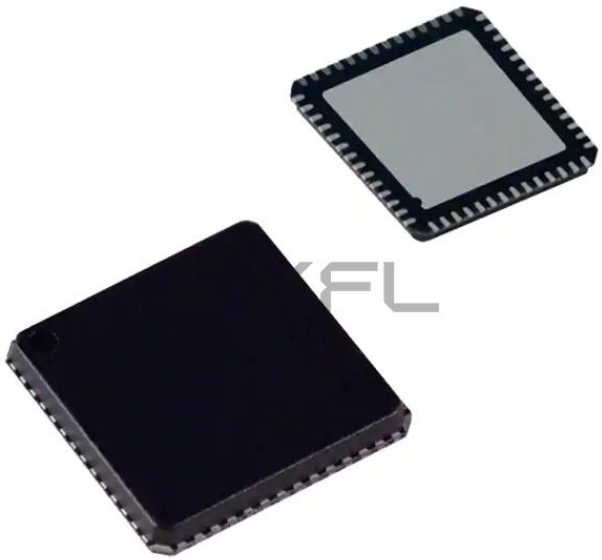

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8052
Core Size	8-Bit
Speed	16.78MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, PSM, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	32KB (32K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	56-VFQFN Exposed Pad, CSP
Supplier Device Package	56-LFCSP-VQ (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc843bcp32z-5

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
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 = 16.78 MHz/8.38 MHz 5 V/3 V
	10	3.5	mA typ	Core CLK = 16.78 MHz/8.38 MHz 5 V/3 V
AV_{DD} Current	3	2	µA typ	Core CLK = 16.78 MHz/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.

Pin No.	Mnemonic	Type[1]	Description
31	P2.3/A11/A19	I/O	Input/Output Port 2 (P2.3). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A11). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A19). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
32	XTAL1	I	Input to the Inverting Oscillator Amplifier.
33	XTAL2	O	Output of the Inverting Oscillator Amplifier.
36	P2.4/A12/A20	I/O	Input/Output Port 2 (P2.4). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A12). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A20). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
37	P2.5/A13/A21	I/O	Input/Output Port 2 (P2.5). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A13). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A21). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
38	P2.6/PWM0/A14/A22	I/O	Input/Output Port 2 (P2.6). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. PWM 0 Voltage Output (PWM0). PWM outputs can be configured to use Port 2.6 and Port 2.7 or Port 3.4 and Port 3.3. External Memory Addresses (A14). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A22). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
39	P2.7/PWM1/A15/A23	I/O	Input/Output Port 2 (P2.7). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. PWM 1 Voltage Output (PWM1). See the CFG841/CFG842 register for further information. External Memory Addresses (A15). Port 2 emits the middle-order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A23). Port 2 emits the high-order address bytes during accesses to the external 24-bit external data memory space.
40	$\overline{EA}$	I	External Access Enable, Logic Input. When held high, this input enables the device to fetch code from internal program memory locations. The devices do not support external code memory. Do not leave this pin floating.
41	$\overline{PSEN}$	O	Program Store Enable, Logic Output. This pin remains low during internal program execution. $\overline{PSEN}$ enables serial download mode when pulled low through a resistor on power-up or reset. On reset, this pin momentarily becomes an input and the status of the pin is sampled. If there is no pull-down resistor in place, the pin goes momentarily high and then user code executes. If a pull-down resistor is in place, the embedded serial download/debug kernel executes.
42	ALE	O	Address Latch Enable, Logic Output. This output latches the low byte and page byte for 24-bit address space accesses of the address into external data memory.

Pin No.	Mnemonic	Type[1]	Description
12	DAC1	O	Voltage Output from DAC1. This pin is a no connect on the ADuC843.
13	P1.4/ADC4		Input Port 1 (P1.0). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 4 (ADC4). Channel selection is via ADCCON2 SFR.
14	P1.5/ADC5/$\overline{SS}$	I	Input Port 1 (P1.5). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 5 (ADC5). Channel selection is via ADCCON2 SFR.
			Slave Select Input for the SPI Interface ($\overline{SS}$).
15	P1.3/ADC6	I	Input Port 1 (P1.3). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 6 (ADC6). Channel selection is via ADCCON2 SFR.
16	P1.7/ADC7	I	Input Port 1(P1.7). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 7 (ADC7). Channel selection is via ADCCON2 SFR.
17	RESET	I	Reset. Digital Input. A high level on this pin for 24 master clock cycles while the oscillator is running resets the device.
18	P3.0/RxD	I/O	Input/Output Port 3 (P3.0). 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.
			Receiver Data Input (Asynchronous) or Data Input/Output (Synchronous) of the Serial (UART) Port (RxD).
19	P3.1/TxD	I/O	Input/Output Port 3 (P3.1). 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.
			Transmitter Data Output (Asynchronous) or Clock Output (Synchronous) of the Serial (UART) Port (TxD).
20	P3.2/$\overline{INT0}$	I/O	Input/Output Port 3 (P3.2). 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.
			Interrupt 0 ($\overline{INT0}$). 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.
21	P3.3/$\overline{INT1}$/MISO/PWM1	I/O	Input/Output Port 3 (P3.3). 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.
			Interrupt 1 ($\overline{INT1}$). 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.
			SPI Master Input/Slave Output Data I/O Pin for SPI Serial Interface (MISO).
			PWM 1 Voltage Output (PWM1). See the CFG841/CFG842 register for further information.
22, 36, 51	DV_{DD}	P	Digital Positive Supply Voltage. 3 V or 5 V nominal.
23, 37, 38, 50	DGND	G	Digital Ground. DGND is the ground reference point for the digital circuitry.

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, that is, +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

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

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.

FUNCTIONAL DESCRIPTION

8052 INSTRUCTION SET

Table 5 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 5. 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

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 8. 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 9. 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 starts again when DMA is started and operates 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.

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.

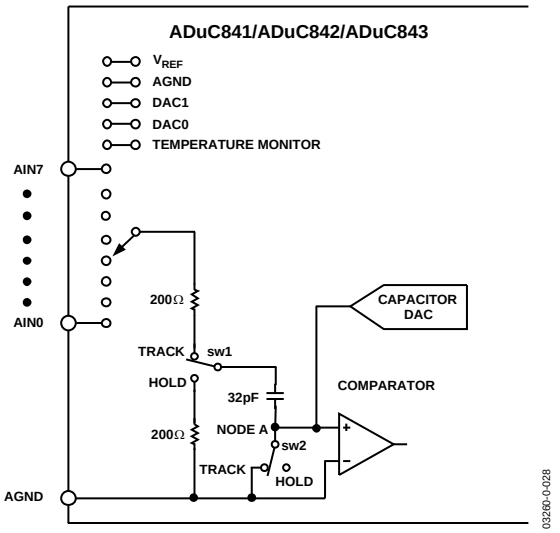

Figure 30. Internal ADC Structure

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.

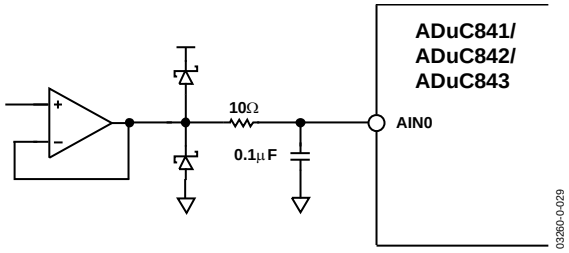

Figure 31. Buffering Analog Inputs

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 11 illustrates examples of how source impedance can affect dc accuracy.

Table 11. 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 12. Check Analog Devices website www.analog.com for details on these and other op amps and instrumentation amps.

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 17. 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 rails. 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 Σ-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 18.

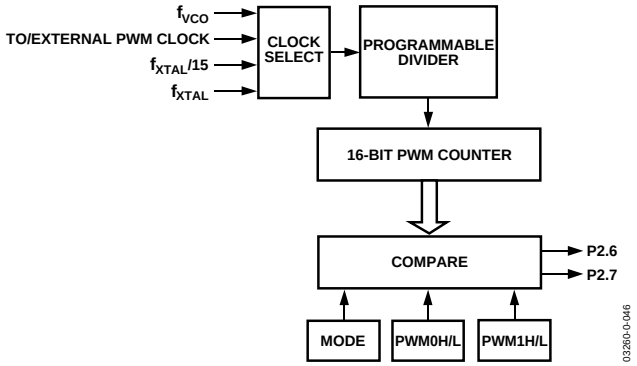

Figure 47. PWM Block Diagram

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 18. 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}	

Using the SPI Interface

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

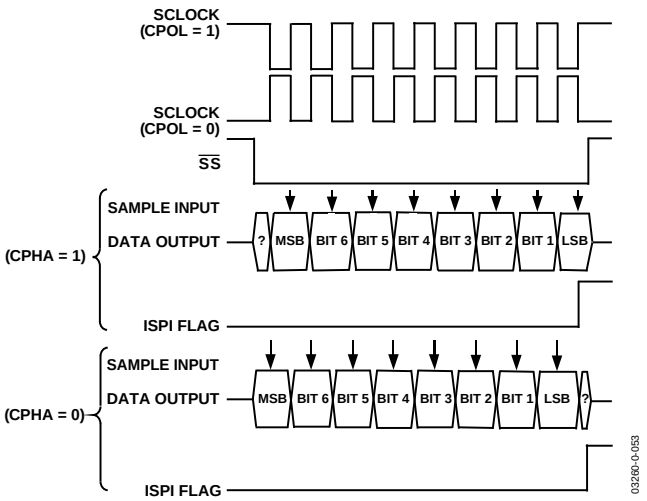

Figure 54. SPI Timing, All Modes

SPI Interface—Master Mode

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

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

SPI Interface—Slave Mode

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

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 22.

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

Table 22. 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
```

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 24. 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 24. 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 is 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)`

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

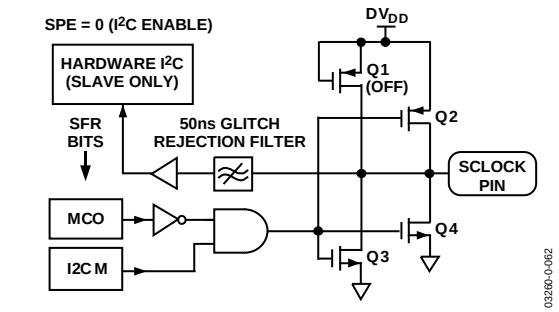

Figure 63. SCLOCK Pin I/O Functional Equivalent in I²C Mode

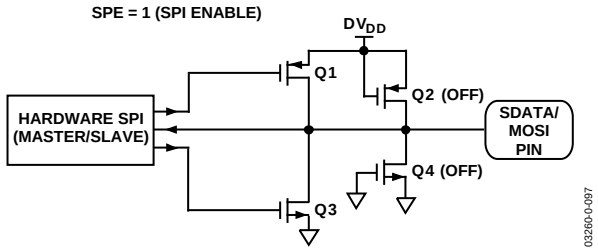

Figure 64. SDATA/MOSI Pin I/O Functional Equivalent in SPI Mode

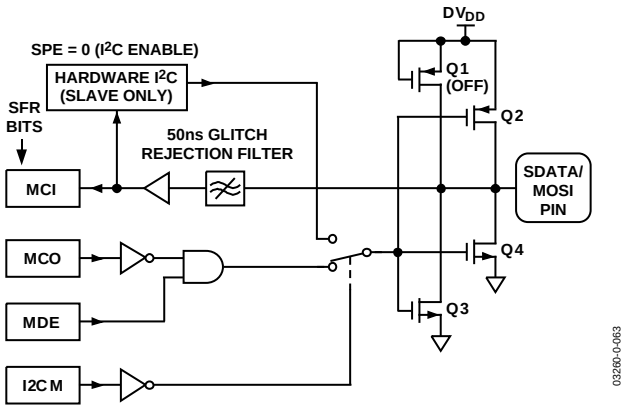

Figure 65. SDATA/MOSI Pin I/O Functional Equivalent in I²C Mode

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 28. Read-Write-Modify Instructions

Instruction	Description
ANL	Logical AND, for example, ANL P1, A
ORL	(Logical OR, for example, ORL P2, A
XRL	(Logical EX-OR, for example, XRL P3, A
JBC	Jump if Bit = 1 and clear bit, for example, JBC P1.1, LABEL
CPL	Complement bit, for example, CPL P3.0
INC	Increment, for example, INC P2
DEC	Decrement, for example, DEC P2
DJNZ	Decrement and Jump if Not Zero, for example, 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.

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.

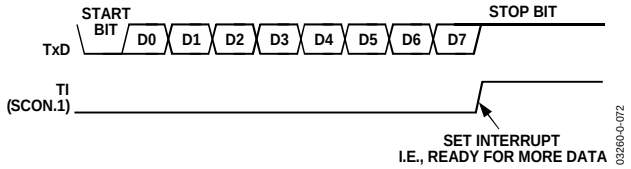

Figure 72. UART Serial Port Transmission, Mode 1

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.

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 free runs, 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 free runs; 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 does not operate if no crystal is present.

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

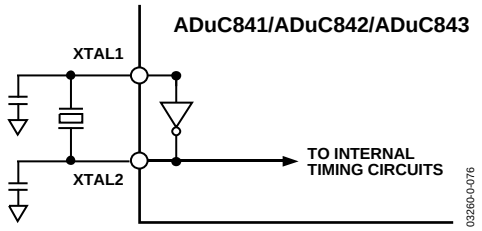

Figure 75. External Parallel Resonant Crystal Connections

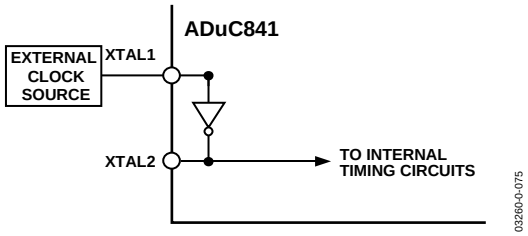

Figure 76. Connecting an External Clock Source (ADuC841)

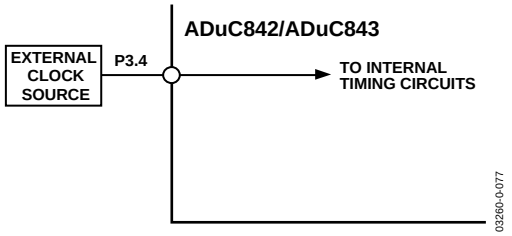

Figure 77. Connecting an External Clock Source (ADuC842/ADuC843)

Whether using the internal PLL or an external clock source, the specified operational clock speed range of the devices 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.

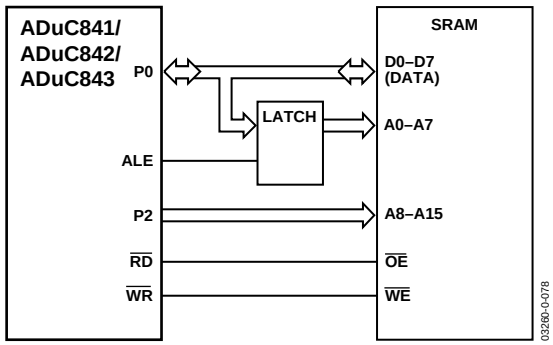

Figure 78. External Data Memory Interface (64 kBytes Address Space)

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.

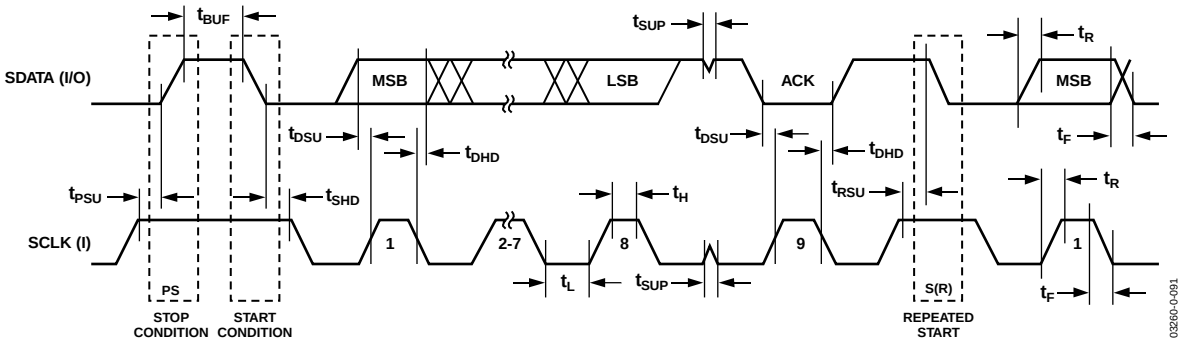

Figure 90. I²C Compatible Interface Timing

Parameter					
SPI MASTER MODE TIMING (CPHA = 0)		**Min**	**Typ**	**Max**	**Unit**
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, that is, core clock frequency = 2.09 MHz.
b. SPI bit-rate selection bits SPR1 and SPR0 in SPICON SFR set to 0 and 0, respectively.

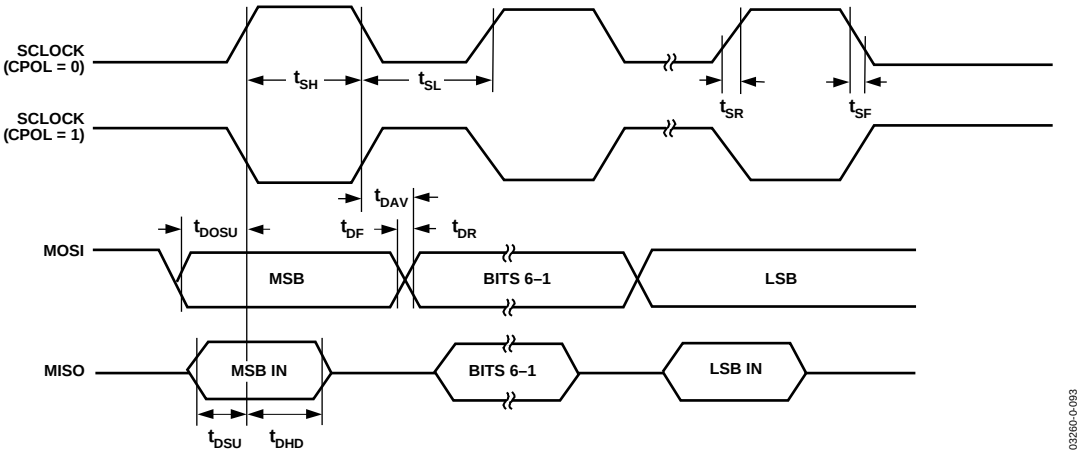

Figure 92. SPI Master Mode Timing (CPHA = 0)