version or the 5 V version. The specifications are given for power supplies within 2.7 V to 3.6 V or ±5% of the nominal 5 V level.

Note that Figure 80 and Figure 81 refer to the PQFP package. For the CSP package, connect the extra DV_{DD}, DGND, AV_{DD}, and AGND in the same manner. Also, the paddle on the bottom of the package should be soldered to a metal plate to provide mechanical stability. This metal plate should not be connected to ground.

Separate analog and digital power supply pins (AV_{DD} and DV_{DD}, respectively) allow AV_{DD} to be kept relatively free of the noisy digital signals that are often present on the system DV_{DD} line. However, though you can power AV_{DD} and DV_{DD} from two separate supplies if desired, you must ensure that they remain within ±0.3 V of one another at all times to avoid damaging the chip (as per the Absolute Maximum Ratings section). Therefore, it is recommended that unless AV_{DD} and DV_{DD} are connected directly together, back-to-back Schottky diodes should be connected between them, as shown in Figure 80.

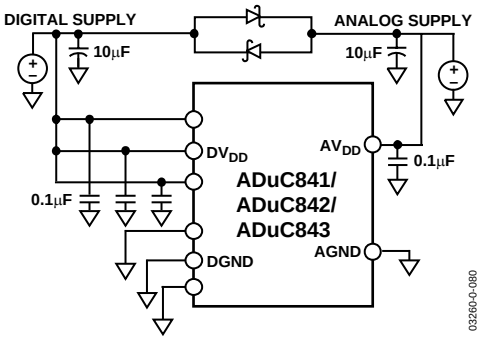


Figure 80. External Dual-Supply Connections

As an alternative to providing two separate power supplies, the user can help keep AV_{DD} quiet by placing a small series resistor and/or ferrite bead between it and DV_{DD}, and then decoupling AV_{DD} separately to ground. An example of this configuration is shown in Figure 81. With this configuration, other analog circuitry (such as op amps and voltage reference) can be powered from the AV_{DD} supply line as well. The user will still want to include back-to-back Schottky diodes between AV_{DD} and DV_{DD} to protect them from power-up and power-down transient conditions that could momentarily separate the two supply voltages.

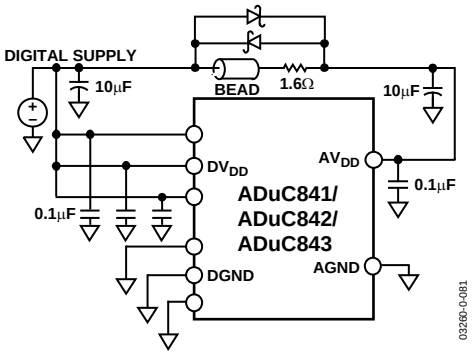


Figure 81. External Single-Supply Connections

Notice that in both Figure 80 and Figure 81, a large value (10 µF) reservoir capacitor sits on DV_{DD} and a separate 10 µF capacitor sits on AV_{DD}. Also, local small-value (0.1 µF) capacitors are located at each V_{DD} pin of the chip. As per standard design practice, be sure to include all of these capacitors, and ensure the smaller capacitors are close to each AV_{DD} pin with trace lengths as short as possible. Connect the ground terminal of each of these capacitors directly to the underlying ground plane. Finally, note that at all times, the analog and digital ground pins on the part must be referenced to the same system ground reference point.

Power Consumption

The currents consumed by the various sections of the part are shown in Table 40. The core values given represent the current drawn by DV_{DD}, while the rest (ADC, DAC, voltage ref) are pulled by the AV_{DD} pin and can be disabled in software when not in use. The other on-chip peripherals (such as the watchdog timer and the power supply monitor) consume negligible current, and are therefore lumped in with the core operating current here. Of course, the user must add any currents sourced by the parallel and serial I/O pins, and sourced by the DAC, in order to determine the total current needed at the supply pins. Also, current drawn from the DV_{DD} supply increases by approximately 10 mA during Flash/EE erase and program cycles.

Table 40. Typical I_{DD} of Core and Peripherals

	V_{DD} = 5 V	V_{DD} = 3 V
Core (Normal Mode)	(2.2 nA × M_{CLK})	(1.4 nA × M_{CLK})
ADC	1.7 mA	1.7 mA
DAC (Each)	250 µA	200 µA
Voltage Ref	200 µA	150 µA

Since operating DV_{DD} current is primarily a function of clock speed, the expressions for core supply current in Table 40 are given as functions of M_{CLK}, the core clock frequency. Plug in a value for M_{CLK} in hertz to determine the current consumed by the core at that oscillator frequency. Since the ADC and DACs can be enabled or disabled in software, add only the currents from the peripherals you expect to use. And again, do not forget to include current sourced by I/O pins, serial port pins, DAC outputs, and so forth, plus the additional current drawn during Flash/EE erase and program cycles. A software switch allows the chip to be switched from normal mode into idle mode, and also into full power-down mode. Brief descriptions of idle and power-down modes follow.

Power Saving Modes

In idle mode, the oscillator continues to run, but the core clock generated from the PLL is halted. The on-chip peripherals continue to receive the clock, and remain functional. The CPU status is preserved with the stack pointer and program counter, and all other internal registers maintain their data during idle mode. Port pins and DAC output pins retain their states in this mode. The chip recovers from idle mode upon receiving any enabled interrupt, or upon receiving a hardware reset.

In full power-down mode, both the PLL and the clock to the core are stopped. The on-chip oscillator can be halted or can continue to oscillate, depending on the state of the oscillator power-down bit in the PLLCON SFR. The TIC, being driven directly from the oscillator, can also be enabled during power-down. All other on-chip peripherals are, however, shut down. Port pins retain their logic levels in this mode, but the DAC output goes to a high impedance state (three-state). During full power-down mode, the part consumes a total of approximately 20 µA. There are five ways of terminating power-down mode:

Asserting the RESET Pin (Pin 15)

Returns to normal mode. All registers are set to their default state and program execution starts at the reset vector once the RESET pin is de-asserted.

Cycling Power

All registers are set to their default state and program execution starts at the reset vector approximately 128 ms later.

Time Interval Counter (TIC) Interrupt

Power-down mode is terminated, and the CPU services the TIC interrupt. The RETI at the end of the TIC ISR returns the core to the instruction after the one that enabled power-down.

I^2C or SPI Interrupt

Power-down mode is terminated, and the CPU services the I^2C/SPI interrupt. The RETI at the end of the ISR returns the core to the instruction after the one that enabled power-down. Note that the I^2C/SPI power-down interrupt enable bit (SERIPD) in the PCON SFR must be set to allow this mode of operation.

$\overline{\text{INT0}}$ Interrupt

Power-down mode is terminated, and the CPU services the $\overline{\text{INT0}}$ interrupt. The RETI at the end of the ISR returns the core to the instruction after the one that enabled power-down. The $\overline{\text{INT0}}$ pin must not be driven low during or within two machine cycles of the instruction that initiates power-down mode. Note that the $\overline{\text{INT0}}$ power-down interrupt enable bit (INT0PD) in the PCON SFR must be set to allow this mode of operation.

Power-On Reset (POR)

An internal POR is implemented on the ADuC841/ADuC842/ADuC843.

3 V Part

For DV_{DD} below 2.45 V, the internal POR holds the part in reset. As DV_{DD} rises above 2.45 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 2.7 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 82 illustrates the operation of the internal POR in detail.

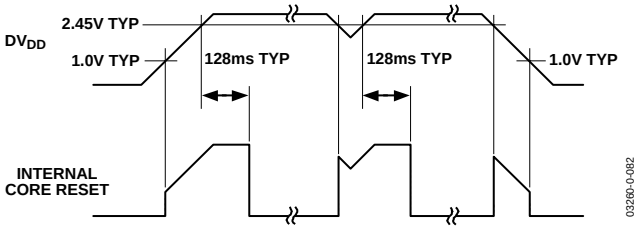


Figure 82. Internal POR Operation

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.

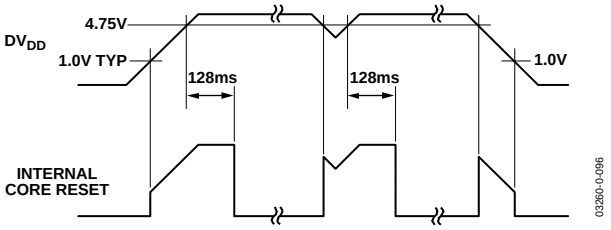


Figure 83. Internal POR Operation

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.

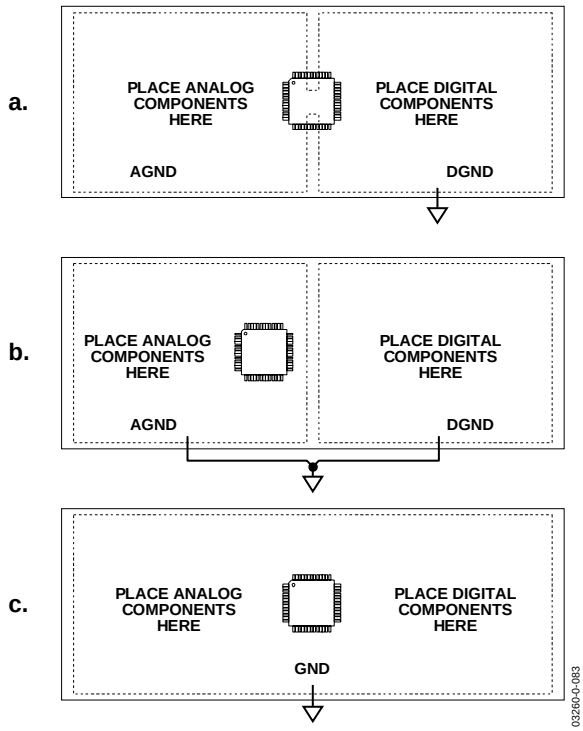


Figure 84. System Grounding Schemes

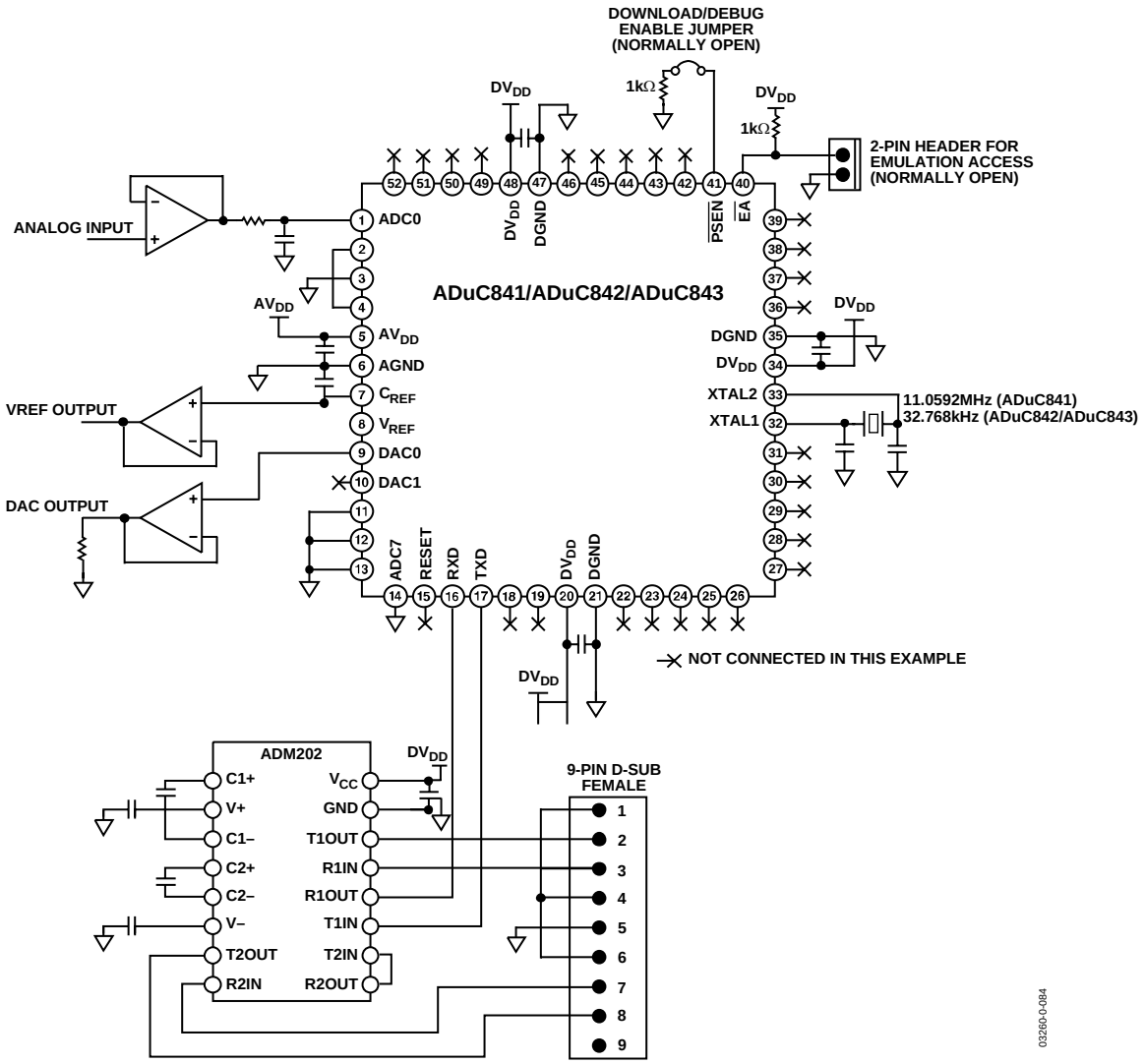


Figure 85. Example System (PQFP Package), DACs Not Present on ADuC843

OTHER HARDWARE CONSIDERATIONS

To facilitate in-circuit programming, plus in-circuit debug and emulation options, users will want to implement some simple connection points in their hardware to allow easy access to download, debug, and emulation modes.

In-Circuit Serial Download Access

Nearly all ADuC841/ADuC842/ADuC843 designs want to take advantage of the in-circuit reprogrammability of the chip. This is accomplished by a connection to the ADuC841/ADuC842/ADuC843's UART, which requires an external RS-232 chip for level translation if downloading code from a PC. Basic configuration of an RS-232 connection is illustrated in Figure 85 with a simple ADM202 based circuit. If users would rather not design an RS-232 chip onto a board, refer to Application Note uC006, *A 4-Wire UART-to-PC Interface*, (at www.analog.com/microconverter) for a simple (and zero-cost-per-board) method of gaining in-circuit serial download access to the part.

In addition to the basic UART connections, users also need a way to trigger the chip into download mode. This is accomplished via a 1 kΩ pull-down resistor that can be jumpered onto the $\overline{\text{PSEN}}$ pin, as shown in Figure 85. To get the part into download mode, simply connect this jumper and power-cycle the device (or manually reset the device, if a manual reset button is available), and it will be ready to serially receive a new program. With the jumper removed, the device comes up in normal mode (and runs the program) whenever power is cycled or RESET is toggled.

Note that $\overline{PSEN}$ 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, i.e., at power-up or upon an external manual reset. Note also that if any external circuitry unintentionally pulls $\overline{PSEN}$ 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 $\overline{PSEN}$ pin low, except for the external $\overline{PSEN}$ 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 $\overline{EA}$ 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 $\overline{EA}$ 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.

TIMING SPECIFICATIONS[1, 2, 3]

Table 41. AV_{DD} =2.7 V to 3.6 V or 4.75 V to 5.25 V, DV_{DD} = 2.7 V to 3.6 V or 4.75 V to 5.25 V; all specifications T_{MIN} to T_{MAX}, unless otherwise noted

Parameter		32.768 kHz External Crystal			
ADuC842/ADuC843 CLOCK INPUT (External Clock Driven XTAL1)		**Min**	**Typ**	**Max**	**Unit**
t_{CK}	XTAL1 Period		30.52		µs
t_{CKL}	XTAL1 Width Low		6.26		µs
t_{CKH}	XTAL1 Width High		6.26		µs
t_{CKR}	XTAL1 Rise Time		9		ns
t_{CKF}	XTAL1 Fall Time		9		ns
$1/t_{CORE}$	ADuC842/ADuC843 Core Clock Frequency[4]	0.131		16.78	MHz
t_{CORE}	ADuC842/ADuC843 Core Clock Period[5]		0.476		µs
t_{CYC}	ADuC842/ADuC843 Machine Cycle Time[6]	0.059	0.476	7.63	µs

[1] AC inputs during testing are driven at DV_{DD} – 0.5 V for a Logic 1 and 0.45 V for Logic 0. Timing measurements are made at V_{IH} min for Logic 1 and V_{IL} max for Logic 0, as shown in Figure 87.
[2] For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs, as shown in Figure 87.
[3] C_{LOAD} for all outputs = 80 pF, unless otherwise noted.
[4] ADuC842/ADuC843 internal PLL locks onto a multiple (512 times) of the 32.768 kHz external crystal frequency to provide a stable 16.78 MHz internal clock for the system. The core can operate at this frequency or at a binary submultiple called Core_Clk, selected via the PLLCON SFR.
[5] This number is measured at the default Core_Clk operating frequency of 2.09 MHz.
[6] ADuC842/ADuC843 machine cycle time is nominally defined as 1/Core_CLK.

Parameter		Variable External Crystal			
ADuC841 CLOCK INPUT (External Clock Driven XTAL1)		**Min**	**Typ**	**Max**	**Unit**
t_{CK}	XTAL1 Period	62.5		1000	ns
t_{CKL}	XTAL1 Width Low	20			ns
t_{CKH}	XTAL1 Width High	20			ns
t_{CKR}	XTAL1 Rise Time			20	ns
t_{CKF}	XTAL1 Fall Time			20	ns
$1/t_{CORE}$	ADuC841 Core Clock Frequency	0.131		20	MHz
t_{CORE}	ADuC841 Core Clock Period		0.476		µs
t_{CYC}	ADuC841 Machine Cycle Time	0.05	0.476	7.63	µs

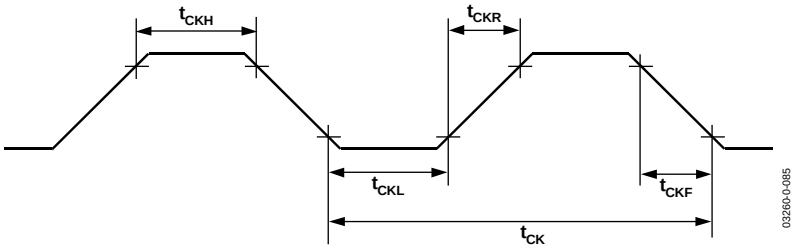


Figure 86. XTAL1 Input

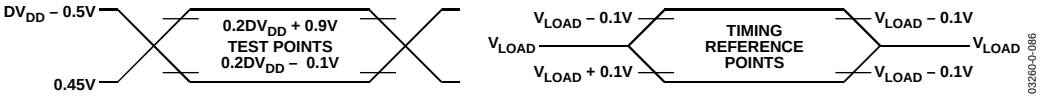


Figure 87. Timing Waveform Characteristics

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

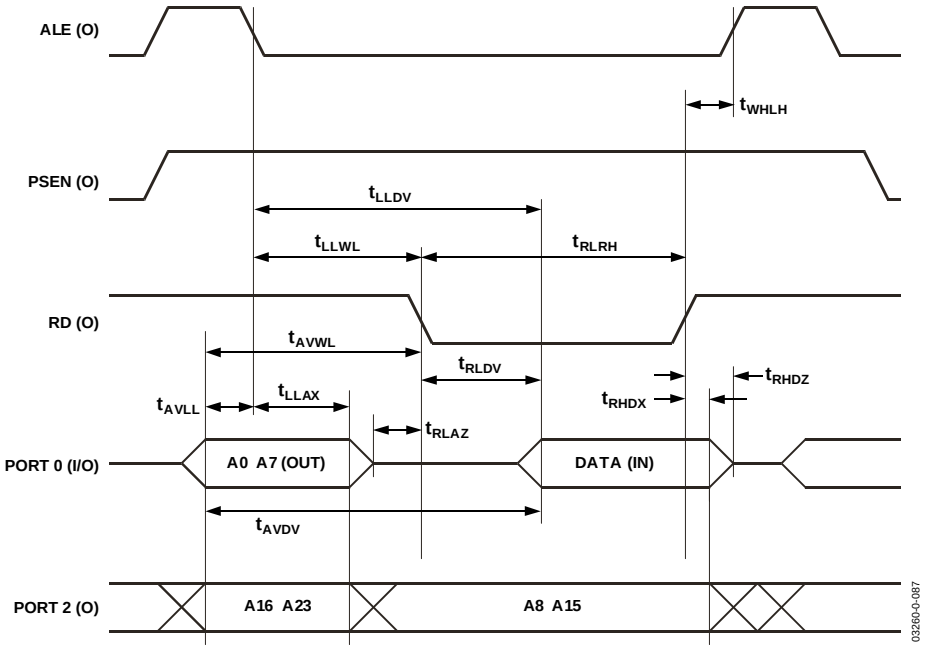


Figure 88. External Data Memory Read Cycle

Parameter		16 MHz Core Clk		8 MHz Core Clock		
EXTERNAL DATA MEMORY WRITE CYCLE		**Min**	**Max**	**Min**	**Max**	**Unit**
t_{WLWH}	$\overline{WR}$ Pulse Width	65		130		ns
t_{AVLL}	Address Valid after ALE Low	60		120		ns
t_{LLAX}	Address Hold after ALE Low	65		135		ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low		130		260	ns
t_{AVWL}	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	190		375		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	60		120		ns
t_{QVWH}	Data Setup before $\overline{WR}$	120		250		ns
t_{WHQX}	Data and Address Hold after $\overline{WR}$	380		755		ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	60		125		ns

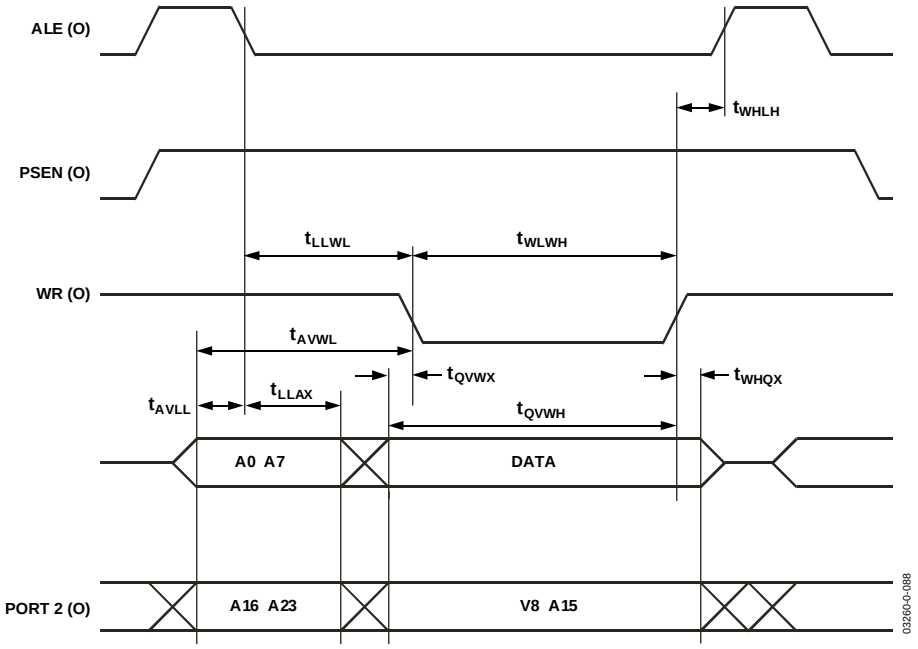


Figure 89. External Data Memory Write Cycle

Parameter				
I²C COMPATIBLE INTERFACE TIMING		**Min**	**Max**	**Unit**
t_L	SCLOCK Low Pulse Width	1.3		µs
t_H	SCLOCK High Pulse Width	0.6		µs
t_{SHD}	Start Condition Hold Time	0.6		µs
t_{DSU}	Data Setup Time	100		µs
t_{DHD}	Data Hold Time		0.9	µs
t_{RSU}	Setup Time for Repeated Start	0.6		µs
t_{PSU}	Stop Condition Setup Time	0.6		µs
t_{BUF}	Bus Free Time between a Stop Conditionand a Start Condition	1.3		µs
t_R	Rise Time of Both SCLOCK and SDATA		300	ns
t_F	Fall Time of Both SCLOCK and SDATA		300	ns
t_{SUP}[1]	Pulse Width of Spike Suppressed		50	ns

[1]Input filtering on both the SCLOCK and SDATA inputs suppresses noise spikes less than 50 ns.

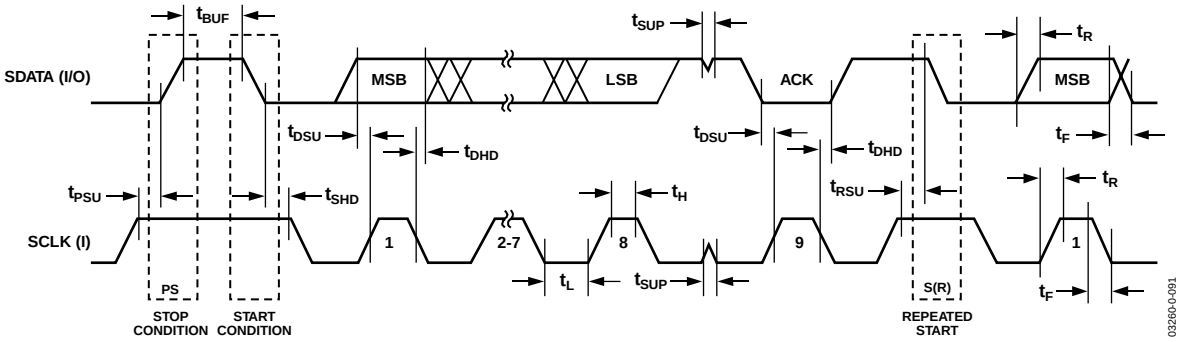


Figure 90. I²C Compatible Interface Timing

Parameter		Min	Typ	Max	Unit
SPI MASTER MODE TIMING (CPHA = 1)					
t_{SL}	SCLOCK Low Pulse Width[1]		476		ns
t_{SH}	SCLOCK High Pulse Width[1]		476		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

[1] Characterized under the following conditions:
a. Core clock divider bits CD2, CD1, and CD0 bits in PLLCON SFR set to 0, 1, and 1, respectively, i.e., core clock frequency = 2.09 MHz.
b. SPI bit-rate selection bits SPR1 and SPR0 in SPICON SFR set to 0 and 0, respectively.

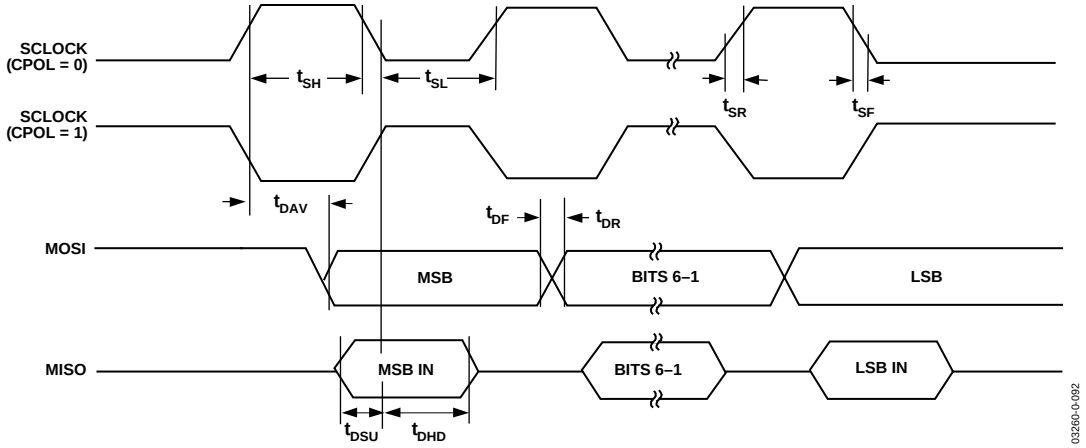


Figure 91. SPI Master Mode Timing (CPHA = 1)

Parameter		Min	Typ	Max	Unit
SPI MASTER MODE TIMING (CPHA = 0)					
t_{SL}	SCLOCK Low Pulse Width[1]		476		ns
t_{SH}	SCLOCK High Pulse Width[1]		476		ns
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns
t_{DOSU}	Data Output Setup before SCLOCK Edge			150	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

[1] Characterized under the following conditions:
a. Core clock divider bits CD2, CD1, and CD0 bits in PLLCON SFR set to 0, 1, and 1, respectively, i.e., core clock frequency = 2.09 MHz.
b. SPI bit-rate selection bits SPR1 and SPR0 in SPICON SFR set to 0 and 0, respectively.

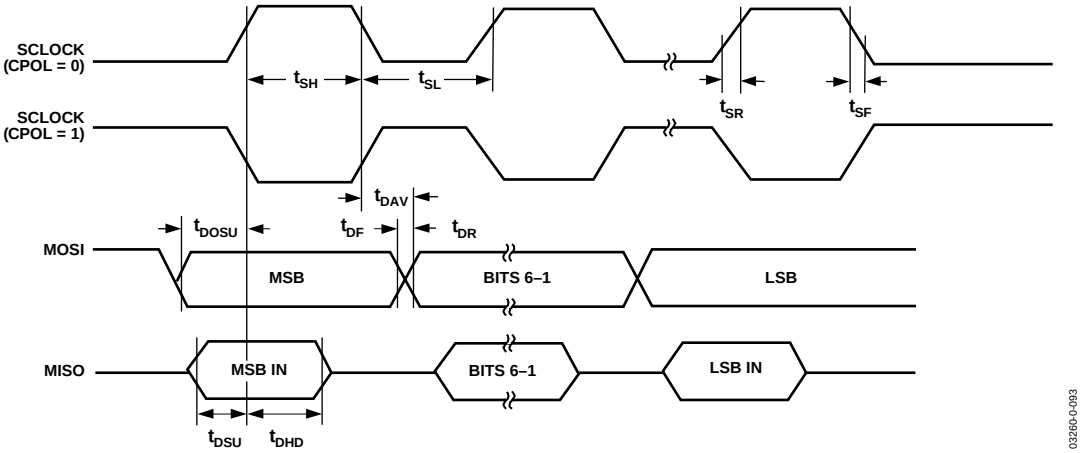


Figure 92. SPI Master Mode Timing (CPHA = 0)

Parameter					
SPI SLAVE MODE TIMING (CPHA = 1)		**Min**	**Typ**	**Max**	**Unit**
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

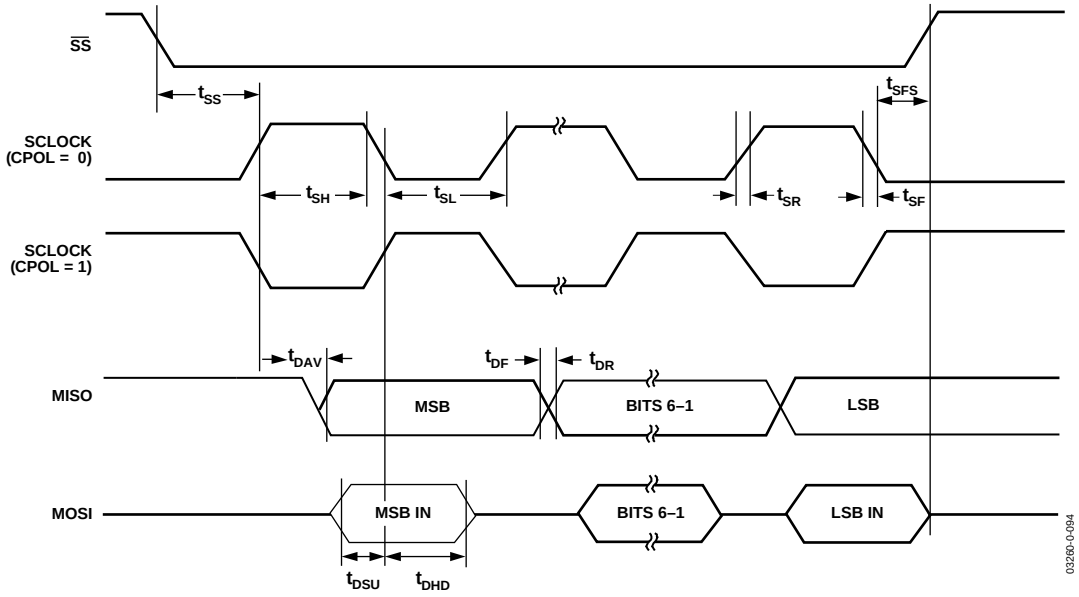


Figure 93. SPI Slave Mode Timing (CPHA = 1)

Parameter					
SPI SLAVE MODE TIMING (CPHA = 0)		**Min**	**Typ**	**Max**	**Unit**
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_{DOSS}	Data Output Valid after $\overline{SS}$ Edge			20	ns
t_{SFS}	$\overline{SS}$ High after SCLOCK Edge				ns

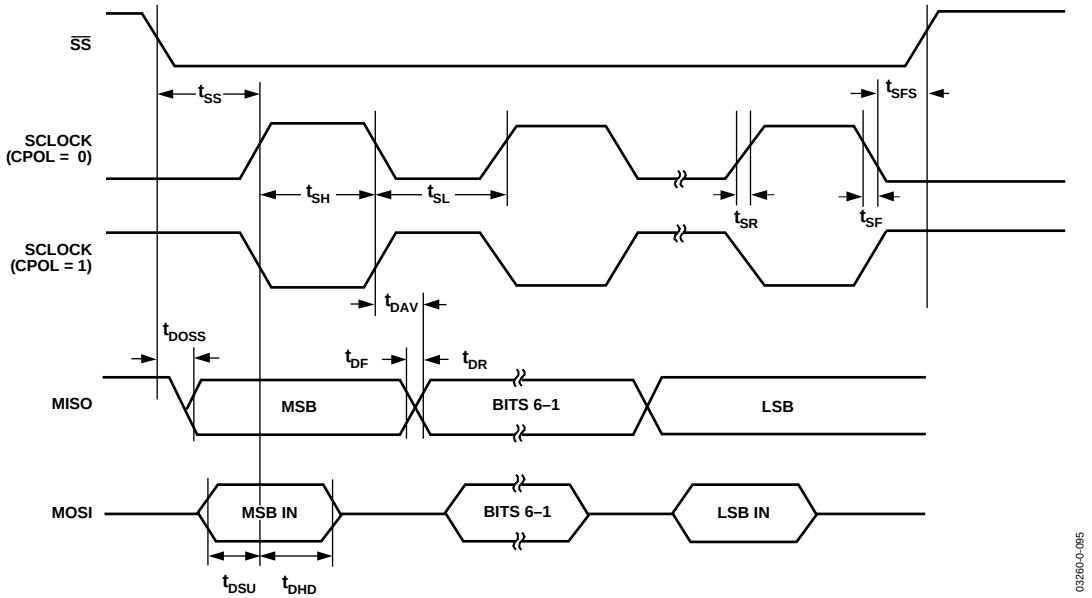
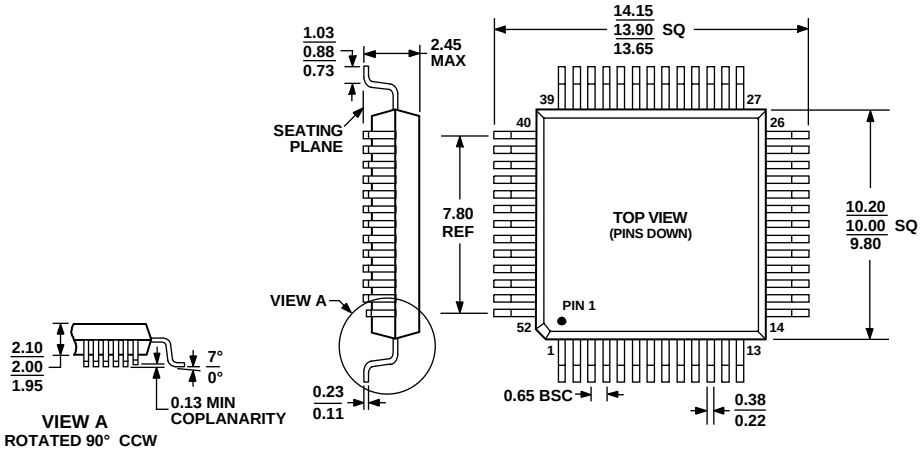


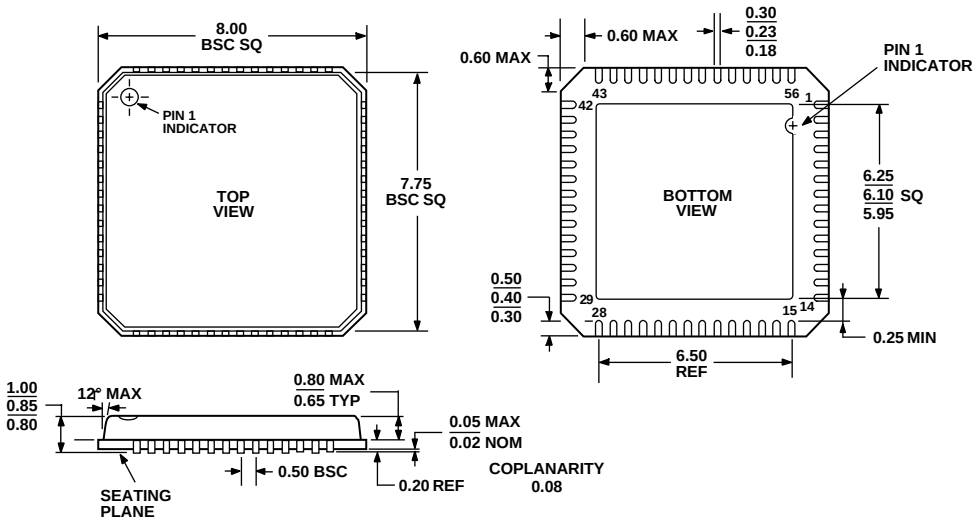
Figure 94. SPI Slave Mode Timing (CPHA = 0)

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-112-AC-1

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



COMPLIANT TO JEDEC STANDARDS MO-220-VLLD-2

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

ORDERING GUIDES

Table 42. ADuC841 Ordering Guide

Model	Supply Voltage V_{DD}	User Program Code Space	Temperature Range	Package Description	Package Option
ADuC841BS62-5	5	62	–40°C to +85°C	52-Lead Plastic Quad Flatpack	S-52
ADuC841BS62-3	3	62	–40°C to +85°C	52-Lead Plastic Quad Flatpack	S-52
ADuC841BCP62-5	5	62	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC841BCP62-3	3	62	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC841BCP8-5	5	8	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC841BCP8-3	3	8	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
EVAL-ADuC841QS	5			QuickStart Development System	
EVAL-ADuC841QSP[2]	5			QuickStart Plus Development System	

Table 43. ADuC842 Ordering Guide

Model	Supply Voltage V_{DD}	User Program Code Space	Temperature Range	Package Description	Package Option
ADuC842BS62-5	5	62	–40°C to +85°C	52-Lead Plastic Quad Flatpack	S-52
ADuC842BS62-3	3	62	–40°C to +85°C	52-Lead Plastic Quad Flatpack	S-52
ADuC842BCP62-5	5	62	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC842BCP62-3	3	62	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC842BCP32-5	5	32	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC842BCP32-3	3	32	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC842BCP8-5	5	8	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC842BCP8-3	3	8	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
EVAL-ADuC842QS	5			QuickStart Development System	
EVAL-ADuC842QSP[2]	5			QuickStart Plus Development System	

Table 44. ADuC843 Ordering Guide

Model	Supply Voltage V_{DD}	User Program Code Space	Temperature Range	Package Description	Package Option
ADuC843BS62-5	5	62	–40°C to +85°C	52-Lead Plastic Quad Flatpack	S-52
ADuC843BS62-3	3	62	–40°C to +85°C	52-Lead Plastic Quad Flatpack	S-52
ADuC843BCP62-5	5	62	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC843BCP62-3	3	62	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC843BCP32-5	5	32	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC843BCP32-3	3	32	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC843BCP8-5	5	8	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
ADuC843BCP8-3	3	8	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
EVAL-ADuC842QS[1]	5			QuickStart Development System	
EVAL-ADuC842QSP[1, 2]	5			QuickStart Plus Development System	

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

[2]The Quickstart Plus system can only be ordered directly from Accutron. It can be purchased from the website www.accutron.com.

Notes

Purchase of licensed I²C components of Analog Devices or one of its sublicensed Associated Companies conveys a license for the purchaser under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

© 2003 Analog Devices, Inc. All rights reserved. Trademarks and registered trademarks are the property of their respective owners.
C03260-0-11/03(0)

ANALOG DEVICES

www.analog.com

ADUC842 Model Options

Model	Package	Pins	Temp Range
ADUC842BCPZ32-3	56	LFCSP	Ind
ADUC842BCPZ32-5	56	LFCSP	Ind
ADUC842BCPZ62-3	56	LFCSP	Ind
ADUC842BCPZ62-5	56	LFCSP	Ind
ADUC842BCPZ8-3	56	LFCSP	Ind
ADUC842BCPZ8-5	56	LFCSP	Ind
ADUC842BSZ62-3	52	MQFP	Ind
ADUC842BSZ62-5	52	MQFP	Ind