MicroConverter®, 12-Bit ADCs and DACs with Embedded 62 kBytes Flash MCU

ADuC832

FEATURES
ANALOG I/O
- **8-Channel, 247 kSPS 12-Bit ADC**
  - **DC Performance: ±1 LSB INL**
  - **AC Performance: 71 dB SNR**
- **DMA Controller for High Speed ADC-to-RAM Capture**
- **2 12-Bit (Monotonic) Voltage Output DACs**
- **Dual Output PWM/Σ-Δ DACs**
- **On-Chip Temperature Sensor Function ±3°C**
- **On-Chip Voltage Reference**

Memory
- **62 kBytes On-Chip Flash/EE Program Memory**
- **4 kBytes On-Chip Flash/EE Data Memory**
- **Flash/EE, 100 Yr Retention, 100 kCycles Endurance**
- **2304 Bytes On-Chip Data RAM**

8051-Based Core
- **8051 Compatible Instruction Set (16 MHz Max)**
- **32 kHz Ext Crystal, On-Chip Programmable PLL**
- **12 Interrupt Sources, 2 Priority Levels**
- **Dual Data Pointer**
- **Extended 11-Bit Stack Pointer**

On-Chip Peripherals
- **Time Interval Counter (TIC)**
- **UART, I²C®, and SPI® Serial I/O**
- **Watchdog Timer (WDT), Power Supply Monitor (PSM)**

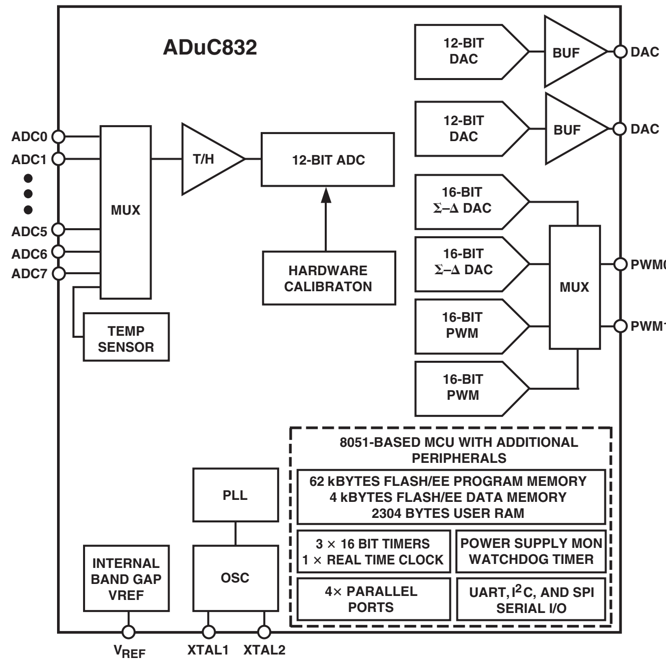
Power
- **Specified for 3 V and 5 V Operation**
- **Normal, Idle, and Power-Down Modes**
- **Power-Down: 25 μA @ 3 V with Wake-Up cct Running**

APPLICATIONS
Optical Networking—Laser Power Control
Base Station Systems
Precision Instrumentation, Smart Sensors
Transient Capture Systems
DAS and Communications Systems

Upgrade to ADuC812 Systems. Runs from 32 kHz External Crystal with On-Chip PLL.

Also Available: ADuC831 Pin Compatible Upgrade to Existing ADuC812 Systems that Require Additional Code or Data Memory. Runs from 1 MHz–16 MHz External Crystal.

FUNCTIONAL BLOCK DIAGRAM



GENERAL DESCRIPTION

The ADuC832 is a complete smart transducer front end, integrating a high performance self-calibrating multichannel 12-bit ADC, dual 12-bit DACs, and programmable 8-bit MCU on a single chip.

The device operates from a 32 kHz crystal with an on-chip PLL generating a high frequency clock of 16.77 MHz. This clock is, in turn, routed through a programmable clock divider from which the MCU core clock operating frequency is generated. The microcontroller core is an 8052 and therefore 8051 instruction set compatible with 12 core clock periods per machine cycle. 62 kBytes of nonvolatile Flash/EE program memory are provided on-chip. 4 kBytes of nonvolatile Flash/EE data memory, 256 bytes RAM, and 2 kBytes of extended RAM are also integrated on-chip.

The ADuC832 also incorporates 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, dual output 16-bit PWM, watchdog timer, time interval counter, three timers/counters, Timer 3 for baud rate generation, and serial I/O ports (SPI, I²C, and UART)

On-chip factory firmware supports in-circuit serial download and debug modes (via UART) as well as single-pin emulation mode via the $\overline{\text{EA}}$ pin. The ADuC832 is supported by QuickStart™ and QuickStart Plus development systems featuring low cost software and hardware development tools. A functional block diagram of the ADuC832 is shown above with a more detailed block diagram shown in Figure 1.

The part is specified for 3 V and 5 V operation over the extended industrial temperature range and is available in a 52-lead plastic quad flatpack package and a 56-lead chip scale package.

MicroConverter is a registered trademark and QuickStart is a trademark of Analog Devices, Inc.
SPI is a registered trademark of Motorola, Inc.
I²C is a registered trademark of Philips Corporation.

REV. 0

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 **www.analog.com**
Fax: 781/326-8703

TABLE OF CONTENTS

SPECIFICATIONS[1] ($AV_{DD} = DV_{DD}$ = 2.7 V to 3.3 V or 4.5 V to 5.5 V; V_{REF} = 2.5 V Internal Reference, F_{CORE} =16.78 MHz; all specifications $T_A = T_{MIN}$ to T_{MAX}, unless otherwise noted.)

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
ADC CHANNEL SPECIFICATIONS				
DC ACCURACY[2, 3]				f_{SAMPLE} = 147 kHz, see Page 11 for Typical Performance at other f_{SAMPLE}
Resolution	12	12	Bits	
Integral Nonlinearity	±1	±1	LSB max	2.5 V Internal Reference
	±0.3	±0.3	LSB typ	
Differential Nonlinearity	±0.9	±0.9	LSB max	2.5 V Internal Reference
	±0.25	±0.25	LSB typ	
Integral Nonlinearity[4]	±1.5	±1.5	LSB max	1 V External Reference
Differential Nonlinearity[4]	+1.5/–0.9	+1.5/–0.9	LSB max	1 V External Reference
Code Distribution	1	1	LSB typ	ADC Input is a DC Voltage
CALIBRATED ENDPOINT ERRORS[5, 6]				
Offset Error	±4	±4	LSB max	
Offset Error Match	±1	±1	LSB typ	
Gain Error	±2	±3	LSB max	
Gain Error Match	–85	–85	dB typ	
DYNAMIC PERFORMANCE				f_{IN} = 10 kHz Sine Wave f_{SAMPLE} = 147 kHz
Signal-to-Noise Ratio (SNR)[7]	71	71	dB typ	
Total Harmonic Distortion (THD)	–85	–85	dB typ	
Peak Harmonic or Spurious Noise	–85	–85	dB typ	
Channel-to-Channel Crosstalk[8]	–80	–80	dB typ	
ANALOG INPUT				
Input Voltage Ranges	0 to V_{REF}	0 to V_{REF}	V	
Leakage Current	±1	±1	µA max	
Input Capacitance	32	32	pF typ	
TEMPERATURE SENSOR[9]				
Voltage Output at 25°C	650	650	mV typ	
Voltage TC	–2.0	–2.0	mV/°C typ	
Accuracy	±3	±3	°C typ	Internal 2.5 V V_{REF}
	±1.5	±1.5	°C typ	External 2.5 V V_{REF}
DAC CHANNEL SPECIFICATIONS				DAC Load to AGND
Internal Buffer Enabled				R_L = 10 kΩ, C_L = 100 pF
DC ACCURACY[10]				
Resolution	12	12	Bits	
Relative Accuracy	±3	±3	LSB typ	
Differential Nonlinearity[11]	–1	–1	LSB max	Guaranteed 12-Bit Monotonic
	±1/2	±1/2	LSB typ	
Offset Error	±50	±50	mV max	V_{REF} Range
Gain Error	±1	±1	% max	AV_{DD} Range
	±1	±1	% typ	V_{REF} Range
Gain Error Mismatch	0.5	0.5	% typ	% of Full-Scale on DAC1
ANALOG OUTPUTS				
Voltage Range_0	0 to V_{REF}	0 to V_{REF}	V typ	DAC V_{REF} = 2.5 V
Voltage Range_1	0 to V_{DD}	0 to V_{DD}	V typ	DAC V_{REF} = V_{DD}
Output Impedance	0.5	0.5	Ω typ	
DAC AC CHARACTERISTICS				
Voltage Output Settling Time	15	15	µs typ	Full-Scale Settling Time to within 1/2 LSB of Final Value
Digital-to-Analog Glitch Energy	10	10	nV sec typ	1 LSB Change at Major Carry

SPECIFICATIONS (continued)

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
DAC CHANNEL SPECIFICATIONS[12, 13]				
Internal Buffer Disabled				
DC ACCURACY[10]				
Resolution	12	12	Bits	
Relative Accuracy	±3	±3	LSB typ	
Differential Nonlinearity[11]	–1	–1	LSB max	Guaranteed 12-Bit Monotonic
	±1/2	±1/2	LSB typ	
Offset Error	±5	±5	mV max	V_{REF} Range
Gain Error	–0.3	–0.3	% typ	V_{REF} Range
Gain Error Mismatch[4]	0.5	0.5	% max	% of Full-Scale on DAC1
ANALOG OUTPUTS				
Voltage Range_0	0 to V_{REF}	0 to V_{REF}	V typ	DAC V_{REF} = 2.5 V
REFERENCE INPUT/OUTPUT				
REFERENCE OUTPUT[14]				
Output Voltage (V_{REF})	2.5	2.5	V	
Accuracy	±2.5	±2.5	% max	Of V_{REF} Measured at the C_{REF} Pin
Power Supply Rejection	47	57	dB typ	
Reference Temperature Coefficient	±100	±100	ppm/°C typ	
Internal V_{REF} Power-On Time	80	80	ms typ	
EXTERNAL REFERENCE INPUT[15]				V_{REF} and C_{REF} Pins Shorted
Voltage Range (V_{REF})[4]	0.1	0.1	V min	
	V_{DD}	V_{DD}	V max	
Input Impedance	20	20	kΩ typ	
Input Leakage	1	1	µA max	Internal Band Gap Deselected via ADCCON1.6
POWER SUPPLY MONITOR (PSM)				
DV_{DD} Trip Point Selection Range	2.63		V min	Four Trip Points Selectable in
	4.37		V max	This Range Programmed via TPD1–0 in PSMCON
DV_{DD} Power Supply Trip Point Accuracy	±3.5		% max	
WATCHDOG TIMER (WDT)[4]				
Timeout Period	0	0	ms min	Nine Timeout Periods
	2000	2000	ms max	Selectable in this Range
FLASH/EE MEMORY RELIABILITY CHARACTERISTICS[16]				
Endurance[17]	100,000	100,000	Cycles min	
Data Retention[18]	100	100	Years min	
DIGITAL INPUTS				
Input High Voltage (V_{INH})[4]	2.4	2	V min	
Input Low Voltage (V_{INL})[4]	0.8	0.4	V max	
Input Leakage Current (Port 0, $\overline{EA}$)	±10	±10	µA max	V_{IN} = 0 V or V_{DD}
	±1	±1	µA typ	V_{IN} = 0 V or V_{DD}
Logic 1 Input Current (All Digital Inputs)	±10	±10	µA max	V_{IN} = V_{DD}
	±1	±1	µA typ	V_{IN} = V_{DD}
Logic 0 Input Current (Port 1, 2, 3)	–75	–25	µA max	
	–40	–15	µA typ	V_{IL} = 450 mV
Logic 1–0 Transition Current (Port 2, 3)	–660	–250	µA max	V_{IL} = 2 V
	–400	–140	µA typ	V_{IL} = 2 V

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
SCLOCK and RESET Only[4]				
(Schmitt-Triggered Inputs)				
V_{T+}	1.3	0.95	V min	
	3.0	2.5	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	16.78	MHz max	Programmable via PLLCON
DIGITAL OUTPUTS				
Output High Voltage (V_{OH})	2.4		V min	V_{DD} = 4.5 V to 5.5 V
	4.0		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	
Floating State Output Capacitance	10	10	pF typ	
START UP TIME				At any Core CLK
At Power-On	500	500	ms typ	
From Idle Mode	100	100	µs typ	
From Power-Down Mode				
Wakeup with $\overline{INT0}$ Interrupt	150	400	µs typ	
Wakeup with SPI/I^2C Interrupt	150	400	µs typ	
Wakeup 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

SPECIFICATIONS (continued)

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
POWER REQUIREMENTS [19, 20]				
Power Supply Voltages				
AV_{DD}/DV_{DD} – AGND		2.7	V min	AV_{DD}/DV_{DD} = 3 V nom
		3.3	V max	
	4.5		V min	AV_{DD}/DV_{DD} = 5 V nom
	5.5		V max	
Power Supply Currents Normal Mode				
DV_{DD} Current[4]	6	3	mA max	Core CLK = 2.097 MHz
AV_{DD} Current	1.7	1.7	mA max	Core CLK = 2.097 MHz
DV_{DD} Current	23	12	mA max	Core CLK = 16.78 MHz
	20	10	mA typ	Core CLK = 16.78 MHz
AV_{DD} Current	1.7	1.7	mA max	Core CLK = 16.78 MHz
Power Supply Currents Idle Mode				
DV_{DD} Current	4	2	mA typ	Core CLK = 2.097 MHz
AV_{DD} Current	0.14	0.14	mA typ	Core CLK = 2.097 MHz
DV_{DD} Current[4]	10	5	mA max	Core CLK = 16.78 MHz
	9	4	mA typ	Core CLK = 16.78 MHz
AV_{DD} Current	0.14	0.14	mA typ	Core CLK = 16.78 MHz
Power Supply Currents Power-Down Mode				Core CLK = 2.097 MHz or 16.78 MHz
DV_{DD} Current[4]	80	25	μA max	Osc. On
	38	14	μA typ	
AV_{DD} Current	2	1	μA typ	
DV_{DD} Current	35	20	μA max	Osc. Off
	25	12	μA typ	
Typical Additional Power Supply Currents				AV_{DD} = DV_{DD} = 5 V
PSM Peripheral	50		μA typ	
ADC	1.5		mA typ	
DAC	150		μA typ	

NOTES

[1] Temperature Range –40°C to +125°C.

[2] ADC linearity is guaranteed during normal MicroConverter core operation.

[3] ADC LSB Size = $V_{REF}/2^{12}$ i.e., for Internal V_{REF} = 2.5 V, 1 LSB = 610 μV and for External V_{REF} = 1 V, 1 LSB = 244 μV.

[4] These numbers are not production tested but are guaranteed by design and/or characterization data on production release.

[5] Offset and Gain Error and Offset and Gain Error Match are measured after factory calibration.

[6] Based on external ADC system components, the user may need to execute a system calibration to remove additional external channel errors and achieve these specifications.

[7] SNR calculation includes distortion and noise components.

[8] Channel-to-channel crosstalk is measured on adjacent channels.

[9] The Temperature Monitor will give a measure of the die temperature directly; air temperature can be inferred from this result.

[10] DAC linearity is calculated using:
Reduced code range of 100 to 4095, 0 to V_{REF} range.
Reduced code range of 100 to 3945, 0 to V_{DD} range.
DAC Output Load = 10 kΩ and 100 pF.

[11] DAC differential nonlinearity specified on 0 to V_{REF} and 0 to V_{DD} ranges.

[12] DAC specification for output impedance in the unbuffered case depends on DAC code.

[13] DAC specifications for I_{SINK}, voltage output settling time and digital-to-analog glitch energy depend on external buffer implementation in unbuffered mode. DAC in unbuffered mode tested with OP270 external buffer, which has a low input leakage current.

[14] Measured with V_{REF} and C_{REF} pins decoupled with 0.1 μF capacitors to ground. Power-up time for the internal reference will be determined by the value of the decoupling capacitor chosen for both the V_{REF} and C_{REF} pins.

[15] When using an external reference device, the internal band gap reference input can be bypassed by setting the ADCCON1.6 bit. In this mode, the V_{REF} and C_{REF} pins need to be shorted together for correct operation.

[16] Flash/EE Memory reliability characteristics apply to both the Flash/EE program memory and the Flash/EE data memory.

[17] Endurance is qualified to 100,000 cycles as per JEDEC Std. 22 method A117 and measured at –40°C, +25°C, and +125°C. Typical endurance at 25°C is 700,000 cycles.

[18] Retention lifetime equivalent at junction temperature (T_J) = 55°C as per JEDEC Std. 22 method A117. Retention lifetime based on an activation energy of 0.6 eV will derate with junction temperature as shown in Figure 18 in the Flash/EE Memory description section.

[19] Power supply current consumption is measured in Normal, Idle, and Power-Down Modes under the following conditions:
Normal Mode: Reset = 0.4 V, Digital I/O pins = open circuit, Core Clk changed via CD bits in PLLCON, Core Executing internal software loop.
Idle Mode: Reset = 0.4 V, Digital I/O pins = open circuit, Core Clk changed via CD bits in PLLCON, PCON.0 = 1, Core Execution suspended in idle mode.
Power-Down Mode: Reset = 0.4 V, All Port 0 pins = 0.4 V, All other digital I/O and Port 1 pins are open circuit, Core Clk changed via CD bits in PLLCON, PCON.0 = 1, Core Execution suspended in power-down mode, OSC turned ON or OFF via OSC_PD bit (PLLCON.7) in PLLCON SFR

[20] DV_{DD} power supply current will increase typically by 3 mA (3 V operation) and 10 mA (5 V operation) during a Flash/EE memory program or erase cycle.

Specifications subject to change without notice.

ABSOLUTE MAXIMUM RATINGS*

(T_A = 25°C, unless otherwise noted.)

AV_{DD} to DV_{DD}	–0.3 V to +0.3 V
AGND to DGND	–0.3 V to +0.3 V
DV_{DD} to DGND, AV_{DD} to AGND	–0.3 V to +7 V
Digital Input Voltage to DGND	–0.3 V to DV_{DD} + 0.3 V
Digital Output Voltage to DGND	–0.3 V to DV_{DD} + 0.3 V
V_{REF} to AGND	–0.3 V to AV_{DD} + 0.3 V
Analog Inputs to AGND	–0.3 V to AV_{DD} + 0.3 V
Operating Temperature Range Industrial	
ADuC832BS	–40°C to +125°C
Operating Temperature Range Industrial	
ADuC832BCP	–40°C to +85°C
Storage Temperature Range	–65°C to +150°C
Junction Temperature	150°C
θ_{JA} Thermal Impedance (ADuC832BS)	90°C/W
θ_{JA} Thermal Impedance (ADuC832BCP)	52°C/W
Lead Temperature, Soldering	
Vapor Phase (60 sec)	215°C
Infrared (15 sec)	220°C

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

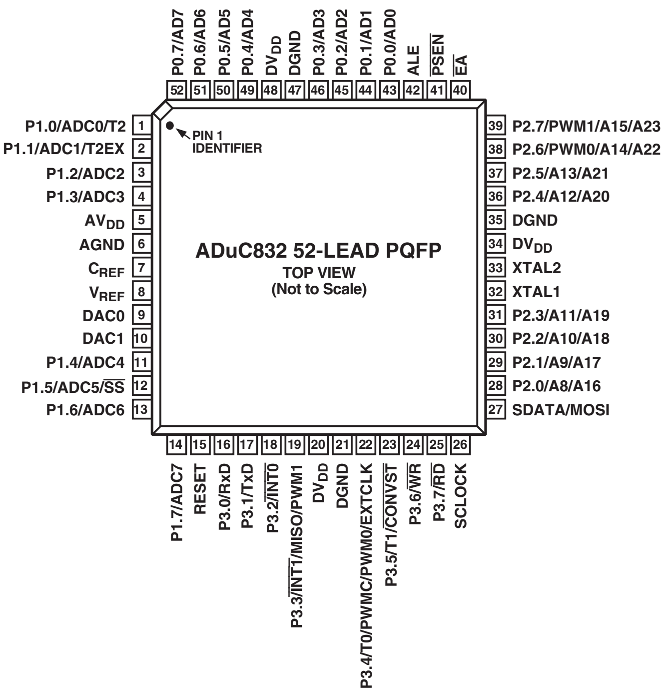
ORDERING GUIDE

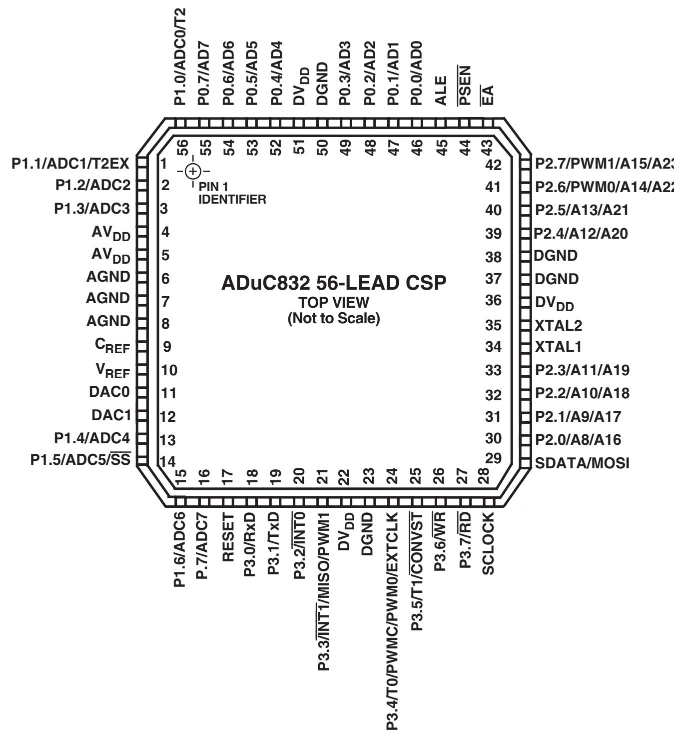
Model	Temperature Range	Package Description	Package Option
ADuC832BS	–40°C to +125°C	52-Lead Plastic Quad Flatpack	S-52
ADuC832BCP	–40°C to +85°C	56-Lead Chip Scale Package	CP-56
EVAL-ADuC832QS		QuickStart Development System	
EVAL-ADuC832QSP		QuickStart Plus Development System	

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADuC832 features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

PIN CONFIGURATION





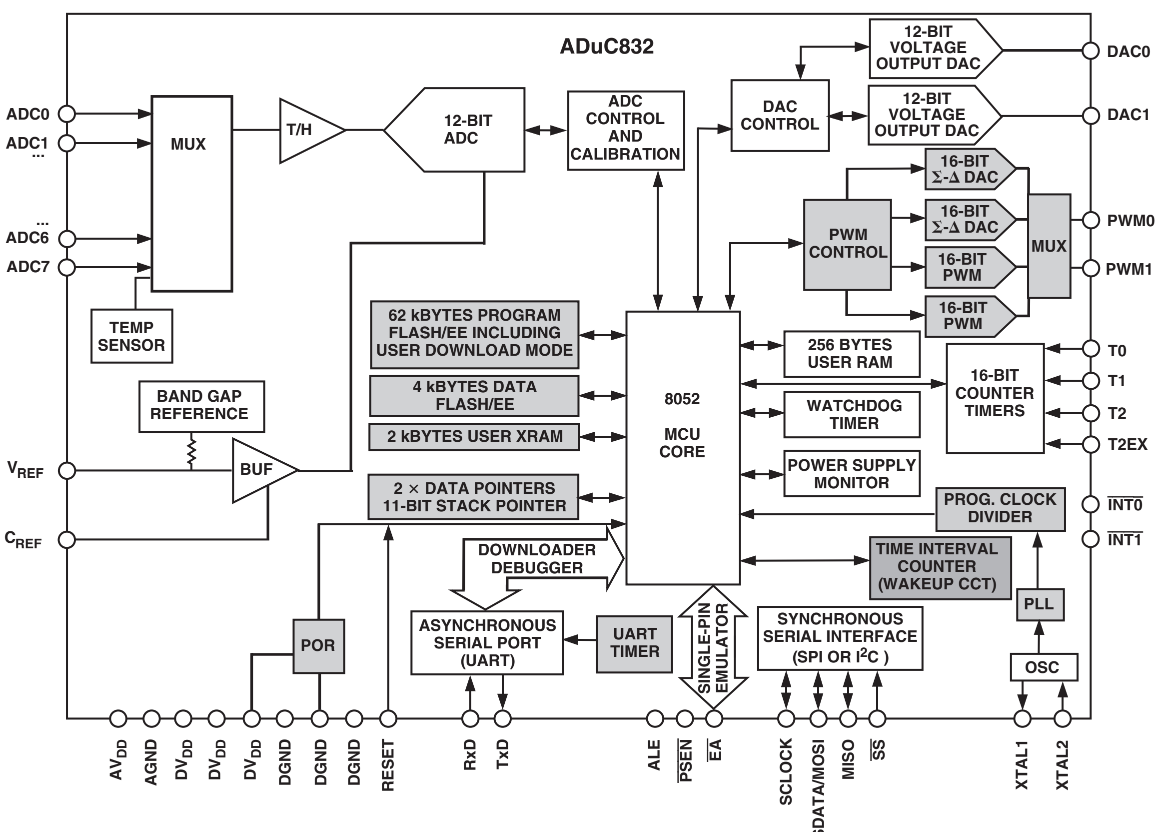


Figure 1. ADuC832 Block Diagram (Shaded Areas are Features Not Present on the ADuC812)

PIN FUNCTION DESCRIPTIONS

Mnemonic	Type	Function
DV_{DD}	P	Digital Positive Supply Voltage, 3 V or 5 V Nominal
AV_{DD}	P	Analog Positive Supply Voltage, 3 V or 5 V Nominal
C_{REF}	I/O	Decoupling Input for On-Chip Reference. Connect 0.1 µF between this pin and AGND.
V_{REF}	I/O	Reference Input/Output. This pin is connected to the internal reference through a series resistor and is the reference source for the analog-to-digital converter. The nominal internal reference voltage is 2.5 V, which appears at the pin. See ADC section on how to connect an external reference.
AGND	G	Analog Ground. Ground reference point for the analog circuitry.
P1.0–P1.7	I	Port 1 is an 8-bit input port only. Unlike other ports, Port 1 defaults to Analog Input mode. To configure any of these Port Pins as a digital input, write a "0" to the port bit. Port 1 pins are multifunction and share the following functionality.
ADC0–ADC7	I	Analog Inputs. Eight single-ended analog inputs. Channel selection is via ADCCON2 SFR.
T2	I	Timer 2 Digital Input. Input to Timer/Counter 2. When enabled, Counter 2 is incremented in response to a 1-to-0 transition of the T2 input.
T2EX	I	Digital Input. Capture/Reload trigger for Counter 2; also functions as an Up/Down control input for Counter 2.
$\overline{SS}$	I	Slave Select Input for the SPI Interface
SDATA	I/O	User Selectable, I^2C Compatible or SPI Data Input/Output Pin
SCLOCK	I/O	Serial Clock Pin for I^2C Compatible or SPI Serial Interface Clock
MOSI	I/O	SPI Master Output/Slave Input Data I/O Pin for SPI Interface
MISO	I/O	SPI Master Input/Slave Output Data I/O Pin for SPI Serial Interface
DAC0	O	Voltage Output from DAC0
DAC1	O	Voltage Output from DAC1
RESET	I	Digital Input. A high level on this pin for 24 master clock cycles while the oscillator is running resets the device.
P3.0–P3.7	I/O	Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low will source current because of the internal pull-up resistors. Port 3 pins also contain various secondary functions that are described below.
PWMC	I	PWM Clock Input
PWM0	O	PWM0 Voltage Output. PWM outputs can be configured to uses ports 2.6 and 2.7 or 3.4 and 3.3
PWM1	O	PWM1 Voltage Output. See CFG832 Register for further information.
RxD	I/O	Receiver Data Input (Asynchronous) or Data Input/Output (Synchronous) of Serial (UART) Port
TxD	O	Transmitter Data Output (Asynchronous) or Clock Output (Synchronous) of Serial (UART) Port
$\overline{INT0}$	I	Interrupt 0, programmable edge or level triggered Interrupt input, can be programmed to one of two priority levels. This pin can also be used as a gate control input to Timer 0.
$\overline{INT1}$	I	Interrupt 1, programmable edge or level triggered Interrupt input, can be programmed to one of two priority levels. This pin can also be used as a gate control input to Timer 1.
T0	I	Timer/Counter 0 Input
T1	I	Timer/Counter 1 Input
$\overline{CONVST}$	I	Active Low Convert Start Logic Input for the ADC Block when the External Convert Start Function is enabled. A low-to-high transition on this input puts the track-and-hold into its hold mode and starts conversion.
EXTCLK	I	Input for External Clock Signal; has to be enabled via CFG832 Register.
$\overline{WR}$	O	Write Control Signal, Logic Output. Latches the data byte from Port 0 into the external data memory.
$\overline{RD}$	O	Read Control Signal, Logic Output. Enables the external data memory to Port 0.
XTAL2	O	Output of the Inverting Oscillator Amplifier
XTAL1	I	Input to the Inverting Oscillator Amplifier
DGND	G	Digital Ground. Ground reference point for the digital circuitry.
P2.0–P2.7 (A8–A15) (A16–A23)	I/O	Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low will source current because of the internal pull-up resistors. Port 2 emits the high order address bytes during fetches from external program memory and middle and high order address bytes during accesses to the external 24-bit external data memory space.

PIN FUNCTION DESCRIPTIONS (continued)

Mnemonic	Type	Function
$\overline{PSEN}$	O	Program Store Enable, Logic Output. This output is a control signal that enables the external program memory to the bus during external fetch operations. It is active every six oscillator periods except during external data memory accesses. This pin remains high during internal program execution. $\overline{PSEN}$ can also be used to enable serial download mode when pulled low through a resistor on power-up or RESET.
ALE	O	Address Latch Enable, Logic Output. This output is used to latch the low byte (and page byte for 24-bit address space accesses) of the address into external memory during normal operation. It is activated every six oscillator periods except during an external data memory access.
$\overline{EA}$	I	External Access Enable, Logic Input. When held high, this input enables the device to fetch code from internal program memory locations 0000H to 1FFFH. When held low, this input enables the device to fetch all instructions from external program memory. This pin should not be left floating.
P0.7–P0.0 (A0–A7)	I/O	Port 0 is an 8-Bit Open-Drain Bidirectional I/O Port. Port 0 pins that have 1s written to them float and in that state can be used as high impedance inputs. Port 0 is also the multiplexed low order address and data bus during accesses to external program or data memory. In this application it uses strong internal pull-ups when emitting 1s.

TERMINOLOGY

ADC SPECIFICATIONS

Integral Nonlinearity

This is the maximum deviation of any code from a straight line passing through the endpoints of the ADC transfer function. The endpoints of the transfer function are zero scale, a point 1/2 LSB below the first code transition, and full scale, a point 1/2 LSB above the last code transition.

Differential Nonlinearity

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

Offset Error

This is the deviation of the first code transition (0000 . . . 000) to (0000 . . . 001) from the ideal, i.e., +1/2 LSB.

Gain Error

This is the deviation of the last code transition from the ideal AIN voltage (Full Scale – 1.5 LSB) after the offset error has been adjusted out.

Signal to (Noise + Distortion) Ratio

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

$$Signal\ to\ (Noise + Distortion) = (6.02N + 1.76)\ dB$$

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

Total Harmonic Distortion

Total Harmonic Distortion is the ratio of the rms sum of the harmonics to the fundamental.

DAC SPECIFICATIONS

Relative Accuracy

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

Voltage Output Settling Time

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

Digital-to-Analog Glitch Impulse

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

The typical performance plots presented in this section illustrate typical performance of the ADuC832 under various operating conditions.

TPC 1 and TPC 2 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 operating at a sampling rate of 152 kHz and the typically worst case errors in both plots are just less than 0.3 LSBs.

TPC 3 and TPC 4 show the variation in worst case positive (WCP) INL and worst case negative (WCN) INL versus external reference input voltage.

TPC 5 and TPC 6 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 operating at a sampling rate of 152 kHz and the typically worst case errors in both plots is just less than 0.2 LSBs.

TPC 7 and TPC 8 show the variation in worst case positive (WCP) DNL and worst case negative (WCN) DNL versus external reference input voltage.

TPC 9 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.

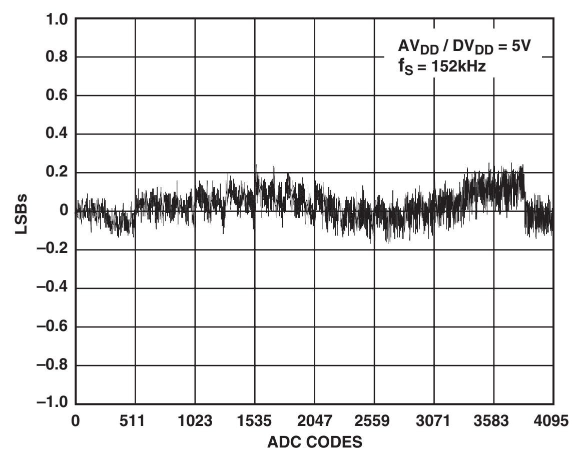
TPC 10 shows a histogram plot of 10,000 ADC conversion results on a dc input for V_{DD} = 3 V. The plot again illustrates a very tight code distribution of 1 LSB with the majority of codes appearing in one output pin.

TPC 11 and TPC 12 show typical FFT plots for the ADuC832. 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 resultant 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.

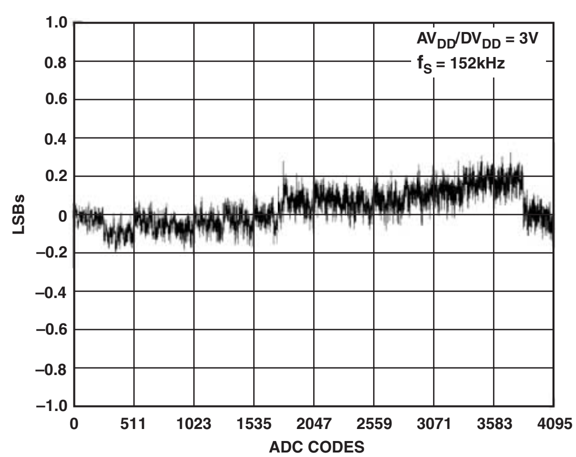
TPC 13 and TPC 14 show typical dynamic performance versus external reference voltages. Again, excellent ac performance can be observed in both plots with some roll-off being observed as V_{REF} falls below 1 V.

TPC 15 shows typical dynamic performance versus sampling frequency. SNR levels of 71 dBs are obtained across the sampling range of the ADuC832.

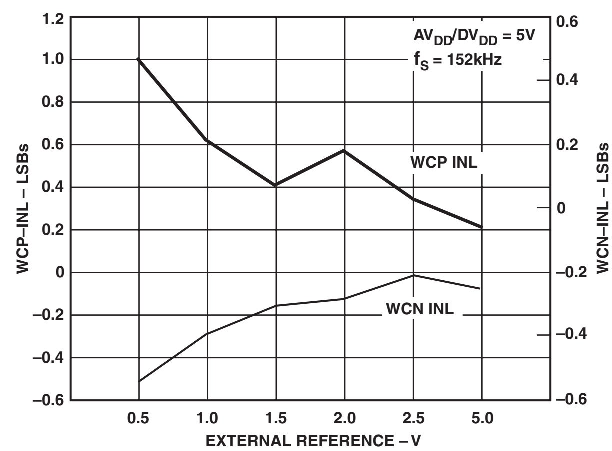
TPC 16 shows the voltage output of the on-chip temperature sensor versus temperature. Although the initial voltage output at 25°C can vary from part to part, the resulting slope of –2 mV/°C is constant across all parts.



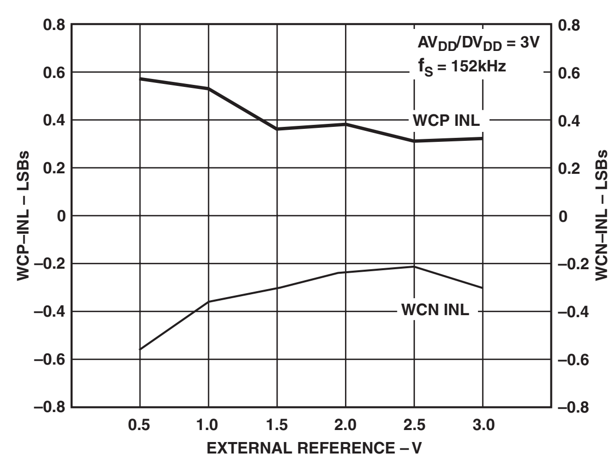
TPC 1. Typical INL Error, V_{DD} = 5 V



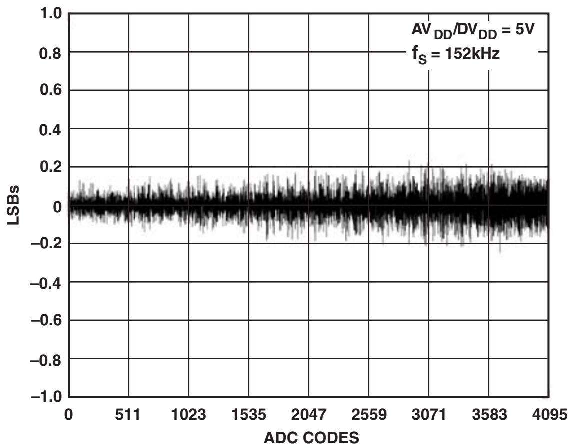
TPC 2. Typical INL Error, V_{DD} = 3 V



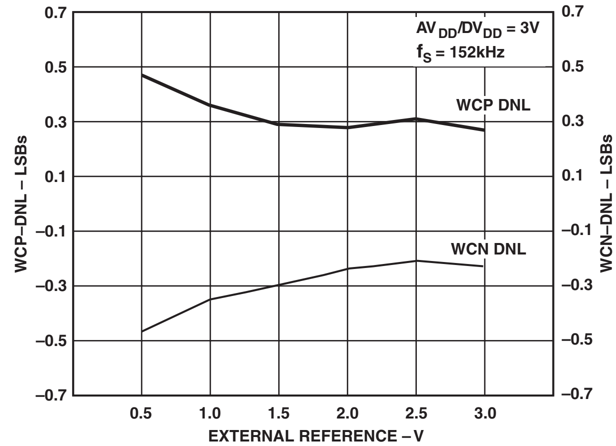
TPC 3. Typical Worst Case INL Error vs. V_{REF}, V_{DD} = 5 V



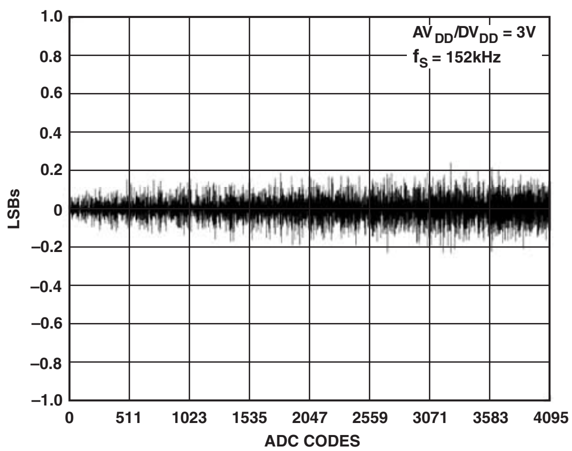
TPC 4. Typical Worst Case INL Error vs. V_{REF}, V_{DD} = 3 V



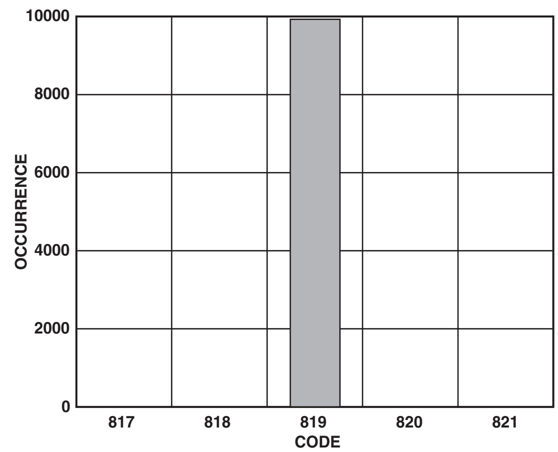
TPC 5. Typical DNL Error, V_{DD} = 5 V



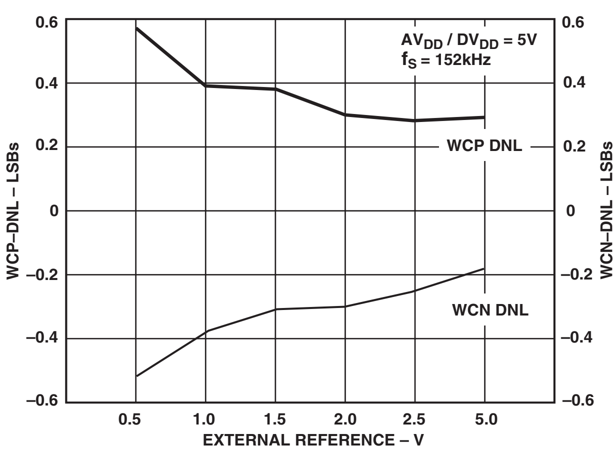
TPC 8. Typical Worst Case DNL Error vs. V_{REF}, V_{DD} = 3 V



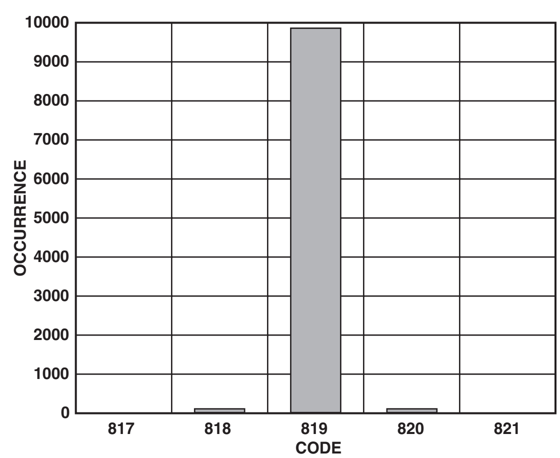
TPC 6. Typical DNL Error, V_{DD} = 3 V



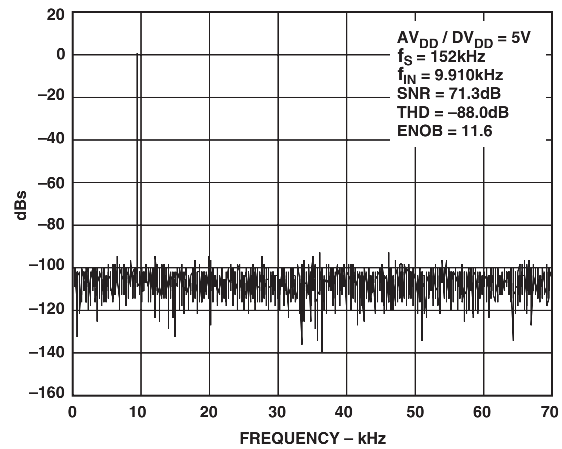
TPC 9. Code Histogram Plot, V_{DD} = 5 V



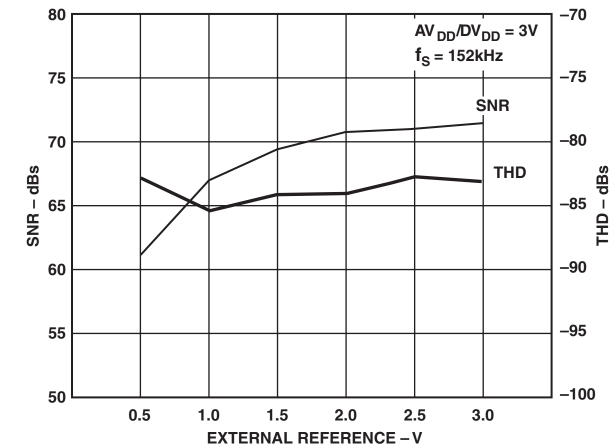
TPC 7. Typical Worst Case DNL Error vs. V_{REF}, V_{DD} = 5 V



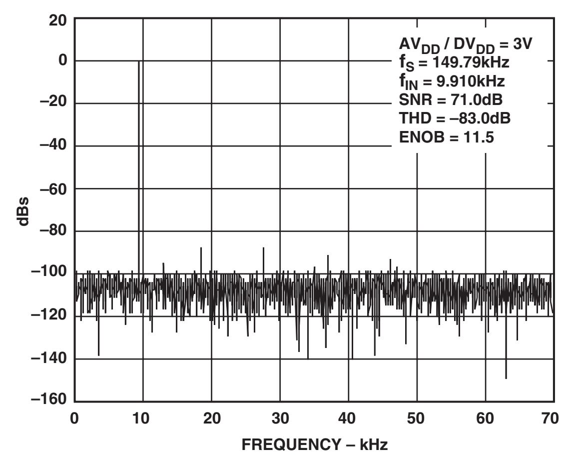
TPC 10. Code Histogram Plot, V_{DD} = 3 V



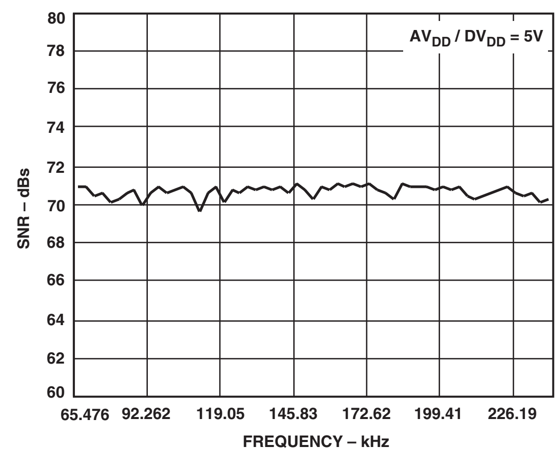
TPC 11. *Dynamic Performance at V_{DD} = 5 V*



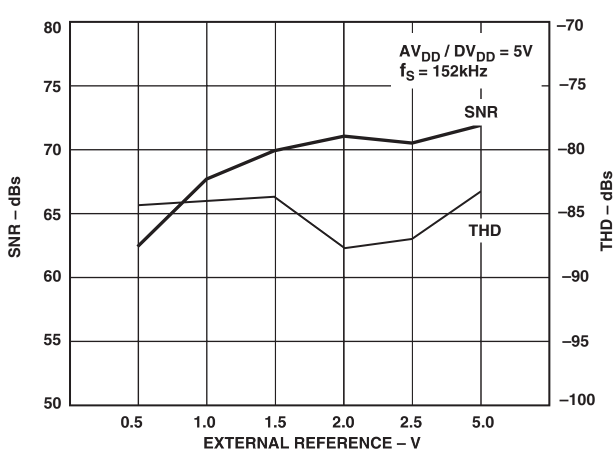
TPC 14. *Typical Dynamic Performance vs. V_{REF}, V_{DD} = 3 V*



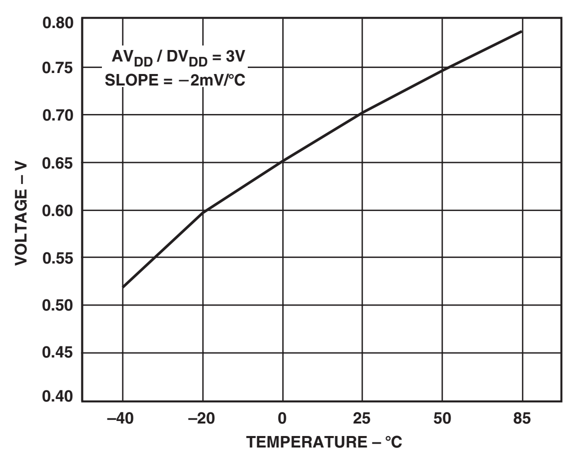
TPC 12. *Dynamic Performance at V_{DD} = 3 V*



TPC 15. *Typical Dynamic Performance vs. Sampling Frequency*



TPC 13. *Typical Dynamic Performance vs. V_{REF}, V_{DD} = 5 V*



TPC 16. *Typical Temperature Sensor Output vs. Temperature*

[redacted]

Flash/EE Program Memory

The ADuC832 provides 62 kBytes of Flash/EE program memory to run user code. The user can choose to run code from this internal memory or from an external program memory.

If the user applies power or resets the device while the $\overline{EA}$ pin is pulled low, the part will execute code from the external program space; otherwise the part defaults to code execution from its internal 62 kBytes of Flash/EE program memory. Unlike the ADuC812, where code execution can overflow from the internal code space to external code space once the PC becomes greater than 1FFFH, the ADuC832 does not support the rollover from F7FFH in internal code space to F800H in external code space. Instead the 2048 bytes between F800H and FFFFH will appear as NOP instructions to user code.

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

Flash/EE Data Memory

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

General-Purpose RAM

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

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

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

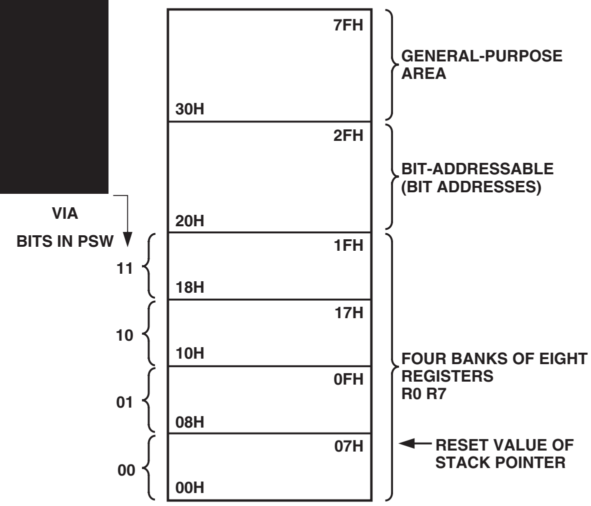


Figure 2. Lower 128 Bytes of Internal Data Memory

The ADuC832 contains 2048 bytes of internal XRAM, 1792 bytes of which can be configured to be used as an extended 11-bit stack pointer.

By default, the stack will operate exactly like an 8052 in that it will roll over from FFH to 00H in the general-purpose RAM. On the ADuC832, however, it is possible (by setting CFG832.7) to enable the 11-bit extended stack pointer. In this case, the stack will roll over from FFH in RAM to 0100H in XRAM.

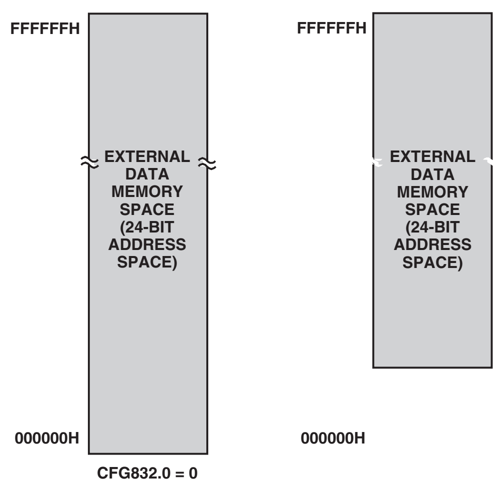
The 11-bit stack pointer is visible in the SP and SPH SFRs. The SP SFR is located at 81H as with a standard 8052. The SPH SFR is located at B7H. The 3 LSBs of this SFR contain the three extra bits necessary to extend the 8-bit stack pointer into an 11-bit stack pointer.BI310.3.5 M 3 reW n1 w 11.5 M /GS3 gs404 554

External Data Memory (External XRAM)
Just like a standard 8051 compatible core, the ADuC832 can access external data memory using a MOVX instruction. The MOVX instruction automatically outputs the various control strobes required to access the data memory.

The ADuC832, however, can access up to 16 MBytes of external data memory. This is an enhancement of the 64 kBytes external data memory space available on a standard 8051 compatible core.

The external data memory is discussed in more detail in the ADuC832 Hardware Design Considerations section.

Internal XRAM
2 kBytes of on-chip data memory exist on the ADuC832. This memory, although on-chip, is also accessed via the MOVX instruction. The 2 kBytes of internal XRAM are mapped into the bottom 2 kBytes of the external address space if the CFG832 bit is set. Otherwise, access to the external data memory will occur just like a standard 8051. When using the internal XRAM, Ports 0 and 2 are free to be used as general-purpose I/O.



Data Pointer (DPTR)

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

The ADuC832 supports dual data pointers. Refer to the Dual Data Pointer section.

Program Status Word (PSW)

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

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

Table I. PSW SFR Bit Designations

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

Power Control SFR (PCON)

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

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

Table II. PCON SFR Bit Designations

Bit	Name	Description
7	SMOD	Double UART Baud Rate
6	SERIPD	I^2C/SPI Power-Down Interrupt Enable
5	INT0PD	$\overline{\text{INT0}}$ Power-Down Interrupt Enable
4	ALEOFF	Disable ALE Output
3	GF1	General-Purpose Flag Bit
2	GF0	General-Purpose Flag Bit
1	PD	Power-Down Mode Enable
0	IDL	Idle Mode Enable

SPECIAL FUNCTION REGISTERS

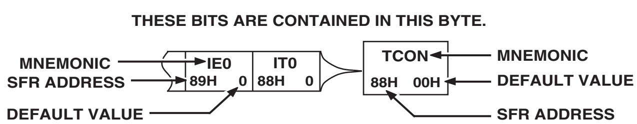
All registers except the program counter and the four general-purpose register banks reside in the special function register (SFR) area. The SFR registers include control, configuration, and data registers that provide an interface between the CPU and other on-chip peripherals.

Figure 6 shows a full SFR memory map and SFR contents on Reset. Unoccupied SFR locations are shown dark-shaded in the figure below (NOT USED). Unoccupied locations in the SFR address space are not implemented i.e., no register exists at this location. If an unoccupied location is read, an unspecified value is returned. SFR locations reserved for on-chip testing are shown lighter shaded below (RESERVED) and should not be accessed by user software. Sixteen of the SFR locations are also bit addressable and denoted by '1' in the figure below, i.e., the bit addressable SFRs are those whose address ends in 0H or 8H.

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

SPICON[1] F8H 04H	DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	RESERVED
B[1] F0H 00H	ADCOFSL[3] F1H 00H	ADCOFSH[3] F2H 20H	ADCGAINL[3] F3H 00H	ADCGAINH[3] F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H
I2CCON[1] E8H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 00H
ACC[1] E0H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
ADCCON2[1] D8H 00H	ADCDATAL D9H 00H	ADCDATAH DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH
PSW[1] D0H 00H	RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	PLLCON D7H 53H
T2CON[1] C8H 00H	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
WDCON[1] C0H 10H	RESERVED	CHIPID C2H 2XH	RESERVED	RESERVED	RESERVED	EDARL C6H 00H	EDARH C7H 00H
IP[1] B8H 00H	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
P3[1] B0H FFH	PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	NOT USED	NOT USED	SPH B7H 00H
IE[1] A8H 00H	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG832 AFH 00H
P2[1] A0H FFH	TIMECON A1H 00H	HTHSEC A2H 00H	SEC A3H 00H	MIN A4H 00H	HOUR A5H 00H	INTVAL A6H 00H	DPCON A7H 00H
SCON[1] 98H 00H	SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	NOT USED
P1[1, 2] 90H FFH	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
TCON[1] 88H 00H	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
P0[1] 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

SFR MAP KEY:



NOTES

[1]SFRs WHOSE ADDRESS ENDS IN 0H OR 8H ARE BIT ADDRESSABLE.
[2]THE PRIMARY FUNCTION OF PORT1 IS AS AN ANALOG INPUT PORT; THEREFORE, TO ENABLE THE DIGITAL SECONDARY FUNCTIONS ON THESE PORT PINS, WRITE A "0" TO THE CORRESPONDING PORT 1 SFR BIT.
[3]CALIBRATION COEFFICIENTS ARE PRECONFIGURED ON POWER-UP TO FACTORY CALIBRATED VALUES.

Figure 6. Special Function Register Locations and Reset Values

ADC CIRCUIT INFORMATION

General Overview

The ADC conversion block incorporates a fast, 8-channel, 12-bit, single-supply ADC. This block provides the user with multichannel mux, track/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 to V_{REF}. A high precision, low drift, and factory calibrated 2.5 V reference is provided on-chip. An external reference can be connected as described later. 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 ADuC832 is 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 ninth ADC channel input) facilitating a temperature sensor implementation.

ADC Transfer Function

The analog input range for the ADC is 0 V to V_{REF}. For this range, the designed code transitions occur midway between successive integer LSB values (i.e., 1/2 LSB, 3/2 LSBs, 5/2 LSBs . . . FS –3/2 LSBs). 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 to V_{REF} range is shown in Figure 7.

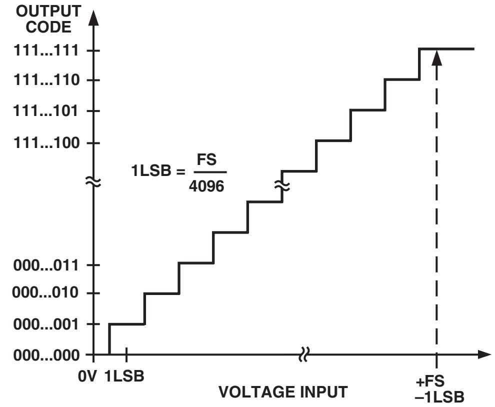


Figure 7. ADC Transfer Function

Typical Operation

Once configured via the ADCCON 1-3 SFRs, the ADC will convert the analog input and provide an ADC 12-bit result word in the ADCDATAH/L SFRs. The top four bits of the ADCDATAH SFR will be written with the channel selection bits so as to identify the channel result. The format of the ADC 12-bit result word is shown in Figure 8.

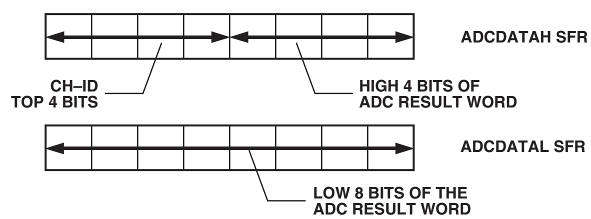


Figure 8. ADC Result Format

ADCCON1 – (ADC Control SFR #1)

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

SFR Address:	EFH
SFR Power-On Default Value:	00H
Bit Addressable:	NO

Table III. ADCCON1 SFR Bit Designations

Bit	Name	Description
ADCCON1.7	MD1	The Mode bit selects the active operating mode of the ADC. Set by the user to power up the ADC. Cleared by the user to power down the ADC.
ADCCON1.6	EXT_REF	Set by the user to select an external reference. Cleared by the user to use the internal reference.
ADCCON1.5 ADCCON1.4	CK1 CK0	The ADC clock divide bits (CK1, CK0) select the divide ratio for the PLL master clock used to generate the ADC clock. To ensure correct ADC operation, the divider ratio must be chosen to reduce the ADC clock to 4.5 MHz and below. A typical ADC conversion will require 17 ADC clocks. The divider ratio is selected as follows: **CK1 CK0 MCLK Divider** 0 0 8 0 1 4 1 0 16 1 1 32
ADCCON1.3 ADCCON1.2	AQ1 AQ0	The ADC acquisition select bits (AQ1, AQ0) select the time provided for the input track-and-hold amplifier to acquire the input signal. An acquisition of three or more ADC clocks is recommended; clocks are selected as follows: **AQ1 AQ0 #ADC Clks** 0 0 1 0 1 2 1 0 3 1 1 4
ADCCON1.1	T2C	The Timer 2 conversion bit (T2C) is set by the user to enable the Timer 2 overflow bit be used as the ADC convert start trigger input.
ADCCON1.0	EXC	The external trigger enable bit (EXC) is set by the user to allow the external Pin P3.5 (CONVST) to be used as the active low convert start input. This input should be an active low pulse (minimum pulsewidth >100 ns) at the required sample rate.

ADCCON2 – (ADC Control SFR #2)
The ADCCON2 register controls ADC channel selection and conversion modes as detailed below.

SFR Address: D8H
SFR Power-On Default Value: 00H
Bit Addressable: YES

Table IV. ADCCON2 SFR Bit Designations

Bit	Name	Description
ADCCON2.7	ADCI	The ADC interrupt bit (ADCI) is set by hardware at the end of a single ADC conversion cycle or at the end of a DMA block conversion. ADCI is cleared by hardware when the PC vectors to the ADC Interrupt Service Routine. Otherwise, the ADCI bit should be cleared by user code.
ADCCON2.6	DMA	The DMA mode enable bit (DMA) is set by the user to enable a preconfigured ADC DMA mode operation. A more detailed description of this mode is given in the ADC DMA Mode section. The DMA bit is automatically set to "0" at the end of a DMA cycle. Setting this bit causes the ALE output to cease, it will start again when DMA is started and will operate correctly after DMA is complete.
ADCCON2.5	CCONV	The continuous conversion bit (CCONV) is set by the user to initiate the ADC into a continuous mode of conversion. In this mode, the ADC starts converting based on the timing and channel configuration already set up in the ADCCON SFRs; the ADC automatically starts another conversion once a previous conversion has completed.
ADCCON2.4	SCONV	The single conversion bit (SCONV) is set to initiate a single conversion cycle. The SCONV bit is automatically reset to "0" on completion of the single conversion cycle.
ADCCON2.3 ADCCON2.2 ADCCON2.1 ADCCON2.0	CS3 CS2 CS1 CS0	The channel selection bits (CS3–0) allow the user to program the ADC channel selection under software control. When a conversion is initiated, the channel converted will be that pointed to by these channel selection bits. In DMA mode, the channel selection is derived from the channel ID written to the external memory.

CS3	CS2	CS1	CS0	CH#	
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	2	
0	0	1	1	3	
0	1	0	0	4	
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0	Temp Monitor	Requires minimum of 1 µs to acquire
1	0	0	1	DAC0	Only use with Internal DAC o/p buffer on
1	0	1	0	DAC1	Only use with Internal DAC o/p buffer on
1	0	1	1	AGND	
1	1	0	0	VREF	
1	1	1	1	DMA STOP	Place in XRAM location to finish DMA sequence, see the section ADC DMA Mode.

All other combinations reserved

ADCCON3 – (ADC Control SFR #3)

The ADCCON3 register controls the operation of various calibration modes as well as giving an indication of ADC busy status.

SFR Address:	F5H
SFR Power-On Default Value:	00H
Bit Addressable:	NO

Table V. ADCCON3 SFR Bit Designations

Bit	Name	Description
ADCCON3.7	BUSY	The ADC Busy Status Bit (BUSY) is a read-only status bit that is set during a valid ADC conversion or calibration cycle. Busy is automatically cleared by the core at the end of conversion or calibration.
ADCCON3.6	GNCLD	Gain Calibration Disable Bit. Set to "0" to Enable Gain Calibration. Set to "1" to Disable Gain Calibration.
ADCCON3.5	AVGS1	Number of Averages Selection Bits.
ADCCON3.4	AVGS0	This bit selects the number of ADC readings averaged during a calibration cycle. AVGS1 AVGS0 Number of Averages 0 0 15 0 1 1 1 0 31 1 1 63
ADCCON3.3	RSVD	Reserved. This bit should always be written as "0."
ADCCON3.2	RSVD	This bit should always be written as "1" by the user when performing calibration.
ADCCON3.1	TYPICAL	Calibration Type Select Bit. This bit selects between Offset (zero-scale) and Gain (full-scale) calibration. Set to "0" for Offset Calibration. Set to "1" for Gain Calibration.
ADCCON3.0	SCAL	Start Calibration Cycle Bit. When set, this bit starts the selected calibration cycle. It is automatically cleared when the calibration cycle is completed.

Driving the A/D Converter

The ADC incorporates a successive approximation (SAR) architecture involving a charge-sampled input stage. Figure 9 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 9. 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 zero, indicating that the sampled charge on the input capacitor is balanced out by the charge being output by the capacitor DAC. The digital value finally 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.

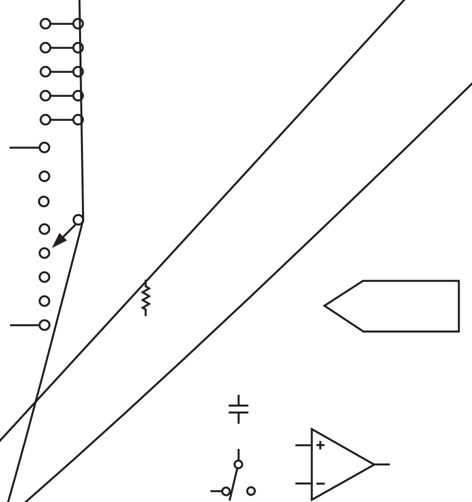


Figure 9. Internal ADC Structure

ground, no amplifier can deliver signals all the way to ground when powered by a single supply. Therefore, if a negative supply is available, you might consider using it to power the front end amplifiers. If you do, however, be sure to include the Schottky diodes shown in Figure 10 (or at least the lower of the two diodes) to protect the analog input from undervoltage conditions. To summarize this section, use the circuit of Figure 10 to drive the analog input pins of the ADuC832.

Voltage Reference Connections

The on-chip 2.5 V band gap voltage reference can be used as the reference source for the ADC and DACs. To ensure the accuracy of the voltage reference, you must decouple the V_{REF} pin to ground with a 0.1 µF capacitor, and the C_{REF} pin to ground with a 0.1 µF capacitor as shown in Figure 11.

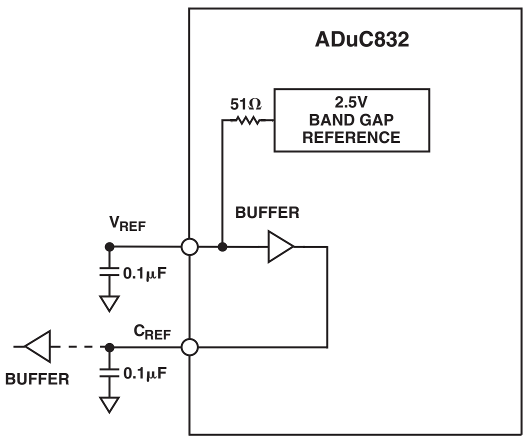


Figure 11. Decoupling V_{REF} and C_{REF}

If the internal voltage reference is to be used as a reference for external circuitry, the C_{REF} output should be used. However, a buffer must be used in this case to ensure that no current is drawn from the C_{REF} pin itself. The voltage on the C_{REF} pin is that of an internal node within the buffer block, and its voltage is critical to ADC and DAC accuracy. On the ADuC812, V_{REF} was the recommended output for the external reference; this can be used but it should be noted that there will be a gain error between this reference and that of the ADC.

The ADuC832 powers up with its internal voltage reference in the "on" state. This is available at the V_{REF} pin, but as noted before there will be a gain error between this and that of the ADC. The C_{REF} output becomes available when the ADC is powered up.

If an external voltage reference is preferred, it should be connected to the V_{REF} and C_{REF} pins as shown in Figure 12. Bit 6 of the ADCCON1 SFR must be set to 1 to switch in the external reference voltage.

To ensure accurate ADC operation, the voltage applied to V_{REF} must be between 1 V and AV_{DD}. In situations where analog input signals are proportional to the power supply (such as some strain gage applications) it may be desirable to connect the C_{REF} and V_{REF} pins directly to AV_{DD}.

Operation of the ADC or DACs with a reference voltage below 1 V, however, may incur loss of accuracy, eventually resulting in missing codes or non-monotonicity. For that reason, do not use a reference voltage less than 1 V.

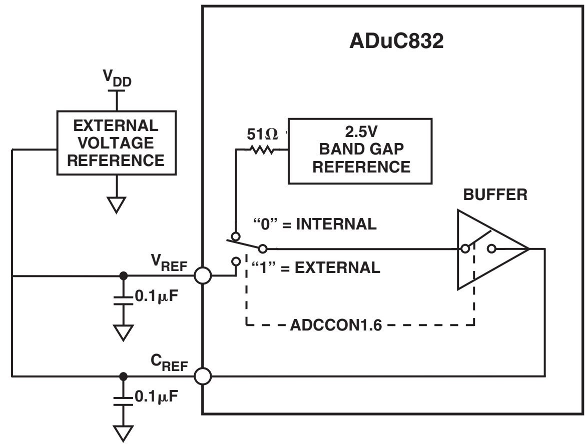


Figure 12. Using an External Voltage Reference

To maintain compatibility with the ADuC812, the external reference may also be connected to the V_{REF} pin as shown in Figure 13, to overdrive the internal reference. Note this introduces a gain error for the ADC that has to be calibrated out; thus the previous method is the recommended one for most users. For this method to work, ADCCON1.6 should be configured to use the internal reference. The external reference will then overdrive this.

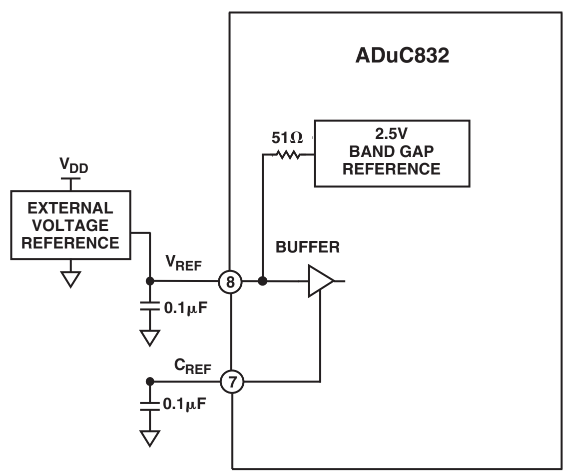


Figure 13. Using an External Voltage Reference

Configuring the ADC

The ADuC832's successive approximation ADC is driven by a divided down version of the master clock. To ensure adequate ADC operation, this ADC clock must be between 400 kHz and 6 MHz, and optimum performance is obtained with ADC clock between 400 kHz and 4.5 MHz. Frequencies within this range can easily be achieved with master clock frequencies from 400 kHz to well above 16 MHz with the four ADC clock divide ratios to choose from. For example, set the ADC clock divide ratio to 4 (i.e., ADCCLK = 16.777216 MHz/8 = 2 MHz) by setting the appropriate bits in ADCCON1 (ADCCON1.5 = 0, ADCCON1.4 = 0).

The total ADC conversion time is 15 ADC clocks, plus 1 ADC clock for synchronization, plus the selected acquisition time (1, 2, 3, or 4 ADC clocks). For the example above, with a 3-clock acquisition time, total conversion time is 19 ADC clocks (or 9.05 μs for a 2 MHz ADC clock).

In continuous conversion mode, a new conversion begins each time the previous one finishes. The sample rate is then simply the inverse of the total conversion time described above. In the example above, the continuous conversion mode sample rate would be 110.3 kHz.

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

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

ADC DMA Mode

The on-chip ADC has been designed to run at a maximum conversion speed of 4 μs (247 kHz sampling rate). When converting at this rate, the ADuC832 MicroConverter has 4 μs to read the ADC result and store the result in memory for further postprocessing, otherwise the next ADC sample could be lost. In an interrupt driven routine, the MicroConverter would also have to jump to the ADC Interrupt Service routine, which will also increase the time required to store the ADC results. In applications where the ADuC832 cannot sustain the interrupt rate, an ADC DMA mode is provided.

To enable DMA mode, Bit 6 in ADCCON2 (DMA) must be set. This allows the ADC results to be written directly to a 16 MByte external static memory SRAM (mapped into data memory space) without any interaction from the ADuC832 core. This mode allows the ADuC832 to capture a contiguous sample stream at full ADC update rates (247 kHz).

A Typical DMA Mode Configuration Example

To set the ADuC832 into DMA mode, a number of steps must be followed:

1. The ADC must be powered down. This is done by ensuring MD1 and MD0 are both set to 0 in ADCCON1.
2. The DMA address pointer must be set to the start address of where the ADC results are to be written. This is done by writing to the DMA mode address pointers DMAL, DMAH, and DMAP. DMAL must be written to first, followed by DMAH, and then by DMAP.
3. The external memory must be preconfigured. This consists of writing the required ADC channel IDs into the top four bits of every second memory location in the external SRAM, starting at the first address specified by the DMA address pointer. As the ADC DMA mode operates independent from the ADuC832 core, it is necessary to provide it with a stop command. This is done by duplicating the last channel ID to be converted followed by "1111" into the next channel selection field. A typical preconfiguration of external memory is as follows:

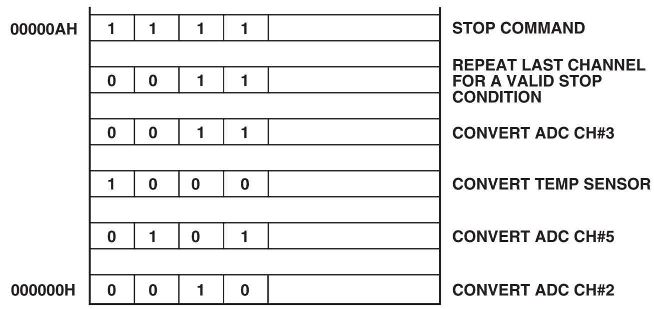


Figure 14. Typical DMA External Memory Preconfiguration

4. The DMA is initiated by writing to the ADC SFRs in the following sequence:
   a. ADCCON2 is written to enable the DMA mode, i.e., MOV ADCCON2, #40H; DMA mode enabled.
   b. ADCCON1 is written to configure the conversion time and power-up of the ADC. It can also enable Timer 2 driven conversions or external triggered conversions if required.
   c. ADC conversions are initiated. This is done by starting single conversions, starting Timer 2, running for Timer 2 conversions, or receiving an external trigger.

When the DMA conversions are completed, the ADC interrupt bit, ADCI, is set by hardware and the external SRAM contains the new ADC conversion results as shown below. It should be noted that no result is written to the last two memory locations.

When the DMA mode logic is active, it takes the responsibility of storing the ADC results away from both the user and ADuC832 core logic. As it writes the results of the ADC conversions to external memory, it takes over the external memory interface from the core. Thus, any core instructions that access the external memory while DMA mode is enabled will not get access to it. The core will execute the instructions and they will take the same time to execute but they will not gain access to the external memory.

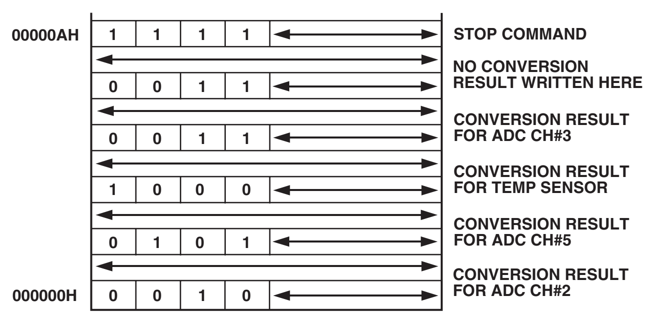


Figure 15. Typical External Memory Configuration Post ADC DMA Operation

The DMA logic operates from the ADC clock and uses pipelining to perform the ADC conversions and 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 the following diagram.

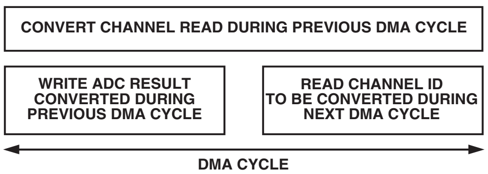


Figure 16. DMA Cycle

From the previous diagram, it can be seen 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 16. 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 ADC DMA mode of operation. This means that even though the instruction that accesses the external ports 0 or 2 will appear to execute, no data will be 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 will be able to 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 will be able to 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 ADuC832 has 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 (eight bits) and the gain calibration coefficient is divided into ADCGAINH (six bits) and ADCGAINL (eight 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

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

Device calibration can be initiated to compensate for significant changes in operating conditions frequency, analog input range, reference voltage, and supply voltages. In this calibration mode, offset calibration uses internal AGND selected via ADCCON2 register bits CS3–CS0 (1011) and gain calibration uses internal V_{REF} selected by CS3–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 using an external reference, tie 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 CS3–CS0 and perform system offset calibration. Select the channel connected to V_{REF} via CS3–CS0 and perform system gain calibration.

The ADC should be configured to use settings for an ADCCLK of divide by 16 and 4 acquisition clocks.

INITIATING CALIBRATION IN CODE

When calibrating the ADC using ADCCON1, the ADC should be set up into the configuration in which it will be used. The ADCCON3 register can then be used to set up the device up and calibrate the ADC offset and gain.

```
MOV ADCCON1,#0ACH    ;ADC on; ADCCLK set
                     ;to divide by 16,4
                     ;acquisition clock
```

To calibrate device offset:

```
MOV ADCCON2,#0BH     ;select internal AGND
MOV ADCCON3,#25H     ;select offset calibration,
                     ;31 averages per bit,
                     ;offset calibration
```

To calibrate device gain:

```
MOV ADCCON2,#0CH     ;select internal VREF
MOV ADCCON3,#27H     ;select offset calibration,
                     ;31 averages per bit,
                     ;offset calibration
```

To calibrate system offset:
Connect system AGND to an ADC channel input (0).

```
MOV ADCCON2,#00H     ;select external AGND
MOV ADCCON3,#25H     ;select offset calibration,
                     ;31 averages per bit
```

To calibrate system gain:
Connect system V_{REF} to an ADC channel input (1).

```
MOV ADCCON2,#01H     ;select external VREF
MOV ADCCON3,#27H     ;select offset calibration,
                     ;31 averages per bit,
                     ;offset calibration
```

The calibration cycle time T_{CAL} is calculated by the following equation:

$$T_{CAL} = 14 \times ADCCLK \times NUMAV \times (16 + T_{ACQ})$$

For an ADCCLK/F_{CORE} divide ratio of 16, a T_{ACQ} = 4 ADCCLK, NUMAV = 15, the calibration cycle time is:

$$T_{CAL} = 14 \times (1/1048576) \times 15 \times (16 + 4)$$
$$T_{CAL} = 4.2\ ms$$

In a calibration cycle, the ADC busy flag (Bit 7), instead of framing an individual ADC conversion as in normal mode, will go high at the start of calibration and only return to zero at the end of the calibration cycle. It can therefore be monitored in code to indicate when the calibration cycle is completed. The following code can be used to monitor the BUSY signal during a calibration cycle:

```
WAIT:
  MOV A, ADCCON3    ;move ADCCON3 to A
  JB ACC.7, WAIT    ;If Bit 7 is set jump to
                    WAIT else continue
```

NONVOLATILE FLASH/EE MEMORY

Flash/EE Memory Overview

The ADuC832 incorporates Flash/EE memory technology on-chip to provide the user with nonvolatile, in-circuit, reprogrammable code and data memory space. Flash/EE memory is a relatively recent type of nonvolatile memory technology and is based on a single transistor cell architecture.

This technology is basically an outgrowth of EPROM technology and was developed through the late 1980s. Flash/EE memory takes the flexible in-circuit reprogrammable features of EEPROM and combines them with the space efficient/density features of EPROM (see Figure 17).

Because Flash/EE technology is based on a single transistor cell architecture, a Flash memory array, like EPROM, can be implemented to achieve the space efficiencies or memory densities required by a given design. Like EEPROM, Flash memory can be programmed in-system at a byte level, although it must first be erased; the erase being performed in page blocks. Thus, Flash memory is often and more correctly referred to as Flash/EE memory.

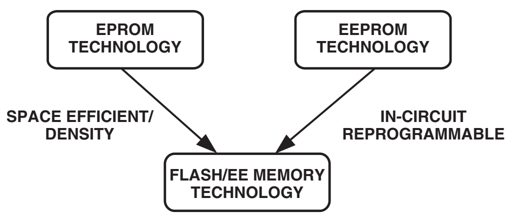


Figure 17. Flash/EE Memory Development

Overall, Flash/EE memory represents a step closer to the ideal memory device that includes nonvolatility, in-circuit programmability, high density, and low cost. Incorporated in the ADuC832, Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one-time programmable (OTP) devices at remote operating nodes.

Flash/EE Memory and the ADuC832

The ADuC832 provides two arrays of Flash/EE memory for user applications. 62 kBytes of Flash/EE program space are provided on-chip to facilitate code execution without any external discrete ROM device requirements. The program memory can be programmed in-circuit using the serial download mode provided, using conventional third party memory programmers, or via a user defined protocol that can configure it as data if required.

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

ADuC832 Flash/EE Memory Reliability

The Flash/EE program and data memory arrays on the ADuC832 are fully qualified for two key Flash/EE memory characteristics, namely Flash/EE Memory Cycling Endurance and Flash/EE Memory Data Retention.

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

a. Initial page erase sequence b. Read/verify sequence c. Byte program sequence d. Second read/verify sequence	A single Flash/EE Memory Endurance Cycle

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

As indicated in the specification pages of this data sheet, the ADuC832 Flash/EE Memory Endurance qualification has been carried out in accordance with JEDEC Specification A117 over the industrial temperature range of –40°C to +25°C and +85°C to +125°C. The results allow the specification of a minimum endurance figure over supply and temperature of 100,000 cycles, with an endurance figure of 700,000 cycles being typical of operation at 25°C.

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. Again, the ADuC832 has been qualified in accordance with the formal JEDEC Retention Lifetime Specification (A117) at a specific junction temperature (T_J = 55°C). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit described above before data retention is characterized. This means that the Flash/EE memory is guaranteed to retain its data for its full specified retention lifetime every time the Flash/EE memory is reprogrammed. It should also be noted that retention lifetime, based on an activation energy of 0.6 eV, will derate with T_J as shown in Figure 18.

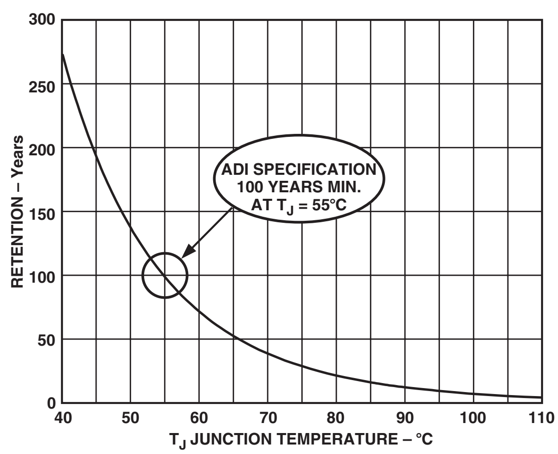


Figure 18. Flash/EE Memory Data Retention

Using the Flash/EE Program Memory

The 62 kByte Flash/EE program memory array is mapped into the lower 62 kBytes of the 64 kBytes program space addressable by the ADuC832, and is used to hold user code in typical applications.

The program memory Flash/EE memory arrays can be programmed in three ways:

(1) Serial Downloading (In-Circuit Programming)

The ADuC832 facilitates code download via the standard UART serial port. The ADuC832 will enter serial download mode after a reset or power cycle if the $\overline{PSEN}$ pin is pulled low through an external 1 kΩ resistor. Once in serial download mode, the user can download code to the full 62 kBytes of Flash/EE program memory while the device is in-circuit in its target application hardware.

A PC serial download executable is provided as part of the ADuC832 QuickStart development system. The Serial Download protocol is detailed in a MicroConverter Application Note uC004.

(2) Parallel Programming

The parallel programming mode is fully compatible with conventional third party Flash or EEPROM device programmers. In this mode, Ports P0, P1, and P2 operate as the external data and address bus interface, ALE operates as the Write Enable strobe, and Port P3 is used as a general configuration port that configures the device for various program and erase operations during parallel programming. The high voltage (12 V) supply required for Flash programming is generated using on-chip charge pumps to supply the high voltage program lines.

The complete parallel programming specification is available on the MicroConverter home page at www.analog.com/microconverter.

(3) User Download Mode (ULOAD)

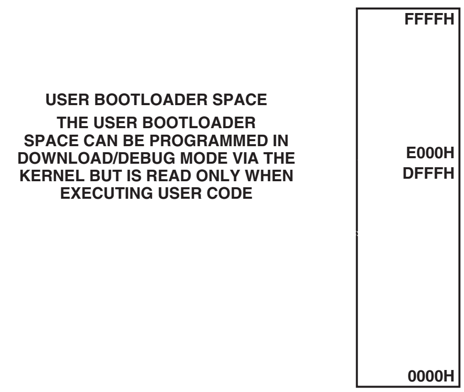
In Figure 19 we can see that it was possible to use the 62 kBytes of Flash/EE program memory available to the user as one single block of memory. In this mode, all of the Flash/EE memory is read only to user code.

However, the Flash/EE program memory can also be written to during runtime simply by entering ULOAD mode. In ULOAD mode, the lower 56 kBytes of program memory can be erased and reprogrammed by user software as shown in Figure 19. ULOAD mode can be used to upgrade your code in the field via any user defined download protocol. Configuring the SPI port on the ADuC832 as a slave, it is possible to completely reprogram the 56 kBytes of Flash/EE program memory in only 5 seconds (see uC007).

Alternatively, ULOAD mode can be used to save data to the 56 kBytes of Flash/EE memory. This can be extremely useful in data logging applications where the ADuC832 can provide up to 60 kBytes of NV data memory on chip (4 kBytes of dedicated Flash/EE data memory also exist).

The upper 6 kBytes of the 62 kBytes of Flash/EE program memory is only programmable via serial download or parallel programming. This means that this space appears as read only to user code. Therefore, it cannot be accidently erased or reprogrammed by erroneous code execution. This makes it very suitable to use the 6 kBytes as a bootloader. A Bootload Enable option exists in the serial downloader to "Always RUN from E000H after Reset." If using a bootloader, this option is recommended to ensure that the bootloader always executes correct code after reset.

Programming the Flash/EE program memory via ULOAD mode is described in more detail in the description of ECON and also in technical note uC007.



USING THE FLASH/EE DATA MEMORY

The 4 kBytes of Flash/EE data memory is configured as 1024 pages, each of four bytes. As with the other ADuC832 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) are 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 20.

ECON—Flash/EE Memory Control SFR

Programming of either the Flash/EE data memory or the 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.

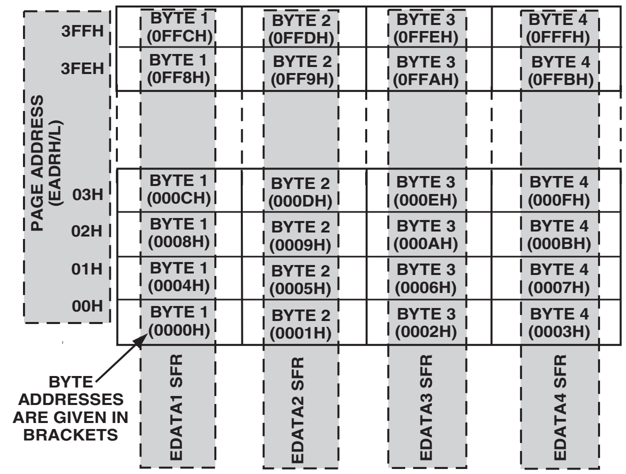


Figure 20. Flash/EE Data Memory Control and Configuration

Table VII. ECON—Flash/EE Memory Commands

ECON VALUE	COMMAND DESCRIPTION (NORMAL MODE) (Power-On Default)	COMMAND DESCRIPTION (ULOAD MODE)
01H READ	Results in four bytes in the Flash/EE data memory, addressed by the page address EADRH/L, being read into EDATA 1 to 4.	Not Implemented. Use the MOVC instruction.
02H WRITE	Results in four 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: The four 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: The 256 bytes in the page being addressed must be pre-erased.
03H	Reserved Command	Reserved Command
04H VERIFY	Verifies if the data in EDATA1–4 is contained in the page address given by EADRH/L. A subsequent read of the ECON SFR will result in a 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 the Erase of 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 the erase of entire 4 kBytes of Flash/EE data memory.	Results in the Erase of the entire 56 kBytes of ULOAD Flash/EE program memory.
81H READBYTE	Results in the byte in the Flash/EE data memory, addressed by the byte address EADRH/L, being read into EDATA1 (0 ≤ EADRH / L ≤ 0FFFH).	Not Implemented. Use the MOVC command.
82H WRITEBYTE	Results in the byte in EDATA1 being written into Flash/EE data memory, at the byte address EADRH/L.	Results in the byte in EDATA1 being written into Flash/EE program memory, at the byte address EADRH/L (0 ≤ EADRH / L ≤ DFFFH).
0FH EXULOAD	Leaves the ECON instructions to operate on the Flash/EE data memory.	Enters NORMAL mode directing subsequent ECON instructions to operate on the Flash/EE data memory.
F0H ULOAD	Enters ULOAD mode, directing subsequent ECON instructions to operate on the Flash/EE program memory.	Leaves the ECON instructions to operate on the Flash/EE program memory.

Example: Programming the Flash/EE Data Memory

A user wishes to program F3H into the second byte on Page 03H of the Flash/EE data memory space while preserving the other three bytes already in this page.

A typical program of the Flash/EE Data array will involve:

1) setting EADRH/L with the page address
2) writing the data to be programmed to the EDATA1–4
3) writing the ECON SFR with the appropriate command

Step 1: Set Up the Page Address

The two address registers EADRH and EADRL hold the high byte address and the low byte address of the page to be addressed. The assembly language to set up the address may appear as:

```
MOV EADRH,#0        ; Set Page Address Pointer
MOV EADRL,#03H
```

Step 2: Set Up the EDATA Registers

We must now write the four values to be written into the page into the four SFRs EDATA1–4. Unfortunately, we do not know three of them. Thus, we must read the current page and overwrite the second byte.

```
MOV  ECON,#1        ; Read Page into EDATA1-4
MOV  EDATA2,#0F3H  ; Overwrite byte 2
```

Step 3: Program Page

A byte in the Flash/EE array can only be programmed if it has previously been erased. To be more specific, a byte can only be programmed if it already holds the value FFH. Because of the Flash/EE architecture, this erase must happen at a page level; therefore, a minimum of four bytes (one page) will be erased when an erase command is initiated. Once the page is erased we can program the four bytes in-page and then perform a verification of the data.

```
MOV  ECON,#5        ; ERASE Page
MOV  ECON,#2        ; WRITE Page
MOV  ECON,#4        ; VERIFY Page
MOV  A,ECON         ; Check if ECON=0 (OK!)
JNZ  ERROR
```

Although the 4 kBytes of Flash/EE data memory are shipped from the factory pre-erased, i.e., byte locations set to FFH, it is nonetheless good programming practice to include an erase-all routine as part of any configuration/setup code running on the ADuC832. An ERASE-ALL command consists of writing "06H" to the ECON SFR, which initiates an erase of the 4 kByte Flash/EE array. This command coded in 8051 assembly would appear as:

```
MOV  ECON,#06H      ; Erase all Command
                    ; 2 ms Duration
```

Flash/EE Memory Timing

Typical program and erase times for the ADuC832 are as follows:

NORMAL MODE (operating on Flash/EE data memory)

READPAGE (4 bytes)	– 5 machine cycles
WRITEPAGE (4 bytes)	– 380 µs
VERIFYPAGE (4 bytes)	– 5 machine cycles
ERASEPAGE (4 bytes)	– 2 ms
ERASEALL (4 kBytes)	– 2 ms
READBYTE (1 byte)	– 3 machine cycle
WRITEBYTE (1 byte)	– 200 µs

ULOAD MODE (operating on Flash/EE program memory)

WRITEPAGE (256 bytes)	– 15 ms
ERASEPAGE (64 bytes)	– 2 ms
ERASEALL (56 kBytes)	– 2 ms
WRITEBYTE (1 byte)	– 200 µs

It should be noted that a given mode of operation is initiated as soon as the command word is written to the ECON SFR. The core microcontroller operation on the ADuC832 is idled until the requested Program/Read or Erase mode is completed.

In practice, this means that even though the Flash/EE memory mode of operation is typically initiated with a two-machine cycle MOV instruction (to write to the ECON SFR), the next instruction will not be executed until the Flash/EE operation is complete. This means that the core will not respond to interrupt requests until the Flash/EE operation is complete, although the core peripheral functions like counter/timers will continue to count and time as configured throughout this period.

ADuC832 Configuration SFR (CFG832)

The CFG832 SFR contains the necessary bits to configure the internal XRAM, External Clock select, PWM output selection, DAC buffer, and the extended SP. By default it configures the user into 8051 mode, i.e., extended SP is disabled, internal XRAM is disabled.

CFG832	**ADuC832 Config SFR**
SFR Address	AFH
Power-On Default Value	00H
Bit Addressable	No

Table VIII. CFG832 SFR Bit Designations

Bit	Name	Description
7	EXSP	Extended SP Enable. When set to "1" by the user, the stack will roll over from SPH/SP = 00FFH to 0100H. When set to "0" by the user, the stack will roll over from SP = FFH to SP = 00H.
6	PWPO	PWM pin out selection Set to "1" by the user = PWM output pins selected as P3.4 and P3.3. Set to "0" by the user = PWM output pins selected as P2.6 and P2.7.
5	DBUF	DAC Output Buffer Set to "1" by the user = DAC Output Buffer Bypassed. Set to "0" by the user = DAC Output Buffer Enabled.
4	EXTCLK	Set by the user to "1" to select an external clock input on P3.4. Set by the user to "0" to use the internal PLL clock.
3	RSVD	Reserved – This bit should always contain 0.
2	RSVD	Reserved – This bit should always contain 0.
1	RSVD	Reserved – This bit should always contain 0.
0	XRAMEN	XRAM Enable Bit When set to "1" by the user, the internal XRAM will be mapped into the lower 2 kBytes of the external address space. When set to "0" by the user, the internal XRAM will not be accessible and the external data memory will be mapped into the lower 2 kBytes of external data memory.

USER INTERFACE TO OTHER ON-CHIP ADuC832 PERIPHERALS

The following 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 ADuC832 incorporates 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, DAC0H/L. It should be noted that in 12-bit asynchronous mode, the DAC voltage output will be 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: for correct DAC operation on the 0 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 Value	04H
Bit Addressable	No

Table IX. DACCON SFR Bit Designations

Bit	Name	Description
7	MODE	The DAC MODE bit sets the overriding operating mode for both DACs. Set to “1” = 8-Bit Mode (Write 8 Bits to DACxL SFR). Set to “0” = 12-Bit Mode.
6	RNG1	DAC1 Range Select Bit. Set to “1” = DAC1 Range 0–V_{DD}. Set to “0” = DAC1 Range 0–V_{REF}.
5	RNG0	DAC0 Range Select Bit. Set to “1” = DAC0 Range 0–V_{DD}. Set to “0” = DAC0 Range 0–V_{REF}.
4	CLR1	DAC1 Clear Bit. Set to “0” = DAC1 Output Forced to 0 V. Set to “1” = DAC1 Output Normal.
3	CLR0	DAC0 Clear Bit. Set to “0” = DAC1 Output Forced to 0 V. Set to “1” = DAC1 Output Normal.
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 will then update simultaneously when the SYNC bit is set to “1.”
1	$\overline{\text{PD1}}$	DAC1 Power-Down Bit. Set to “1” = Power-On DAC1. Set to “0” = Power-Off DAC1.
0	PD0	DAC0 Power-Down Bit. Set to “1” = Power-On DAC0. Set to “0” = Power-Off DAC0.

DACxH/L	**DAC Data Registers**		
Function	DAC Data Registers, written by user to update the DAC output.		
SFR Address	DAC0L (DAC0 Data Low Byte)	→ F9H; DAC1L (DAC1 Data Low Byte)	→ FBH
	DAC0H (DAC0 Data High Byte)	→ FAH; DAC1H(DAC1 Data High Byte)	→ FCH
Power-On Default Value	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 eight bits, and the lower nibble of DACxH contains the upper four bits.

Using the DAC

The on-chip DAC architecture consists of a resistor string DAC followed by an output buffer amplifier, the functional equivalent of which is illustrated in Figure 21. Details of the actual DAC architecture can be found in U.S. Patent Number 5969657 (www.uspto.gov). Features of this architecture include inherent guaranteed monotonicity and excellent differential linearity.

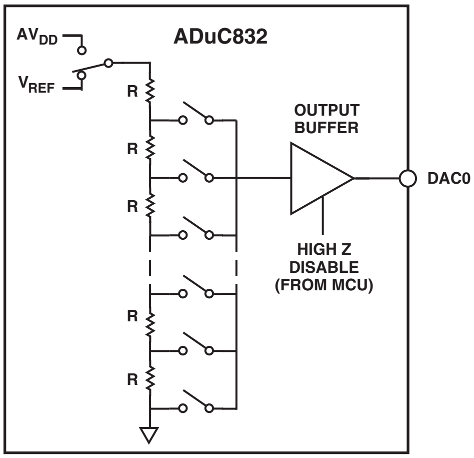


Figure 21. Resistor String DAC Functional Equivalent

As illustrated in Figure 21, the reference source for each DAC is user selectable in software. It can be either AV_{DD} or V_{REF}. In 0-to-AV_{DD} mode, the DAC output transfer function spans from 0 V to the voltage at the AV_{DD} pin. In 0-to-V_{REF} mode, the DAC output transfer function spans from 0 V to the internal V_{REF} or, if an external reference is applied, the voltage at the V_{REF} pin. The DAC output buffer amplifier features a true rail-to-rail output stage implementation. This means that, unloaded, each output is capable of swinging to within less than 100 mV of both AV_{DD} and ground. Moreover, the DAC's linearity specification (when driving a 10 kΩ resistive load to ground) is guaranteed through the full transfer function *except* codes 0 to 100, and, in 0-to-AV_{DD} mode only, codes 3995 to 4095. Linearity degradation near ground and V_{DD} is caused by saturation of the output amplifier, and a general representation of its effects (neglecting offset and gain error) is illustrated in Figure 22. The dotted line in Figure 22 indicates the *ideal* transfer function, and the solid line represents what the transfer function might look like with endpoint nonlinearities due to saturation of the output amplifier. Note that Figure 22 represents a transfer function in 0-to-V_{DD} mode only. In 0-to-V_{REF} mode (with $V_{REF} < V_{DD}$) the lower nonlinearity would be similar, but the upper portion of the transfer function would follow the "ideal" line right to the end (V_{REF} in this case, not V_{DD}), showing no signs of endpoint linearity errors.

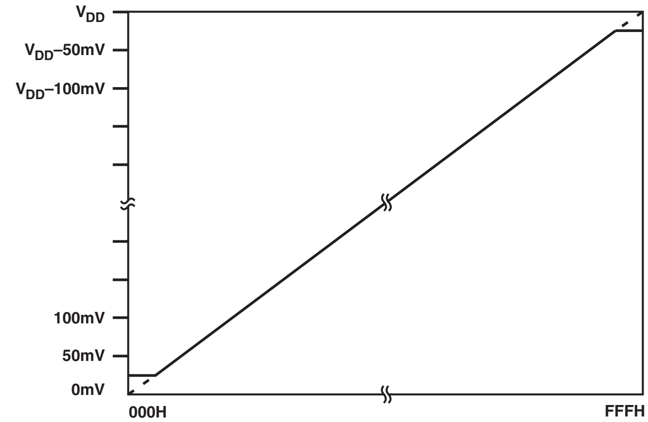


Figure 22. Endpoint Nonlinearities Due to Amplifier Saturation

The endpoint nonlinearities conceptually illustrated in Figure 22 get worse as a function of output loading. Most of the ADuC832's specifications assume a 10 kΩ resistive load to ground at the DAC output. As the output is forced to source or sink more current, the nonlinear regions at the top or bottom (respectively) of Figure 22 become larger. With larger current demands, this can significantly limit output voltage swing. Figures 23 and 24 illustrate this behavior. It should be noted that the upper trace in each of these figures is only valid for an output range selection of 0-to-AV_{DD}. In 0-to-V_{REF} mode, DAC loading will not cause highside voltage drops as long as the reference voltage remains below the upper trace in the corresponding figure. For example, if AV_{DD} = 3 V and V_{REF} = 2.5 V, the high side voltage will not be affected by loads less than 5 mA. But somewhere around 7 mA, the upper curve in Figure 24 drops below 2.5 V (V_{REF}), indicating that at these higher currents the output will not be capable of reaching V_{REF}.

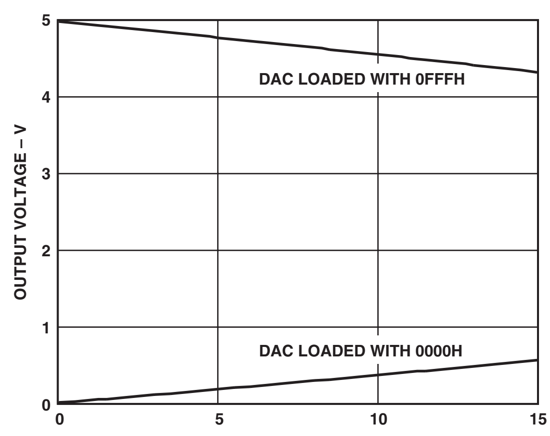


Figure 23. Source and Sink Current Capability with $V_{REF} = V_{DD} = 5$ V

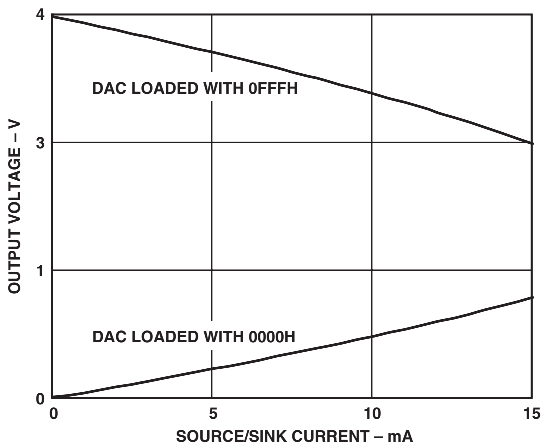


Figure 24. Source and Sink Current Capability with $V_{REF} = V_{DD} = 3$ V

To reduce the effects of the saturation of the output amplifier at values close to ground and to give reduced offset and gain errors, the internal buffer can be bypassed. This is done by setting the DBUF bit in the CFG832 register. This allows a full rail-to-rail output from the DAC, which should then be buffered externally using a dual supply op amp in order to get a rail-to-rail output. This external buffer should be located as near as physically possible to the DAC output pin on the PCB. Note that the unbuffered mode only works in the 0 to V_{REF} range.

To drive significant loads with the DAC outputs, external buffering may be required (even with the internal buffer enabled), as illustrated in Figure 25. A list of recommended op amps is in Table VI.

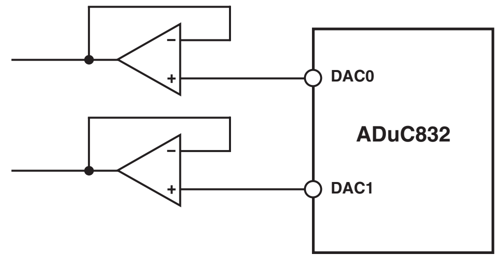


Figure 25. Buffering the DAC Outputs

The DAC output buffer also features a high impedance disable function. In the chip's default power-on state, both DACs are disabled, and their outputs are in a high impedance state (or "three-state") where they remain inactive until enabled in software. This means that if a zero output is desired during power-up or power-down transient conditions, then a pull-down resistor must be added to each DAC output. Assuming this resistor is in place, the DAC outputs will remain at ground potential whenever the DAC is disabled.

ON-CHIP PLL

The ADuC832 is 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 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 above choice of frequencies ensures that the modulators and the core will be synchronous, regardless of the core clock rate. The PLL control register is PLLCON.

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

Table X. PLLCON SFR Bit Designations

Bit	Name	Description
7	OSC_PD	Oscillator Power-Down Bit. Set by user to halt the 32 kHz oscillator in power-down mode. Cleared by 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 the PLL loop is correctly tracking the crystal clock. If the external crystal becomes subsequently disconnected, the PLL will rail and the core will halt. Cleared automatically at power-on to indicate 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 for future use; should be written with "0."
4	----	Reserved for future use; should be written with "0."
3	FINT	Fast Interrupt Response Bit Set by 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 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 will operate.
0	CD0	(see below)

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

PULSEWIDTH MODULATOR (PWM)

The PWM on the ADuC832 is a highly flexible PWM offering programmable resolution and an input clock, and can be configured for any one of six different modes of operation. Two of these modes allow the PWM to be configured as a Σ-Δ DAC with up to 16 bits of resolution. A block diagram of the PWM is shown in Figure 26.

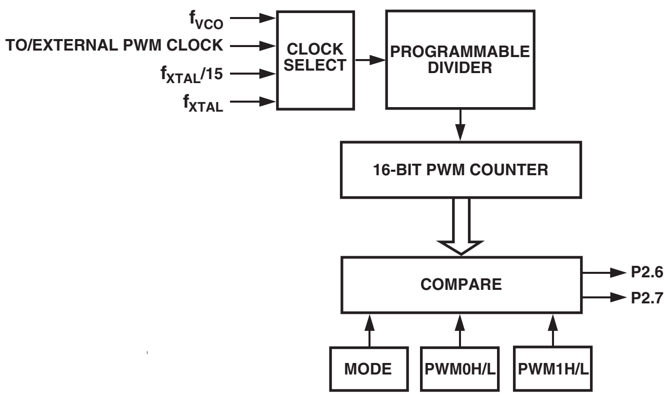


Figure 26. PWM Block Diagram

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

PWMCON (as described below) controls the different modes of operation of the PWM as well as the PWM clock frequency. PWM0H/L and PWM1H/L are the data registers that determine the duty cycles of the PWM outputs. The output pins that the PWM uses are determined by the CFG832 register, and can be either P2.6 and P2.7 or P3.4 and P3.3. In this section of the data sheet, it is assumed that P2.6 and P2.7 are selected as the PWM outputs.

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

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

Table XI. PWMCON SFR Bit Designations

Bit	Name	Description
7	SNGL	Turns off PWM Output at P2.6 or P3.4 Leaving Port Pin Free for Digital I/O.
6	MD2	PWM Mode Bits
5	MD1	The MD2/1/0 bits choose the PWM mode as follows:
4	MD0	MD2 MD1 MD0 Mode
		0 0 0 Mode 0: PWM Disabled
		0 0 1 Mode 1: Single variable resolution PWM on P2.7 or P3.3
		0 1 0 Mode 2: Twin 8-bit PWM
		0 1 1 Mode 3: Twin 16-bit PWM
		1 0 0 Mode 4: Dual NRZ 16-bit Σ-Δ DAC
		1 0 1 Mode 5: Dual 8-bit PWM
		1 1 0 Mode 6: Dual RZ 16-bit Σ-Δ DAC
		1 1 1 Reserved for future use
3	CDIV1	PWM Clock Divider
2	CDIV0	Scale the clock source for the PWM counter as shown below:
		CDIV1 CDIV0 Description
		0 0 PWM Counter = Selected Clock /1
		0 1 PWM Counter = Selected Clock /4
		1 0 PWM Counter = Selected Clock /16
		1 1 PWM Counter = Selected Clock /64
1	CSEL1	PWM Clock Divider
0	CSEL0	Select the clock source for the PWM as shown below:
		CSEL1 CSEL0 Description
		0 0 PWM Clock = $f_{XTAL}/15$
		0 1 PWM Clock = f_{XTAL}
		1 0 PWM Clock = External input at P3.4/T0
		1 1 PWM Clock = f_{VCO} = 16.777216 MHz

PWM MODES OF OPERATION

MODE 0: PWM Disabled

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

MODE 1: Single Variable Resolution PWM

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

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

PWM0H/L sets the duty cycle of the PWM output waveform, as shown in Figure 27.

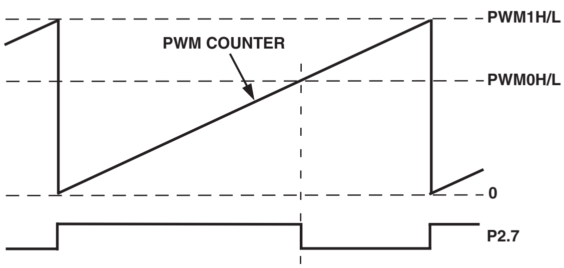


Figure 27. ADuC832 PWM in Mode 1

MODE 2: Twin 8-Bit PWM

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

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

The outputs of the PWM at P2.6 and P2.7 are shown in Figure 28. As can be seen, the output of PWM0 (P2.6) goes low when the PWM counter equals PWM0L. The output of PWM1 (P2.7) goes high when the PWM counter equals PWM1H and goes low again when the PWM counter equals PWM0H. Setting PWM1H to 0 ensures that both PWM outputs start simultaneously.

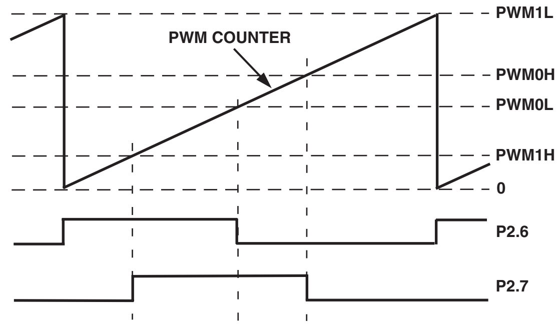


Figure 28. PWM Mode 2

MODE 3: Twin 16-Bit PWM

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

As shown in Figure 29, while the PWM counter is less than PWM0H/L, the output of PWM0 (P2.6) is high. Once the PWM counter equals PWM0H/L, PWM0 (P2.6) goes low and remains low until the PWM counter rolls over.

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

In this mode, both PWM outputs are synchronized, i.e., once the PWM counter rolls over to 0, both PWM0 (P2.6) and PWM1 (P2.7) will go high.

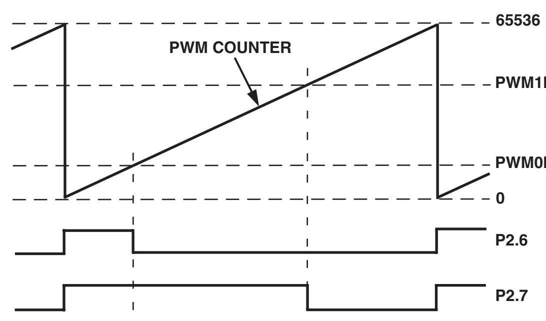
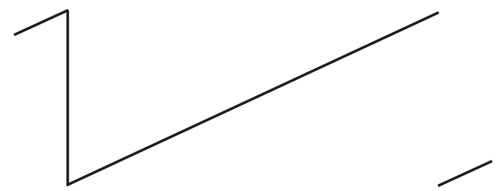


Figure 29. PWM Mode 3



MODE 4: Dual NRZ 16-Bit Σ-Δ DAC

Mode 4 provides a high speed PWM output similar to that of a Σ-Δ DAC. Typically, this mode will be 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 was set to 4010H (slightly above one quarter of FS) then typically P2.7 will be low for three clocks and high for one clock (each clock is approximately 60 ns). Over every 65536 clocks, the PWM will compensate 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.

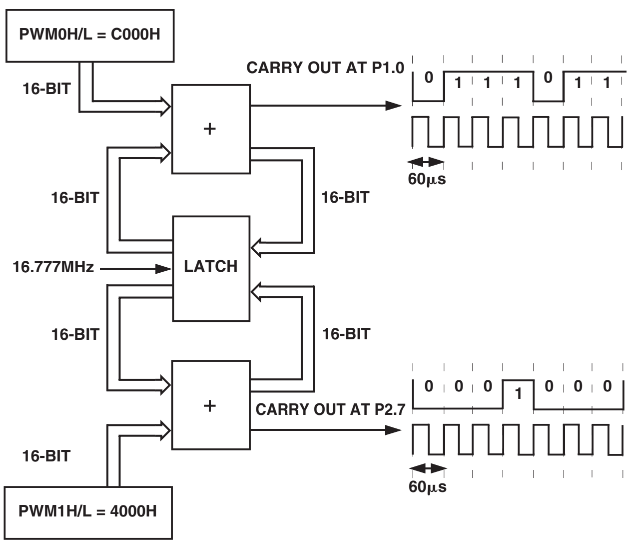


Figure 30. PWM Mode 4

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 then write 0s to the four LSBs. This means that a 12-bit accurate S-D DAC output can occur at 4.096 kHz. Similarly writing 0s to the eight LSBs gives an 8-bit accurate S-D 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 eight 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 same clock source and clock divider.

SERIAL PERIPHERAL INTERFACE

The ADuC832 integrates a complete hardware Serial Peripheral Interface (SPI) on-chip. SPI is an industry standard synchronous serial interface that allows eight bits of data to be synchronously transmitted and received simultaneously, i.e., full duplex. It should be noted that the SPI pins are shared with the I^2C pins. Therefore, the user can only enable one or the other interface at any given time (see SPE in Table XII). The SPI port can be configured for Master or Slave operation and typically consists of four pins, namely:

MISO (Master In, Slave Out Data I/O Pin)
The MISO (master in slave out) pin is configured as an input line in master mode and 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 (master out slave in) pin is configured as an output line in master mode and 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 XII). In slave mode the SPICON register will have to 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 the CPHA and CPOL are configured the same for the master and slave devices.

$\overline{SS}$ (Slave Select Input Pin)
The Slave Select ($\overline{SS}$) input pin is shared with the ADC5 input. In order to configure this pin as a digital input, the bit must be cleared, e.g., CLR P1.5.

This line is active low. Data is only received or transmitted in slave mode when the $\overline{SS}$ pin is low, allowing the ADuC832 to be used in single master, multislave SPI configurations. If CPHA = 1 then the $\overline{SS}$ input may be permanently pulled low. With 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 following SFR registers are used to control the SPI interface.

SPICON	**SPI Control Register**
SFR Address	F8H
Power-On Default Value	04H
Bit Addressable	Yes

Table XII. SPICON SFR Bit Designations

Bit	Name	Description
7	ISPI	SPI Interrupt Bit. Set by MicroConverter at the end of each SPI transfer. Cleared directly by user code or indirectly by reading the SPIDAT SFR.
6	WCOL	Write Collision Error Bit. Set by MicroConverter if SPIDAT is written to while an SPI transfer is in progress. Cleared by user code.
5	SPE	SPI Interface Enable Bit. Set by user to enable the SPI interface. Cleared by user to enable the I^2C pins.
4	SPIM	SPI Master/Slave Mode Select Bit. Set by user to enable Master Mode operation (SCLOCK is an output). Cleared by user to enable Slave Mode operation (SCLOCK is an input).
3	CPOL	Clock Polarity Select Bit. Set by user if SCLOCK idles high. Cleared by user if SCLOCK idles low.
2	CPHA	Clock Phase Select Bit. Set by user if leading SCLOCK edge is to transmit data. Cleared by user if trailing SCLOCK edge is to transmit data.
1	SPR1	SPI Bit-Rate Select Bits.
0	SPR0	These bits select the SCLOCK rate (bitrate) in master mode as follows: SPR1 SPR0 Selected Bit Rate 0 0 $f_{OSC}/2$ 0 1 $f_{OSC}/4$ 1 0 $f_{OSC}/8$ 1 1 $f_{OSC}/16$ In SPI Slave Mode, i.e., SPIM = 0, the logic level on the external $\overline{SS}$ pin can be read via the SPR0 bit.

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

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

Using the SPI Interface

Depending on the configuration of the bits in the SPICON SFR shown in Table XIII, the ADuC832 SPI interface will transmit or receive data in a number of possible modes. Figure 33 shows all possible ADuC832 SPI configurations 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.

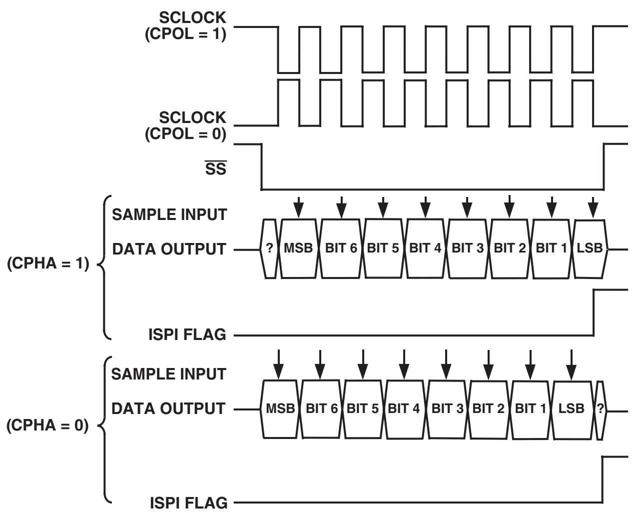


Figure 33. SPI Timing, All Modes

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. It should also be noted that the $\overline{SS}$ pin is not used in master mode. If the ADuC832 needs to assert the $\overline{SS}$ pin on an external slave device, a port digital output pin should be used.

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

SPI Interface—Slave Mode

In slave mode the SCLOCK is an input. The $\overline{SS}$ pin must also be driven low externally during the byte communication.

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

I^{2}C COMPATIBLE INTERFACE

The ADuC832 supports a fully licensed* I^{2}C serial interface. The I^{2}C interface is implemented as a full hardware slave and software master. SDATA is the data I/O pin and SCLOCK is the serial clock. These two pins are shared with the MOSI and SCLOCK pins of the on-chip SPI interface. Therefore, the user can only enable one or the other interface at any given time (see SPE in SPICON previously). Application Note uC001 describes the operation of this interface as implemented is available from the MicroConverter website at www.analog.com/microconverter.

Three SFRs are used to control the I^{2}C interface. These are described below:

I2CCON	**I^{2}C Control Register**
SFR Address	E8H
Power-On Default Value	00H
Bit Addressable	Yes

Table XIII. I2CCON SFR Bit Designations

Bit	Name	Description
7	MDO	I^{2}C Software Master Data Output Bit (Master Mode Only). This data bit is used to implement a master I^{2}C transmitter interface in software. Data written to this bit will be output on the SDATA pin if the data output enable (MDE) bit is set.
6	MDE	I^{2}C Software Master Data Output Enable Bit (Master Mode Only). Set by user to enable the SDATA pin as an output (Tx). Cleared by the user to enable SDATA pin as an input (Rx).
5	MCO	I^{2}C Software Master Clock Output Bit (Master Mode Only). This data bit is used to implement a master I^{2}C transmitter interface in software. Data written to this bit will be output on the SCLOCK pin.
4	MDI	I^{2}C Software Master Data Input Bit (Master Mode Only). This data bit is used to implement a master I^{2}C receiver interface in software. Data on the SDATA pin is latched into this bit on SCLOCK if the Data Output Enable (MDE) bit is "0."
3	I2CM	I^{2}C Master/Slave Mode Bit Set by user to enable I^{2}C software master mode. Cleared by user to enable I^{2}C hardware slave mode.
2	I2CRS	I^{2}C Reset Bit (Slave Mode Only). Set by user to reset the I^{2}C interface. Cleared by user code for normal I^{2}C operation.
1	I2CTX	I^{2}C Direction Transfer Bit (Slave Mode Only). Set by the MicroConverter if the interface is transmitting. Cleared by the MicroConverter if the interface is receiving.
0	I2CI	I^{2}C Interrupt Bit (Slave Mode Only). Set by the MicroConverter after a byte has been transmitted or received. Cleared automatically when user code reads the I2CDAT SFR (see I2CDAT below).

I2CADD	**I^{2}C Address Register**
Function	Holds the I^{2}C peripheral address for the part. It may be overwritten by user code. Technical Note uC001 at www.analog.com/microconverter describes the format of the I^{2}C standard 7-bit address in detail.
SFR Address	9BH
Power-On Default Value	55H
Bit Addressable	No

I2CDAT	**I^{2}C Data Register**
Function	The I2CDAT SFR is written by the user to transmit data over the I^{2}C interface or read by user code to read data just received by the I^{2}C interface. Accessing I2CDAT automatically clears any pending I^{2}C interrupt and the I2CI bit in the I2CCON SFR. User software should only access I2CDAT once per interrupt cycle.
SFR Address	9AH
Power-On Default Value	00H
Bit Addressable	No

*Purchase of licensed I^{2}C components of Analog Devices or one of its sublicensed associated companies conveys a license for the purchaser under the Philips I^{2}C Patent Rights to use the ADuC832 in an I^{2}C system, provided that the system conforms to the I^{2}C Standard Specification as defined by Philips.

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

- Only two bus lines are required; a serial data line (SDATA) and a serial clock line (SCLOCK).
- An I^{2}C master can communicate with multiple slave devices. Because each slave device has a unique 7-bit address, single master/slave relationships can exist at all times even in a multislave environment (Figure 34).
- On-chip filtering rejects <50 ns spikes on the SDATA and the SCLOCK lines to preserve data integrity.

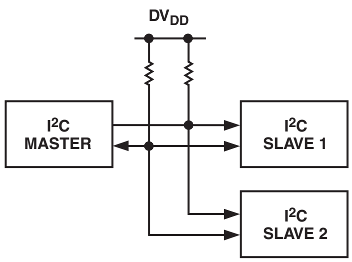


Figure 34. Typical I^{2}C System

Software Master Mode

The ADuC832 can be used as an I^{2}C master device by configuring the I^{2}C peripheral in master mode and writing software to output the data bit by bit. This is referred to as a software master. Master mode is enabled by setting the I2CM bit in the I2CCON register.

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

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

Software must control MDO, MCO and MDE appropriately to generate the START condition, slave address, acknowledge bits, data bytes, and STOP conditions appropriately. These functions are provided in technical note uC001.

Hardware Slave Mode

After reset the ADuC832 defaults to hardware slave mode. The I^{2}C interface is enabled by clearing the SPE bit in SPICON. Slave mode is enabled by clearing the I2CM bit in I2CCON. The ADuC832 has a full hardware slave. In slave mode the I^{2}C address is stored in the I2CADD register. Data received or to be transmitted is stored in the I2CDAT register.

Once enabled in I^{2}C slave mode the slave controller waits for a START condition. If the ADuC832 detects a valid start condition, followed by a valid address, followed by the R/$\overline{W}$ bit, the I2CI interrupt bit will automatically be set by hardware.

The I^{2}C peripheral will only generate a core interrupt if the user has preconfigured the I^{2}C interrupt enable bit in the IEIP2 SFR, as well as the global interrupt bit EA in the IE SFR.

```
; Enabling I2C Interrupts for the ADuC832
MOV  IEIP2,#01H     ; enable I2C interrupt
SETB EA
```

On the ADuC832 an autoclear of the I2CI bit is implemented so this bit is cleared automatically on a read or write access to the I2CDAT SFR.

```
MOV     I2CDAT, A   ; I2CI autocleared
MOV     A, I2CDAT   ; I2CI autocleared
```

If for any reason the user tries to clear the interrupt more than once i.e., access the data SFR more than once per interrupt then the I^{2}C controller will halt. The interface will then have to be reset using the I2CRS bit.

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

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

Once the ADuC832 has received a valid address, hardware will hold SCLOCK low until the I2CI bit is cleared by software. This allows the master to wait for the slave to be ready before transmitting the clocks for the next byte.

The I2CI interrupt bit will be set every time a complete data byte is received or transmitted, provided it is followed by a valid ACK. If the byte is followed by a NACK an interrupt is NOT generated. The ADuC832 will continue to issue interrupts for each complete data byte transferred until a STOP condition is received or the interface is reset.

When a STOP condition is received, the interface will reset to a state where it is waiting to be addressed (idle). Similarly, if the interface receives a NACK at the end of a sequence it also returns to the default idle state. The I2CRS bit can be used to reset the I^{2}C interface. This bit can be used to force the interface back to the default idle state.

It should be noted that there is no way (in hardware) to distinguish between an interrupt generated by a received START + valid address and an interrupt generated by a received data byte. User software must be used to distinguish between these interrupts.

DUAL DATA POINTER

The ADuC832 incorporates two data pointers. The second data pointer is a shadow data pointer and is selected via the data pointer control SFR (DPCON). DPCON also includes some nice features such as automatic hardware post-increment and post-decrement as well as automatic data pointer toggle. DPCON is described in Table XIV.

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

Table XIV. DPCON SFR Bit Designations

Bit	Name	Description
7	----	Reserved for Future Use.
6	DPT	Data Pointer Automatic Toggle Enable. Cleared by user to disable auto swapping of the DPTR. Set in user software to enable automatic toggling of the DPTR after each each MOVX or MOVC instruction.
5	DP1m1	Shadow Data Pointer Mode.
4	DP1m0	These two bits enable extra modes of the shadow data pointer operation, allowing for more compact and more efficient code size and execution. m1 m0 Behavior of the Shadow Data Pointer 0 0 8052 Behavior 0 1 DPTR is post-incremented after a MOVX or a MOVC instruction. 1 0 DPTR is post-decremented after a MOVX or MOVC instruction. 1 1 DPTR LSB is toggled after a MOVX or MOVC instruction. (This instruction can be useful for moving 8-bit blocks to/from 16-bit devices.)
3	DP0m1	Main Data Pointer Mode.
2	DP0m0	These two bits enable extra modes of the main data pointer operation, allowing for more compact and more efficient code size and execution. m1 m0 Behavior of the Main Data Pointer 0 0 8052 Behavior 0 1 DPTR is post-incremented after a MOVX or a MOVC instruction. 1 0 DPTR is post-decremented after a MOVX or MOVC instruction. 1 1 DPTR LSB is toggled after a MOVX or MOVC instruction. (This instruction can be useful for moving 8-bit blocks to/from 16-bit devices.)
1	----	This bit is not implemented to allow the INC DPCON instruction toggle the data pointer without incrementing the rest of the SFR.
0	DPSEL	Data Pointer Select. Cleared by user to select the main data pointer. This means that the contents of this 24-bit register are placed into the three SFRs DPL, DPH, and DPP. Set by the user to select the shadow data pointer. This means that the contents of a separate 24-bit register appears in the three SFRs DPL, DPH, and DPP.

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

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

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

The following code uses 16 bytes and 2054 cycles. To perform this on a standard 8051 requires approximately 33 bytes and 7172 cycles (depending on how it is implemented).

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

POWER SUPPLY MONITOR

As its name suggests, the Power Supply Monitor, once enabled, monitors the DV_{DD} supply on the ADuC832. It will indicate when any of the supply pins drop below one of four user-selectable voltage trip points from 2.63 V to 4.37 V. For correct operation of the Power Supply Monitor function, AV_{DD} must be equal to or greater than 2.7 V. Monitor function is controlled via the PSMCON SFR. If enabled via the IEIP2 SFR, the monitor will interrupt the core using the PSMI bit in the PSMCON SFR. This bit will not be cleared until the failing power supply has returned above the trip point for at least 250 ms. This monitor function allows the user to save working registers to avoid possible data loss due to the low supply condition, and also ensures that normal code execution will not resume until a safe supply level has been well established. The supply monitor is also protected against spurious glitches triggering the interrupt circuit.

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

Table XV. PSMCON SFR Bit Designations

Bit	Name	Description
7	----	Reserved.
6	CMPD	DV_{DD} Comparator Bit. This is a read-only bit and directly reflects the state of the DV_{DD} comparator. Read "1" indicates the DV_{DD} supply is above its selected trip point. Read "0" indicates the DV_{DD} supply is below its selected trip point.
5	PSMI	Power Supply Monitor Interrupt Bit. This bit will be set high by the MicroConverter if either CMPA or CMPD is low, indicating low analog or digital supply. The PSMI bit can be used to interrupt the processor. Once CMPD and/or CMPA return (and remain) high, a 250 ms counter is started. When this counter times out, the PSMI interrupt is cleared. PSMI can also be written by the user. However, if either comparator output is low, it is not possible for the user to clear PSMI.
4	TPD1	DV_{DD} Trip Point Selection Bits.
3	TPD0	These bits select the DV_{DD} trip point voltage as follows: TPD1 TPD0 Selected DV_{DD} Trip Point (V) 0 0 4.37 0 1 3.08 1 0 2.93 1 1 2.63
2	----	Reserved
1	----	Reserved
0	PSMEN	Power Supply Monitor Enable Bit. Set to "1" by the user to enable the Power Supply Monitor Circuit. Cleared to "0" by the user to disable the Power Supply Monitor Circuit.

WATCHDOG TIMER

The purpose of the watchdog timer is to generate a device reset or interrupt within a reasonable amount of time if the ADuC832 enters an erroneous state, possibly due to a programming error or electrical noise. The watchdog function can be disabled by clearing the WDE (Watchdog Enable) bit in the Watchdog Control (WDCON) SFR. When enabled, the watchdog circuit will generate a system reset or interrupt (WDS) if the user program fails to set the watchdog (WDE) bit within a predetermined amount of time (see PRE3–0 bits in WDCON). The watchdog timer itself is a 16-bit counter that is clocked directly from the 32.768 kHz external crystal. The watchdog time out interval can be adjusted via the PRE3–0 bits in WDCON. Full control and status of the watchdog timer function can be controlled via the watchdog timer control SFR (WDCON). The WDCON SFR can only be written by user software if the double write sequence described in WDWR below is initiated on every write access to the WDCON SFR.

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

Table XVI. WDCON SFR Bit Designations

Bit	Name	Description
7	PRE3	Watchdog Timer Prescale Bits.
6	PRE2	The Watchdog timeout period is given by the equation: $t_{WD} = (2^{PRE} \times (2^9/f_{XTAL}))$
5	PRE1	($0 \leq PRE \leq 7$; f_{XTAL} = 32.768 kHz)
4	PRE0	

PRE3	PRE2	PRE1	PRE0	Timeout Period (ms)	Action
0	0	0	0	15.6	Reset or Interrupt
0	0	0	1	31.2	Reset or Interrupt
0	0	1	0	62.5	Reset or Interrupt
0	0	1	1	125	Reset or Interrupt
0	1	0	0	250	Reset or Interrupt
0	1	0	1	500	Reset or Interrupt
0	1	1	0	1000	Reset or Interrupt
0	1	1	1	2000	Reset or Interrupt
1	0	0	0	0.0	Immediate Reset
PRE3–0 > 1000					Reserved

Bit	Name	Description
3	WDIR	Watchdog Interrupt Response Enable Bit. If this bit is set by the user, the watchdog will generate an interrupt response instead of a system reset when the watchdog timeout period has expired. This interrupt is not disabled by the CLR EA instruction and it is also a fixed, high priority interrupt. If the watchdog is not being used to monitor the system, it can alternatively be used as a timer. The prescaler is used to set the timeout period in which an interrupt will be generated.
2	WDS	Watchdog Status Bit. Set by the Watchdog Controller to indicate that a watchdog timeout has occurred. Cleared by writing a "0" or by an external hardware reset. It is not cleared by a watchdog reset.
1	WDE	Watchdog Enable Bit. Set by user to enable the watchdog and clear its counters. If this bit is not set by the user within the watchdog timeout period, the watchdog will generate a reset or interrupt, depending on WDIR. Cleared under the following conditions: User writes "0," Watchdog Reset (WDIR = "0"); Hardware Reset; PSM Interrupt.
0	WDWR	Watchdog Write Enable Bit. To write data into the WDCON SFR involves a double instruction sequence. The WDWR bit must be set and the very next instruction must be a write instruction to the WDCON SFR. For example:

```
CLR  EA          ;disable interrupts while writing
                  ;to WDT
SETB WDWR        ;allow write to WDCON
MOV  WDCON, #72H ;enable WDT for 2.0s timeout
SETB EA          ;enable interrupts again (if rqd)
```

TIME INTERVAL COUNTER (TIC)

A time interval counter is provided on-chip for counting longer intervals than the standard 8051 compatible timers are capable of. The TIC is capable of timeout intervals ranging from 1/128 second to 255 hours. Furthermore, this counter is clocked by the external 32.768 kHz crystal rather than the core clock and has the ability to remain active in power-down mode and time long power-down intervals. This has obvious applications for remote battery-powered sensors where regular widely spaced readings are required. Note: Instructions to the TIC SFRs are also clocked at 32.768 kHz, sufficient time must be allowed for in user code for these instructions to execute.

Six SFRs are associated with the time interval counter, TIMECON being its control register. Depending on the configuration of the IT0 and IT1 bits in TIMECON, the selected time counter register overflow will clock the interval counter. When this counter is equal to the time interval value loaded in the INTVAL SFR, the TII bit (TIMECON.2) is set and generates an interrupt if enabled. If the ADuC832 is in power-down mode, again with TIC interrupt enabled, the TII bit will wake up the device and resume code execution by vectoring directly to the TIC interrupt service vector address at 0053H. The TIC-related SFRs are described below. Note also that the timebase SFRs can be written initially with the current time; the TIC can then be controlled and accessed by user software. In effect, this facilitates the implementation of a real-time clock. A block diagram of the TIC is shown in Figure 35.

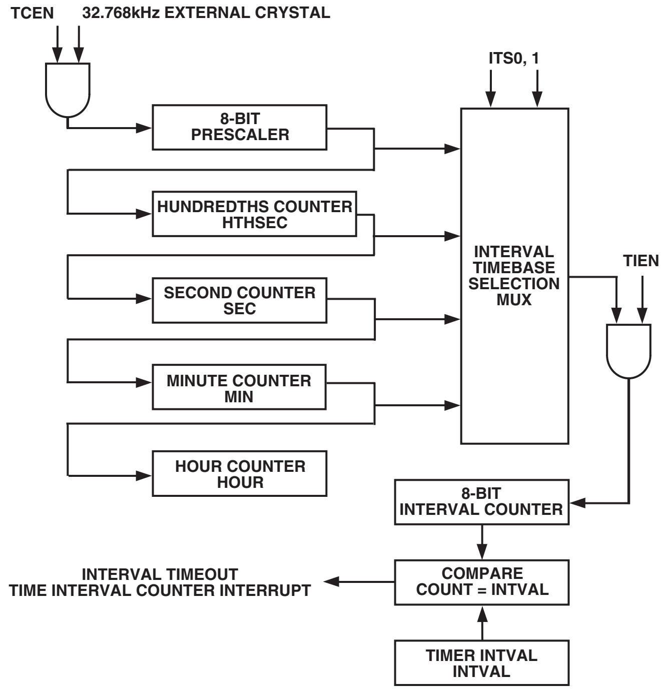


Figure 35. TIC, Simplified Block Diagram

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

Table XVII. TIMECON SFR Bit Designations

Bit	Name	Description
7	----	Reserved for Future Use.
6	TFH	Twenty-Four Hour Select Bit.
		Set by the user to enable the Hour counter to count from 0 to 23.
		Cleared by the user to enable the Hour counter to count from 0 to 255.
5	ITS1	Interval Timebase Selection Bits.
4	ITS0	Written by user to determine the interval counter update rate.
		ITS1 ITS0 Interval Timebase
		0 0 1/128 Second
		0 1 Seconds
		1 0 Minutes
		1 1 Hours
3	STI	Single Time Interval Bit.
		Set by the thberva0.0.5932 0 allow.8(Written by user to 0 b6 Tutom.)-cally 4load Tiand start user 0.0terval T6847

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 Value	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 Value	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 Value	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 counts from 0 to 59 before rolling over to increment the HOUR time register
SFR Address	A4H
Power-On Default Value	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 Value	00H
Bit Addressable	No
Valid Value	0 to 23 decimal

8052 COMPATIBLE ON-CHIP PERIPHERALS

This section gives a brief overview of the various secondary peripheral circuits that are also available to the user on-chip. These remaining functions are mostly 8052 compatible (with a few additional features) and are controlled via standard 8052 SFR bit definitions.

Parallel I/O

The ADuC832 uses four input/output ports to exchange data with external devices. In addition to performing general-purpose I/O, some ports are capable of external memory operations while others are multiplexed with alternate functions for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general-purpose I/O pin.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port that is directly controlled via the Port 0 SFR. Port 0 is also the multiplexed low order address and data bus during accesses to external program or data memory.

Figure 36 shows a typical bit latch and I/O buffer for a Port 0 port pin. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal. See the following Read-Modify-Write Instructions section for more details.

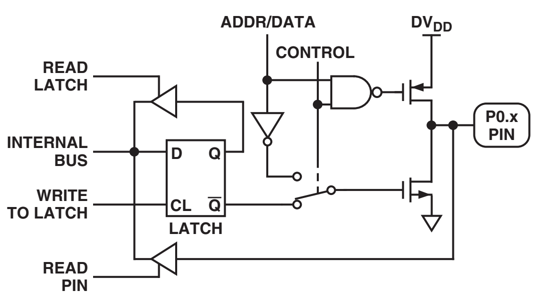


Figure 36. Port 0 Bit Latch and I/O Buffer

As shown in Figure 36, the output drivers of Port 0 pins are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P0 SFR gets 1s written to it (i.e., all of its bit latches become 1). When accessing external memory, the CONTROL signal in Figure 36 goes high, enabling push-pull operation of the output pin from the internal address or data bus (ADDR/DATA line). Therefore, no external pull-ups are required on Port 0 in order for it to access external memory.

In general-purpose I/O port mode, Port 0 pins that have 1s written to them via the Port 0 SFR will be configured as "open drain" and will therefore float. In this state, Port 0 pins can be used as high impedance inputs. This is represented in Figure 36 by the NAND gate whose output remains high as long as the CONTROL signal is low, thereby disabling the top FET. External pull-up resistors are therefore required when Port 0 pins are used as general-purpose outputs. Port 0 pins with 0s written to them will drive a logic low output voltage (V_{OL}) and will be capable of sinking 1.6 mA.

Port 1

Port 1 is also an 8-bit port directly controlled via the P1 SFR. Port 1 digital output capability is not supported on this device. Port 1 pins can be configured as digital inputs or analog inputs.

By (power-on) default, these pins are configured as analog inputs, i.e., "1" written in the corresponding Port 1 register bit. To configure any of these pins as digital inputs, the user should write a "0" to these port bits to configure the corresponding pin as a high impedance digital input.

These pins also have various secondary functions described in Table XVIII.

Table XVIII. Port 1, Alternate Pin Functions

Pin	Alternate Function
P1.0	T2 (Timer/Counter 2 External Input)
P1.1	T2EX (Timer/Counter 2 Capture/Reload Trigger)
P1.5	$\overline{\text{SS}}$ (Slave Select for the SPI Interface)

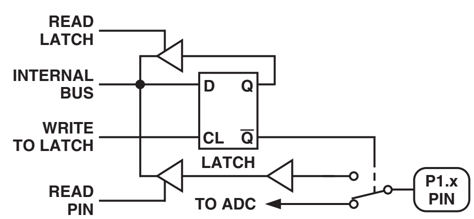


Figure 37. Port 1 Bit Latch and I/O Buffer

Port 2

Port 2 is a bidirectional port with internal pull-up resistors directly controlled via the P2 SFR. Port 2 also emits the high order address bytes during fetches from external program memory and middle and high order address bytes during accesses to the 24-bit external data memory space.

As shown in Figure 38, the output drivers of Ports 2 are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses (as for Port 0). In external memory addressing mode (CONTROL = 1), the port pins feature push-pull operation controlled by the internal address bus (ADDR line). However, unlike the P0 SFR during external memory accesses, the P2 SFR remains unchanged.

In general-purpose I/O port mode, Port 2 pins that have 1s written to them are pulled high by the internal pull-ups (Figure 39) and, in that state, can be used as inputs. As inputs, Port 2 pins being pulled externally low will source current because of the internal pull-up resistors. Port 2 pins with 0s written to them will drive a logic low output voltage (V_{OL}) and will be capable of sinking 1.6 mA.

P2.6 and P2.7 can also be used as PWM outputs. In the case that they are selected as the PWM outputs via the CFG832 SFR, the PWM outputs will overwrite anything written to P2.6 or P2.7.

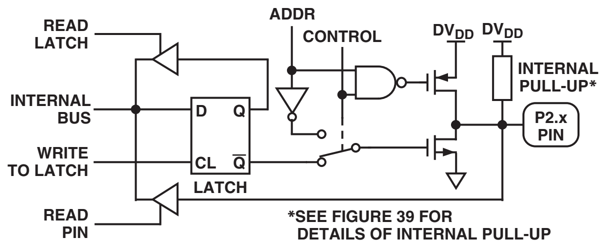


Figure 38. Port 2 Bit Latch and I/O Buffer

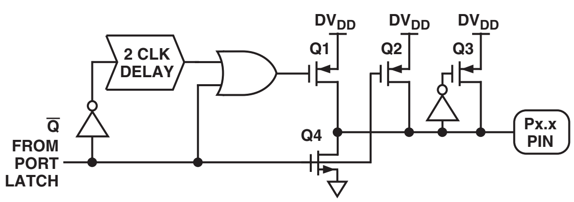


Figure 39. Internal Pull-Up Configuration

Port 3

Port 3 is a bidirectional port with internal pull-ups directly controlled via the P3 SFR. Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and, in that state, can be used as inputs. As inputs, Port 3 pins being pulled externally low will source current because of the internal pull-ups. Port 3 pins with 0s written to them will drive a logic low output voltage (V_{OL}) and will be capable of sinking 4 mA.

Port 3 pins also have various secondary functions described in Table XIX. The alternate functions of Port 3 pins can only be activated if the corresponding bit latch in the P3 SFR contains a 1. Otherwise, the port pin is stuck at 0.

Table XIX. Port 3, Alternate Pin Functions

Pin	Alternate Function
P3.0	RxD (UART Input Pin)(or Serial Data I/O in Mode 0)
P3.1	TxD (UART Output Pin) (or Serial Clock Output in Mode 0)
P3.2	$\overline{\text{INT0}}$ (External Interrupt 0)
P3.3	$\overline{\text{INT1}}$ (External Interrupt 1)/PWM 1/MISO
P3.4	T0 (Timer/Counter 0 External Input) PWM External Clock/PWM 0
P3.5	T1 (Timer/Counter 1 External Input)
P3.6	$\overline{\text{WR}}$ (External Data Memory Write Strobe)
P3.7	$\overline{\text{RD}}$ (External Data Memory Read Strobe)

P3.3 and P3.4 can also be used as PWM outputs. In the case that they are selected as the PWM outputs via the CFG832 SFR, the PWM outputs will overwrite anything written to P3.4 or P3.3.

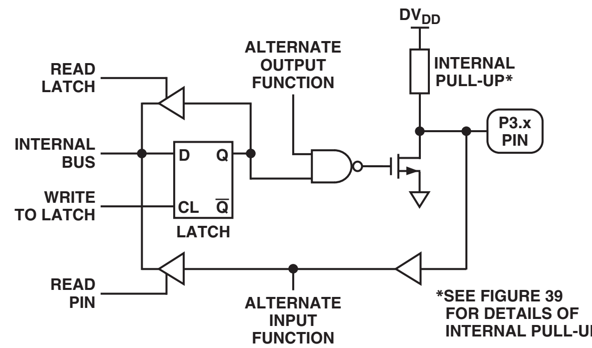


Figure 40. Port 3 Bit Latch and I/O Buffer

Additional Digital I/O

In addition to the port pins, the dedicated SPI/I^{2}C pins (SCLOCK and SDATA/MOSI) also feature both input and output functions. Their equivalent I/O architectures are illustrated in Figure 41 and Figure 43, respectively, for SPI operation and in Figure 42 and Figure 44 for I^{2}C operation.

Notice that in I^{2}C mode (SPE = 0), the strong pull-up FET (Q1) is disabled, leaving only a weak pull-up (Q2) present. By contrast, in SPI mode (SPE = 1) the strong pull-up FET (Q1) is controlled directly by SPI hardware, giving the pin push-pull capability.

In I^{2}C mode (SPE = 0), two pull-down FETs (Q3 and Q4) operate in parallel in order to provide an extra 60% or 70% of current sinking capability. In SPI mode, however, (SPE = 1) only one of the pull-down FETs (Q3) operates on each pin resulting in sink capabilities identical to that of Port 0 and Port 2 pins.

On the input path of SCLOCK, notice that a Schmitt trigger conditions the signal going to the SPI hardware to prevent false triggers (double triggers) on slow incoming edges. For incoming signals from the SCLOCK and SDATA pins going to I^{2}C hardware, a filter conditions the signals in order to reject glitches of up to 50 ns in duration.

Notice also that direct access to the SCLOCK and SDATA/MOSI pins is afforded through the SFR interface in I^{2}C master mode. Therefore, if you are not using the SPI or I^{2}C functions, you can use these two pins to give additional high current digital outputs.

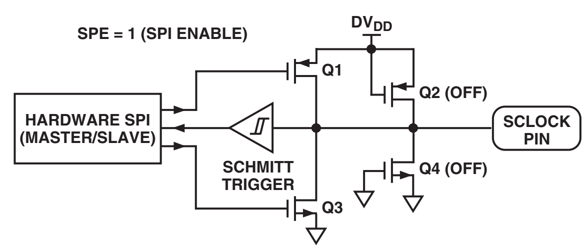


Figure 41. SCLOCK Pin I/O Functional Equivalent in SPI Mode

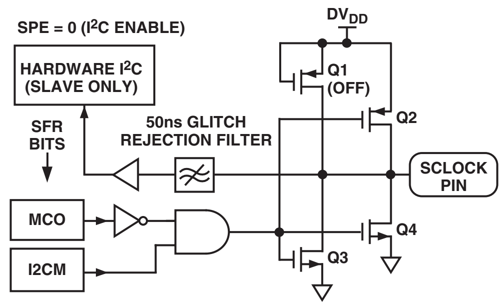


Figure 42. SCLOCK Pin I/O Functional Equivalent in I^2C Mode

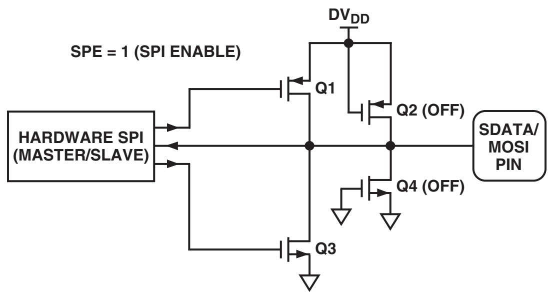


Figure 43. SDATA/MOSI Pin I/O Functional Equivalent in SPI Mode

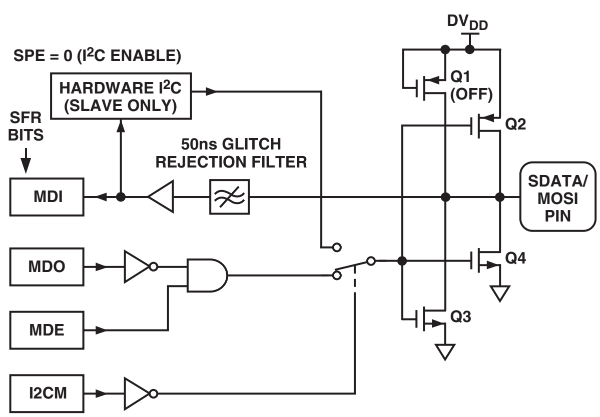


Figure 44. SDATA/MOSI Pin I/O Functional Equivalent in I^2C Mode

MISO is shared with P3.3 and as such has the same configuration as that shown in Figure 40.

Read-Modify-Write Instructions

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

ANL	(Logical AND, e.g., ANL P1, A)
ORL	(Logical OR, e.g., ORL P2, A)
XRL	(Logical EX-OR, e.g., XRL P3, A)
JBC	(Jump if bit = 1 and clear bit, e.g., JBC P1.1, LABEL)
CPL	(Complement bit, e.g., CPL P3.0)
INC	(increment, e.g., INC P2)
DEC	(Decrement, e.g., DEC P2)
DJNZ	(Decrement and jump if not zero, e.g., DJNZ P3, LABEL)
MOV PX.Y, C*	(Move carry to Bit Y of Port X)
CLR PX.Y*	(Clear Bit Y of Port X)
SETB PX.Y*	(Set Bit Y of Port X)

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

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

Timers/Counters

The ADuC832 has three 16-bit Timer/Counters: Timer 0, Timer 1, and Timer 2. The Timer/Counter hardware has been 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 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 consists of 12 core clock periods, the maximum count rate is 1/12 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. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes two machine cycles (24 core clock periods) to recognize a 1-to-0 transition, the maximum count rate is 1/24 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 Value 00H

Bit Addressable No

Table XX. TMOD SFR Bit Designations

Bit	Name	Description
7	Gate	Timer 1 Gating Control. Set by software to enable Timer/Counter 1 only while $\overline{INT1}$ pin is high and TR1 control bit is set. Cleared by software to enable Timer 1 whenever 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 Auto-Reload 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 $\overline{INT0}$ pin is high and TR0 control bit is set. Cleared by software to enable Timer 0 whenever 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 Auto-Reload Timer/Counter. TH0 holds a value that is to be reloaded into TL0 each time it overflows. 1 1 TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only, controlled by Timer 1 control bits.

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

Table XXI. TCON SFR Bit Designations

Bit	Name	Description
7	TF1	Timer 1 Overflow Flag. Set by hardware on a Timer/Counter 1 overflow. Cleared by hardware when the Program Counter (PC) vectors to the interrupt service routine.
6	TR1	Timer 1 Run Control Bit. Set by the user to turn on Timer/Counter 1. Cleared by the user to turn off Timer/Counter 1.
5	TF0	Timer 0 Overflow Flag. Set by hardware on a Timer/Counter 0 overflow. Cleared by hardware when the PC vectors to the interrupt service routine.
4	TR0	Timer 0 Run Control Bit. Set by the user to turn on Timer/Counter 0. Cleared by the user to turn off Timer/Counter 0.
3	IE1*	External Interrupt 1 ($\overline{INT1}$) Flag. Set by hardware by a falling edge or zero level being applied to external interrupt Pin $\overline{INT1}$, depending on bit IT1 state. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
2	IT1*	External Interrupt 1 (IE1) Trigger Type. Set by software to specify edge-sensitive detection (i.e., 1-to-0 transition). Cleared by software to specify level-sensitive detection (i.e., zero level).
1	IE0*	External Interrupt 0 ($\overline{INT0}$) Flag. Set by hardware by a falling edge or zero level being applied to external interrupt Pin $\overline{INT0}$, depending on bit IT0 state. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
0	IT0*	External Interrupt 0 (IE0) Trigger Type. Set by software to specify edge-sensitive detection (i.e., 1-to-0 transition). Cleared by software to specify level-sensitive detection (i.e., zero level).

*These bits are not used in the control of timer/counter 0 and 1, but are used instead in the control and monitoring of the external INT0 and INT1 interrupt pins.

Timer/Counter 0 and 1 Data Registers
Each timer consists of two 8-bit registers. These can be used as independent registers or combined to be a single 16-bit register depending on the timer mode configuration.

TH0 and TL0
Timer 0 high byte and low byte.
SFR Address = 8CH, 8AH, respectively.

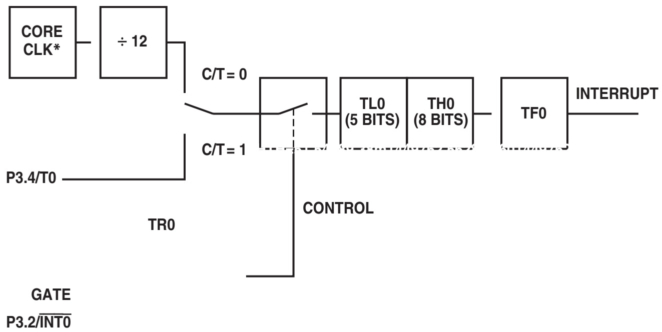
TH1 and TL1
Timer 1 high byte and low byte.
SFR Address = 8DH, 8BH, respectively.

TIMER/COUNTER 0 AND 1 OPERATING MODES

The following paragraphs describe the operating modes for Timer/Counters 0 and 1. Unless otherwise noted, it should be assumed that these modes of operation are the same for Timer 0 as for Timer 1.

Mode 0 (13-Bit Timer/Counter)

Mode 0 configures an 8-bit timer/counter with a divide-by-32 prescaler. Figure 45 shows Mode 0 operation.



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

Table XXII. T2CON SFR Bit Designations

Bit	Name	Description
7	TF2	Timer 2 Overflow Flag. Set by hardware on a Timer 2 overflow. TF2 will not 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 timer capture/reload registers.

TH2 and TL2

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

RCAP2H and RCAP2L

Timer 2, Capture/Reload byte and low byte.
SFR Address = CBH, CAH respectively.

Timer/Counter Operation Modes
The following paragraphs describe the operating modes for Timer/Counter 2. The operating modes are selected by bits in the T2CON SFR as shown in Table XXIII.

Table XXIII. 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
In Autoreload mode, there are two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still performs the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2. The Autoreload mode is illustrated in Figure 49.

16-Bit Capture Mode
In the Capture mode, there are again two options, which 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. The Capture mode is illustrated in Figure 50.

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1.

In either case, if Timer 2 is being used to generate the baud rate, the TF2 interrupt flag will not occur. Therefore, Timer 2 interrupts will not occur so they do not have to be disabled. In this mode the EXF2 flag, however, can still cause interrupts and this can be used as a third external interrupt.

Baud rate generation will be described as part of the UART serial port operation in the following pages.

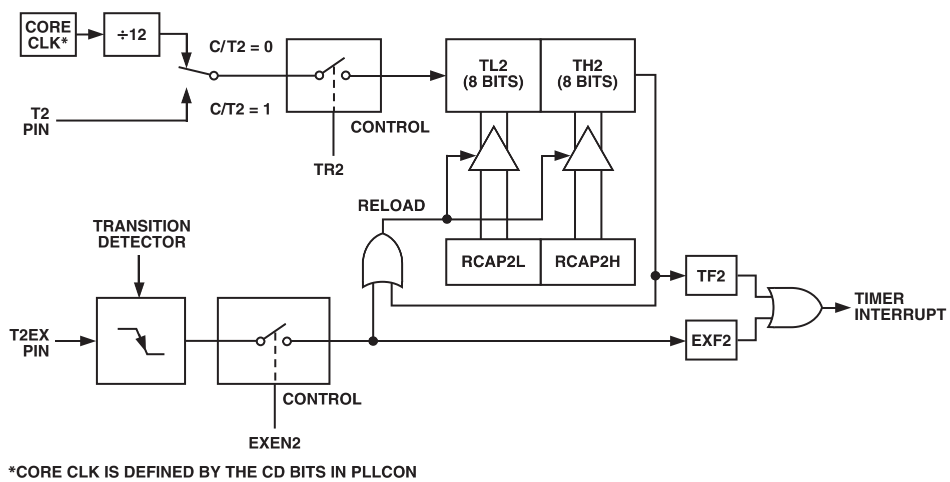


Figure 49. Timer/Counter 2, 16-Bit Autoreload Mode

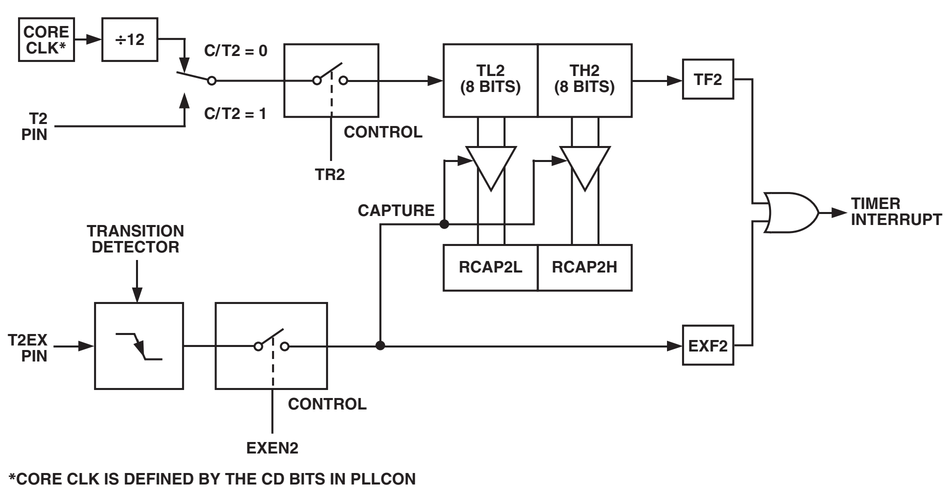


Figure 50. Timer/Counter 2, 16-Bit Capture Mode

UART SERIAL INTERFACE

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

SBUF

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

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

Table XXIV. SCON SFR Bit Designations

Bit	Name	Description
7	SM0	UART Serial Mode Select Bits.
6	SM1	These bits select the Serial Port operating mode as follows: SM0 SM1 Selected Operating Mode 0 0 Mode 0: Shift Register, fixed baud rate (Core_Clk/2) 0 1 Mode 1: 8-bit UART, variable baud rate 1 0 Mode 2: 9-bit UART, fixed baud rate (Core_Clk/64) or (Core_Clk/32) 1 1 Mode 3: 9-bit UART, variable baud rate
5	SM2	Multiprocessor Communication Enable Bit. Enables multiprocessor communication in Modes 2 and 3. In Mode 0, SM2 should be cleared. In Mode 1, if SM2 is set, RI will not be activated if a valid stop bit was not received. If SM2 is cleared, RI will be set as soon as the byte of data has been received. In Modes 2 or 3, if SM2 is set, RI will not be activated if the received ninth data bit in RB8 is 0. If SM2 is cleared, RI will be set as soon as the byte of data has been received.
4	REN	Serial Port Receive Enable Bit. Set by user software to enable serial port reception. Cleared by user software to disable serial port reception.
3	TB8	Serial Port Transmit (Bit 9). The data loaded into TB8 will be the ninth data bit that will be transmitted in Modes 2 and 3.
2	RB8	Serial Port Receiver Bit 9. The ninth data bit received in Modes 2 and 3 is latched into RB8. For Mode 1 the stop bit is latched into RB8.
1	TI	Serial Port Transmit Interrupt Flag. Set by hardware at the end of the eighth bit in Mode 0, or at the beginning of the stop bit in Modes 1, 2, and 3. TI must be cleared by user software.
0	RI	Serial Port Receive Interrupt Flag. Set by hardware at the end of the eighth bit in Mode 0, or halfway through the stop bit in Modes 1, 2, and 3. RI must be cleared by software.

Mode 0: 8-Bit Shift Register Mode
Mode 0 is selected by clearing both the SM0 and SM1 bits in the SFR SCON. Serial data enters and exits through RxD. TxD outputs the shift clock. Eight data bits are transmitted or received. Transmission is initiated by any instruction that writes to SBUF. The data is shifted out of the RxD line. The eight bits are transmitted with the least-significant bit (LSB) first, as shown in Figure 51.

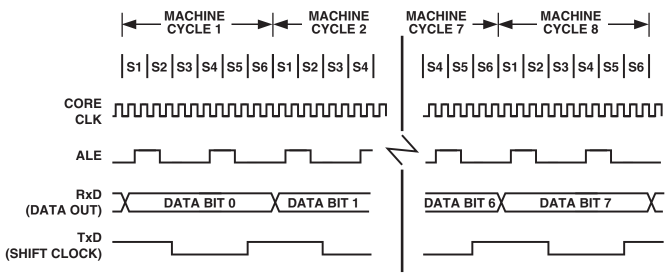


Figure 51. UART Serial Port Transmission, Mode 0

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 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 ninth 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 52.

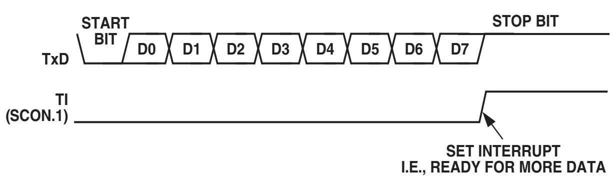


Figure 52. UART Serial Port Transmission, Mode 0

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

The eight bits in the receive shift register are latched into SBUF.

The ninth bit (Stop bit) is clocked into RB8 in SCON.

The Receiver Interrupt flag (RI) is set.

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

RI = 0, and either SM2 = 0 or SM2 = 1, and the received stop bit = 1.

If either 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/64 by default, although by setting the SMOD bit in PCON, the frequency can be doubled to Core_Clk/32. Eleven bits are transmitted or received, a start bit (0), eight data bits, a programmable ninth bit, and a stop bit (1). The ninth bit is most often used as a parity bit, although it can be used for anything, including a ninth data bit if required.

To transmit, the eight data bits must be written into SBUF. The ninth bit must be written to TB8 in SCON. When transmission is initiated, the eight data bits (from SBUF) are loaded onto the transmit shift register (LSB first). The contents of TB8 are loaded into the ninth bit position of the transmit shift register. The transmission will start 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 eight data bytes are input at RxD (LSB first) and loaded onto the receive shift register. When all eight bits have been clocked in, the following events occur:

The eight bits in the receive shift register are latched into SBUF.

The ninth data bit is latched into RB8 in SCON.

The Receiver Interrupt flag (RI) is set.

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

RI = 0, and either SM2 = 0 or SM2 = 1, and the received stop bit = 1.

If either 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 1/64 of the core clock. If SMOD = 1, the baud rate is 1/32 of the core clock:

$$Mode\ 2\ Baud\ Rate = (2^{SMOD}/64) \times (Core\ Clock\ Frequency)$$

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 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:

$$\textit{Modes 1 and 3 Baud Rate} = (2^{SMOD}/32) \times (\textit{Timer 1 Overflow Rate})$$

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:

$$\textit{Modes 1 and 3 Baud Rate} = (2^{SMOD}/32) \times (\textit{Core Clock}/(12 \times [256 - TH1]))$$

Table XXV shows some commonly used baud rates and how they might be calculated from a core clock frequency of 16.78 MHz and 2.0971 MHz. Generally speaking, a 5% error is tolerable using asynchronous (start/stop) communications.

Table XXV. Commonly Used Baud Rates, Timer 1

Ideal Baud	Core CLK (MHz)	SMOD Value	TH1-Reload Value	Actual Baud	% Error
9600	16.78	1	–9 (F9H)	9709	1.14
2400	16.78	1	–36 (DCH)	2427	1.14
1200	16.78	1	–73 (B7H)	1197	0.25
1200	2.10	0	–9 (F4H)	1213	1.14

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.

$$\textit{Modes 1 and 3 Baud Rate} = (1/16) \times (\textit{Timer 2 Overflow Rate})$$

Therefore, when Timer 2 is used to generate baud rates, the timer increments every two clock cycles and not 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 53.

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

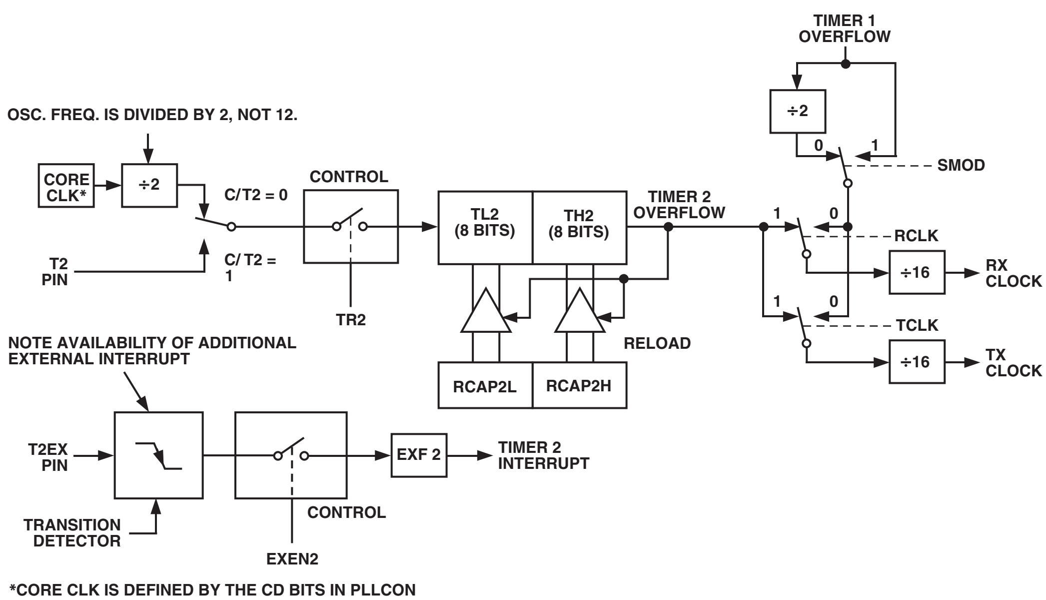


Figure 53. Timer 2, UART Baud Rates

Timer 3 Generated Baud Rates

The high integer dividers in a UART block mean that high speed baud rates are not always possible using some particular crystals. For example, using a 12 MHz crystal, a baud rate of 115200 is not possible. To address this problem, the ADuC832 has added a dedicated baud rate timer (Timer 3) specifically for generating highly accurate baud rates.

Timer 3 can be used instead of Timer 1 or Timer 2 for generating very accurate high speed UART baud rates including 115200 and 230400. Timer 3 also allows a much wider range of baud rates to be obtained. In fact, every desired bit rate from 12 bit/s to 393216 bit/s can be generated to within an error of ±0.8%. Timer 3 also frees up the other three timers, allowing them to be used for different applications. A block diagram of Timer 3 is shown in Figure 54.

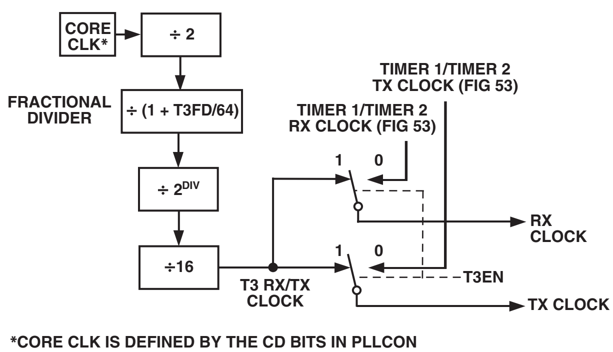


Figure 54. Timer 3, UART Baud Rates

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

Table XXVII. T3CON SFR Bit Designations

Bit	Name	Description
7	T3BAUDEN	T3UARTBAUD Enable Set to enable Timer 3 to generate the baud rate. When set PCON.7, T2CON.4 and T2CON.5 are ignored. Cleared to let the baud rate be generated as per a standard 8052.
6		-
5		-
4		-
3		-
2	DIV2	Binary Divider Factor
1	DIV1	DIV2 DIV1 DIV0 Bin Divider
0	DIV0	0 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1 1

The appropriate value to write to the DIV2-1-0 bits can be calculated using the following formula where f_{CORE} defined in PLLCON SFR:

Note: The DIV value must be rounded down.

$$DIV = \frac{\log\left(\frac{f_{CORE}}{32 \times Baud\ Rate}\right)}{\log(2)}$$

T3FD is the fractional divider ratio required to achieve the required baud rate. We can calculate the appropriate value for T3FD using the following formula:

Note: T3FD should be rounded to the nearest integer.

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

Once the values for DIV and T3FD are calculated the actual baud rate can be calculated using the following formula:

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

For example, to get a baud rate of 115200 while operating at 16.7 MHz

$$DIV = LOG\left(11059200 / (32 \times 115200)\right) / LOG2 = 1.58 = 1$$

$$T3FD = (2 \times 11059200) / (2^1 \times 115200) - 64 = 32 = 20H$$

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

Table XXVIII. Commonly Used Baud Rates Using Timer 3

Ideal Baud	CD	DIV	T3CON	T3FD	% Error
230400	0	1	81H	09H	0.25
115200	0	2	82H	09H	0.25
115200	1	1	81H	09H	0.25
115200	2	0	80H	09H	0.25
57600	0	3	83H	09H	0.25
57600	1	2	82H	09H	0.25
57600	2	1	81H	09H	0.25
57600	3	0	80H	09H	0.25
38400	0	3	83H	2DH	0.2
38400	1	2	82H	2DH	0.2
38400	2	1	81H	2DH	0.2
38400	3	0	80H	2DH	0.2
19200	0	4	84H	2DH	0.2
19200	1	3	83H	2DH	0.2
19200	2	2	82H	2DH	0.2
19200	3	1	81H	2DH	0.2
19200	4	0	80H	2DH	0.2
9600	0	5	85H	2DH	0.2
9600	1	4	84H	2DH	0.2
9600	2	3	83H	2DH	0.2
9600	3	2	82H	2DH	0.2
9600	4	1	81H	2DH	0.2
9600	5	0	80H	2DH	0.2

INTERRUPT SYSTEM

The ADuC832 provides a total of nine interrupt sources with two priority levels. The control and configuration of the interrupt system is carried out through three interrupt-related SFRs.

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

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

Table XXIX. IE SFR Bit Designations

Bit	Name	Description
7	EA	Written by User to Enable "1" or Disable "0" All Interrupt Sources
6	EADC	Written by User to Enable "1" or Disable "0" ADC Interrupt
5	ET2	Written by User to Enable "1" or Disable "0" Timer 2 Interrupt
4	ES	Written by User to Enable "1" or Disable "0" UART Serial Port Interrupt
3	ET1	Written by User to Enable "1" or Disable "0" Timer 1 Interrupt
2	EX1	Written by User to Enable "1" or Disable "0" External Interrupt 1
1	ET0	Written by User to Enable "1" or Disable "0" Timer 0 Interrupt
0	EX0	Written by User to Enable "1" or Disable "0" External Interrupt 0

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

Table XXX. IP SFR Bit Designations

Bit	Name	Description
7	----	Reserved for Future Use
6	PADC	Written by User to Select ADC Interrupt Priority ("1" = High; "0" = Low)
5	PT2	Written by User to Select Timer 2 Interrupt Priority ("1" = High; "0" = Low)
4	PS	Written by User to Select UART Serial Port Interrupt Priority ("1" = High; "0" = Low)
3	PT1	Written by User to Select Timer 1 Interrupt Priority ("1" = High; "0" = Low)
2	PX1	Written by User to Select External Interrupt 1 Priority ("1" = High; "0" = Low)
1	PT0	Written by User to Select Timer 0 Interrupt Priority ("1" = High; "0" = Low)
0	PX0	Written by User to Select External Interrupt 0 Priority ("1" = High; "0" = Low)

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

Table XXXI. IEIP2 SFR Bit Designations

Bit	Name	Description
7	----	Reserved for Future Use
6	PTI	Priority for Time Interval Interrupt
5	PPSM	Priority for Power Supply Monitor Interrupt
4	PSI	Priority for SPI/I^{2}C Interrupt
3	----	This Bit Must Contain Zero.
2	ETI	Written by User to Enable "1" or Disable "0" Time Interval Counter Interrupt.
1	EPSMI	Written by User to Enable "1" or Disable "0" Power Supply Monitor Interrupt.
0	ESI	Written by User to Enable "1" or Disable "0" SPI or I^{2}C Serial Port Interrupt.

Interrupt Priority

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

Table XXXII. Priority within an Interrupt Level

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

Interrupt Vectors

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

Table XXXIII. Interrupt Vector Addresses

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

ADuC832 HARDWARE DESIGN CONSIDERATIONS

This section outlines some of the key hardware design considerations that must be addressed when integrating the ADuC832 into any hardware system.

Clock Oscillator

The clock source for the ADuC832 can be generated by the internal PLL or by an external clock input. To use the internal PLL, connect a 32.768 kHz parallel resonant crystal between XTAL1 and XTAL2, and connect a capacitor from each pin to ground as shown below. This crystal allows the PLL to lock correctly to give a f_{VCO} of 16.777216 MHz. If no crystal is present, the PLL will free run, giving a f_{VCO} of 16.7 MHz ±20%. This is useful if an external clock input is required. The part will power up and the PLL will free run; the user then in software writes to the CFG832 SFR to enable the external clock input on P3.4.

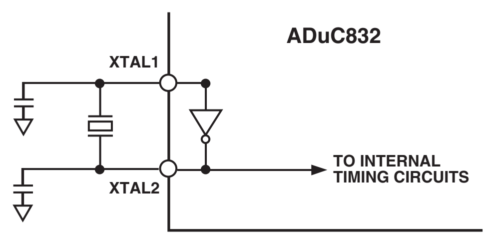


Figure 55. External Parallel Resonant Crystal Connections

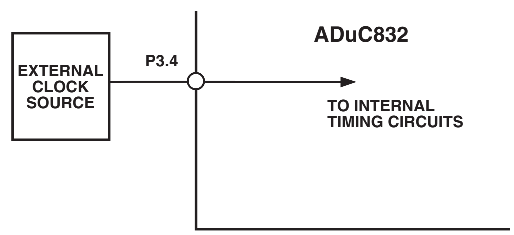


Figure 56. Connecting an External Clock Source

Whether using the internal PLL or an external clock source, the ADuC832's specified operational clock speed range is 400 kHz to 16.777216 MHz. The core itself is static, and will function all the way down to dc. But at clock speeds slower that 400 kHz, the ADC will no longer function correctly. Therefore, to ensure specified operation, use a clock frequency of at least 400 kHz and no more than 16.777216 MHz.

External Memory Interface

In addition to its internal program and data memories, the ADuC832 can access up to 64 kBytes of external program memory (ROM/PROM/and so on.) and up to 16 MBytes of external data memory (SRAM).

To select from which code space (internal or external program memory) to begin executing instructions, tie the $\overline{EA}$ (external access) pin high or low, respectively. When $\overline{EA}$ is high (pulled up to V_{DD}), user program execution will start at address 0 of the internal 62 kBytes Flash/EE code space. When $\overline{EA}$ is low (tied to ground) user program execution will start at address 0 of the external code space.

A second very important function of the $\overline{EA}$ pin is described in the Single Pin Emulation Mode section.

External program memory (if used) must be connected to the ADuC832 as illustrated in Figure 57. Note that 16 I/O lines (Ports 0 and 2) are dedicated to bus functions during external program memory fetches. Port 0 (P0) serves as a multiplexed address/data bus. It emits the low byte of the program counter (PCL) as an address, and then goes into a float state awaiting the arrival of the code byte from the program memory. During the time that the low byte of the program counter is valid on P0, the signal ALE (Address Latch Enable) clocks this byte into an address latch. Meanwhile, Port 2 (P2) emits the high byte of the program counter (PCH), then $\overline{PSEN}$ strobes the EPROM and the code byte is read into the ADuC832.

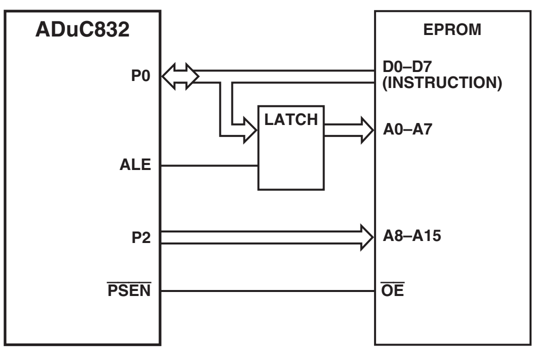


Figure 57. External Program Memory Interface

Note that program memory addresses are always 16 bits wide, even in cases where the actual amount of program memory used is less than 64 kBytes. External program execution sacrifices two of the 8-bit ports (P0 and P2) to the function of addressing the program memory. While executing from external program memory, Ports 0 and 2 can be used simultaneously for read/write access to external data memory, but not for general-purpose I/O.

Though both external program memory and external data memory are accessed by some of the same pins, the two are completely independent of each other from a software point of view. For example, the chip can read/write external data memory while executing from external program memory.

Figure 58 shows a hardware configuration for accessing up to 64 kBytes of external RAM. This interface is standard to any 8051 compatible MCU.

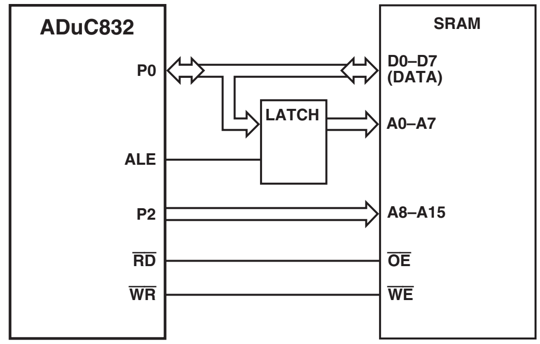


Figure 58. External Data Memory Interface (64 K Address Space)

If access to more than 64 kBytes of RAM is desired, a feature unique to the ADuC832 allows addressing up to 16 MBytes of external RAM simply by adding an additional latch, as illustrated in Figure 59.

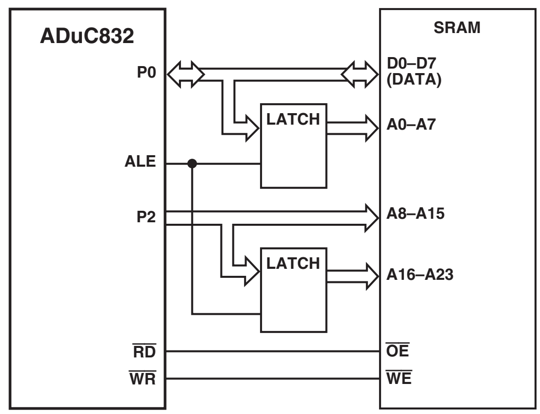


Figure 59. External Data Memory Interface (16 MBytes Address Space)

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 ADuC832 (write operation) or 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 ADuC832's operational power supply voltage range is 2.7 V to 5.25 V. Although the guaranteed data sheet specifications are given only for power supplies within 2.7 V to 3.6 V or ±10% of the nominal 5 V level, the chip will function equally well at any power supply level between 2.7 V and 5.5 V.

Note that Figures 60 and 61 refer to the PQFP package, for the CSP package connect the extra DV_{DD}, DGND, AV_{DD}, and AGND in the same manner.

Separate analog and digital power supply pins (AV_{DD} and DV_{DD}, respectively) allow AV_{DD} to be kept relatively free of noisy digital signals 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 in order 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, you connect back-to-back Schottky diodes between them as shown in Figure 60.

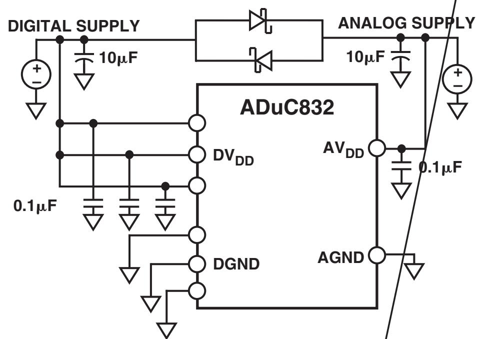
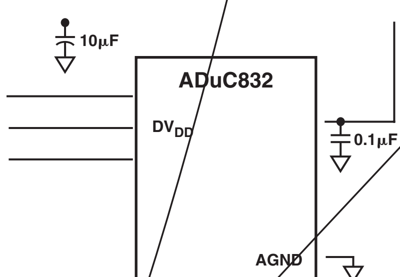


Figure 60. External Dual-Supply Connections

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 61. With this configuration other analog circuitry (such as op amps, voltage reference, and so on) can be powered from the AV_{DD} supply line as well. The user will still want to include back-to-back Schottky diodes between AV_{DD} and DV_{DD} in order to protect from power-up and power-down transient conditions that could separate the two supply voltages momentarily.



the oscillator, can also be enabled during power down. All other on-chip peripherals however, are 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 ADuC832 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 deasserted.

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 will return 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 will return the core to the instruction after the one that enabled power-down. It should be noted that the I^2C/SPI power-down interrupt enable bit (SERIPD) in the PCON SFR must first be set to allow this mode of operation.

$\overline{INT0}$ Interrupt
Power-down mode is terminated and the CPU services the $\overline{INT0}$ interrupt. The RETI at the end of the ISR will return the core to the instruction after the one that enabled power-down. The $\overline{INT0}$ pin must not be driven low during or within 2 machine cycles of the instruction that initiates power-down mode. It should be noted that the $\overline{INT0}$ power-down interrupt enable bit (INT0PD) in the PCON SFR must first be set to allow this mode of operation.

Power-On Reset
An internal POR (Power-On Reset) is implemented on the ADuC832. For DV_{DD} below 2.45 V, the internal POR will hold the ADuC832 in reset. As DV_{DD} rises above 2.45 V, an internal timer will timeout for 128 ms approximately 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 will hold the ADuC832 in reset until the power supply has dropped below 1 V. Figure 62 illustrates the operation of the internal POR in detail.

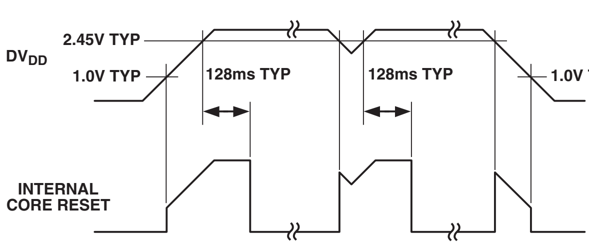


Figure 62. Internal POR Operation

Grounding and Board Layout Recommendations
As with all high resolution data converters, special attention must be paid to grounding and PC board layout of ADuC832-based designs in order to achieve optimum performance from the ADC and DACs.

Although the ADuC832 has 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 ADuC832, as illustrated in the simplified example of Figure 63a. In systems where digital and analog ground planes are connected together somewhere else (at the system's power supply for example), they cannot be connected again near the ADuC832 since a ground loop would result. In these cases, tie the ADuC832's AGND and DGND pins all to the analog ground plane, as illustrated in Figure 63b. 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 ADuC832 can then be placed between the digital and analog sections, as illustrated in Figure 63c.

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 the currents took to reach their destinations. For example, do not power components on the analog side of Figure 63b 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 placed a noisy digital chip on the left half of the board in Figure 63c. Whenever possible, avoid large discontinuities in the ground plane(s) (such as are 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 ADuC832's digital inputs, add a series resistor to each relevant line to keep rise and fall times longer than 5 ns at the ADuC832 input pins. A value of 100 Ω or 200 Ω is usually sufficient to prevent high speed signals from coupling capacitively into the ADuC832 and affecting the accuracy of ADC conversions.

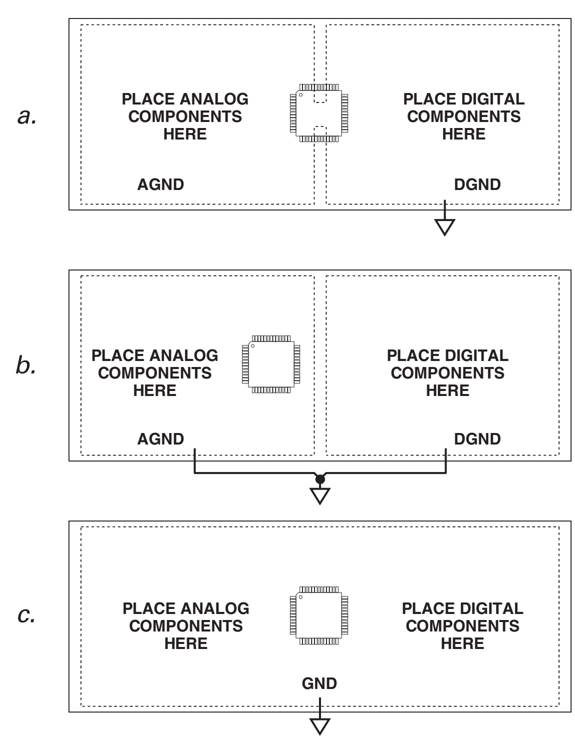


Figure 63. System Grounding Schemes

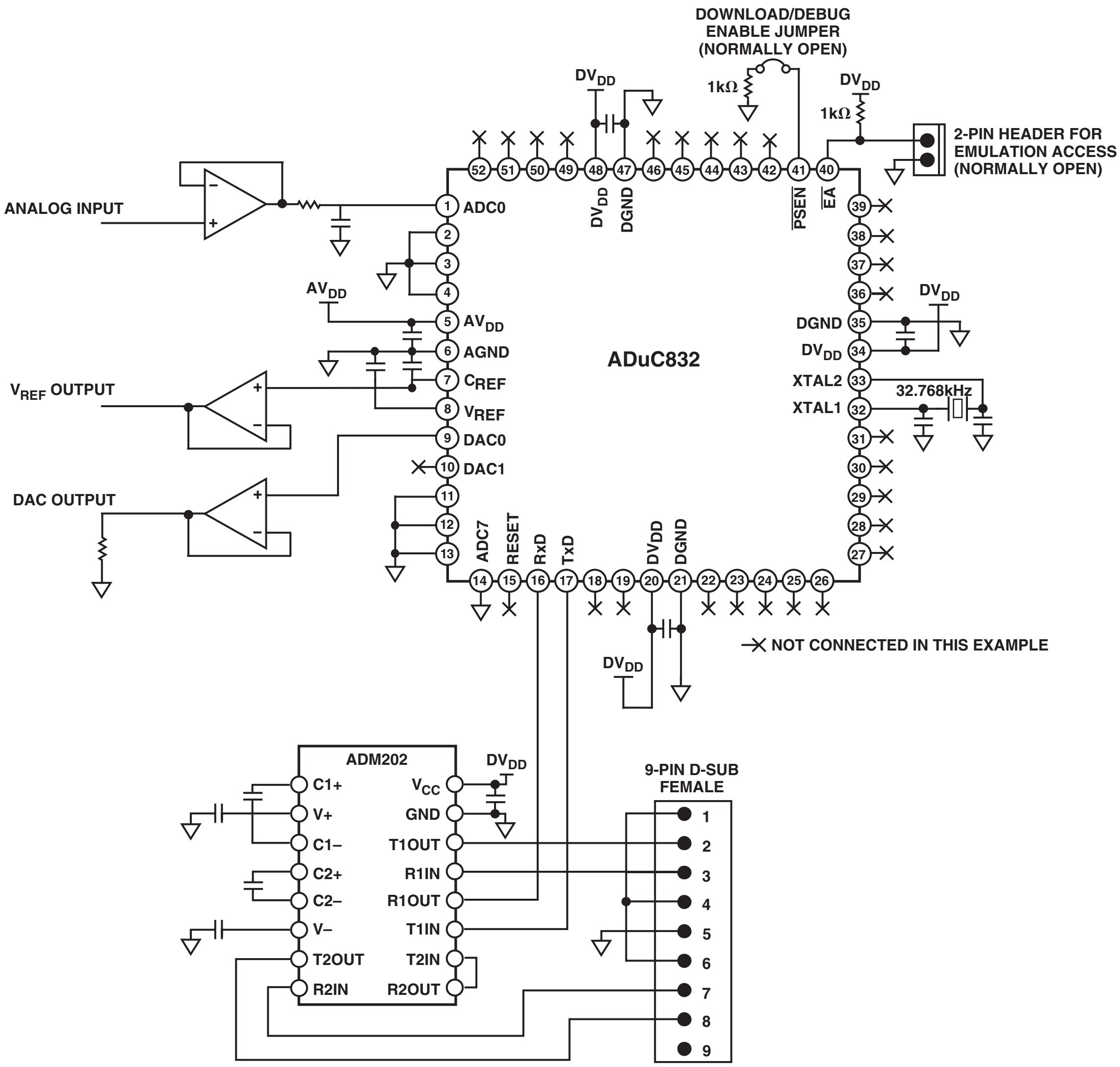


Figure 64. Example ADuC832 System (PQFP Package)

OTHER HARDWARE CONSIDERATIONS

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

In-Circuit Serial Download Access

Nearly all ADuC832 designs will want to take advantage of the in-circuit reprogrammability of the chip. This is accomplished by a connection to the ADuC832'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 64 with a simple ADM202-based circuit. If users would rather not design an RS-232 chip onto a board, refer to the application note "uC006–A 4-Wire UART-to-PC Interface"* for a simple (and zero-cost-per-board) method of gaining in-circuit serial download access to the ADuC832.

In addition to the basic UART connections, users will 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 64. To get the ADuC832 into download mode, simply connect this jumper and power-cycle the device (or manually reset the device, if a manual reset button is available) and it will be ready to receive a new program serially. With the jumper removed, the device will come up in normal mode (and run 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 (i.e., 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.

*Application Note uC006 is available at www.analog.com/microconverter

Embedded Serial Port Debugger

From a hardware perspective, entry into serial port debug mode is identical to the serial download entry sequence described above. 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 ADuC832 device, (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 ADuC832 is a dedicated controller for single-pin in-circuit emulation (ICE) using standard production ADuC832 devices. In this mode, emulation access is gained by connection to a single pin, the $\overline{EA}$ pin. Normally, this pin is hardwired either high or low to select execution from internal or external program memory space, as described earlier. To enable single-pin emulation mode, however, users will need to pull the $\overline{EA}$ pin high through a 1 kΩ resistor as shown in Figure 64. The emulator will then connect to the 2-pin header also shown in Figure 64. 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"

Windows is a registered trademark of Microsoft Corporation.

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:	ADuC832 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.

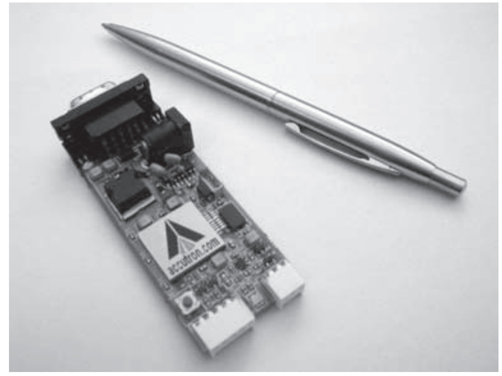


Figure 67. Accutron Single Pin Emulator

TIMING SPECIFICATIONS[1, 2, 3] (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 Min	Typ	Max	Unit	Figure
CLOCK INPUT (External Clock Driven XTAL1)						
t_{CK}	XTAL1 Period		30.52		µs	68
t_{CKL}	XTAL1 Width Low		15.16		µs	68
t_{CKH}	XTAL1 Width High		15.16		µs	68
t_{CKR}	XTAL1 Rise Time		20		ns	68
t_{CKF}	XTAL1 Fall Time		20		ns	68
$1/t_{CORE}$	ADuC832 Core Clock Frequency[4]	0.131		16.78	MHz	
t_{CORE}	ADuC832 Core Clock Period[5]		0.476		µs	
t_{CYC}	ADuC832 Machine Cycle Time[6]	0.72	5.7	91.55	µs	

NOTES

[1]AC inputs during testing are driven at DV_{DD} – 0.5 V for a Logic 1 and 0.45 V for a Logic 0. Timing measurements are made at V_{IH} min for a Logic 1 and V_{IL} max for a Logic 0, as shown in Figure 69.

[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 69.

[3]C_{LOAD} for all outputs = 80 pF, unless otherwise noted.

[4]ADuC832 internal PLL locks onto a multiple (512 times) the external crystal frequency of 32.768 kHz 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]ADuC832 Machine Cycle Time is nominally defined as 12/Core_CLK.

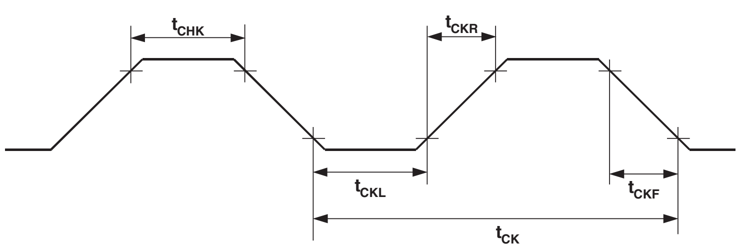


Figure 68. XTAL1 Input

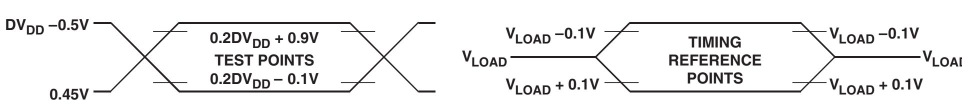


Figure 69. Timing Waveform Characteristics

Parameter		16.78 MHz Core Clk Min	Max	Variable Clock Min	Max	Unit	Figure
EXTERNAL PROGRAM MEMORY READ CYCLE							
t_{LHLL}	ALE Pulsewidth	79		$2t_{CK} - 40$		ns	70
t_{AVLL}	Address Valid to ALE Low	19		$t_{CK} - 40$		ns	70
t_{LLAX}	Address Hold after ALE Low	29		$t_{CK} - 30$		ns	70
t_{LLIV}	ALE Low to Valid Instruction In		138		$4t_{CK} - 100$	ns	70
t_{LLPL}	ALE Low to $\overline{\text{PSEN}}$ Low	29		$t_{CK} - 30$		ns	70
t_{PLPH}	$\overline{\text{PSEN}}$ Pulsewidth	133		$3t_{CK} - 45$		ns	70
t_{PLIV}	$\overline{\text{PSEN}}$ Low to Valid Instruction In		73		$3t_{CK} - 105$	ns	70
t_{PXIX}	Input Instruction Hold after $\overline{\text{PSEN}}$	0		0		ns	70
t_{PXIZ}	Input Instruction Float after $\overline{\text{PSEN}}$		34		$t_{CK} - 25$	ns	70
t_{AVIV}	Address to Valid Instruction In		193		$5t_{CK} - 105$	ns	70
t_{PLAZ}	$\overline{\text{PSEN}}$ Low to Address Float		25		25	ns	70
t_{PHAX}	Address Hold after $\overline{\text{PSEN}}$ High	0		0		ns	70

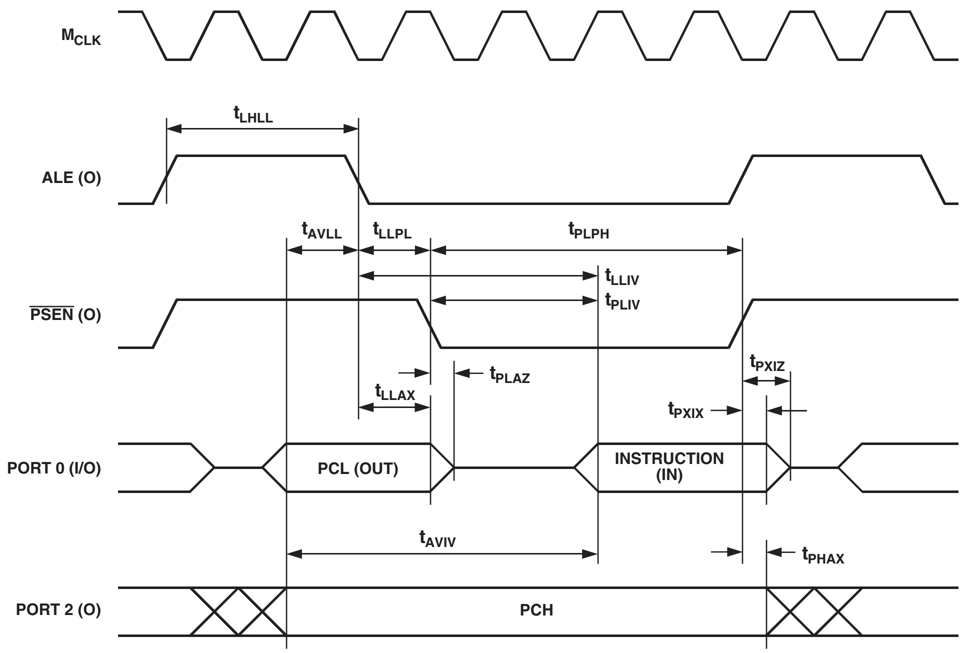


Figure 70. External Program Memory Read Cycle

Parameter		16.78 MHz Core Clk Min	Max	Variable Clock Min	Max	Unit	Figure
EXTERNAL DATA MEMORY READ CYCLE							
t_{RLRH}	$\overline{RD}$ Pulsewidth	257		$6t_{CK} - 100$		ns	71
t_{AVLL}	Address Valid after ALE Low	19		$t_{CK} - 40$		ns	71
t_{LLAX}	Address Hold after ALE Low	24		$t_{CK} - 35$		ns	71
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		133		$5t_{CK} - 165$	ns	71
t_{RHDX}	Data and Address Hold after $\overline{RD}$	0		0		ns	71
t_{RHDZ}	Data Float after $\overline{RD}$		49		$2t_{CK} - 70$	ns	71
t_{LLDV}	ALE Low to Valid Data In		326		$8t_{CK} - 150$	ns	71
t_{AVDV}	Address to Valid Data In		371		$9t_{CK} - 165$	ns	71
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	128	228	$3t_{CK} - 50$	$3t_{CK} + 50$	ns	71
t_{AVWL}	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	108		$4t_{CK} - 130$		ns	71
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns	71
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	19	257	$t_{CK} - 40$	$6t_{CK} - 100$	ns	71

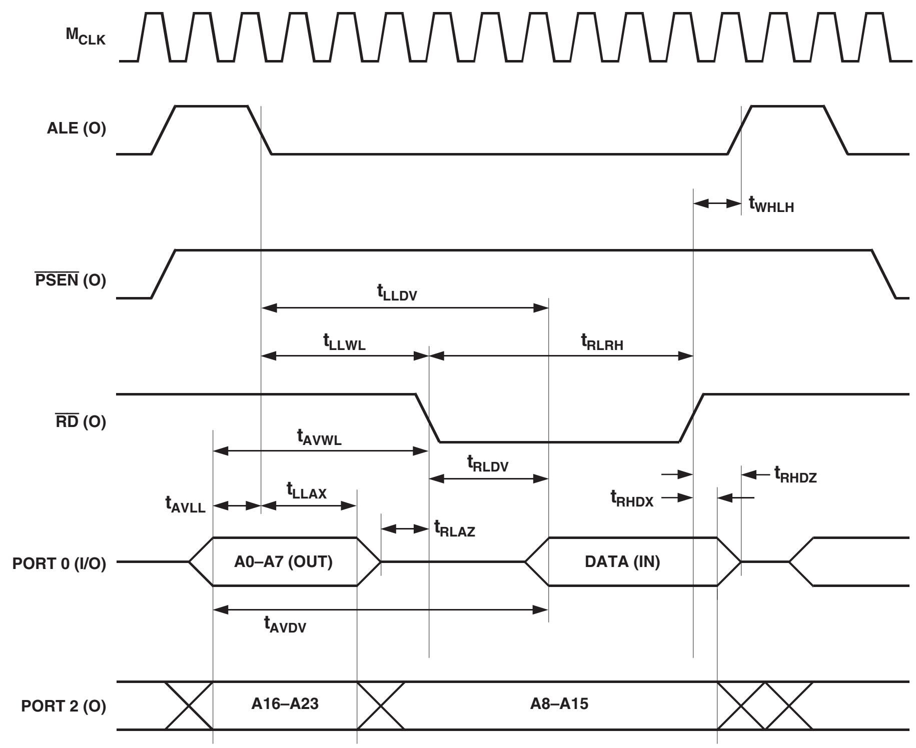


Figure 71. External Data Memory Read Cycle

Parameter		16.78 MHz Core Clk Min	Max	Variable Clock Min	Max	Unit	Figure
EXTERNAL DATA MEMORY WRITE CYCLE							
t_{WLWH}	$\overline{WR}$ Pulsewidth	257		$6t_{CK} - 100$		ns	72
t_{AVLL}	Address Valid after ALE Low	19		$t_{CK} - 40$		ns	72
t_{LLAX}	Address Hold after ALE Low	24		$t_{CK} - 35$		ns	72
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	128	228	$3t_{CK} - 50$	$3t_{CK} + 50$	ns	72
t_{AVWL}	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	108		$4t_{CK} - 130$		ns	72
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	9		$t_{CK} - 50$		ns	72
t_{QVWH}	Data Setup before $\overline{WR}$	267		$7t_{CK} - 150$		ns	72
t_{WHQX}	Data and Address Hold after $\overline{WR}$	9		$t_{CK} - 50$		ns	72
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	19	257	$t_{CK} - 40$	$6t_{CK} - 100$	ns	72

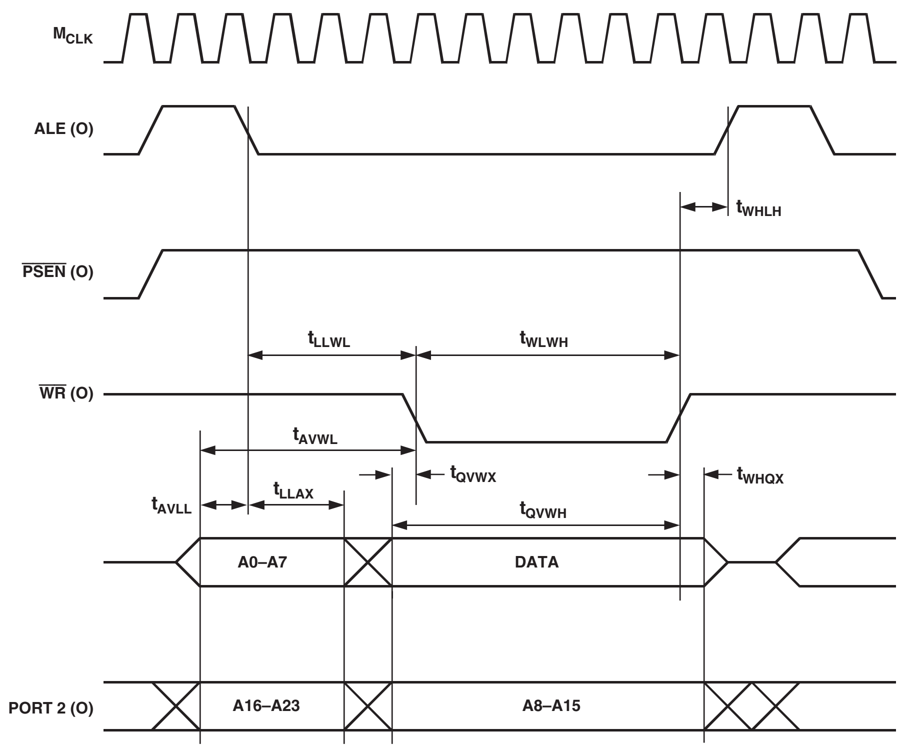


Figure 72. External Data Memory Write Cycle

Parameter		16.78 MHz Core Clk Min	Typ	Max	Variable Clock Min	Typ	Max	Unit	Figure
UART TIMING (Shift Register Mode)									
t_{XLXL}	Serial Port Clock Cycle Time		715			$12t_{CK}$		µs	73
t_{QVXH}	Output Data Setup to Clock	463			$10t_{CK} - 133$			ns	73
t_{DVXH}	Input Data Setup to Clock	252			$2t_{CK} + 133$			ns	73
t_{XHDX}	Input Data Hold after Clock	0			0			ns	73
t_{XHQX}	Output Data Hold after Clock	22			$2t_{CK} - 117$			ns	73

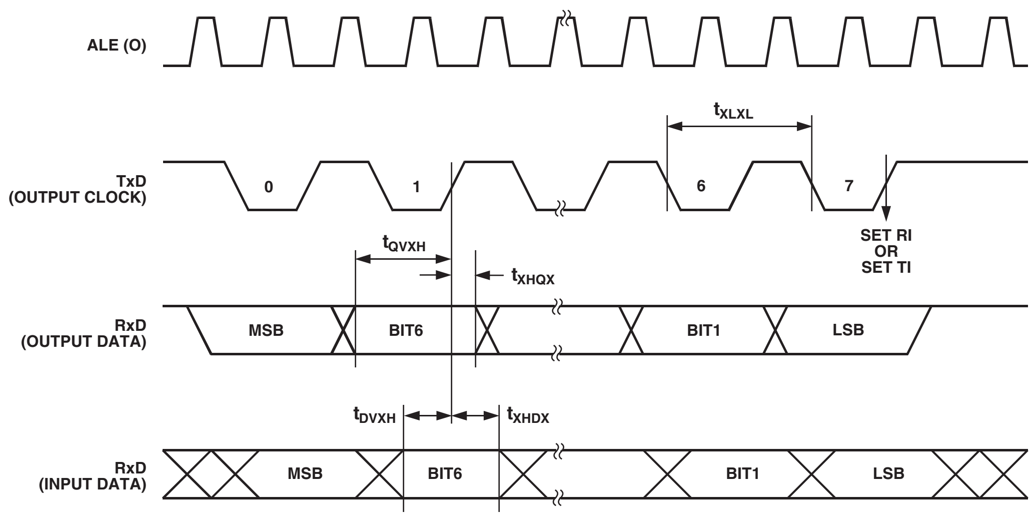


Figure 73. UART Timing in Shift Register Mode

Parameter		Min	Max	Unit	Figure
I²C COMPATIBLE INTERFACE TIMING					
t_L	SCLOCK Low Pulsewidth	4.7		µs	74
t_H	SCLOCK High Pulsewidth	4.0		µs	74
t_{SHD}	Start Condition Hold Time	0.6		µs	74
t_{DSU}	Data Setup Time	100		µs	74
t_{DHD}	Data Hold Time		0.9	µs	74
t_{RSU}	Setup Time for Repeated Start	0.6		µs	74
t_{PSU}	Stop Condition Setup Time	0.6		µs	74
t_{BUF}	Bus Free Time between a STOP Condition and a START Condition	1.3		µs	74
t_R	Rise Time of Both SCLOCK and SDATA		300	ns	74
t_F	Fall Time of Both SCLOCK and SDATA		300	ns	74
t_{SUP}*	Pulsewidth of Spike Suppressed		50	ns	74

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

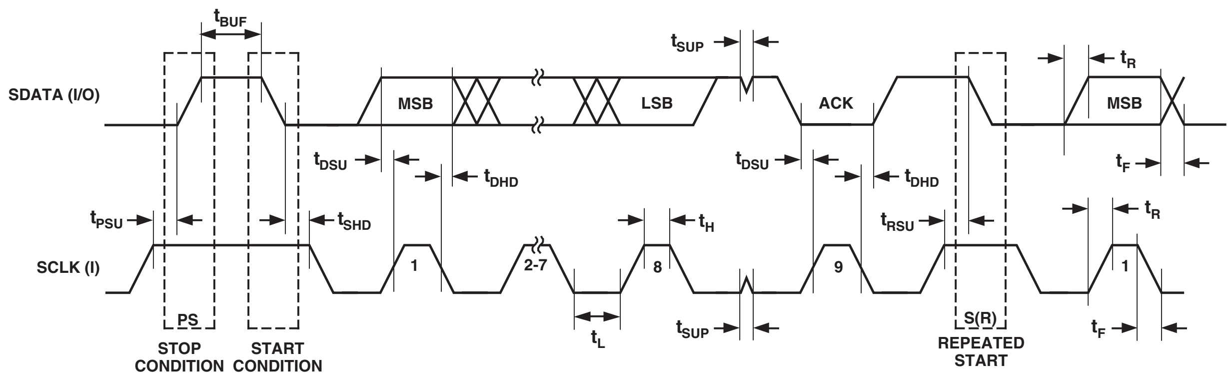
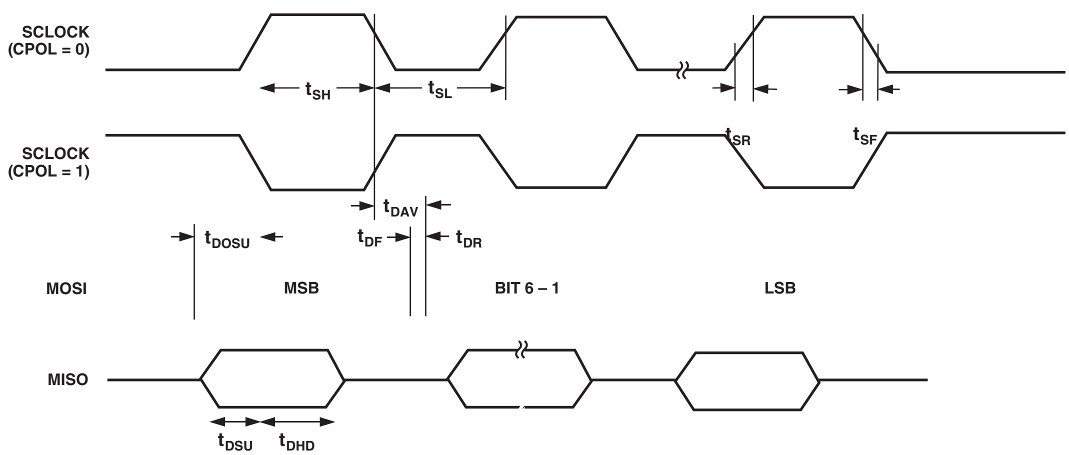


Figure 74. I²C Compatible Interface Timing

Parameter		Min	Typ	Max	Unit	Figure
SPI MASTER MODE TIMING (CPHA = 1)						
t_{SL}	SCLOCK Low Pulsewidth*		476		ns	75
t_{SH}	SCLOCK High Pulsewidth*		476		ns	75
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns	75
t_{DSU}						

Parameter		Min	Typ	Max	Unit	Figure
SPI MASTER MODE TIMING (CPHA = 0)						
t_{SL}	SCLOCK Low Pulsewidth*		476		ns	76
t_{SH}	SCLOCK High Pulsewidth*		476		ns	76
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns	76
t_{DOSU}	Data Output Setup before SCLOCK Edge			150	ns	76
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns	76
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns	76
t_{DF}	Data Output Fall Time		10	25	ns	76
t_{DR}	Data Output Rise Time		10	25	ns	76
t_{SR}	SCLOCK Rise Time		10	25	ns	76
t_{SF}	SCLOCK Fall Time		10	25	ns	76

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



Parameter		Min	Typ	Max	Unit	Figure
SPI SLAVE MODE TIMING (CPHA = 1)						
t_{SS}	$\overline{SS}$ to SCLOCK Edge	0			ns	77
t_{SL}	SCLOCK Low Pulsewidth		330		ns	77
t_{SH}	SCLOCK High Pulsewidth		330		ns	77
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns	77
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns	77
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns	77
t_{DF}	Data Output Fall Time		10	25	ns	77
t_{DR}	Data Output Rise Time		10	25	ns	77
t_{SR}	SCLOCK Rise Time		10	25	ns	77
t_{SF}	SCLOCK Fall Time		10	25	ns	77
t_{SFS}	$\overline{SS}$ High after SCLOCK Edge	0			ns	77

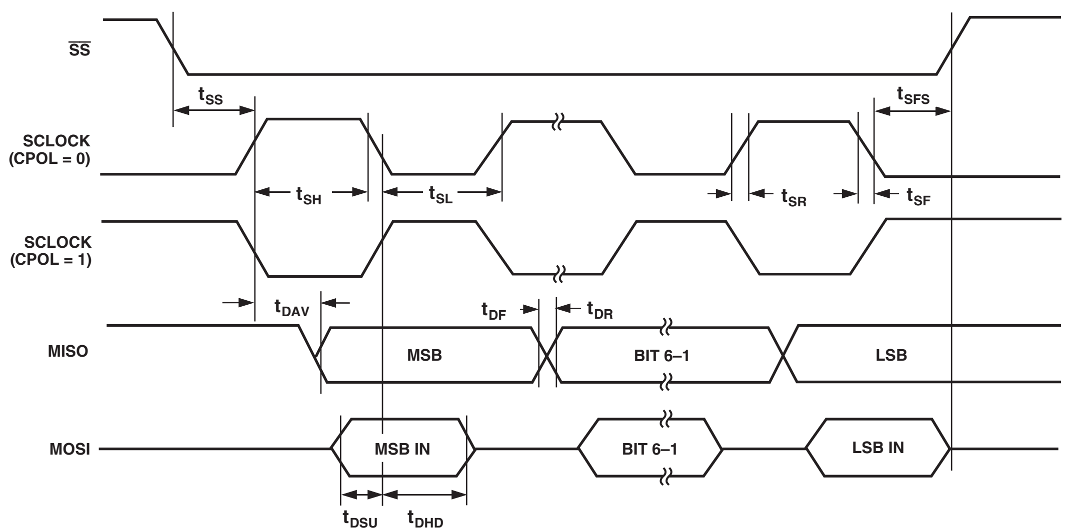


Figure 77. SPI Slave Mode Timing (CPHA = 1)

Parameter		Min	Typ	Max	Unit	Figure
SPI SLAVE MODE TIMING (CPHA = 0)						
t_{SS}	$\overline{SS}$ to SCLOCK Edge	0			ns	78
t_{SL}	SCLOCK Low Pulsewidth		330		ns	78
t_{SH}	SCLOCK High Pulsewidth		330		ns	78
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns	78
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns	78
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns	78
t_{DF}	Data Output Fall Time		10	25	ns	78
t_{DR}	Data Output Rise Time		10	25	ns	78
t_{SR}	SCLOCK Rise Time		10	25	ns	78
t_{SF}	SCLOCK Fall Time		10	25	ns	78
t_{DOSS}	Data Output Valid after $\overline{SS}$ Edge			20	ns	78
t_{SFS}	$\overline{SS}$ High after SCLOCK Edge				ns	78

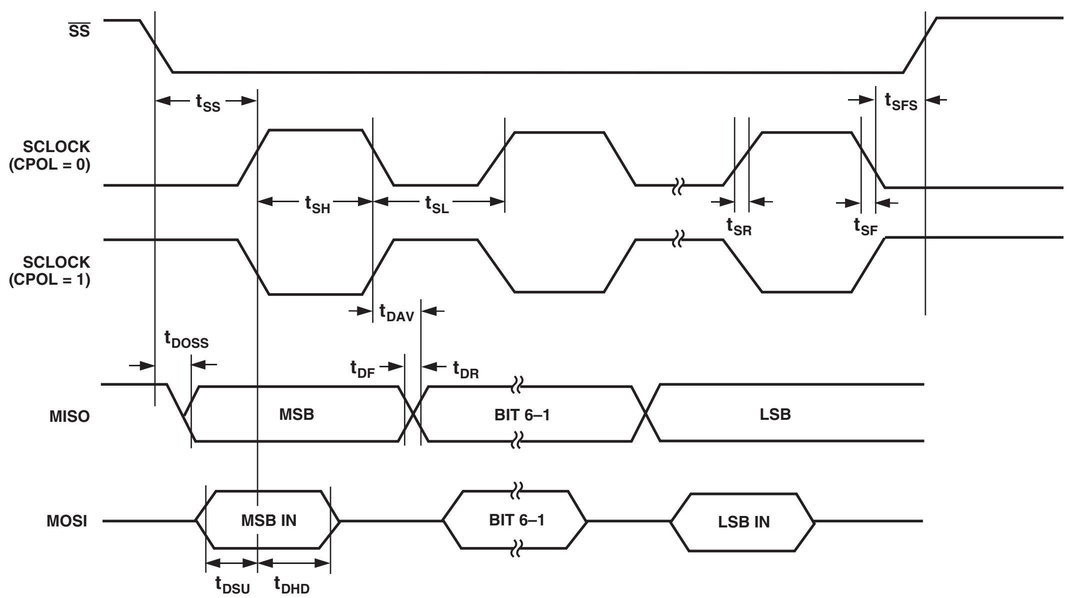
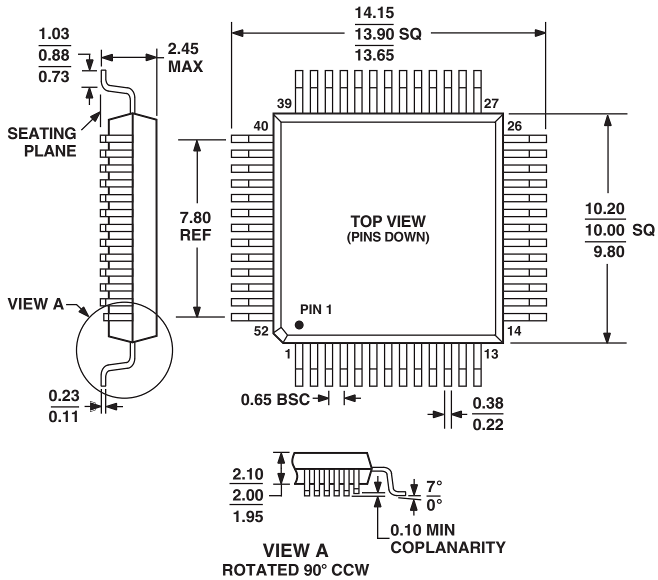


Figure 78. SPI Slave Mode Timing (CPHA = 0)

OUTLINE DIMENSIONS

52-Lead Plastic Quad Flatpack [MQFP] (S-52)

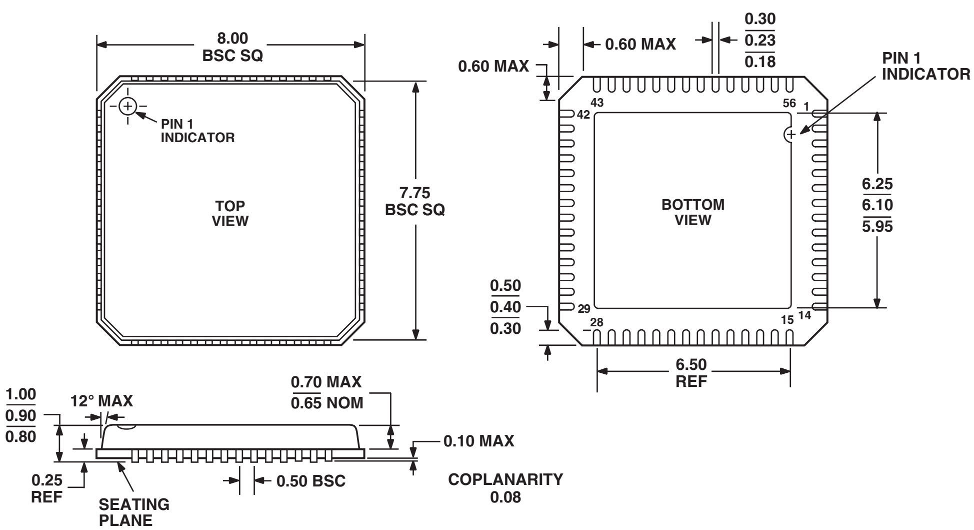
Dimensions shown in millimeters



COMPLIANT TO JEDEC STANDARDS MO-112-AC-1

56-Lead Frame Chip Scale Package [LFCSP] 8 × 8 mm Body (CP-56)

Dimensions shown in millimeters



COMPLIANT TO JEDEC STANDARDS MO-220-VLLD-2

C02987–0–11/02(0)

PRINTED IN U.S.A.