Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8052
Core Size	8-Bit
Speed	8.38MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, PSM, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	62KB (62K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	52-QFP
Supplier Device Package	80-PQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc843bsz62-3

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

Pin 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.

Pin No.	Mnemonic	Type[1]	Description
46	P0.0/A0	I/O	Input/Output Port 0 (P0.0). 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. External Memory Address (A0). 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.
47	P0.1/A1	I/O	Input/Output Port 0 (P0.1). 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. External Memory Address (A1). 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-up resistors when emitting 1s.
48	P0.2/A2	I/O	Input/Output Port 0 (P0.2). 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. External Memory Address (A2). 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-up resistors when emitting 1s.
49	P0.3/A3	I/O	Input/Output Port 0 (P0.3). 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. External Memory Address (A3). 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-up resistors when emitting 1s.
52	P0.4/A4	I/O	Input/Output Port 0 (P0.4). 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. External Memory Address (A4). 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-up resistors when emitting 1s.
53	P0.5/A5	I/O	Input/Output Port 0 (P0.5). 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. External Memory Address (A5). 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-up resistors when emitting 1s.
54	P0.6/A6	I/O	Input/Output Port 0 (P0.6). 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. External Memory Address (A6). 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-up resistors when emitting 1s.
55	P0.7/A7	I/O	Input/Output Port 0 (P0.7). 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. External Memory Address (A7). 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-up resistors when emitting 1s.
56	P1.0/ADC0/T2	I	Input Port 1 (P1.0). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit. Single-Ended Analog Input (ADC0). Channel selection is via ADCCON2 SFR. Timer 2 Digital Input (T2). Input to Timer/Counter 2. When enabled, Counter 2 is incremented in response to a 1-to-0 transition of the T2 input.
	EPAD		Exposed Pad. The LFCSP has an exposed pad that must be soldered to the metal plate on the printed circuit board (PCB) for mechanical reasons and to DGND.

[1] P = power, G = ground, I = input, O = output, and NC = no connect.

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.

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, that is, 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.

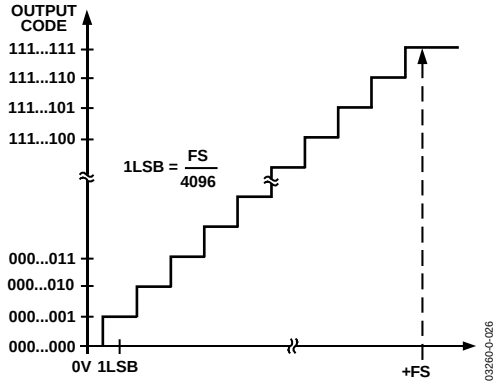

Figure 28. ADC Transfer Function

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.

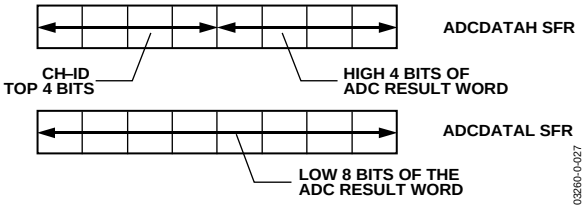

Figure 29. ADC Result Word Format

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.

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.

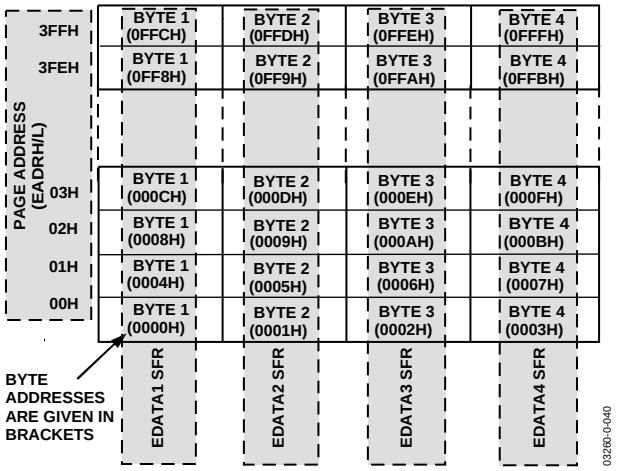

Figure 41. Flash/EE Data Memory Control and Configuration

Table 13. 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.

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 16. 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.

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

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 P2.7 is typically 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.

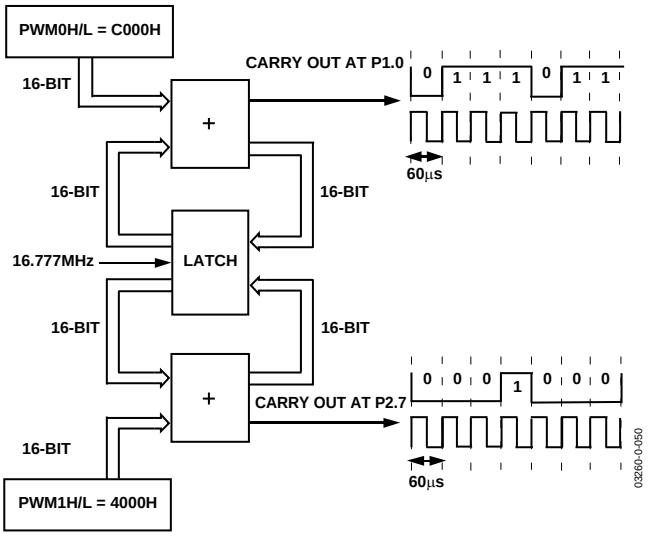

Figure 51. PWM Mode 4

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.

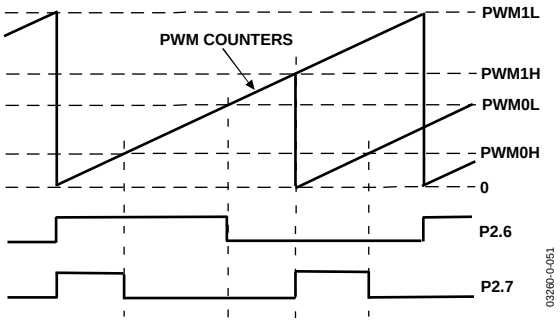

Figure 52. PWM Mode 5

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 do 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), P2.7 is typically 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 compensates 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.

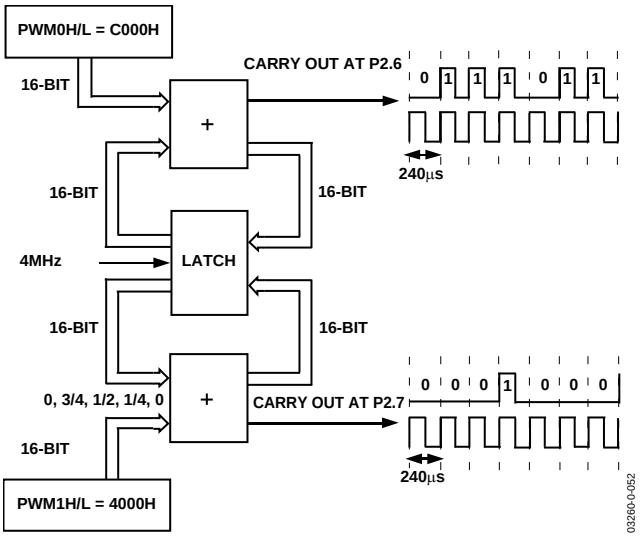

Figure 53. PWM Mode 6

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

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 32.

Table 32. 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 also triggers 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 does not occur. Therefore, Timer 2 interrupts does 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.

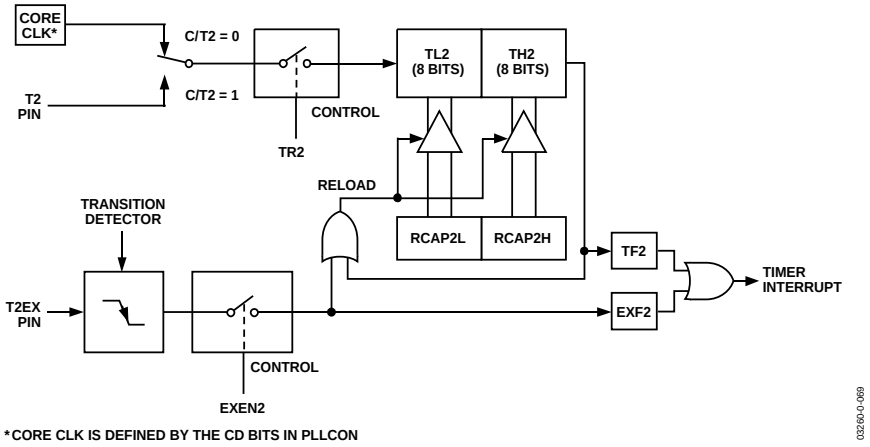

Figure 70. Timer/Counter 2, 16-Bit Autoreload Mode

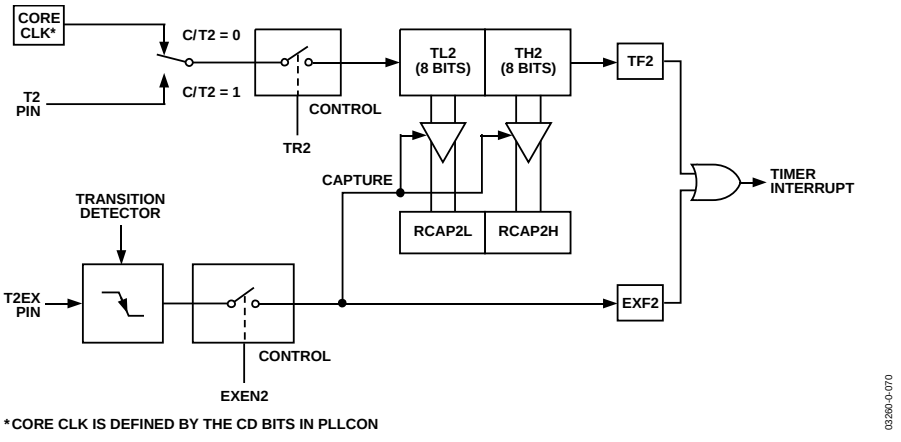

Figure 71. Timer/Counter 2, 16-Bit Capture Mode

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.

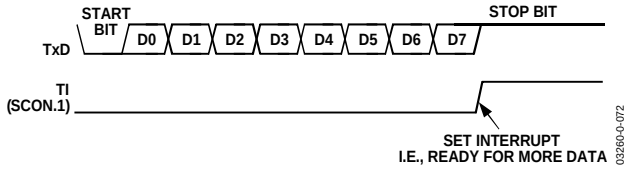

Figure 72. UART Serial Port Transmission, Mode 1

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*)])

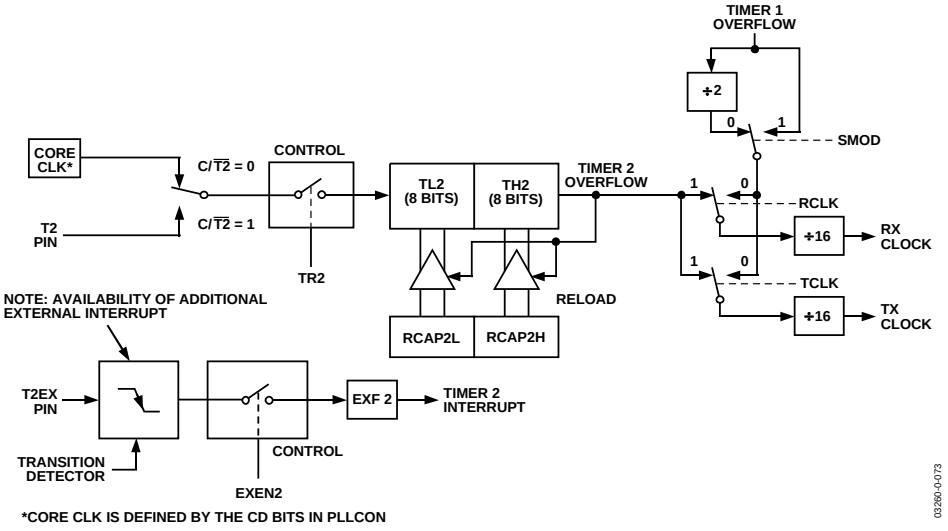

Figure 73. Timer 2, UART Baud Rates

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.

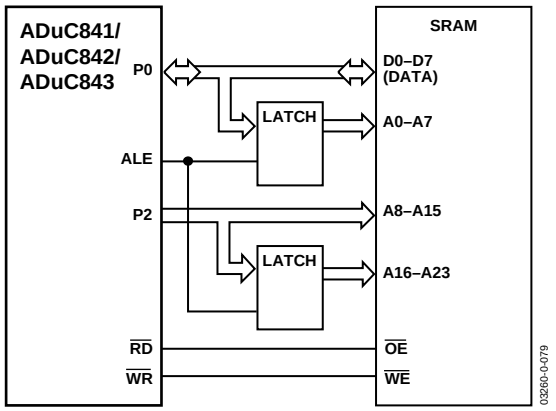

Figure 79. External Data Memory Interface (16 MBytes Address Space)

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.

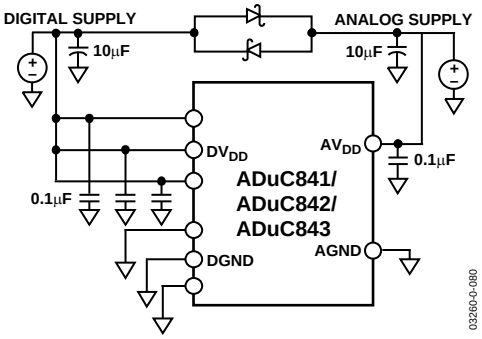

Figure 80. External Dual-Supply Connections

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 still needs 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.

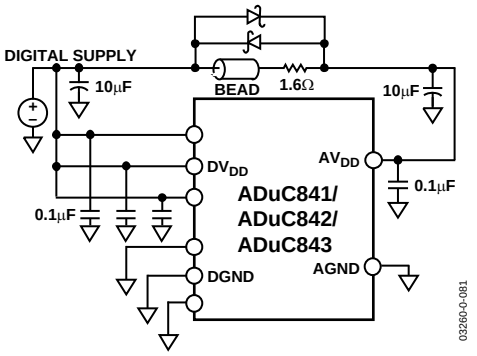

Figure 81. External Single-Supply Connections

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 41. 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 41. 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 41 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.

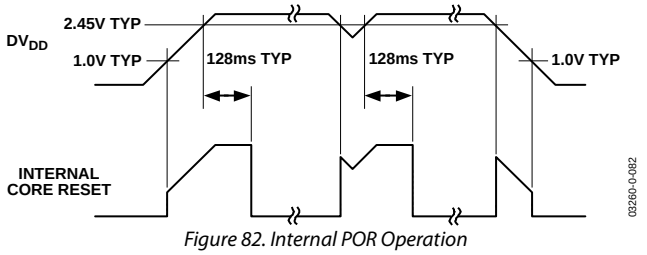

Figure 82. Internal POR Operation

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.

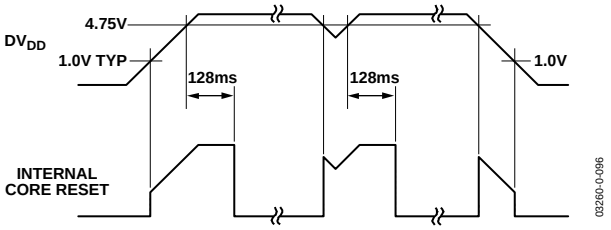

Figure 83. Internal POR Operation

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.

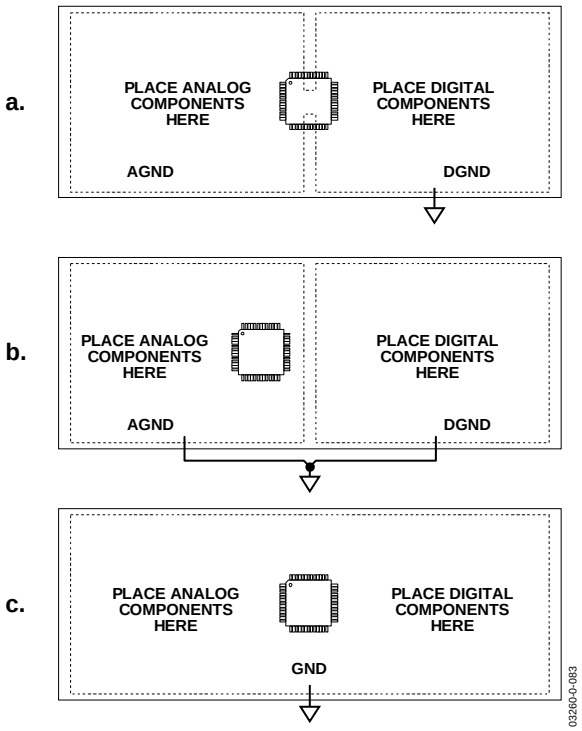

Figure 84. System Grounding Schemes

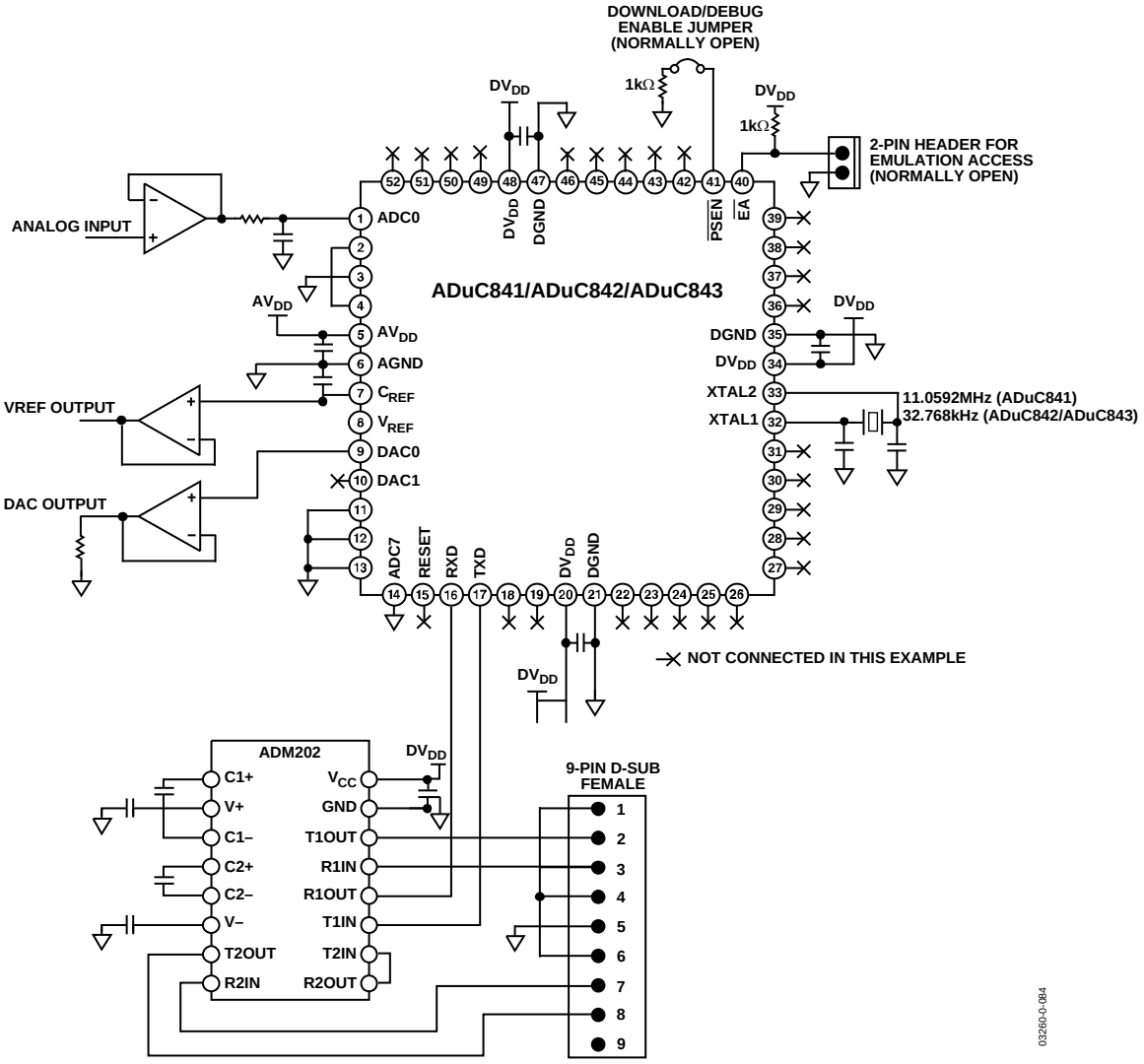

Figure 85. Example System (PQFP Package), DACs Not Present on ADuC843

OTHER HARDWARE CONSIDERATIONS

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

Embedded Serial Port Debugger

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

Single-Pin Emulation Mode

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

Typical System Configuration

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

DEVELOPMENT TOOLS

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

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

These systems are described briefly in the following sections.

QUICKSTART DEVELOPMENT SYSTEM

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

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

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

Download—In-Circuit Serial Downloader

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

ASPIRE—IDE

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

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

QuickStart Plus Development System

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

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

TIMING SPECIFICATIONS[1, 2, 3]

Table 42. 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

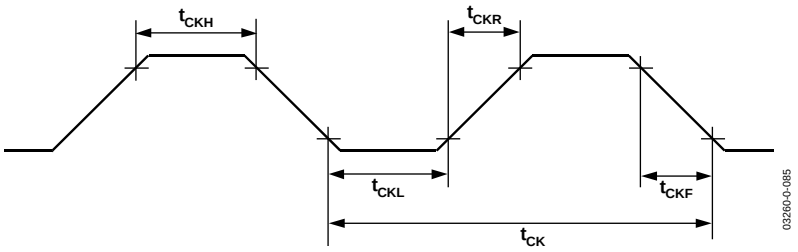

Figure 86. XTAL1 Input

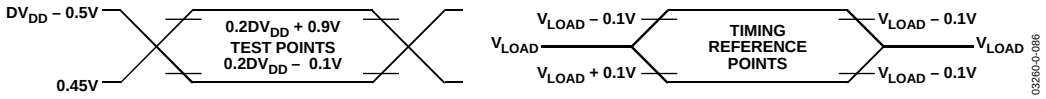

Figure 87. Timing Waveform Characteristics