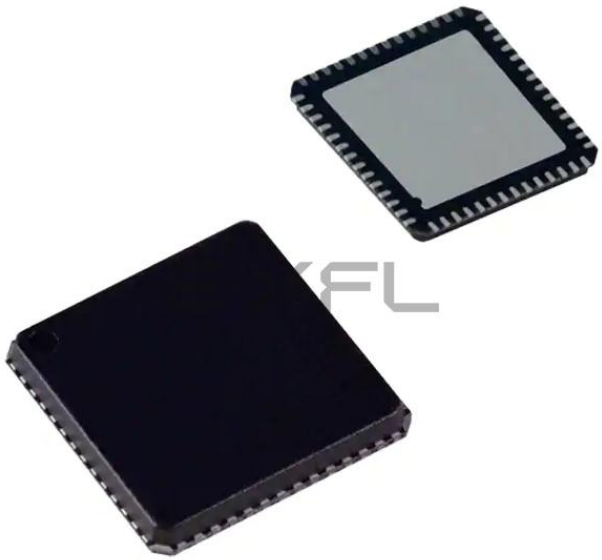

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8052
Core Size	8-Bit
Speed	20MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, PSM, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	62KB (62K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	A/D 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	56-VFQFN Exposed Pad, CSP
Supplier Device Package	56-LFCSP-VQ (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc841bcpz62-5

Pin No.	Mnemonic	Type[1]	Description
11	P1.4/ADC4		Input Port 1 (P1.4). 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.
12	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 ($\overline{SS}$).
13	P1.6/ADC6	I	Input Port 1 (P1.6). 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.
14	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.
15	RESET	I	Reset. Digital Input. A high level on this pin for 24 master clock cycles while the oscillator is running resets the device.
16	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).
17	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).
18	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.
19	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.
20, 34, 48	DV_{DD}	P	Digital Positive Supply Voltage. 3 V or 5 V nominal.
21, 35, 47	DGND	G	Digital Ground. DGND is the ground reference point for the digital circuitry.

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

TYPICAL PERFORMANCE CHARACTERISTICS

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

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

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

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

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

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

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

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

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

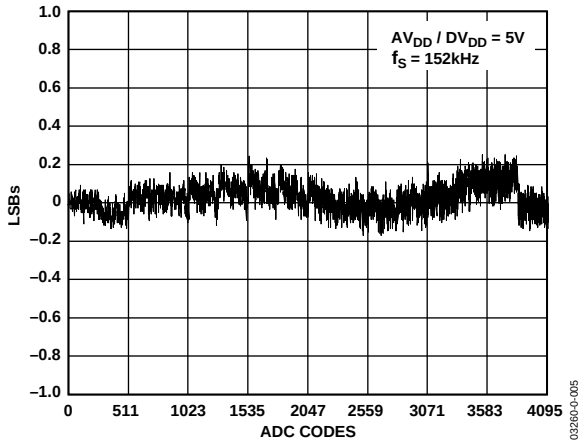

Figure 5. Typical INL Error, V_{DD} = 5 V, f_s = 152 kHz

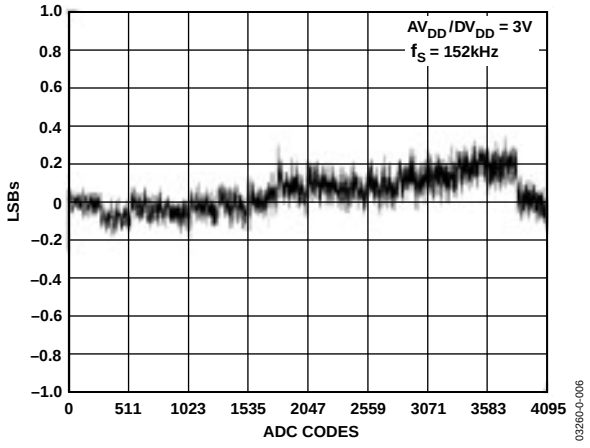

Figure 6. Typical INL Error, V_{DD} = 3 V, f_s = 152 kHz

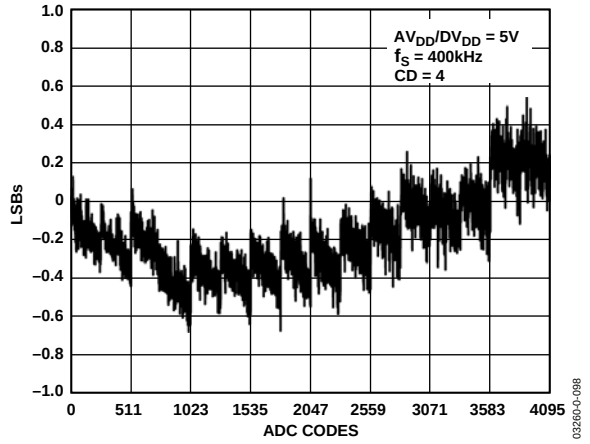

Figure 7. Typical INL Error, V_{DD} = 5 V, f_S = 400 kHz

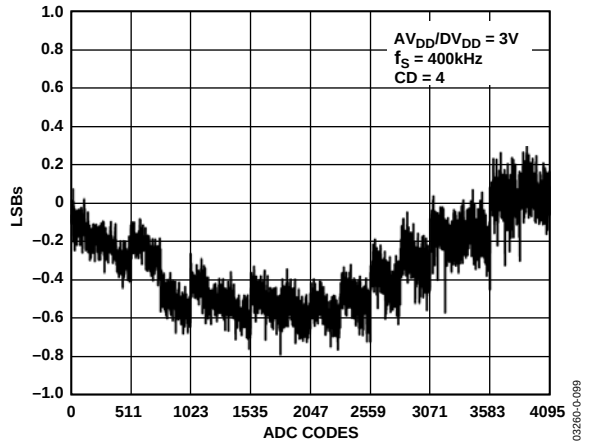

Figure 8. Typical INL Error, V_{DD} = 3 V, f_S = 400 kHz

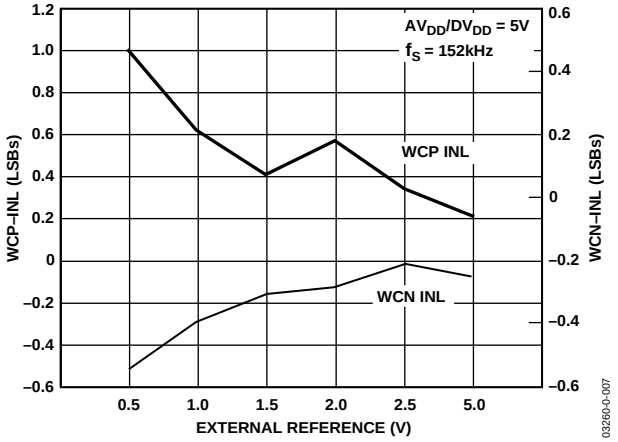

Figure 9. Typical Worst-Case INL Error vs. V_{REF}, V_{DD} = 5 V

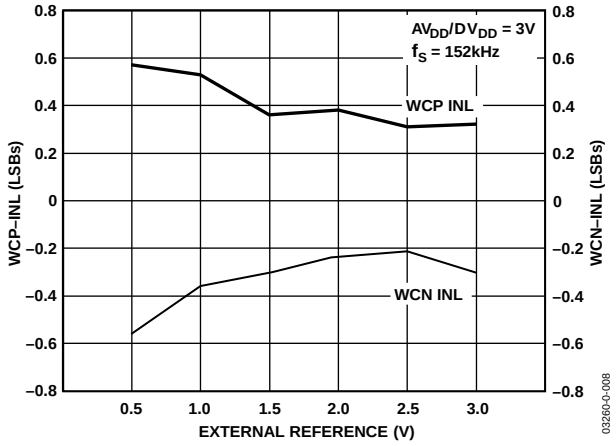

Figure 10. Typical Worst-Case INL Error vs. V_{REF}, V_{DD} = 3 V

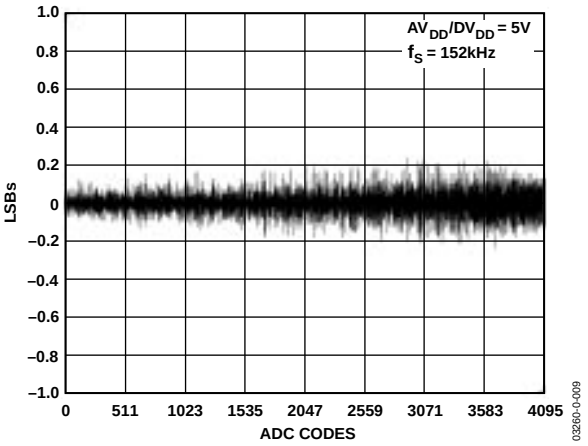

Figure 11. Typical DNL Error, V_{DD} = 5 V

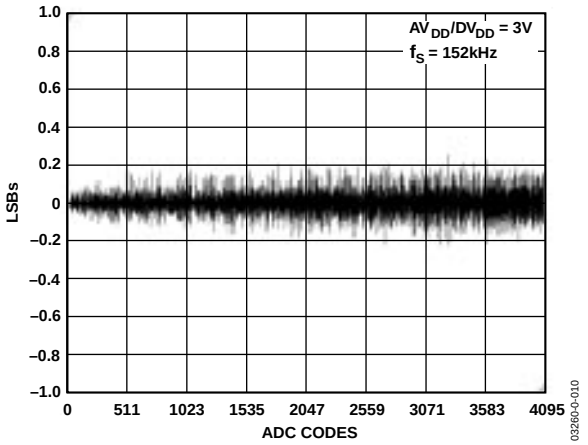

Figure 12. Typical DNL Error, V_{DD} = 3 V

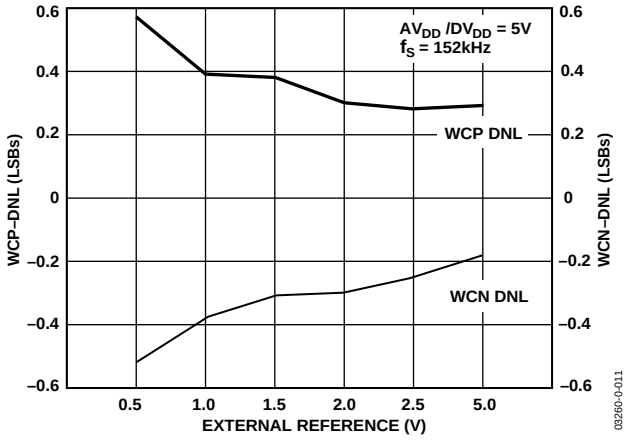

Figure 13. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 5 V

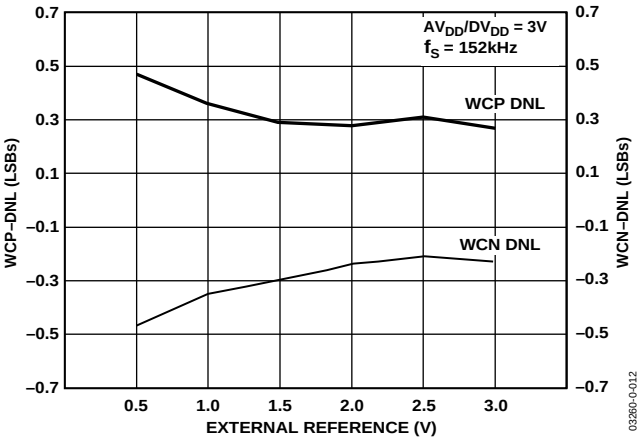

Figure 14. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 3 V

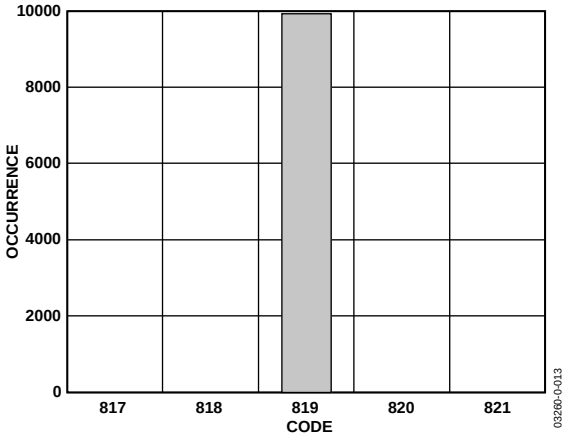

Figure 15. Code Histogram Plot, V_{DD} = 5 V

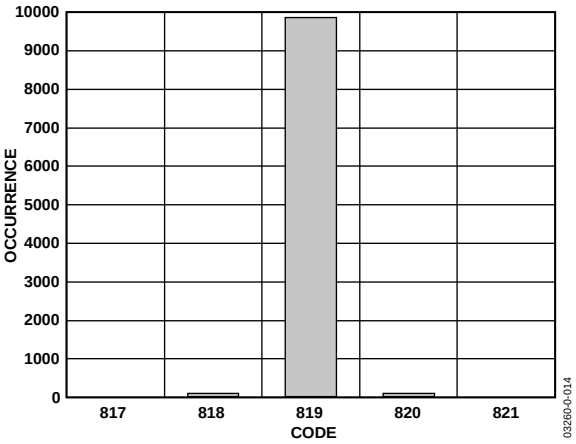

Figure 16. Code Histogram Plot, V_{DD} = 3 V

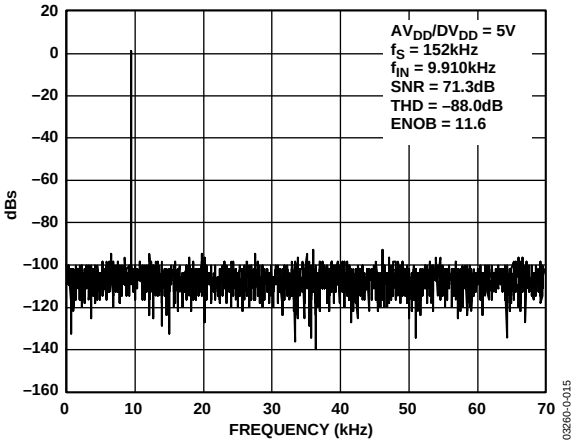

Figure 17. Dynamic Performance at V_{DD} = 5 V

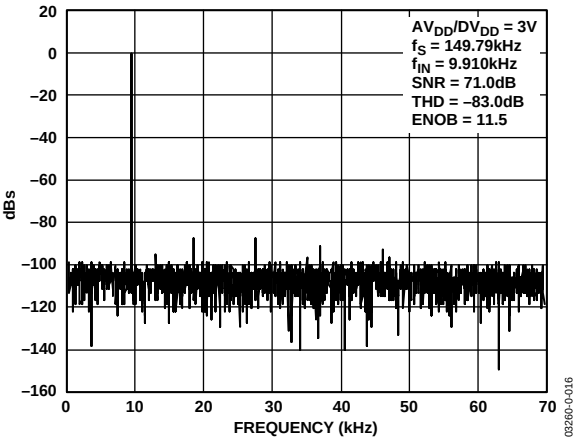

Figure 18. Dynamic Performance at V_{DD} = 3 V

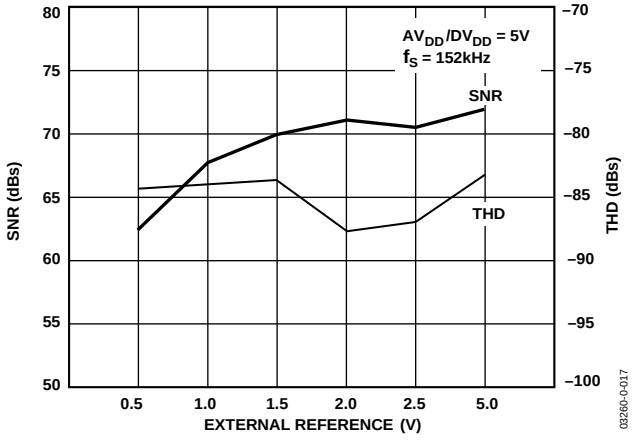

Figure 19. Typical Dynamic Performance vs. V_{REF}, V_{DD} = 5 V

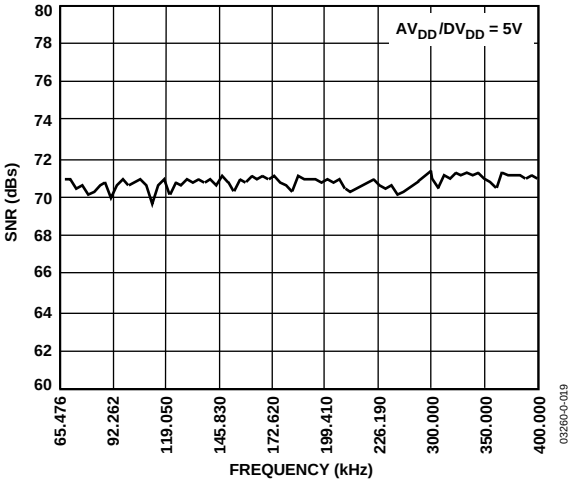

Figure 21. Typical Dynamic Performance vs. Sampling Frequency

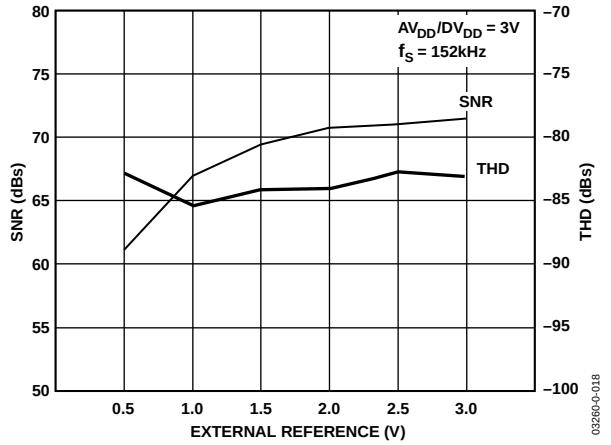

Figure 20. Typical Dynamic Performance vs. V_{REF}, V_{DD} = 3 V

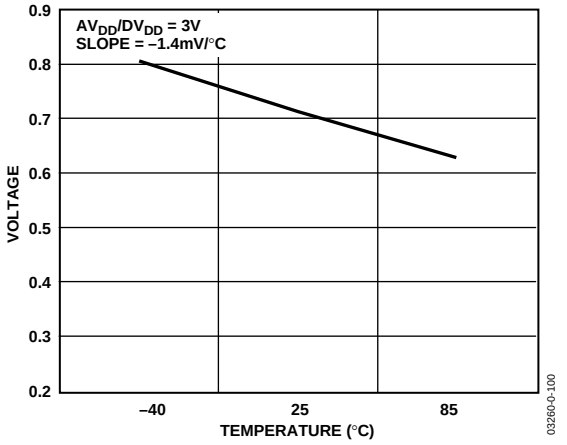

Figure 22. Typical Temperature Sensor Output vs. Temperature

GENERAL DESCRIPTION (*continued*)

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

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

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

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

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

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.

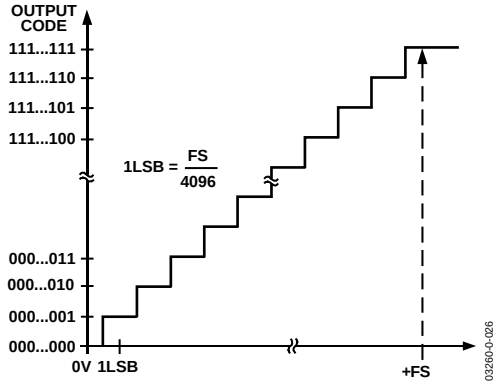

Figure 28. ADC Transfer Function

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.

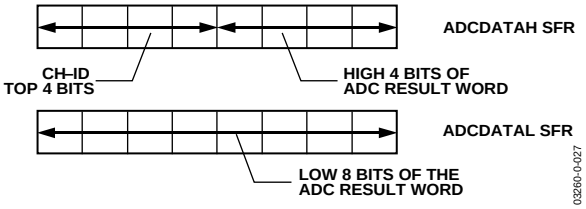

Figure 29. ADC Result Word Format

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.

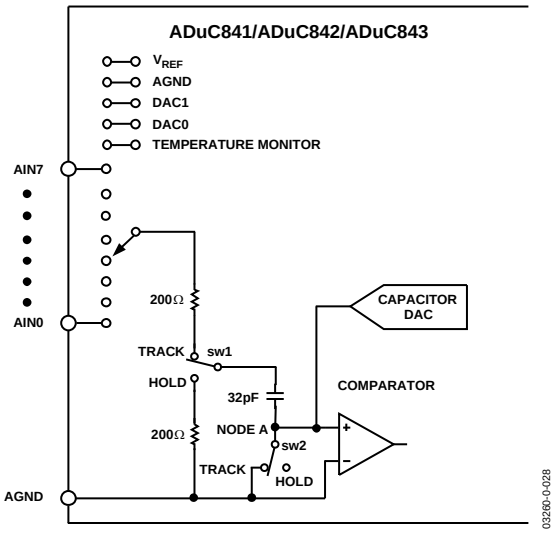

Figure 30. Internal ADC Structure

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.

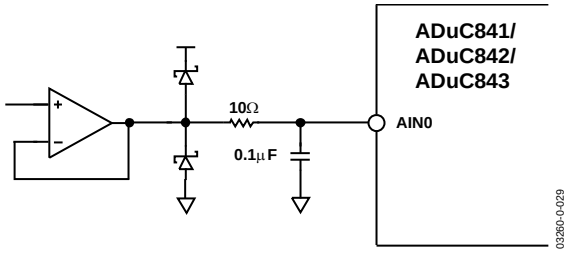

Figure 31. Buffering Analog Inputs

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.

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

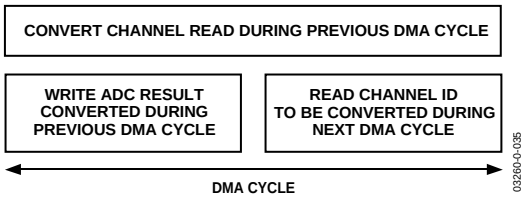

Figure 36. DMA Cycle

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

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

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

Micro Operation during ADC DMA Mode

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

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

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

ADC Offset and Gain Calibration Coefficients

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

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

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

CALIBRATING THE ADC

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

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

SERIAL PERIPHERAL INTERFACE (SPI)

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

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

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

MOSI (Master Out, Slave In Pin)

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

SCLOCK (Serial Clock I/O Pin)

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

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

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

Using the SPI Interface

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

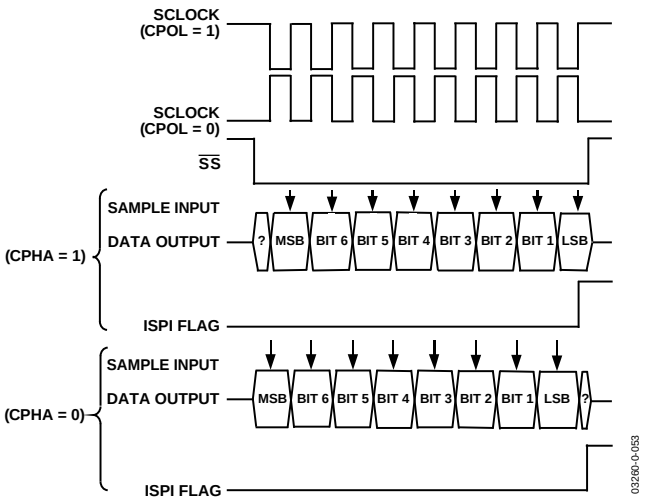

Figure 54. SPI Timing, All Modes

SPI Interface—Master Mode

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

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

SPI Interface—Slave Mode

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

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

Timers/Counters

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

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

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

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

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

Table 29. TMOD SFR Bit Designations

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

TIMER/COUNTER 0 AND 1 OPERATING MODES

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

Mode 0 (13-Bit Timer/Counter)

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

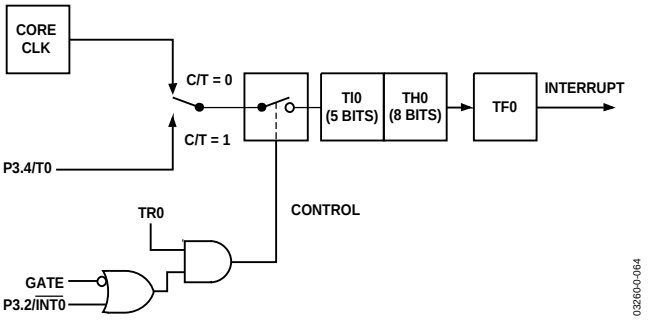

Figure 66. Timer/Counter 0, Mode 0

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

Mode 1 (16-Bit Timer/Counter)

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

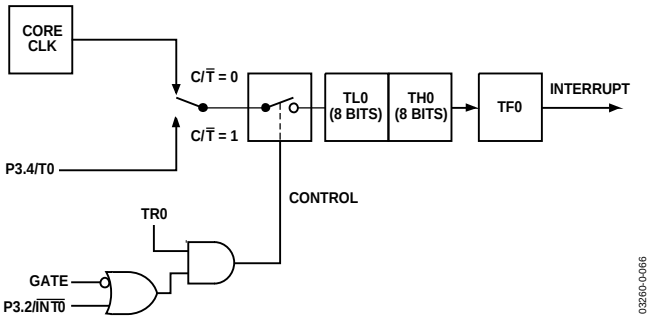

Figure 67. Timer/Counter 0, Mode 1

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

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

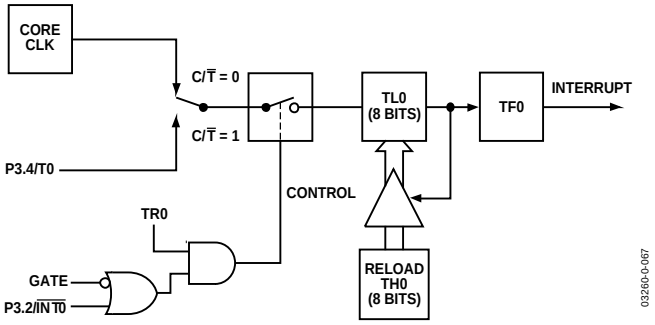

Figure 68. Timer/Counter 0, Mode 2

Mode 3 (Two 8-Bit Timer/Counters)

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

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

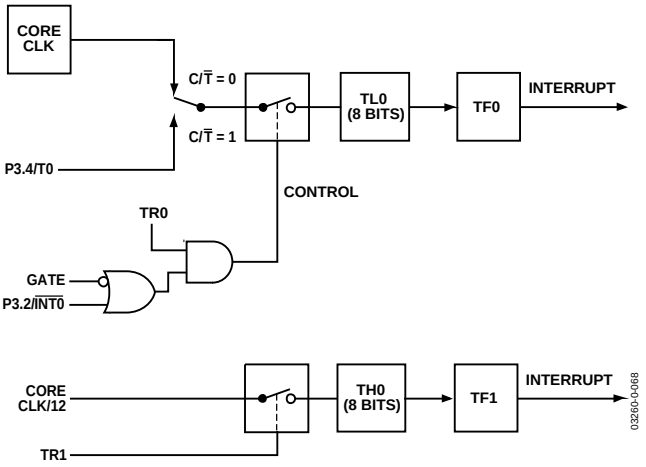

Figure 69. Timer/Counter 0, Mode 3

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

Table 31. T2CON SFR Bit Designations

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

Timer/Counter 2 Data Registers

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

TH2 and TL2

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

RCAP2H and RCAP2L

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

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*)])

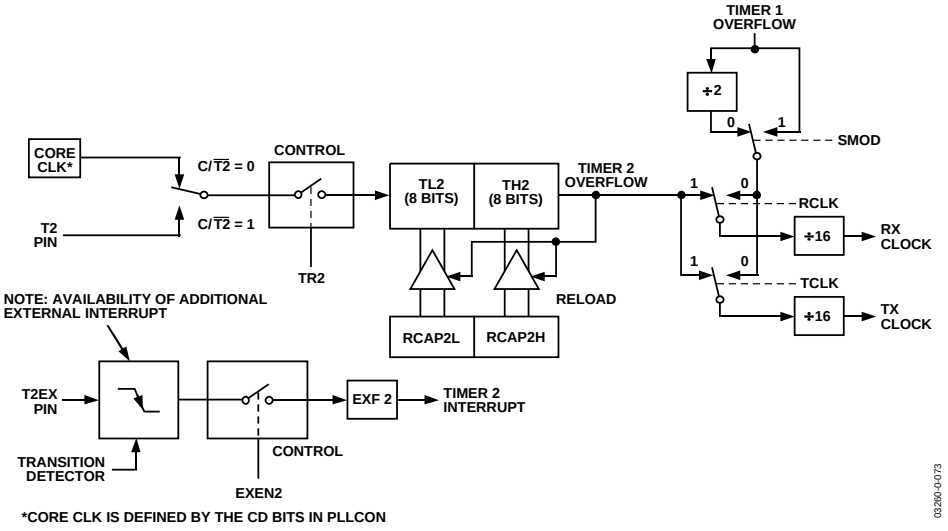

Figure 73. Timer 2, UART Baud Rates

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.

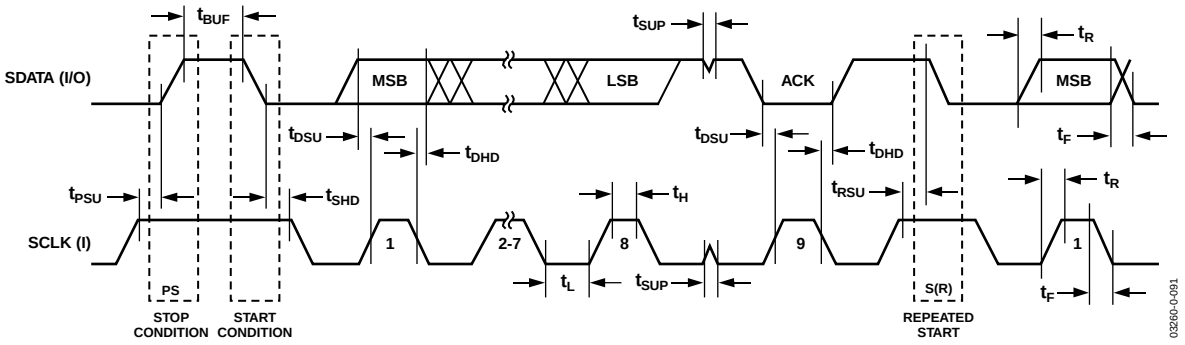

Figure 90. I²C Compatible Interface Timing