MicroConverter®, Multichannel 12-Bit ADC with Embedded Flash MCU

ADuC812

FEATURES

Analog I/O
- 8-Channel, High Accuracy 12-Bit ADC
- On-Chip, 100 ppm/°C Voltage Reference
- High Speed 200 kSPS
- DMA Controller for High Speed ADC-to-RAM Capture
- 2 12-Bit Voltage Output DACs
- On-Chip Temperature Sensor Function

Memory
- 8K Bytes On-Chip Flash/EE Program Memory
- 640 Bytes On-Chip Flash/EE Data Memory
- 256 Bytes On-Chip Data RAM
- 16M Bytes External Data Address Space
- 64K Bytes External Program Address Space

8051 Compatible Core
- 12 MHz Nominal Operation (16 MHz Max)
- 3 16-Bit Timer/Counters
- High Current Drive Capability—Port 3
- 9 Interrupt Sources, 2 Priority Levels

Power
- Specified for 3 V and 5 V Operation
- Normal, Idle, and Power-Down Modes

On-Chip Peripherals
- UART and SPI® Serial I/O
- 2-Wire (400 kHz I^2C® Compatible) Serial I/O
- Watchdog Timer

Power Supply Monitor

APPLICATIONS

Intelligent Sensors Calibration and Conditioning
Battery-Powered Systems (Portable PCs, Instruments, Monitors)
Transient Capture Systems
DAS and Communications Systems
Control Loop Monitors (Optical Networks/Base Stations)

GENERAL DESCRIPTION

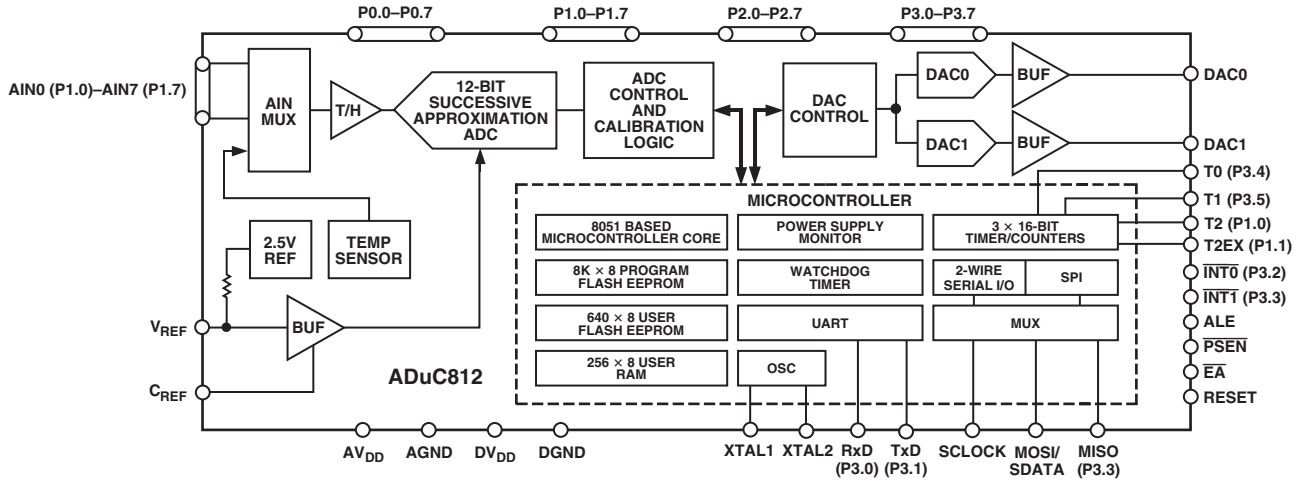
The ADuC812 is a fully integrated 12-bit data acquisition system incorporating a high performance self-calibrating multichannel ADC, dual DAC, and programmable 8-bit MCU (8051 instruction set compatible) on a single chip.

The programmable 8051 compatible core is supported by 8K bytes Flash/EE program memory, 640 bytes Flash/EE data memory, and 256 bytes data SRAM on-chip.

Additional MCU support functions include Watchdog Timer, Power Supply Monitor, and ADC DMA functions. Thirty-two programmable I/O lines, I^2C compatible SPI and Standard UART Serial Port I/O are provided for multiprocessor interfaces and I/O expansion.

Normal, idle, and power-down operating modes for both the MCU core and analog converters allow flexible power management schemes suited to low power applications. The part is specified for 3 V and 5 V operation over the industrial temperature range and is available in a 52-lead, plastic quad flatpack package, and in a 56-lead, chip scale package.

FUNCTIONAL BLOCK DIAGRAM



REV. E

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective companies.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781/329-4700 www.analog.com
Fax: 781/326-8703 © 2003 Analog Devices, Inc. All rights reserved.

TABLE OF CONTENTS

FEATURES 1
APPLICATONS 1
GENERAL DESCRIPTION 1
SPECIFICATIONS 3
ABSOLUTE MAXIMUM RATINGS 6
PIN CONFIGURATIONS 6
ORDERING GUIDE 6
PIN FUNCTION DESCRIPTIONS 7
TERMINOLOGY 8
ADC SPECIFICATIONS 8
Integral Nonlinearity 8
Differential Nonlinearity 8
Offset Error 8
Full-Scale Error 8
Signal to (Noise + Distortion) Ratio 8
Total Harmonic Distortion 8
DAC SPECIFICATIONS 8
Relative Accuracy 8
Voltage Output Settling Time 8
Digital-to-Analog Glitch Impulse 8
ARCHITECTURE, MAIN FEATURES 9
MEMORY ORGANIZATION 9
OVERVIEW OF MCU-RELATED SFRs 10
Accumulator SFR 10
B SFR 10
Stack Pointer SFR 10
Data Pointer 10
Program Status Word SFR 10
Power Control SFR 10
SPECIAL FUNCTION REGISTERS 11
ADC CIRCUIT INFORMATION 12
General Overview 12
ADC Transfer Function 12
Typical Operation 12
ADCCON1—(ADC Control SFR #1) 13
ADCCON2—(ADC Control SFR #2) 14
ADCCON3—(ADC Control SFR #3) 14
Driving the ADC 15
Voltage Reference Connections 16
Configuring the ADC 16
ADC DMA Mode 16
DMA Mode Configuration Example 17
Micro Operation during ADC DMA Mode 17
Offset and Gain Calibration Coefficients 17
Calibration 18
NONVOLATILE FLASH MEMORY 18
Flash Memory Overview 18
Flash/EE Memory and the ADuC812 18
ADuC812 Flash/EE Memory Reliability 18
Using the Flash/EE Program Memory 19
Using the Flash/EE Data Memory 19
ECON—Flash/EE Memory Control SFR 20
Flash/EE Memory Timing 20
Using the Flash/EE Memory Interface 20
Erase-All 20
Program a Byte 20
USER INTERFACE TO OTHER ON-CHIP
ADuC812 PERIPHERALS 21
Using the DAC 22
WATCHDOG TIMER 24
POWER SUPPLY MONITOR 24
SERIAL PERIPHERAL INTERFACE 25
MISO (Master In, Slave Out Data I/O Pin) 25
MOSI (Master Out, Slave In Pin) 26
SCLOCK (Serial Clock I/O Pin) 26
$\overline{SS}$ (Slave Select Input Pin) 26
Using the SPI Interface 27
SPI Interface—Master Mode 27
SPI Interface—Slave Mode 27
I^2C COMPATIBLE INTERFACE 28
8051 COMPATIBLE ON-CHIP PERIPHERALS 29
Parallel I/O Ports 0–3 29
Timers/Counters 29
Timer/Counters 0 and 1 Data Registers 31
TH0 and TL0 31
TH1 and TL1 31
TIMER/COUNTERS 0 AND 1 OPERATING MODES 32
Mode 0 (13-Bit Timer/Counter) 32
Mode 1 (16-Bit Timer/Counter) 32
Mode 2 (8-Bit Timer/Counter with Auto Reload) 32
Mode 3 (Two 8-Bit Timer/Counters) 32
Timer/Counter 2 Data Registers 33
TH2 and TL2 33
RCAP2H and RCAP2L 33
Timer/Counter Operation Modes 34
16-Bit Autoreload Mode 34
16-Bit Capture Mode 34
UART SERIAL INTERFACE 35
Mode 0 (8-Bit Shift Register Mode) 36
Mode 1 (8-Bit UART, Variable Baud Rate) 36
Mode 2 (9-Bit UART with Fixed Baud Rate) 36
Mode 3 (9-Bit UART with Variable Baud Rate) 36
UART Serial Port Baud Rate Generation 36
Timer 1 Generated Baud Rates 37
Timer 2 Generated Baud Rates 37
INTERRUPT SYSTEM 38
Interrupt Priority 39
Interrupt Vectors 39
ADuC812 HARDWARE DESIGN CONSIDERATIONS 40
Clock Oscillator 40
External Memory Interface 40
Power-On Reset Operation 41
Power Supplies 41
Power Consumption 42
Grounding and Board Layout Recommendations 43
OTHER HARDWARE CONSIDERATIONS 44
In-Circuit Serial Download Access 44
Embedded Serial Port Debugger 44
Single-Pin Emulation Mode 45
Enhanced-Hooks Emulation Mode 45
Typical System Configuration 45
QUICKSTART DEVELOPMENT SYSTEM 45
Download—In-Circuit Serial Downloader 45
DeBug—In-Circuit Debugger 45
ADSIM—Windows Simulator 45
TIMING SPECIFICATIONS 46
OUTLINE DIMENSIONS 56
Revision History 57

SPECIFICATIONS[1, 2]

(AV_{DD} = DV_{DD} = 3.0 V or 5.0 V ± 10%, REF_{IN}/REF_{OUT} = 2.5 V Internal Reference, MCLKIN = 11.0592 MHz, f_{SAMPLE} = 200 kHz, DAC V_{OUT} Load to AGND; R_L = 2 kΩ, C_L = 100 pF. All specifications T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

Parameter	ADuC812BS V_{DD} = 5 V	ADuC812BS V_{DD} = 3 V	Unit	Test Conditions/Comments
ADC CHANNEL SPECIFICATIONS				
DC ACCURACY[3, 4]				
Resolution	12	12	Bits	
Integral Nonlinearity	±1/2	±1/2	LSB typ	f_{SAMPLE} = 100 kHz
	±1.5	±1.5	LSB max	f_{SAMPLE} = 100 kHz
	±1.5	±1.5	LSB typ	f_{SAMPLE} = 200 kHz
Differential Nonlinearity	±1	±1	LSB typ	f_{SAMPLE} = 100 kHz. Guaranteed No Missing Codes at 5 V
CALIBRATED ENDPOINT ERRORS[5, 6]				
Offset Error	±5	±5	LSB max	
	±1	±1	LSB typ	
Offset Error Match	1	1	LSB typ	
Gain Error	±6	±6	LSB max	
	±1	±1	LSB typ	
Gain Error Match	1.5	1.5	LSB typ	
USER SYSTEM CALIBRATION[7]				
Offset Calibration Range	±5	±5	% of V_{REF} typ	
Gain Calibration Range	±2.5	±2.5	% of V_{REF} typ	
DYNAMIC PERFORMANCE				f_{IN} = 10 kHz Sine Wave, f_{SAMPLE} = 100 kHz
Signal-to-Noise Ratio (SNR)[8]	70	70	dB typ	
Total Harmonic Distortion (THD)	–78	–78	dB typ	
Peak Harmonic or Spurious Noise	–78	–78	dB typ	
ANALOG INPUT				
Input Voltage Ranges	0 to V_{REF}	0 to V_{REF}	V	
Leakage Current	±1	±1	µA max	
	±0.1	±0.1	µA typ	
Input Capacitance[9]	20	20	pF max	
TEMPERATURE SENSOR[10]				
Voltage Output at 25°C	600	600	mV typ	Can vary significantly (> ±20%)
Voltage TC	–3.0	–3.0	mV/°C typ	from device to device
DAC CHANNEL SPECIFICATIONS				
DC ACCURACY[11]				
Resolution	12	12	Bits	
Relative Accuracy	±3	±3	LSB typ	
Differential Nonlinearity	±0.5	±1	LSB typ	Guaranteed 12-Bit Monotonic
Offset Error	±60	±60	mV max	
	±15	±15	mV typ	
Full-Scale Error	±30	±30	mV max	
	±10	±10	mV typ	
Full-Scale 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	
Voltage Range_1	0 to V_{DD}	0 to V_{DD}	V typ	
Resistive Load	10	10	kΩ typ	
Capacitive Load	100	100	pF typ	
Output Impedance	0.5	0.5	Ω typ	
I_{SINK}	50	50	µA typ	

SPECIFICATIONS[1, 2] (continued)

Parameter	ADuC812BS V_{DD} = 5 V	ADuC812BS V_{DD} = 3 V	Unit	Test Conditions/Comments
DAC AC CHARACTERISTICS				
Voltage Output Settling Time	15	15	µs typ	Full-Scale Settling Time to within 1/2 LSB of Final Value
Digital-to-Analog Glitch Energy	10	10	nV sec typ	1 LSB Change at Major Carry
REFERENCE INPUT/OUTPUT				
REF_{IN} Input Voltage Range[9]	2.3/V_{DD}	2.3/V_{DD}	V min/max	
Input Impedance	150	150	kΩ typ	
REF_{OUT} Output Voltage	2.5 ± 2.5%	2.5 ± 2.5%	V min/max	Initial Tolerance @ 25°C
	2.5	2.5	V typ	
REF_{OUT} Tempco	100	100	ppm/°C typ	
FLASH/EE MEMORY PERFORMANCE CHARACTERISTICS[12, 13]				
Endurance	10,000		Cycles min	
	50,000	50,000	Cycles typ	
Data Retention	10		Years min	
WATCHDOG TIMER CHARACTERISTICS				
Oscillator Frequency	64	64	kHz typ	
POWER SUPPLY MONITOR CHARACTERISTICS				
Power Supply Trip Point Accuracy	±2.5	±2.5	% of Selected Nominal Trip Point Voltage max	
	±1.0	±1.0	% of Selected Nominal Trip Point Voltage typ	
DIGITAL INPUTS				
Input High Voltage (V_{INH})	2.4	2.4	V min	
XTAL1 Input High Voltage (V_{INH}) Only	4		V min	
Input Low Voltage (V_{INL})	0.8	0.8	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)	–80	–40	µA max	
	–40	–20	µA typ	V_{IL} = 450 mV
Logic 1-0 Transition Current (Port 1, 2, 3)	–700	–500	µA max	V_{IL} = 2 V
	–400	–200	µA typ	V_{IL} = 2 V
Input Capacitance	10	10	pF typ	

Parameter	ADuC812BS V_{DD} = 5 V	ADuC812BS V_{DD} = 3 V	Unit	Test Conditions/Comments
DIGITAL OUTPUTS				
Output High Voltage (V_{OH})	2.4	2.4	V min	V_{DD} = 4.5 V to 5.5 V I_{SOURCE} = 80 μA
	4.0	2.6	V typ	V_{DD} = 2.7 V to 3.3 V I_{SOURCE} = 20 μA
Output Low Voltage (V_{OL})				
ALE, $\overline{PSEN}$, 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} = 8 mA
	0.2	0.2	V typ	I_{SINK} = 8 mA
Floating State Leakage Current	±10	±10	μA max	
	±1	±1	μA typ	
Floating State Output Capacitance	10	10	pF typ	
POWER REQUIREMENTS[14, 15, 16]				
I_{DD} Normal Mode[17]	43	25	mA max	MCLKIN = 16 MHz
	32	16	mA typ	MCLKIN = 16 MHz
	26	12	mA typ	MCLKIN = 12 MHz
	8	3	mA typ	MCLKIN = 1 MHz
I_{DD} Idle Mode	25	10	mA max	MCLKIN = 16 MHz
	18	6	mA typ	MCLKIN = 16 MHz
	15	6	mA typ	MCLKIN = 12 MHz
	7	2	mA typ	MCLKIN = 1 MHz
I_{DD} Power-Down Mode[18]	30	15	μA max	
	5	5	μA typ	

NOTES

[1]Specifications apply after calibration.

[2]Temperature range –40°C to +85°C.

[3]Linearity is guaranteed during normal MicroConverter core operation.

[4]Linearity may degrade when programming or erasing the 640 byte Flash/EE space during ADC conversion times due to on-chip charge pump activity.

[5]Measured in production at V_{DD} = 5 V after Software Calibration Routine at 25°C only.

[6]User may need to execute Software Calibration Routine to achieve these specifications, which are configuration dependent.

[7]The offset and gain calibration spans are defined as the voltage range of user system offset and gain errors that the ADuC812 can compensate.

[8]SNR calculation includes distortion and noise components.

[9]Specification is not production tested, but is supported by characterization data at initial product release.

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

[11]DAC linearity is calculated using:
Reduced code range of 48 to 4095, 0 to V_{REF} range
Reduced code range of 48 to 3995, 0 to V_{DD} range
DAC output load = 10 kΩ and 50 pF.

[12]Flash/EE Memory Performance Specifications are qualified as per JEDEC Specification (Data Retention) and JEDEC Draft Specification A117 (Endurance).

[13]Endurance Cycling is evaluated under the following conditions:
Mode = Byte Programming, Page Erase Cycling
Cycle Pattern = 00H to FFH
Erase Time = 20 ms
Program Time = 100 μs

[14]I_{DD} at other MCLKIN frequencies is typically given by:
Normal Mode (V_{DD} = 5 V): I_{DD} = (1.6 nAs × MCLKIN) + 6 mA
Normal Mode (V_{DD} = 3 V): I_{DD} = (0.8 nAs × MCLKIN) + 3 mA
Idle Mode (V_{DD} = 5 V): I_{DD} = (0.75 nAs × MCLKIN) + 6 mA
Idle Mode (V_{DD} = 3 V): I_{DD} = (0.25 nAs × MCLKIN) + 3 mA
where MCLKIN is the oscillator frequency in MHz and resultant I_{DD} values are in mA.

[15]I_{DD} currents are expressed as a summation of analog and digital power supply currents during normal MicroConverter operation.

[16]I_{DD} is not measured during Flash/EE program or erase cycles; I_{DD} will typically increase by 10 mA during these cycles.

[17]Analog I_{DD} = 2 mA (typ) in normal operation (internal V_{REF}, ADC, and DAC peripherals powered on).

[18]EA = Port0 = DV_{DD}, XTAL1 (Input) tied to DV_{DD}, during this measurement.

Typical specifications are not production tested, but are supported by characterization data at initial product release.

Timing Specifications—See Pages 46–55.

Specifications subject to change without notice.
Please refer to User Guide, Quick Reference Guide, Application Notes, and Silicon Errata Sheet at www.analog.com/microconverter for additional information.

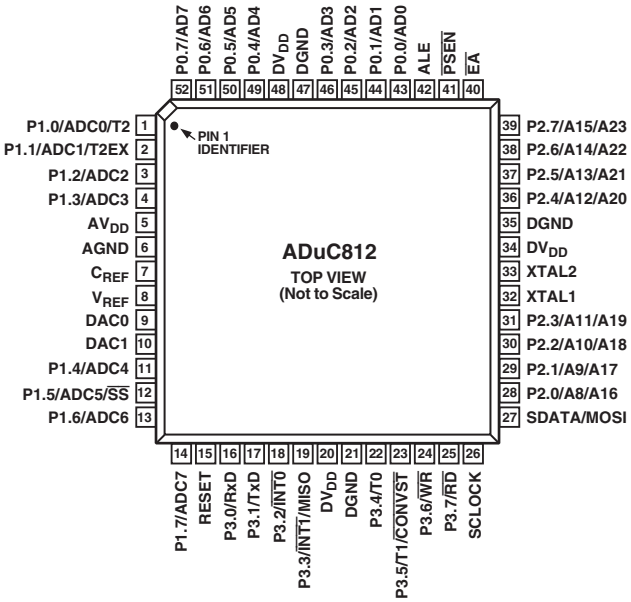
ABSOLUTE MAXIMUM RATINGS*
(T_A = 25°C, unless otherwise noted.)

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 (B Version)
. . . . . –40°C to +85°C
Storage Temperature Range . . . . . –65°C to +150°C
Junction Temperature . . . . . 150°C
θ_{JA} Thermal Impedance . . . . . 90°C/W
Lead Temperature, Soldering
Vapor Phase (60 sec) . . . . . 215°C
Infrared (15 sec) . . . . . 220°C

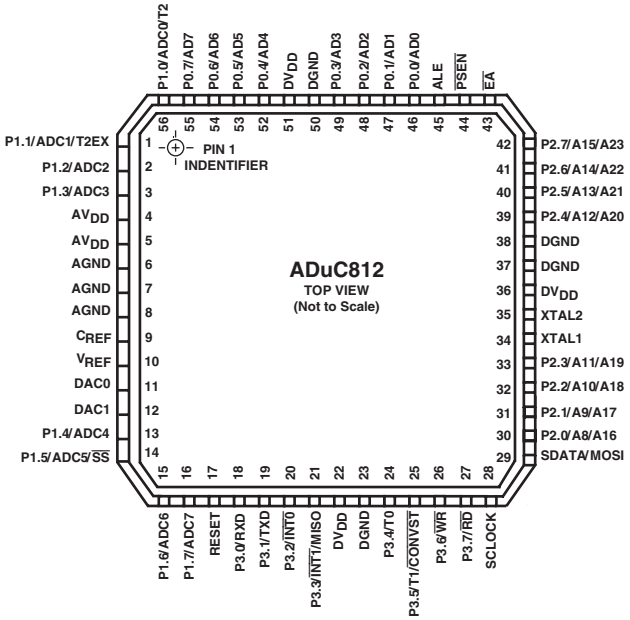
*Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those listed in the operational sections of this specification is not implied. Exposure to absolute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

PIN CONFIGURATIONS

52-Lead MQFP



56-Lead LFCSP



ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
ADuC812BS	–40°C to +85°C	52-Lead Metric Quad Flat Package	S-52
ADuC812BS	–40°C to +85°C	56-Lead Lead Frame Chip Scale Package	CP-56
EVAL-ADuC812QS		QuickStart Development System	
EVAL-ADuC812QSP		QuickStart Development System Plus	

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 ADuC812 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 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	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 ADC. The nominal internal reference voltage is 2.5 V, which appears at the pin. This pin can be overdriven by 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 multifunctional 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. External power-on reset (POR) circuity must be implemented to drive the RESET pin as described in the Power-On Reset Operation section.
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; in that state they 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.
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, $\overline{INT0}$ 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, $\overline{INT1}$ 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.
$\overline{WR}$	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 and to the Internal Clock Generator Circuits.
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; in that state they 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
$\overline{\text{PSEN}}$	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. $\overline{\text{PSEN}}$ 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.
$\overline{\text{EA}}$	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.
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.

Full-Scale 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:

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

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-scale 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.

ARCHITECTURE, MAIN FEATURES

The ADuC812 is a highly integrated, true 12-bit data acquisition system. At its core, the ADuC812 incorporates a high performance 8-bit (8052 compatible) MCU with on-chip reprogrammable nonvolatile Flash program memory controlling a multichannel (eight input channels) 12-bit ADC.

The chip incorporates all secondary functions to fully support the programmable data acquisition core. These secondary functions include User Flash Memory, Watchdog Timer (WDT), Power Supply Monitor (PSM), and various industry-standard parallel and serial interfaces.

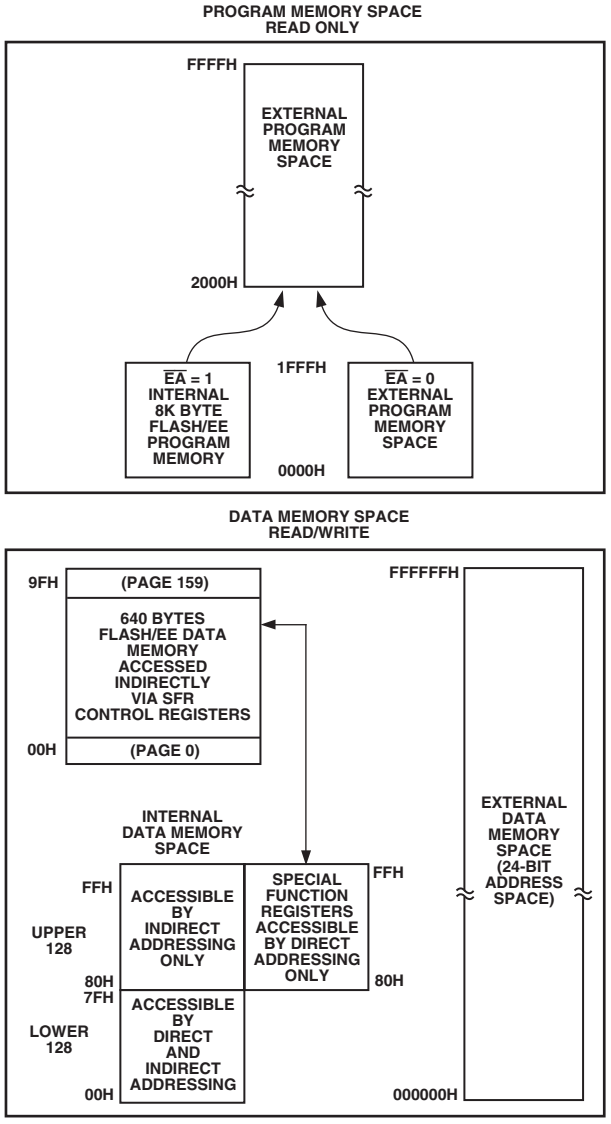


Figure 1. Program and Data Memory Maps

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) above the register banks form a block of bit addressable memory space at bit addresses 00H through 7FH.

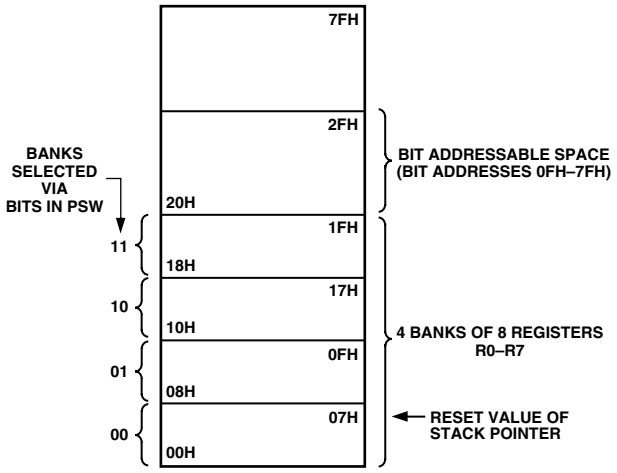


Figure 2. Lower 128 Bytes of Internal RAM

MEMORY ORGANIZATION

As with all 8052 compatible devices, the ADuC812 has separate address spaces for program and data memory as shown in Figure 1. Also as shown in Figure 1, an additional 640 bytes of User Data Flash EEPROM are available to the user. The User Data Flash Memory area is accessed indirectly via a group of control registers mapped in the Special Function Register (SFR) area in the Data Memory Space.

The SFR space is mapped in the upper 128 bytes of internal data memory space. The SFR area is accessed by direct addressing only and provides an interface between the CPU and all on-chip peripherals. A block diagram showing the programming model of the ADuC812 via the SFR area is shown in Figure 3.

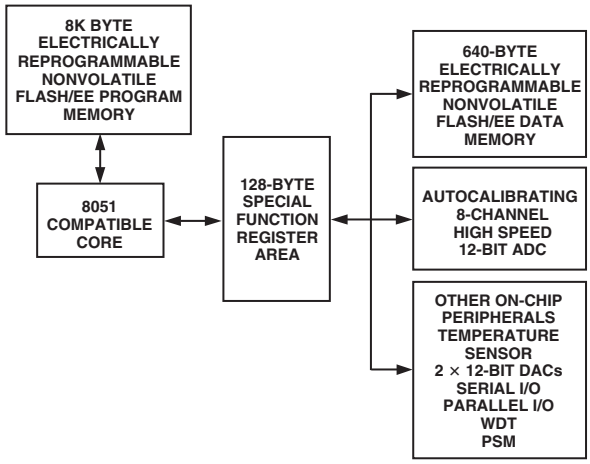


Figure 3. Programming Model

OVERVIEW OF MCU-RELATED SFRs

Accumulator SFR

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

B SFR

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

Stack Pointer SFR

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

Data Pointer

The Data Pointer is made up of three 8-bit registers: 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, and DPL).

Program Status Word SFR

The PSW register is the Program Status Word that 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

CY	AC	F0	RS1	RS0	OV	F1	P

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

The Power Control (PCON) register 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

SMOD	———	———	ALEOFF	GF1	GF0	PD	IDL

Table II. PCON SFR Bit Designations

Bit	Name	Description
7	SMOD	Double UART Baud Rate
6	———	Reserved
5	———	Reserved
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

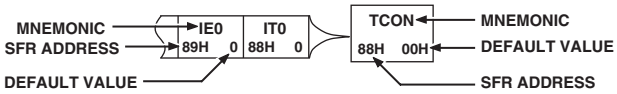
All registers except the program counter and the four general-purpose register banks reside in the special function register (SFR) area. The SFR registers include control, configuration, and data registers that provide an interface between the CPU and other on-chip peripherals.

Figure 4 shows a full SFR memory map and SFR contents on reset. Unoccupied SFR locations are shown dark shaded (NOT USED). Unoccupied locations in the SFR address space are not implemented, i.e., no register exists at this location. If an unoccupied location is read, an unspecified value is returned. SFR locations reserved for on-chip testing are shown lighter shaded (RESERVED) and should not be accessed by user software. Sixteen of the SFR locations are also bit addressable and denoted by "1" 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 0	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON[1] F8H 00H
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B[1] F0H 00H
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	I2CCON[1] E8H 00H
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC[1] E0H 00H
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	ADCCON2[1] D8H 00H
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	PSW[1] D0H 00H
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	T2CON[1] C8H 00H
PRE2 C7H 0	PRE1 C6H 0	PRE0 C5H 0	C4H 0	WDR1 C3H 0	WDR2 C2H 0	WDS C1H 0	WDE C0H 0	BITS	WDCON[1] C0H 00H
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	IP[1] B8H 00H
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	P3[1] B0H FFH
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	IE[1] A8H 00H
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2[1] A0H FFH
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	SCON[1] 98H 00H
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1[1, 3] 90H FFH
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	TCON[1] 88H 00H
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0[1] 80H FFH

DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	NOT USED
ADCOFSL[2] F1H 00H	ADCOFSH[2] F2H 20H	ADCGAINL[2] F3H 00H	ADCGAINH[2] F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 20H
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
ADCDATAL D9H 00H	ADCDATAH DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH
RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	RESERVED
RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
NOT USED	NOT USED	NOT USED	ETIM3 C4H C9H	RESERVED	EDARL C6H 00H	RESERVED
ECON B9H 00H	ETIM1 BAH 52H	ETIM2 BBH 04H	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H
NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
IE2 A9H 00H	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	NOT USED	NOT USED	NOT USED
NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	NOT USED	NOT USED
SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

SFR MAP KEY:



SFR NOTES

[1]SFRs WHOSE ADDRESS ENDS IN 0H OR 8H ARE BIT ADDRESSABLE.
[2]CALIBRATION COEFFICIENTS ARE PRECONFIGURED ON POWER-UP TO FACTORY CALIBRATED VALUES.
[3]THE PRIMARY FUNCTION OF PORT 1 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.

Figure 4. Special Function Register Locations and Reset Values

ADC CIRCUIT INFORMATION

General Overview

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

The ADC consists of a conventional successive-approximation converter based around a capacitor DAC. The converter accepts an analog input range of 0 V to V_{REF}. A high precision, low drift and factory calibrated 2.5 V reference is provided on-chip. The internal reference may be overdriven via the external V_{REF} pin. This external reference can be in the range 2.3 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 ADuC812 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. A software calibration routine is provided to allow the user to overwrite the factory programmed calibration coefficients if required, thus minimizing the impact of endpoint errors in the user's target system.

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 5.

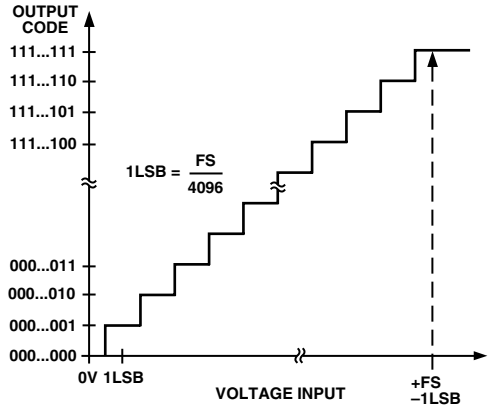


Figure 5. ADC Transfer Function

Typical Operation

Once configured via the ADCCON 1–3 SFRs (shown on the following page), 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 to identify the channel result. The format of the ADC 12-bit result word is shown in Figure 6.

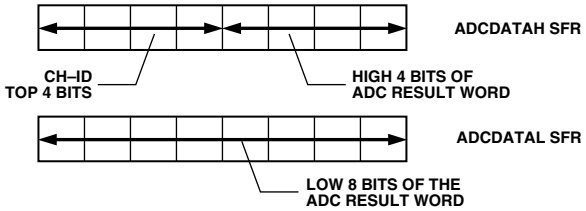


Figure 6. ADC Result Format

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 20H

MD1	MD0	CK1	CK0	AQ1	AQ0	T2C	EXC

Table III. ADCCON1 SFR Bit Designations

Bit	Name	Description
ADCCON1.7 ADCCON1.6	MD1 MD0	The mode bits (MD1, MD0) select the active operating mode of the ADC as follows: MD1 MD0 Active Mode 0 0 ADC powered down 0 1 ADC normal mode 1 0 ADC powered down if not executing a conversion cycle 1 1 ADC standby if not executing a conversion cycle Note: In power-down mode the ADC V_{REF} circuits are maintained on, whereas all ADC peripherals are powered down, thus minimizing current consumption.
ADCCON1.5 ADCCON1.4	CK1 CK0	The ADC clock divide bits (CK1, CK0) select the divide ratio for the master clock used to generate the ADC clock. A typical ADC conversion will require 17 ADC clocks. The divider ratio is selected as follows: CK1 CK0 MCLK Divider 0 0 1 0 1 2 1 0 4 1 1 8
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, and are selected as follows: AQ1 AQ0 #ADC Clks 0 0 1 0 1 2 1 0 4 1 1 8
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. ADC conversions are initiated on the second Timer 2 overflow.
ADCCON1.0	EXC	The external trigger enable bit (EXC) is set by the user to allow the external CONVST pin 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

ADCI	DMA	CCONV	SCONV	CS3	CS2	CS1	CS0

Table IV. ADCCON2 SFR Bit Designations

Location	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.
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.
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 the one pointed to by these channel selection bits. In DMA mode, the channel selection is derived from the channel ID written to the external memory. CS3 CS2 CS1 CS0 CH# 0 0 0 0 0 0 0 0 1 1 0 0 1 0 2 0 0 1 1 3 0 1 0 0 4 0 1 0 1 5 0 1 1 0 6 0 1 1 1 7 1 0 0 0 Temp Sensor 1 1 1 1 DMA STOP All other combinations reserved.

ADCCON3—(ADC Control SFR #3)
The ADCCON3 register gives user software an indication of ADC busy status.

SFR Address F5H
SFR Power-On Default Value 00H

BUSY	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD	RSVD

Table V. ADCCON3 SFR Bit Designations

Bit Location	Bit Status	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 ADCCON3.5 ADCCON3.4 ADCCON3.3 ADCCON3.2 ADCCON3.1 ADCCON3.0	RSVD RSVD RSVD RSVD RSVD RSVD RSVD	ADCCON3.0–3.6 are reserved (RSVD) for internal use. These bits will read as "0" and should only be written as "0" by user software.

Driving the ADC

The ADC incorporates a successive approximation (SAR) architecture involving a charge-sampled input stage. Figure 7 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 7. 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.

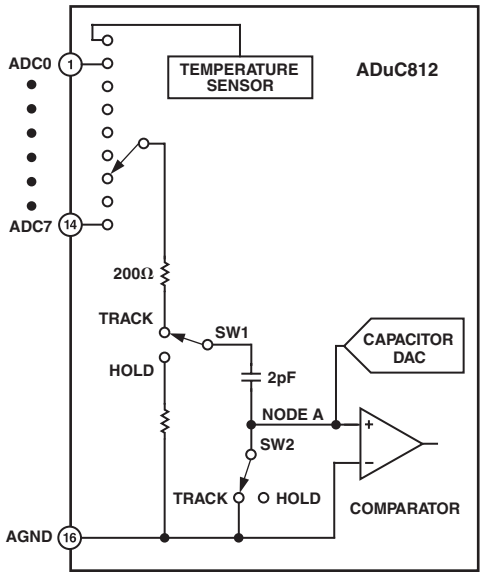


Figure 7. Internal ADC Structure

Note that whenever a new input channel is selected, a residual charge from the 2 pF sampling capacitor places a transient on the newly selected input. The signal source must be capable of recovering from this transient before the sampling switches click into "hold" mode. Delays can be inserted in software (between channel selection and conversion request) to account for input stage settling, but a hardware solution will alleviate this burden from the software design task and will ultimately result in a cleaner system implementation. One hardware solution would be to choose a very fast settling op amp to drive each analog input. Such an op amp would need to settle fully from a small signal transient in less than 300 ns to guarantee adequate settling under all software configurations. A better solution, recommended for use with any amplifier, is shown in Figure 8.

Though at first glance the circuit in Figure 8 may look like a simple antialiasing filter, it actually serves no such purpose since its corner frequency is well above the Nyquist frequency, even at a 200 kHz sample rate. Though the R/C does help to reject some incoming high frequency noise, its primary function is to ensure that the transient demands of the ADC input stage are met. It does so by providing a capacitive bank from which the 2 pF

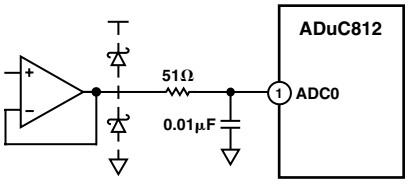


Figure 8. Buffering Analog Inputs

sampling capacitor can draw its charge. Since the 0.01 μF capacitor in Figure 8 is more than 4096 times the size of the 2 pF sampling capacitor, its voltage will not change by more than one count (1/4096) of the 12-bit transfer function when the 2 pF charge from a previous channel is dumped onto it. A larger capacitor can be used if desired, but not a larger resistor (for reasons described below).

The Schottky diodes in Figure 8 may be necessary to limit the voltage applied to the analog input pin as per the Absolute Maximum Ratings. They are not necessary if the op amp is powered from the same supply as the ADuC812 since in that case, the op amp is unable to generate voltages above V_{DD} or below ground. An op amp is necessary unless the signal source is very low impedance to begin with. DC leakage currents at the ADuC812's analog inputs can cause measurable dc errors with external source impedances of as little as 100 Ω. To ensure accurate ADC operation, keep the total source impedance at each analog input less than 61 Ω. The table below illustrates examples of how source impedance can affect dc accuracy.

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

Although Figure 8 shows the op amp operating at a gain of 1, you can configure it for any gain needed. Also, you can use an instrumentation amplifier in its place to condition differential signals. Use any modern amplifier that is capable of delivering the signal (0 to V_{REF}) with minimal saturation. Some single-supply, rail-to-rail op amps that are useful for this purpose include, but are not limited to, the ones given in Table VI. Check Analog Devices literature (CD ROM data book, and so on) for details about these and other op amps and instrumentation amps.

Table VI. Some Single-Supply Op Amps

Op Amp Model	Characteristics
OP181/OP281/OP481	Micropower
OP191/OP291/OP491	I/O Good up to V_{DD}, Low Cost
OP196/OP296/OP496	I/O to V_{DD}, Micropower, Low Cost
OP183/OP283	High Gain-Bandwidth Product
OP162/OP262/OP462	High GBP, Micro Package
AD820/AD822/AD824	FET Input, Low Cost
AD823	FET Input, High GBP

Keep in mind that the ADC's transfer function is 0 V to V_{REF}, and any signal range lost to amplifier saturation near ground will impact dynamic range. Though the op amps in Table VI are capable of delivering output signals very closely approaching ground, no amplifier can deliver signals all the way to ground when powered by a single supply. Therefore, if a negative supply is available, consider using it to power the front end amplifiers.

However, be sure to include the Schottky diodes shown in Figure 8 (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 8 to drive the analog input pins of the ADuC812.

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, decouple both the V_{REF} pin and the C_{REF} pin to ground with 0.1 μF ceramic chip capacitors as shown in Figure 9.

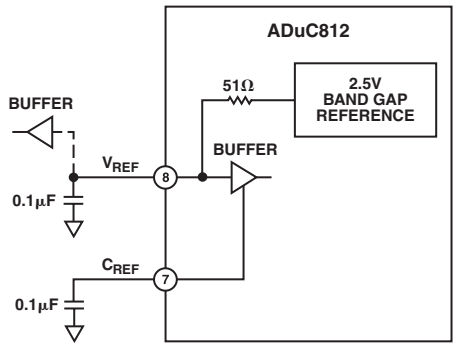


Figure 9. Decoupling V_{REF} and C_{REF}

The internal voltage reference can also be tapped directly from the V_{REF} pin, if desired, to drive external circuitry. However, a buffer must be used to ensure that no current is drawn from the V_{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. Do not connect anything to this pin except the capacitor, and be sure to keep trace-lengths short on the C_{REF} capacitor, decoupling the node straight to the underlying ground plane.

The ADuC812 powers up with its internal voltage reference in the "off" state. The voltage reference turns on automatically whenever the ADC or either DAC gets enabled in software. Once enabled, the voltage reference requires approximately 65 ms to power up and settle to its specified value. Be sure that your software allows this time to elapse before initiating any conversions. If an external voltage reference is preferred, connect it to the V_{REF} pin as shown in Figure 10 to overdrive the internal reference.

To ensure accurate ADC operation, the voltage applied to V_{REF} must be between 2.3 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 V_{REF} pin directly to AV_{DD}. In such a configuration, the user must also connect the C_{REF} pin directly to AV_{DD} to circumvent internal buffer headroom limitations. This allows the ADC input transfer function to span the full range of 0 V to AV_{DD} accurately.

Operation of the ADC or DACs with a reference voltage below 2.3 V, however, may incur loss of accuracy resulting in missing codes or nonmonotonicity. For that reason, do not use a reference voltage less than 2.3 V.

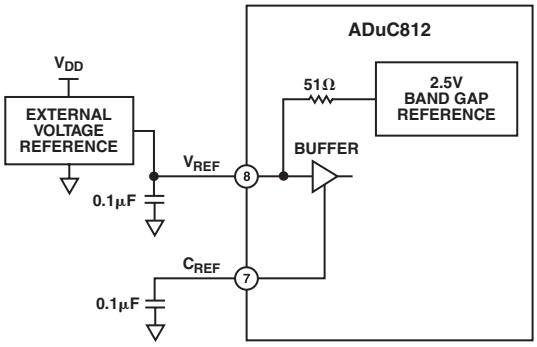


Figure 10. Using an External Voltage Reference

Configuring the ADC

The three SFRs (ADCCON1, ADCCON2, ADCCON3) configure the ADC. In nearly all cases, an acquisition time of one ADC clock (ADCCON1.2 = 0, ADCCON1.3 = 0) will provide plenty of time for the ADuC812 to acquire its signal before switching the internal track-and-hold amplifier into hold mode. The only exception would be a high source impedance analog input, but these should be buffered first anyway since source impedances of greater than 610 Ω can cause dc errors as well.

The ADuC812'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 4 MHz, and optimum performance is obtained with ADC clock between 400 kHz and 3 MHz. Frequencies within this range can 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, with a 12 MHz master clock, set the ADC clock divide ratio to 4 (i.e., ADCCLK = MCLK/4 = 3 MHz) by setting the appropriate bits in ADCCON1 (ADCCON1.5 = 1, ADCCON1.4 = 0).

The total ADC conversion time is 15 ADC clocks, plus one ADC clock for synchronization, plus the selected acquisition time (1, 2, 3, or 4 ADC clocks). For the example above, with a one clock acquisition time, total conversion time is 17 ADC clocks (or 5.67 μs for a 3 MHz ADC clock).

In continuous conversion mode, a new conversion begins each time the previous one finishes. The sample rate is the inverse of the total conversion time described above. In the example above, the continuous conversion mode sample rate would be 176.5 kHz.

ADC DMA Mode

The on-chip ADC has been designed to run at a maximum conversion speed of 5 μs (200 kHz sampling rate). When converting at this rate, the ADuC812 MicroConverter has 5 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 ADuC812 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 ADuC812 core. This mode allows the ADuC812 to capture a contiguous sample stream at full ADC update rates (200 kHz).

DMA Mode Configuration Example

To set the ADuC812 into DMA mode, a number of steps must be followed.

1. The ADC must be powered down by setting MD1 and MD0 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 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 independently of the ADuC812 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. Figure 11 shows a typical preconfiguration of external memory.

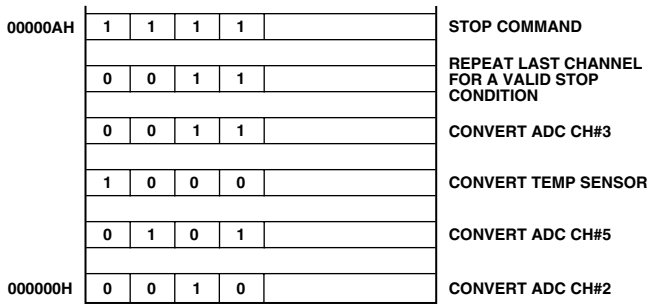


Figure 11. Typical DMA External Memory Preconfiguration

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 by starting single/continuous conversions, starting Timer 2 running for Timer 2 conversions, or by 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 in Figure 12. It should be noted that no result is written to the last two memory locations.

When the DMA mode logic is active, it is responsible for storing the ADC results away from both the user and ADuC812 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 gain 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.

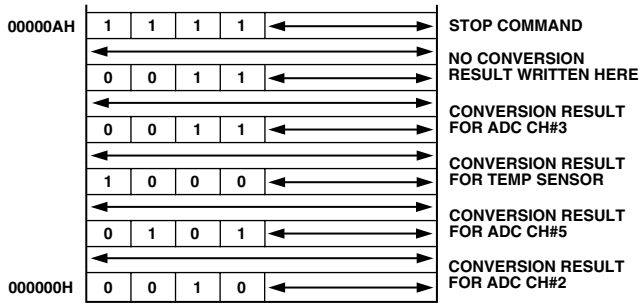


Figure 12. Typical External Memory Configuration Post ADC DMA Operation

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 Figure 13.

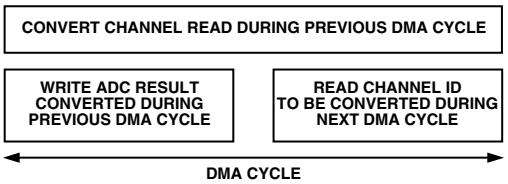


Figure 13. DMA Cycle

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 are 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 13. 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, it should be noted that MCU core accesses to Ports 0 and 2 (which 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.

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.

Offset and Gain Calibration Coefficients

The ADuC812 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, 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 ±2.5% of the reference voltage.

Calibration

Each ADuC812 is calibrated in the factory prior to shipping, and the offset and gain calibration coefficients are stored in a hidden area of FLASH/EE memory. Each time the ADuC812 powers up, an internal power-on configuration routine copies these coefficients into the offset and gain calibration registers in the SFR area.

The MicroConverter ADC accuracy may vary from system to system due to board layout, grounding, clock speed, and so on. To get the best ADC accuracy in your system, perform the software calibration routine described in Application Note uC005, available from the MicroConverter homepage at www.analog.com/microconverter.

NONVOLATILE FLASH MEMORY

Flash Memory Overview

The ADuC812 incorporates Flash memory technology on-chip to provide the user with a nonvolatile, in-circuit reprogrammable code and data memory space.

Flash/EE memory is a relatively new type of nonvolatile memory technology based on a single transistor cell architecture.

This technology is basically an outgrowth of EPROM technology and was developed in 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 14).

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 in page blocks. Thus, Flash memory is often and more correctly referred to as Flash/EE memory.

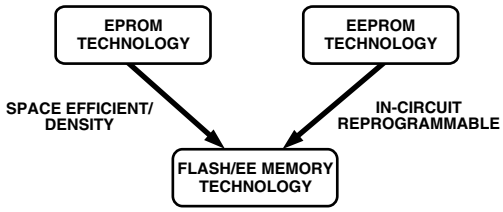


Figure 14. Flash Memory Development

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 ADuC812, Flash/EE memory technology allows the user to update program code space in-circuit without replacing one-time programmable (OTP) devices at remote operating nodes.

Flash/EE Memory and the ADuC812

The ADuC812 provides two arrays of Flash/EE memory for user applications. 8K bytes 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 using conventional third party memory programmers. This array can also be programmed in-circuit, using the serial download mode provided.

A 640 byte Flash/EE data memory space is also provided on-chip as a general-purpose nonvolatile scratchpad area. User access to this area is via a group of six SFRs.

ADuC812 Flash/EE Memory Reliability

The Flash/EE program and data memory arrays on the ADuC812 are fully qualified for two key Flash/EE memory characteristics: Flash/EE Memory Cycling Endurance and Flash/EE Memory Data Retention.

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

a. Initial Page Erase Sequence
b. Read/Verify Sequence
c. Byte Program Sequence
d. Second Read/Verify Sequence

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

As indicated in the Specification tables, the ADuC812 Flash/EE Memory Endurance qualification has been carried out in accordance with JEDEC Specification A117 over the industrial temperature ranges of –40°C, +25°C, and +85°C. The results allow the specification of a minimum endurance figure over supply and temperature of 10,000 cycles, with an endurance figure of 50,000 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 ADuC812 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.

Using the Flash/EE Program Memory

This 8K byte Flash/EE program memory array is mapped into the lower 8K bytes of the 64K bytes program space addressable by the ADuC812 and will be used to hold user code in typical applications.

The program memory array can be programmed in one of two modes:

Serial Downloading (In-Circuit Programming)

As part of its embedded download/debug kernel, the ADuC812 facilitates serial code download via the standard UART serial port. Serial download mode is automatically entered on power-up if the external pin $\overline{\text{PSEN}}$ is pulled low through an external resistor as shown in Figure 15. Once in this mode, the user can download code to the program memory array while the device is sited in its target application hardware. A PC serial download executable is provided as part of the ADuC812 QuickStart development system.

The Serial Download protocol is detailed in a MicroConverter Applications Note uC004, available from the ADI MicroConverter website at www.analog.com/micronverter.

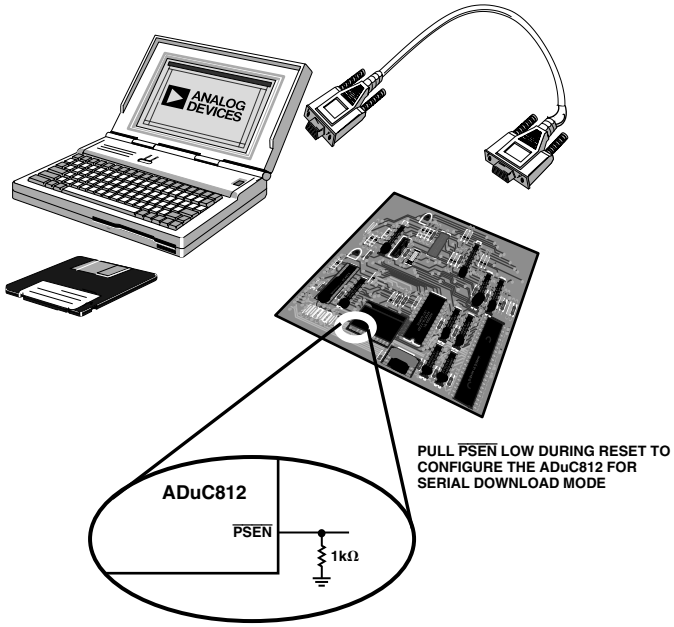


Figure 15. Flash/EE Memory Serial Download Mode Programming

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 homepage at www.analog.com/microconverter.

Using the Flash/EE Data Memory

The user Flash/EE data memory array consists of 640 bytes that are configured into 160 (Page 00H to Page 9FH) 4-byte pages, as shown in Figure 16.

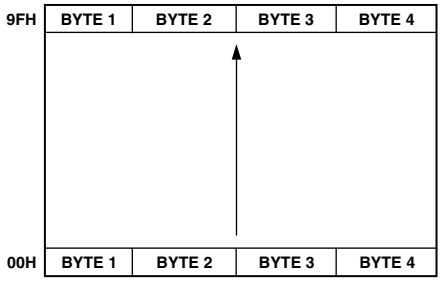


Figure 16. User Flash/EE Memory Configuration

As with other ADuC812 user peripheral circuits, the interface to this memory space is via a group of registers mapped in the SFR space. A group of four data registers (EDATA1–4) is used to hold the 4-byte page being accessed. EADRL is used to hold the 8-bit address of the page being accessed. Finally, ECON is an 8-bit control register that may be written with one of five Flash/EE memory access commands to trigger various read, write, erase, and verify functions. These register can be summarized as follows:

ECON:	SFR Address	B9H
	Function	Controls access to 640 bytes Flash/EE data space.
	Default	00H
EADRL:	SFR Address	C6H
	Function	Holds the Flash/EE data page address. 0H through 9FH
	Default	00H
EDATA1–4:		
	SFR Address	BCH to BFH, respectively
	Function	Holds the Flash/EE data memory page write or page read data bytes.
	Default	EDATA1–4→00H

A block diagram of the SFR registered interface to the data Flash/EE memory array is shown in Figure 17.

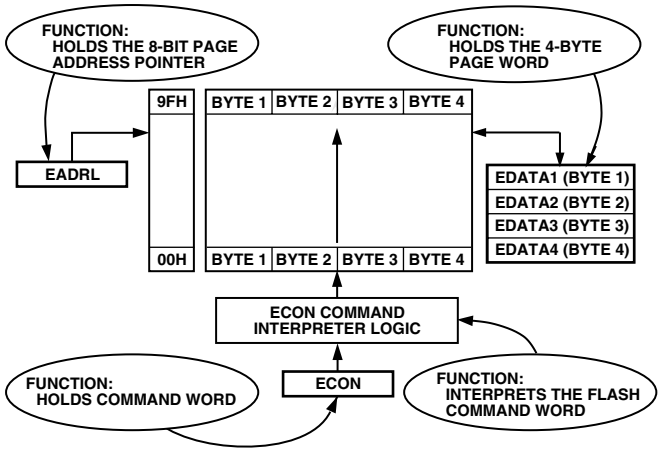


Figure 17. User Flash/EE Memory Control and Configuration

ECON—Flash/EE Memory Control SFR
This SFR acts as a command interpreter and may be written with one of five command modes to enable various read, program, and erase cycles as detailed in Table VII.

Table VII. ECON—Flash/EE Memory Control Register Command Modes

Command Byte	Command Mode
01H	READ COMMAND Results in four bytes being read into EDATA1–4 from memory page address contained in EADRL.
02H	PROGRAM COMMAND Results in four bytes (EDATA1–4) being written to memory page address in EADRL. This write command assumes the designated "write" page has been pre-erased.
03H	RESERVED FOR INTERNAL USE 03H should not be written to the ECON SFR.
04H	VERIFY COMMAND Allows the user to verify if data in EDATA1–4 is contained in page address designated by EADRL. A subsequent read of the ECON SFR will result in a zero being read if the verification is valid; a nonzero value will be read to indicate an invalid verification.
05H	ERASE COMMAND Results in an erase of the 4-byte page designated in EADRL.
06H	ERASE-ALL COMMAND Results in erase of the full Flash/EE data memory 160-page (640 bytes) array.
07H to FFH	RESERVED COMMANDS Commands reserved for future use.

Flash/EE Memory Timing
The typical program/erase times for the Flash/EE data memory are:

Erase Full Array (640 Bytes) – 20 ms
Erase Single Page (4 Bytes) – 20 ms
Program Page (4 Bytes) – 250 μs
Read Page (4 Bytes) – Within Single Instruction Cycle

Flash/EE erase and program timing is derived from the master clock. When using a master clock frequency of 11.0592 MHz, it is not necessary to write to the ETIM registers at all. However, when operating at other master clock frequencies (f_{CLK}), you must change the values of ETIM1 and ETIM2 to avoid degrading data Flash/EE endurance and retention. ETIM1 and ETIM2 form a 16-bit word, ETIM2 being the high byte and ETIM1 the low byte. The value of this 16-bit word must be set as follows to ensure optimum data Flash/EE endurance and retention.

$$ETIM2, ETIM1 = 100\ \mu s \times f_{CLK}$$

ETIM3 should always remain at its default value of 201 dec/C9 hex.

Using the Flash/EE Memory Interface
As with all Flash/EE memory architectures, the array can be programmed in system at a byte level, although it must be erased first, the erasure being performed in page blocks (4-byte pages in this case).

A typical access to the Flash/EE array will involve setting up the page address to be accessed in the EADRL SFR, configuring the EDATA1–4 with data to be programmed to the array (the EDATA SFRs will not be written for read accesses), and finally writing the ECON command word that initiates one of the six modes shown in Table VII. 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 ADuC812 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 (250 μs or 20 ms later). 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 pseudo-idle period.

Erase-All
Although the 640-byte user Flash/EE array is 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 ADuC812. An ERASE-ALL command consists of writing 06H to the ECON SFR, which initiates an erase of all 640 byte locations in the Flash/EE array. This command coded in 8051 assembly would appear as:

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

Program a Byte
In general terms, 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 erasure must happen at a page level; therefore, a minimum of four bytes (1 page) will be erased when an erase command is initiated. A more specific example of the Program-Byte process is shown below. In this example, the user writes 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. As the user is only required to modify one of the page bytes, the full page must be first read so that this page can then be erased without the existing data being lost. This example, coded in 8051 assembly, would appear as:

```
MOV  EADRL, #03H    ; Set Page Address Pointer
MOV  ECON, #01H     ; Read Page
MOV  EDATA2, #0F3H  ; Write New Byte
MOV  ECON, #05H     ; Erase Page
MOV  ECON, #02H     ; Write Page (Program
                      Flash/EE)
```

USER INTERFACE TO OTHER ON-CHIP ADuC812 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 ADuC812 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.

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

MODE	RNG1	RNG0	CLR1	CLR0	SYNC	$\overline{PD1}$	PD0

Table VIII. 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 eight 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	$\overline{PD1}$	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 DACL contains the lower eight bits, and the lower nibble of DACH 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 18. 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.

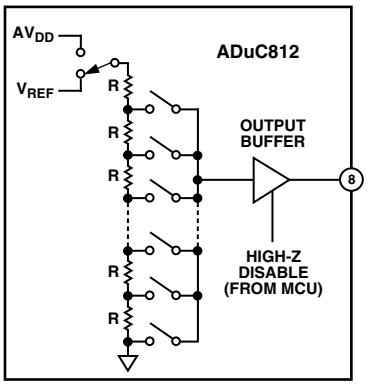


Figure 18. Resistor String DAC Functional Equivalent

As illustrated in Figure 18, 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 48, 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 19. The dotted line in Figure 19 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 19 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.

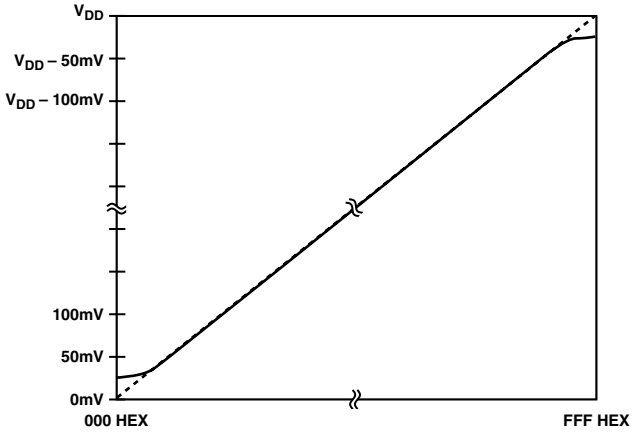


Figure 19. Endpoint Nonlinearities Due to Amplifier Saturation

The endpoint nonlinearities conceptually illustrated in Figure 19 get worse as a function of output loading. Most of the ADuC812's data sheet 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 19 become larger. With larger current demands, this can significantly limit output voltage swing. Figure 20 and Figure 21 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 high-side voltage drops as long as the reference voltage remains below the upper trace in the corresponding figure. For example, if AV_{DD} = 3 V and V_{REF} = 2.5 V, the high-side voltage will not be affected by loads less than 5 mA. But somewhere around 7 mA the upper curve in Figure 21 drops below 2.5 V (V_{REF}), indicating that at these higher currents the output will not be capable of reaching V_{REF}.

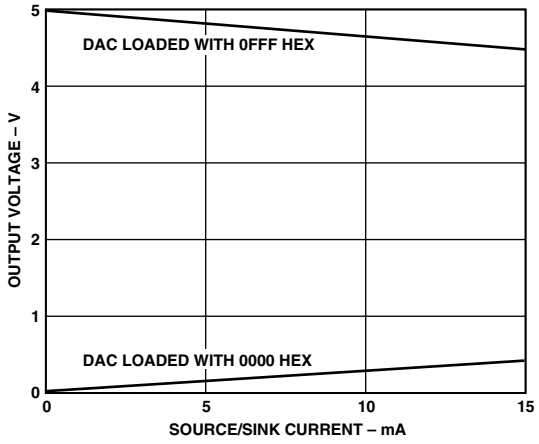


Figure 20. Source and Sink Current Capability with $V_{REF} = V_{DD} = 5$ V

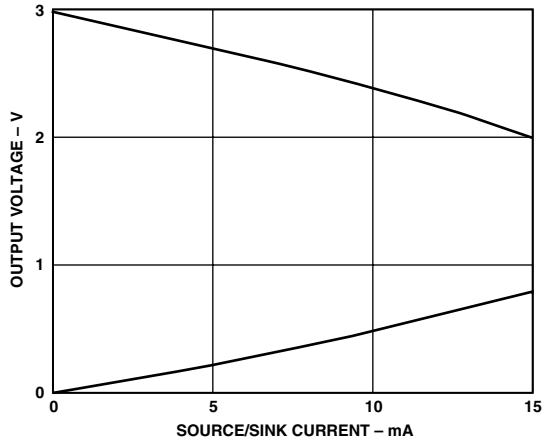


Figure 21. Source and Sink Current Capability with $V_{REF} = V_{DD} = 3$ V

To drive significant loads with the DAC outputs, external buffering may be required, as illustrated in Figure 22.

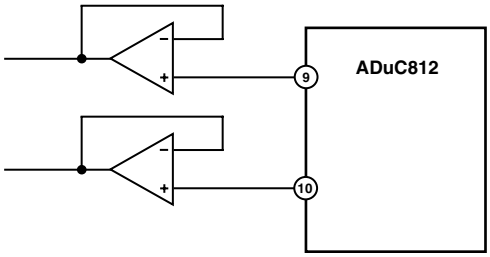


Figure 22. Buffering the DAC Outputs

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. However, each DAC output will still spike briefly when power is first applied to the chip, and again when each DAC is first enabled in software. Typical scope shots of these spikes are given in Figure 23 and Figure 24, respectively.

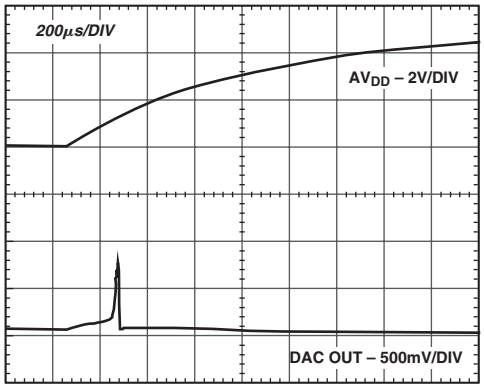


Figure 23. DAC Output Spike at Chip Power-Up

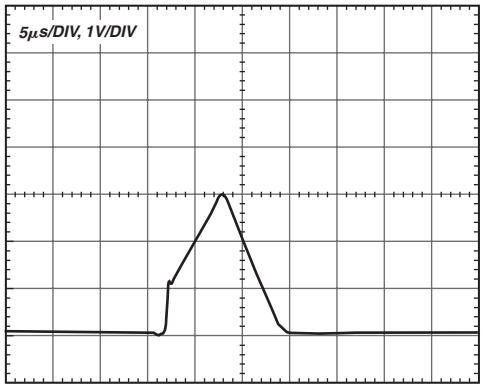


Figure 24. DAC Output Spike at DAC Enable

WATCHDOG TIMER

The purpose of the watchdog timer is to generate a device reset within a reasonable amount of time if the ADuC812 enters an erroneous state, possibly due to a programming error. 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 if the user program fails to set the watchdog timer refresh bits (WDR1, WDR2) within a predetermined amount of time (see PRE2–0 bits in WDCON). The watchdog timer itself is a 16-bit counter. The watchdog timeout interval can be adjusted via the PRE2–0 bits in WDCON. Full Control and Status of the watchdog timer function can be controlled via the watchdog timer control SFR (WDCON).

	Watchdog Timer
WDCON	**Control Register**
SFR Address	C0H
Power-On Default Value	00H
Bit Addressable	Yes

PRE2	PRE1	PRE0	—	WDR1	WDR2	WDS	WDE

Table IX. WDCON SFR Bit Designations

Bit	Name	Description
7	PRE2	Watchdog Timer Prescale Bits.
6	PRE1	
5	PRE0	PRE2 PRE1 PRE0 Timeout Period (ms) 0 0 0 16 0 0 1 32 0 1 0 64 0 1 1 128 1 0 0 256 1 0 1 512 1 1 0 1024 1 1 1 2048
4	—	Not Used.
3	WDR1	Watchdog Timer Refresh Bits. Set sequentially to refresh the watchdog timer.
2	WDR2	
1	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.
0	WDE	Watchdog Enable Bit. Set by user to enable the watchdog and clear its counters.

Example

To set up the watchdog timer for a timeout period of 2048 ms, the following code would be used:

```
MOV    WDCON,#0E0h  ;2.048 second
                    ;timeout period
SETB   WDE          ;enable watchdog timer
```

To prevent the watchdog timer from timing out, the timer refresh bits need to be set before 2.048 seconds has elapsed.

```
SETB   WDR1       ;refresh watchdog timer..
SETB   WDR2       ; ..bits must be set in this
                  ;order
```

POWER SUPPLY MONITOR

As its name suggests, the Power Supply Monitor, once enabled, monitors both supplies (AV_{DD} and DV_{DD}) on the ADuC812. It will indicate when either power supply drops below one of five user selectable voltage trip points from 2.63 V to 4.63 V. For correct operation of the Power Supply Monitor function, AV_{DD} must be equal to or greater than 2.7 V. The Power Supply Monitor function is controlled via the PSMCON SFR. If enabled via the IE2 SFR, the Power Supply 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 256 ms. This ensures that the power supply has fully settled before the bit is cleared. 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.

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

—	CMP	PSMI	TP2	TP1	TP0	PSF	PSMEN

Table X. PSMCON SFR Bit Designations

Bit	Name	Description
7	—	Not Used.
6	CMP	AV_{DD} and DV_{DD} Comparator Bit. This is a read-only bit and directly reflects the state of the AV_{DD} and DV_{DD} comparators. Read "1" indicates that both the AV_{DD} and DV_{DD} supplies are above their selected trip points. Read "0" indicates that either the AV_{DD} or 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 CMP is low, indicating low analog or digital supply. The PSMI bit can be used to interrupt the processor. Once CMPD and/or CMP return (and remain) high, a 256 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	TP2	V_{DD} Trip Point Selection Bits.
3	TP1	
2	TP0	These bits select the AV_{DD} and DV_{DD} trip point voltage as follows: TP2 TP1 TP0 Selected DV_{DD} Trip Point (V) 0 0 0 4.63 0 0 1 4.37 0 1 0 3.08 0 1 1 2.93 1 0 0 2.63
1	PSF	AV_{DD}/DV_{DD} Fault Indicator. Read "1" indicates that the AV_{DD} supply caused the fault condition. Read "0" indicates that the DV_{DD} supply caused the fault condition.
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.

Example

To configure the PSM for a trip point of 4.37 V, the following code would be used:

```
MOV     PSMCON,#005h     ;enable PSM with
                         ;4.37V threshold
SETB    EA               ;enable interrupts
MOV     IE2,#002h        ;enable PSM
                         ;interrupt
```

If the supply voltage falls below this level, the PC would vector to the ISR.

```
ORG          0043h            ;PSM ISR
CHECK:MOV    A,PSMCON         ;PSMCON.5 is the
                              ;PSM interrupt
                              ;bit..
JB           ACC.5,CHECK      ;..it is cleared
                              ;only when Vdd
                              ;has remained
                              ;above the trip
                              ;point for 256ms
                              ;or more.
RETI         ; return only when "all's well"
```

SERIAL PERIPHERAL INTERFACE

The ADuC812 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 interface, and therefore the user can only enable one or the other interface at any given time (see SPE in Table XI). 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 XI). 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. 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 ADuC812 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	OOH
Bit Addressable	Yes

ISPI	WCOL	SPE	SPIM	CPOL	CPHA	SPR1	SPR0

Table XI. 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 I^2C interface.
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 (bit rate) in Master mode as follows: SPR1 SPR0 Selected Bit Rate 0 0 $f_{OSC}/4$ 0 1 $f_{OSC}/8$ 1 0 $f_{OSC}/32$ 1 1 $f_{OSC}/64$ 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 XI, the ADuC812 SPI interface will transmit or receive data in a number of possible modes. Figure 25 shows all possible ADuC812 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.

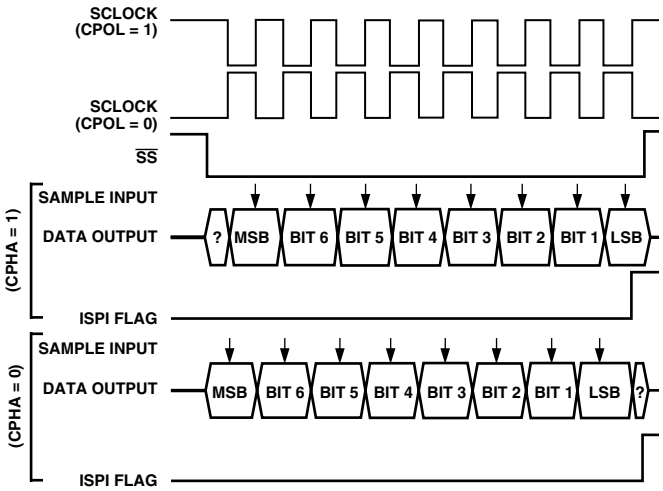


Figure 25. SPI Timing, All Modes

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 ADuC812 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.

I²C* COMPATIBLE INTERFACE

The ADuC812 supports a 2-wire serial interface mode that is I²C compatible. The I²C compatible interface shares its pins with the on-chip SPI interface and therefore the user can only enable one or the other interface at any given time (see SPE in Table IX). An application note describing the operation of this interface as implemented is available from the MicroConverter website at www.analog.com/microconverter. This interface can be configured as a software master or hardware slave, and uses two pins in the interface.

SDATA	Serial Data I/O Pin
SCLOCK	Serial Clock

Three SFRs are used to control the I²C compatible interface. These are described below:

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

MDO	MDE	MCO	MDI	I2CM	I2CRS	I2CTX	I2CI

Table XII. I2CCON SFR Bit Designations

Bit	Name	Description
7	MDO	I²C Software Master Data Output Bit (Master Mode Only). This data bit is used to implement a master I²C transmitter interface in software. Data written to this bit will be output on the SDATA pin if the data output enable (MDE) bit is set.
6	MDE	I²C Software Master Data Output Enable Bit (Master Mode Only). Set by the user to enable the SDATA pin as an output (Tx). Cleared by the user to enable SDATA pin as an input (Rx).
5	MCO	I²C Software Master Data Output Bit (Master Mode Only). This data bit is used to implement a master I²C transmitter interface in software. Data written to this bit will be output on the SCLOCK pin.
4	MDI	I²C Software Master Data Input Bit (Master Mode Only). This data bit is used to implement a master I²C receiver interface in software. Data on the SDATA pin is latched into this bit on SCLOCK if the Data Output Enable (MDE) = 0.
3	I2CM	I²C Master/Slave Mode Bit. Set by user to enable I²C software master mode. Cleared by user to enable I²C hardware slave mode.
2	I2CRS	I²C Reset Bit (Slave Mode Only). Set by user to reset the I²C interface. Cleared by user for normal I²C operation.
1	I2CTX	I²C 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	I²C Interrupt Bit (Slave Mode Only). Set by the MicroConverter after a byte has been transmitted or received. Cleared by user software.

I2CADD	**I²C Address Register**
Function	Holds the I²C peripheral address for the part. It may be overwritten by the user code. Application note uC001 at www.analog.com/microconverter describes the format of the I²C standard 7-bit address in detail.
SFR Address	9BH
Power-On Default Value	55H
Bit Addressable	No

I2CDAT	**I²C Data Register**
Function	The I2CDAT SFR is written by the user to transmit data over the I²C interface or read by user code to read data just received by the I²C interface. User software should only access I2CDAT once per interrupt cycle.
SFR Address	9AH
Power-On Default Value	00H
Bit Addressable	No

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

8051 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 fully 8051 compatible and are controlled via standard 8051 SFR bit definitions.

Parallel I/O Ports 0–3

The ADuC812 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; others are multiplexed with an alternate function 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 is an 8-bit, open-drain, bidirectional I/O port that is directly controlled via the P0 SFR (SFR address = 80H). 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 that state, Port 0 pins can be used as high impedance inputs. An external pull-up resistor will be required on Port 0 outputs to force a valid logic high level externally. 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.

Port 1 is also an 8-bit port directly controlled via the P1 SFR (SFR address = 90H). Port 1 is an input only port. 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 XIII.

Table XIII. 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	$\overline{SS}$ (Slave Select for the SPI Interface)

Port 2 is a bidirectional port with internal pull-up resistors directly controlled via the P2 SFR (SFR address = A0H). 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 24-bit external data memory space.

Port 3 is a bidirectional port with internal pull-ups directly controlled via the P3 SFR (SFR address = B0H). 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 also have various secondary functions described in Table XIV.

Table XIV. 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	$\overline{INT0}$ (External Interrupt 0)
P3.3	$\overline{INT1}$ (External Interrupt 1)
P3.4	T0 (Timer/Counter 0 External Input)
P3.5	T1 (Timer/Counter 1 External Input)
P3.6	$\overline{WR}$ (External Data Memory Write Strobe)
P3.7	$\overline{RD}$ (External Data Memory Read Strobe)

The alternate functions of P1.0, P1.1, P1.5, and Port 3 pins can be activated only if the corresponding bit latch in the P1 and P3 SFRs contains a 1. Otherwise, the port pin is stuck at 0.

Timers/Counters

The ADuC812 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, think of it as counting machine cycles. Since a machine cycle consists of 12 core clock periods, the maximum count rate is 1/12 of 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 of 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

Gate	C/T	M1	M0	Gate	C/T	M1	M0

Table XV. 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 Autoreload Timer/Counter. TH1 holds a value that is to be reloaded into TL1 each time it overflows. 1 1 Timer/Counter 1 Stopped.
3	Gate	Timer 0 Gating Control. Set by software to enable Timer/Counter 0 only while $\overline{INT0}$ 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 5-bit prescaler. 0 1 16-Bit Timer/Counter. TH0 and TL0 are cascaded; there is no prescaler. 1 0 8-Bit Autoreload Timer/Counter. TH0 holds a value that is to be reloaded into TL0 each time it overflows. 1 1 TL0 is an 8-bit timer/counter controlled by the standard timer 0 control bits. TH0 is an 8-bit timer only, controlled by Timer 1 control bits.

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

TF1	TR1	TF0	TR0	IE1*	IT1*	IE0*	IT0*

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

Table XVI. 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 user to turn on Timer/Counter 1. Cleared by 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 user to turn on Timer/Counter 0. Cleared by 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 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).

Timer/Counters 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/COUNTERS 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 26 shows Mode 0 operation.

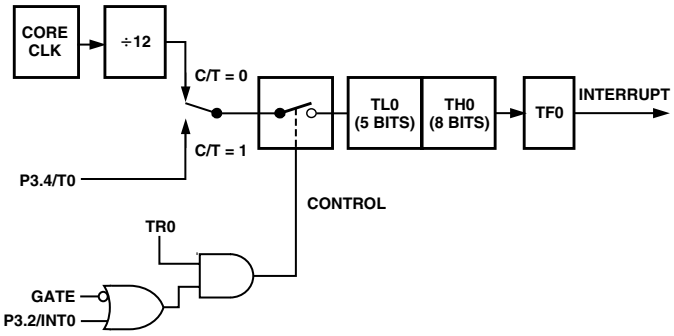


Figure 26. Timer/Counter 0, Mode 0

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

Mode 1 (16-Bit Timer/Counter)

Mode 1 is the same as Mode 0, except that the timer register is running with all 16 bits. Mode 1 is shown in Figure 27.

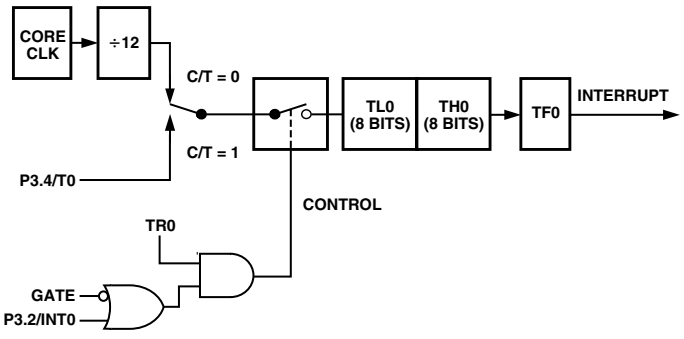


Figure 27. Timer/Counter 0, Mode 1

Mode 2 (8-Bit Timer/Counter with Auto Reload)

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in Figure 28. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is preset by software. The reload leaves TH0 unchanged.

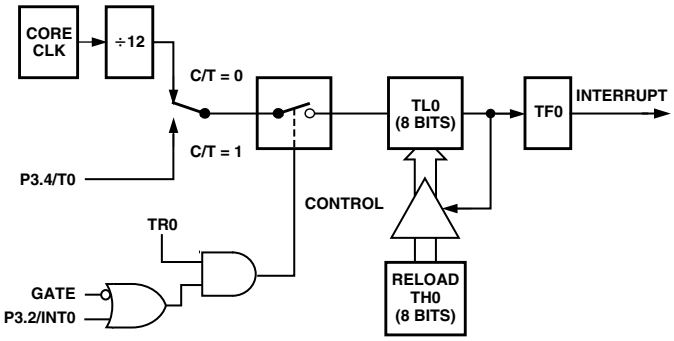


Figure 28. Timer/Counter 0, Mode 2

Mode 3 (Two 8-Bit Timer/Counters)

Mode 3 has different effects on Timer 0 and Timer 1. Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. This configuration is shown in Figure 29. TL0 uses the Timer 0 control bits: C/T, Gate, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt. Mode 3 is provided for applications requiring an extra 8-bit timer or counter.

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

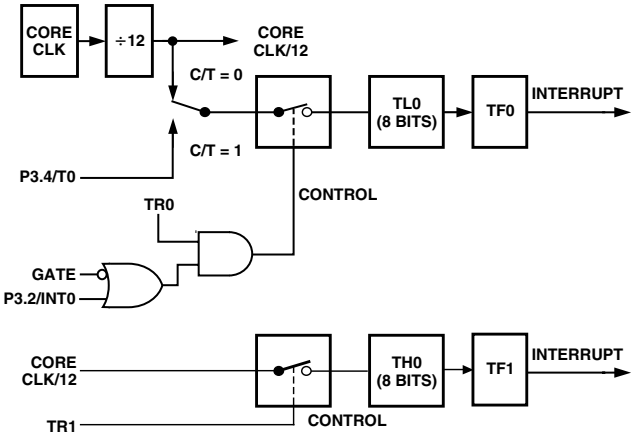


Figure 29. Timer/Counter 0, Mode 3

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

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	CNT2	CAP2

Table XVII. 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 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 user to enable Timer 1 overflow to be used for the receive clock.
4	TCLK	Transmit Clock Enable Bit. Set by 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 user to enable Timer 1 overflow to be used for the transmit clock.
3	EXEN2	Timer 2 External Enable Flag. Set by 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 user for Timer 2 to ignore events at T2EX.
2	TR2	Timer 2 Start/Stop Control Bit. Set by user to start Timer 2. Cleared by 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 user to enable captures on negative transitions at T2EX if EXEN2 = 1. Cleared by 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 high 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 XVIII.

Table XVIII. TIMECON SFR Bit Designations

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 reload 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 30.

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, that 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 31.

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 however, the EXF2 flag 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.

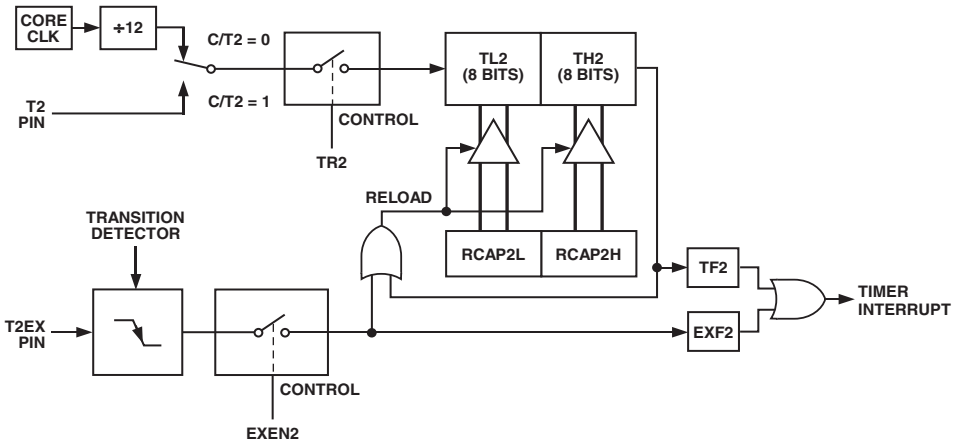


Figure 30. Timer/Counter 2, 16-Bit Autoreload Mode

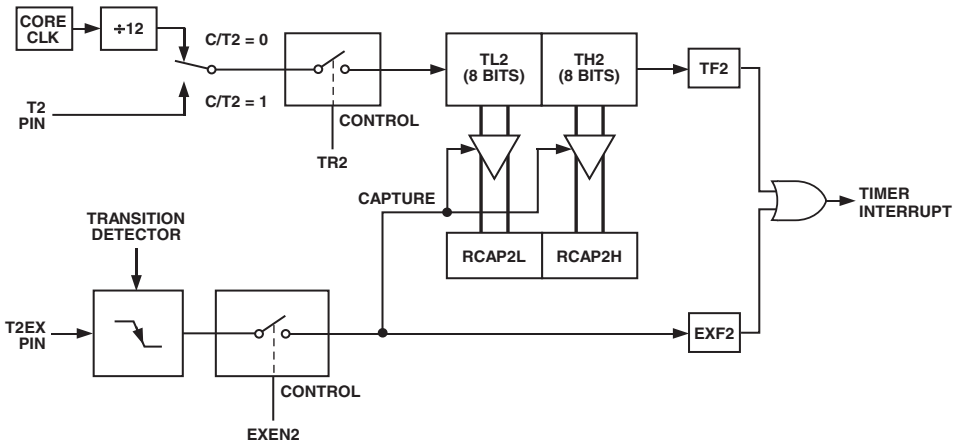


Figure 31. Timer/Counter 2, 16-Bit Capture Mode

UART SERIAL INTERFACE

The serial port is full-duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can begin receiving a second byte before a previously received byte has been read from the receive register. However, if the first byte still has not been read by the time reception of the second byte is complete, the first byte 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

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Table XIX. 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 32.

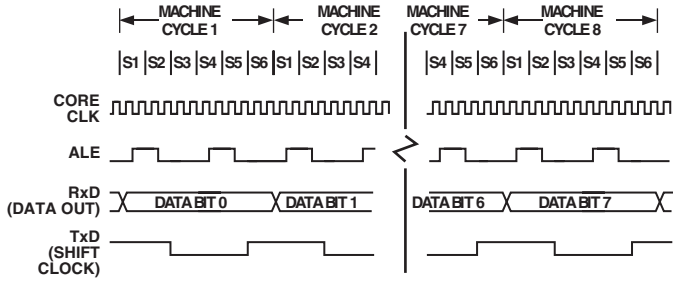


Figure 32. UART Serial Port Transmission, Mode 0

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 33.

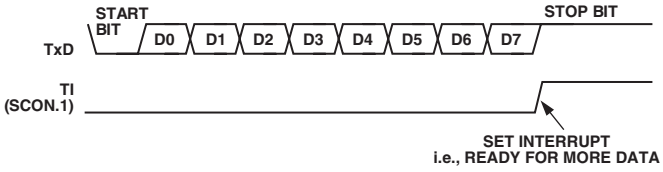


Figure 33. UART Serial Port Transmission, Mode 0

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:

$$Mode\ 0\ Baud\ Rate = \left(Core\ Clock\ Frequency/12\right)$$

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:

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

Mode 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:

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

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:

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

Table XX shows some commonly used baud rates and how they might be calculated from a core clock frequency of 11.0592 MHz and 12 MHz. Generally speaking, a 5% error is tolerable using asynchronous (start/stop) communications.

Table XX. Commonly Used Baud Rates, Timer 1

Ideal Baud	Core CLK	SMOD Value	TH1-Reload Value	Actual Baud	% Error
9600	12	1	–7 (F9H)	8929	7
19200	11.0592	1	–3 (FDH)	19200	0
9600	11.0592	0	–3 (FDH)	9600	0
2400	11.0592	0	–12 (F4H)	2400	0

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.

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

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. Therefore, it increments six times faster than Timer 1, and 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 34.

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

$$\textit{Modes 1 and 3 Baud Rate} = \left(\textit{Core Clk}\right)/\left(32 \times \left[65536 - \left(RCAP2H, RCAP2L\right)\right]\right)$$

Table XXI shows some commonly used baud rates and how they might be calculated from a core clock frequency of 11.0592 MHz and 12 MHz.

Table XXI. Commonly Used Baud Rates, Timer 2

Ideal Baud	Core CLK	RCAP2H Value	RCAP2L Value	Actual Baud	% Error
19200	12	–1 (FFH)	–20 (ECH)	19661	2.4
9600	12	–1 (FFH)	–41 (D7H)	9591	0.1
2400	12	–1 (FFH)	–164 (5CH)	2398	0.1
1200	12	–2 (FEH)	–72 (B8H)	1199	0.1
19200	11.0592	–1 (FFH)	–18 (EEH)	19200	0
9600	11.0592	–1 (FFH)	–36 (DCH)	9600	0
2400	11.0592	–1 (FFH)	–144 (70H)	2400	0
1200	11.0592	–2 (FFH)	–32 (E0H)	1200	0

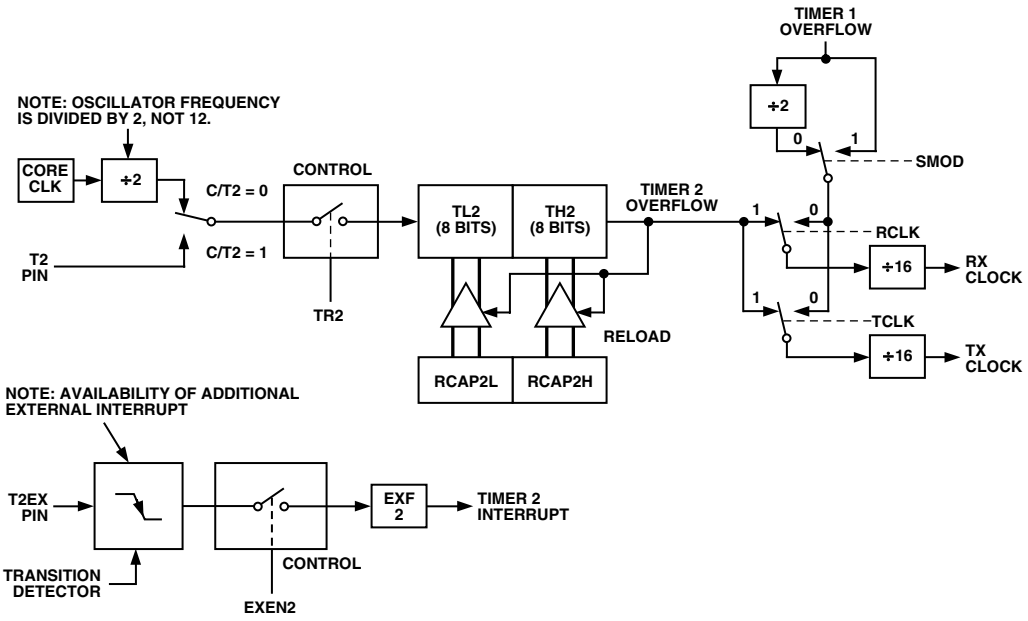


Figure 34. Timer 2, UART Baud Rates

INTERRUPT SYSTEM

The ADuC812 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
IE2	Secondary Interrupt Enable Register

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

EA	EADC	ET2	ES	ET1	EX1	ET0	EX0

Table XXII. 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

PSI	PADC	PT2	PS	PT1	PX1	PT0	PX0

Table XXIII. IP SFR Bit Designations

Bit	Name	Description
7	PSI	Written by user to select I^2C/SPI priority ("1" = High; "0" = Low).
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).

IE2 — **Secondary Interrupt Enable Register**

SFR Address A9H
Power-On Default Value 00H
Bit Addressable No

—	—	—	—	—	—	EPSMI	ESI

Table XXIV. IE2 SFR Bit Designations

Bit	Name	Description
7	—	Reserved for future use.
6	—	Reserved for future use.
5	—	Reserved for future use.
4	—	Reserved for future use.
3	—	Reserved for future use.
2	—	Reserved for future use.
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" I^2C/SPI 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 high priority may interrupt the service routine of a low priority interrupt. If two interrupts of different priorities occur at the same time, the higher level interrupt will be served 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 XXV.

Table XXV. Priority within an Interrupt Level

Source	Priority	Description
PSMI	1 (Highest)	Power Supply Monitor 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
I2CI + ISPI	7	I^2C/SPI Interrupt
RI + TI	8	Serial Interrupt
TF2 + EXF2	9 (Lowest)	Timer/Counter 2 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 the Table XXVI.

Table XXVI. Interrupt Vector Addresses

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

ADuC812 HARDWARE DESIGN CONSIDERATIONS

This section outlines some of the key hardware design considerations that must be addressed when integrating the ADuC812 into any hardware system.

Clock Oscillator

The clock source for the ADuC812 can come either from an external source or from the internal clock oscillator. To use the internal clock oscillator, connect a parallel resonant crystal between Pins 32 and 33, and connect a capacitor from each pin to ground as shown below.

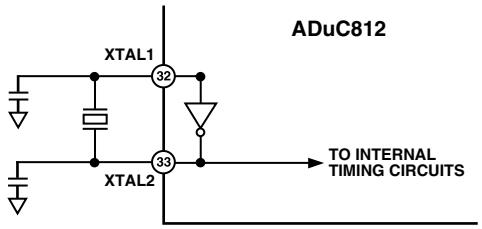


Figure 35. External Parallel Resonant Crystal Connections

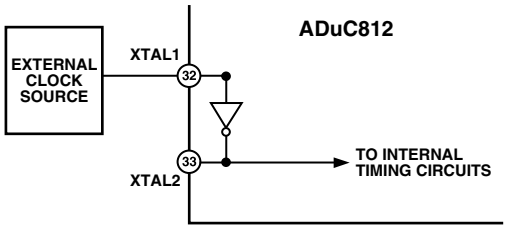


Figure 36. Connecting an External Clock Source

Whether using the internal oscillator or an external clock source, the ADuC812's specified operational clock speed range is 300 kHz to 16 MHz. The core 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 MHz.

External Memory Interface

In addition to its internal program and data memories, the ADuC812 can access up to 64 K bytes of external program memory (ROM, PROM, etc.) and up to 16 M bytes 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 8 K bytes 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. In either case, addresses above 1FFFH (8K) are mapped to the external space.

Note that 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 ADuC812 as illustrated in Figure 37. 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 ADuC812.

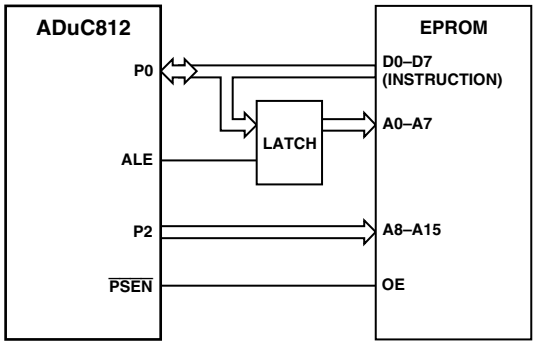


Figure 37. External Program Memory Interface

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 K bytes. 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 38 shows a hardware configuration for accessing up to 64 K bytes of external RAM. This interface is standard to any 8051 compatible MCU.

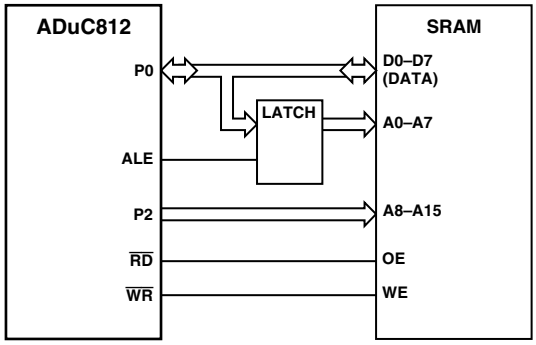


Figure 38. External Data Memory Interface (64K Address Space)

If access to more than 64K bytes of RAM is desired, a feature unique to the ADuC812 allows addressing up to 16 MBytes of external RAM simply by adding an additional latch as illustrated in Figure 39.

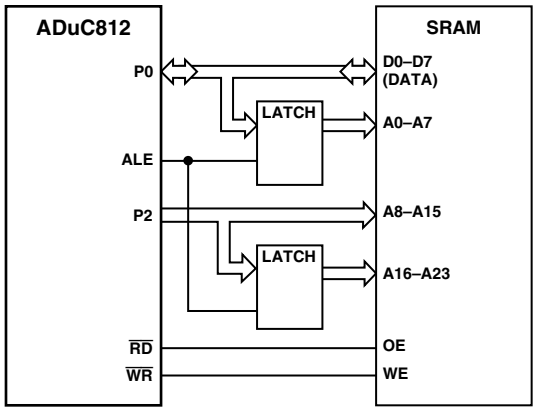


Figure 39. External Data Memory Interface (16 M Bytes Address Space)

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 ADuC812 (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 64K byte external data memory access is maintained.

Detailed timing diagrams of external program and data memory read and write access can be found in the Timing Specification sections.

Power-On Reset Operation

External POR (power-on reset) circuitry must be implemented to drive the RESET pin of the ADuC812. The circuit must hold the RESET pin asserted (high) whenever the power supply (DV_{DD}) is below 2.5 V. Furthermore, V_{DD} must remain above 2.5 V for at least 10 ms before the RESET signal is deasserted (low), by which time the power supply must have reached at least a 2.7 V level. The external POR circuit must be operational down to 1.2 V or less. The timing diagram in Figure 40 illustrates this functionality under three separate events: power-up, brownout, and power-down. Notice that when RESET is asserted (high), it tracks the voltage on DV_{DD}. These recommendations must be adhered to through the manufacturing flow of your ADuC812 based system as well as during its normal power-on operation. Failure to adhere to these recommendations can result in permanent damage to device functionality.

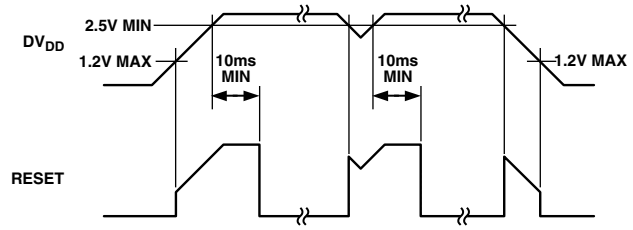


Figure 40. External POR Timing

The best way to implement an external POR function to meet the above requirements involves the use of a dedicated POR chip, such as the ADM809/ADM810 SOT-23 packaged PORs from Analog Devices. Recommended connection diagrams for both active high ADM810 and active low ADM809 PORs are shown in Figure 41 and Figure 42, respectively.

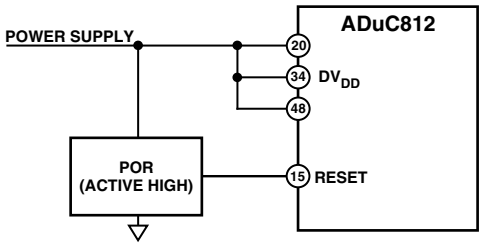


Figure 41. External Active High POR Circuit

Some active-low POR chips, such as the ADM809, can be used with a manual push-button as an additional reset source as illustrated by the dashed line connection in Figure 42.

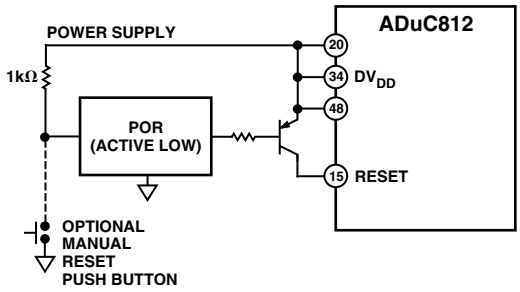


Figure 42. External Active Low POR Circuit

Power Supplies

The ADuC812'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.

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 43.

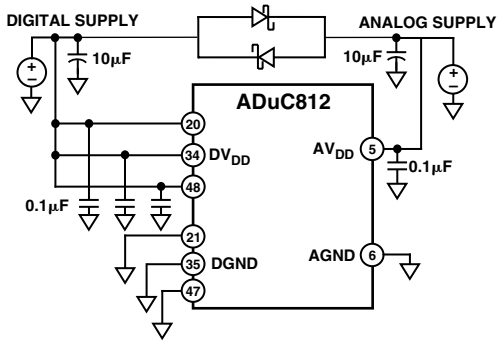


Figure 43. External Dual-Supply Connections

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 44. 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.

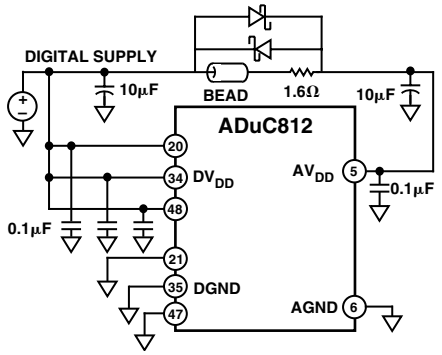


Figure 44. External Single-Supply Connections

Notice that in both Figure 43 and Figure 44, a large value (10 µF) reservoir capacitor sits on DV_{DD} and a separate 10 µF capacitor sits on AV_{DD}. Also, local small value (0.1 µF) capacitors are located at each V_{DD} pin of the chip. As per standard design practice, be sure to include all of these capacitors, and ensure the smaller capacitors are close to each AV_{DD} pin with trace lengths as short as possible. Connect the ground terminal of each of these capacitors directly to the underlying ground plane. Finally, it should also be noted that, at all times, the analog and digital ground pins on the ADuC812 must be referenced to the same system ground reference point.

Power Consumption

The currents consumed by the various sections of the ADuC812 are shown in Table XXVII. The CORE values given represent the current drawn by DV_{DD}, while the rest (ADC, DAC, Voltage Reference) are pulled by the AV_{DD} pin and can be disabled in software when not in use. The other on-chip peripherals (watchdog timer, power supply monitor, and so on) consume negligible current and are therefore lumped in with the CORE operating current here. Of course, the user must add any currents sourced by the DAC or the parallel and serial I/O pins, in order to determine the total current needed at the ADuC812's supply pins. Also, current drawn from the DV_{DD} supply will increase by approximately 10 mA during Flash/EE erase and program cycles.

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

	V_{DD} = 5 V	V_{DD} = 3 V
CORE (Normal Mode)	(1.6 nAs × MCLK) + 6 mA	(0.8 nAs × MCLK) + 3 mA
CORE (Idle Mode)	(0.75 nAs × MCLK) + 5 mA	(0.25 nAs × MCLK) + 3 mA
ADC	1.3 mA	1.0 mA
DAC (Each)	250 µA	200 µA
Voltage Ref	200 µA	150 µA

Since operating DV_{DD} current is primarily a function of clock speed, the expressions for CORE supply current in Table XXVII are given as functions of MCLK, the oscillator frequency. Plug in a value for MCLK in hertz to determine the current consumed by the core at that oscillator frequency. Since the ADC and DACs can be enabled or disabled in software, add only the currents from the peripherals you expect to use. The internal voltage reference is automatically enabled whenever either the ADC or at least one DAC is enabled. And again, do not forget to include current sourced by I/O pins, serial port pins, DAC outputs, and so forth, plus the additional current drawn during Flash/EE erase and program cycles.

A software switch allows the chip to be switched from normal mode into idle mode, and also into full power-down mode. Below are brief descriptions of power-down and idle modes.

In idle mode, the oscillator continues to run but is gated off to the core only. The on-chip peripherals continue to receive the clock, and remain functional. Port pins and DAC output pins retain their states in this mode. The chip will recover from idle mode upon receiving any enabled interrupt, or upon receiving a hardware reset.

In full power-down mode, the on-chip oscillator stops, and all on-chip peripherals are shut down. Port pins retain their logic levels in this mode, but the DAC output goes to a high impedance state (three-state). The chip will only recover from power-down mode upon receiving a hardware reset or when power is cycled. During full power-down mode, the ADuC812 consumes a total of approximately 5 µA.

Grounding and Board Layout Recommendations

As with all high resolution data converters, special attention must be paid to grounding and PC board layout of ADuC812 based designs in order to achieve optimum performance from the ADC and DACs.

Although the ADuC812 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 ADuC812, as illustrated in the simplified example of Figure 45a. In systems where digital and analog ground planes are connected together somewhere else (for example, at the system's power supply), they cannot be connected again near the ADuC812 since a ground loop would result. In these cases, tie the ADuC812's AGND and DGND pins all to the analog ground plane, as illustrated in Figure 45b. 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 ADuC812 can then be placed between the digital and analog sections, as illustrated in Figure 45c.

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 45b 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 45c. Whenever possible, avoid large discontinuities in the ground plane(s) (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 ADuC812's digital inputs, add a series resistor to each relevant line to keep rise and fall times longer than 5 ns at the ADuC812 input pins. A value of 100 or 200 is usually sufficient to prevent high speed signals from coupling capacitively into the ADuC812 and affecting the accuracy of ADC conversions.

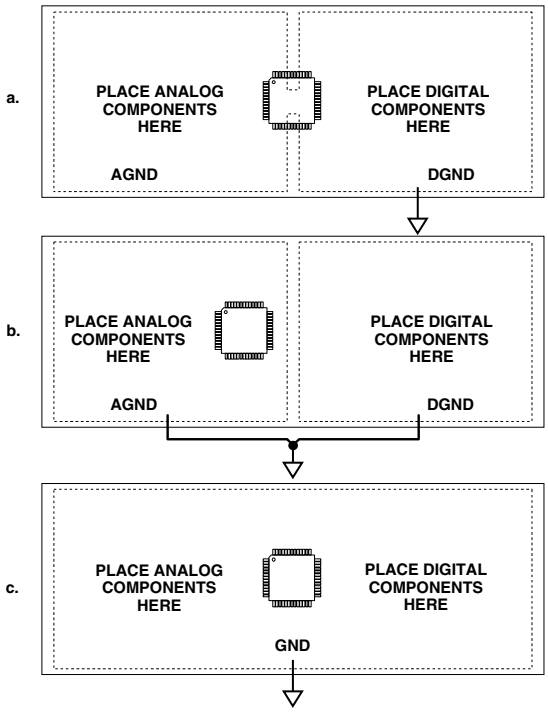


Figure 45. System Grounding Schemes

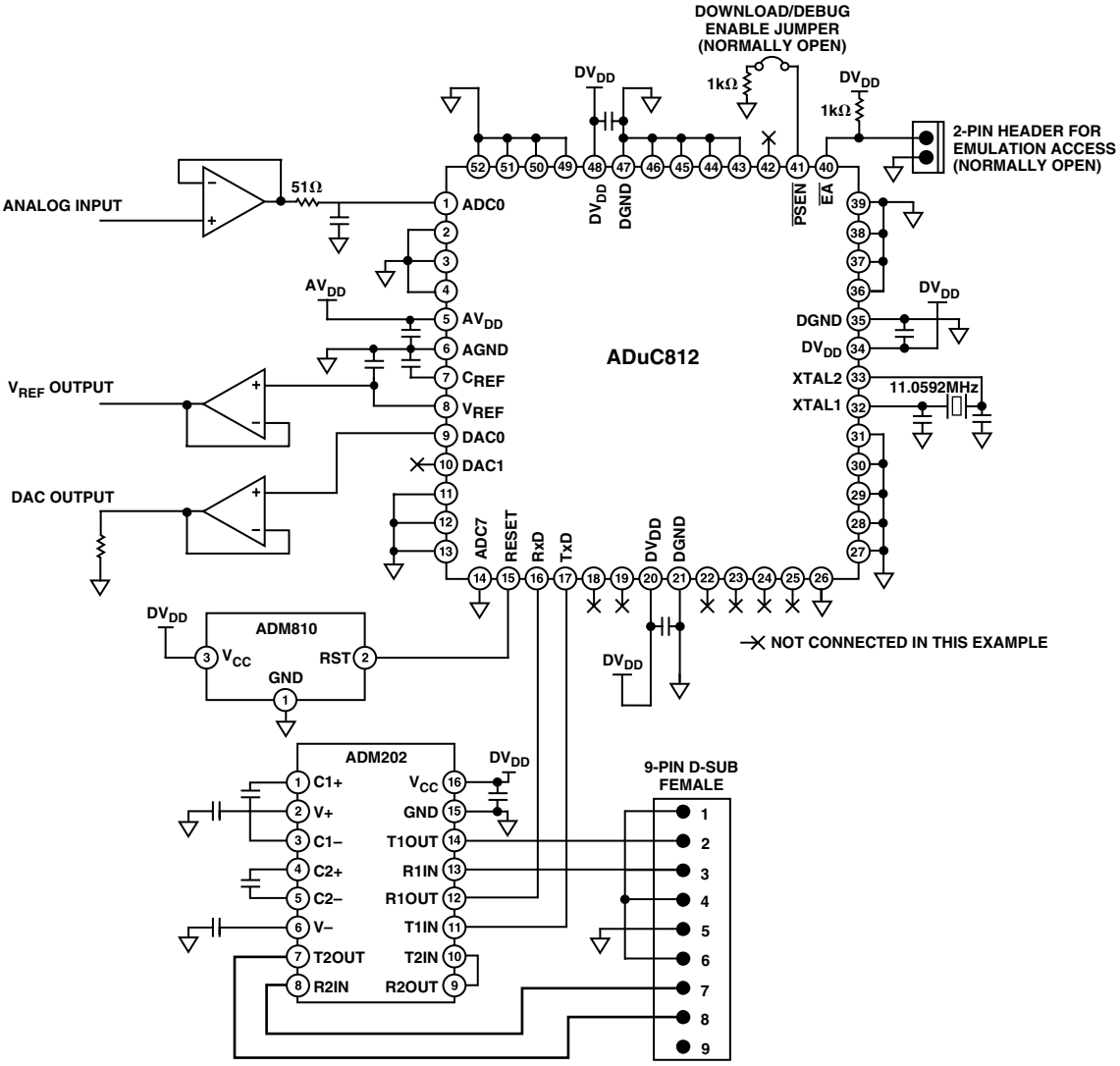


Figure 46. Typical System Configuration

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 ADuC812 designs will want to take advantage of the in-circuit reprogrammability of the chip. This is accomplished by a connection to the ADuC812'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 46 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, (available at www.analog.com/microconverter) for a simple (and zero-cost-per-board) method of gaining in-circuit serial download access to the ADuC812.

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 k pull-down resistor that can be jumpered onto the $\overline{\text{PSEN}}$ pin, as shown in Figure 46. To get the ADuC812 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.

Embedded Serial Port Debugger

From a hardware perspective, entry to 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 ADuC812 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 ADuC812 is a dedicated controller for single-pin in-circuit emulation (ICE) using standard production ADuC812 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 k resistor, as shown in Figure 46. The emulator will then connect to the 2-pin header also shown in Figure 46. To be compatible with the standard connector that comes with the single-pin emulator available from Accutron Limited (www.accutron.com), use a 2-pin 0.1 inch pitch "Friction Lock" header from Molex (www.molex.com) such as their part number 22-27-2021. Be sure to observe the polarity of this header. As represented in Figure 46, when the Friction Lock tab is at the right, the ground pin should be the lower of the two pins (when viewed from the top).

Enhanced-Hooks Emulation Mode
ADuC812 also supports enhanced-hooks emulation mode. An enhanced-hooks based emulator is available from Metalink Corporation (www.metaice.com). No special hardware support for these emulators needs to be designed onto the board since these are pod-style emulators where users must replace the chip on their board with a header device that the emulator pod plugs into. The only hardware concern is then one of determining if adequate space is available for the emulator pod to fit into the system enclosure.

Typical System Configuration
A typical ADuC812 configuration is shown in Figure 46. It summarizes some of the hardware considerations discussed in the previous paragraphs.

QUICKSTART DEVELOPMENT SYSTEM

The QuickStart Development System is a full featured, low cost development tool suite supporting the ADuC812. The system consists of the following PC based (Windows® compatible) hardware and software development tools.

Hardware:	ADuC812 Evaluation Board, Plug-In Power Supply and Serial Port Cable
Code Development:	8051 Assembler
Code Functionality:	Windows Based Simulator
In-Circuit Code Download:	Serial Downloader
In-Circuit Debugger:	Serial Port Debugger
Miscellaneous Other:	CD-ROM Documentation and Two Additional Prototype Devices

Figure 47 shows the typical components of a QuickStart Development System. A brief description of some of the software tools components in the QuickStart Development System is given in the following sections.



Figure 47. Components of the QuickStart Development System

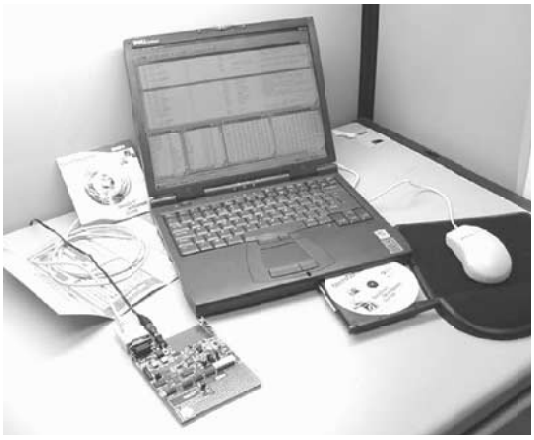


Figure 48. Typical Debug Session

Download—In-Circuit Serial Downloader
The Serial Downloader is a Windows application that allows the user to serially download an assembled program (Intel Hex format file) to the on-chip program FLASH memory via the serial COM1 port on a standard PC. Application Note uC004 detailing this serial download protocol is available at www.analog.com/microconverter.

DeBug—In-Circuit Debugger
The Debugger is a Windows application that allows the user to debug code execution on silicon using the MicroConverter UART serial port. The debugger provides access to all on-chip peripherals during a typical debug session as well as single-step and breakpoint code execution control.

ADSIM—Windows Simulator
The Simulator is a Windows application that fully simulates all the MicroConverter functionality including ADC and DAC peripherals. The simulator provides an easy-to-use, intuitive interface to the MicroConverter functionality and integrates many standard debug features including multiple breakpoints, single stepping, and code execution trace capability. This tool can be used both as a tutorial guide to the part as well as an efficient way to prove code functionality before moving to a hardware platform.

The QuickStart development tool suite software is freely available at the Analog Devices MicroConverter website, www.analog.com/microconverter.

TIMING SPECIFICATIONS[1, 2, 3] (AV_{DD} = DV_{DD} = 3.0 V or 5.0 V ± 10%. All specifications T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

Parameter		12 MHz Min	12 MHz Typ	12 MHz Max	Variable Clock Min	Variable Clock Typ	Variable Clock Max	Unit
CLOCK INPUT (External Clock Driven XTAL1)								
t_{CK}	XTAL1 Period		83.33		62.5		1000	ns
t_{CKL}	XTAL1 Width Low	20			20			ns
t_{CKH}	XTAL1 Width High	20			20			ns
t_{CKR}	XTAL1 Rise Time			20			20	ns
t_{CKF}	XTAL1 Fall Time			20			20	ns
t_{CYC}[4]	ADuC812 Machine Cycle Time		1			$12t_{CK}$		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.
[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.
[3]C_{LOAD} for Port 0, ALE, $\overline{PSEN}$ outputs = 100 pF; C_{LOAD} for all other outputs = 80 pF, unless otherwise noted.
[4]ADuC812 Machine Cycle Time is nominally defined as MCLKIN/12.

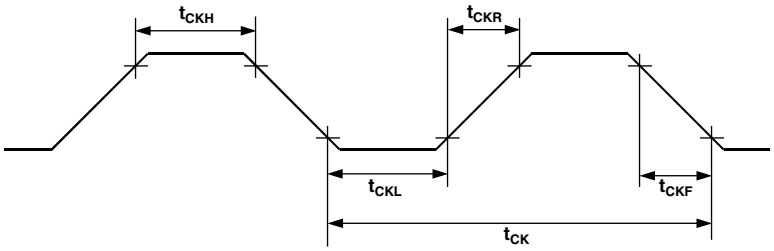


Figure 49. XTAL 1 Input

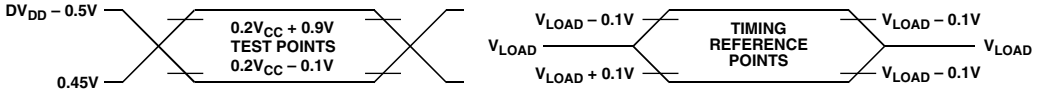


Figure 50. Timing Waveform Characteristics

Parameter		12 MHz Min	12 MHz Max	Variable Clock Min	Variable Clock Max	Unit
EXTERNAL PROGRAM MEMORY READ CYCLE						
t_{LHLL}	ALE Pulsewidth	127		$2t_{CK} - 40$		ns
t_{AVLL}	Address Valid to ALE Low	43		$t_{CK} - 40$		ns
t_{LLAX}	Address Hold after ALE Low	53		$t_{CK} - 30$		ns
t_{LLIV}	ALE Low to Valid Instruction In		234		$4t_{CK} - 100$	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	53		$t_{CK} - 30$		ns
t_{PLPH}	$\overline{PSEN}$ Pulsewidth	205		$3t_{CK} - 45$		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CK} - 105$	ns
t_{PXIX}	Input Instruction Hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float after $\overline{PSEN}$		59		$t_{CK} - 25$	ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CK} - 105$	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		25		25	ns
t_{PHAX}	Address Hold after $\overline{PSEN}$ High	0		0		ns

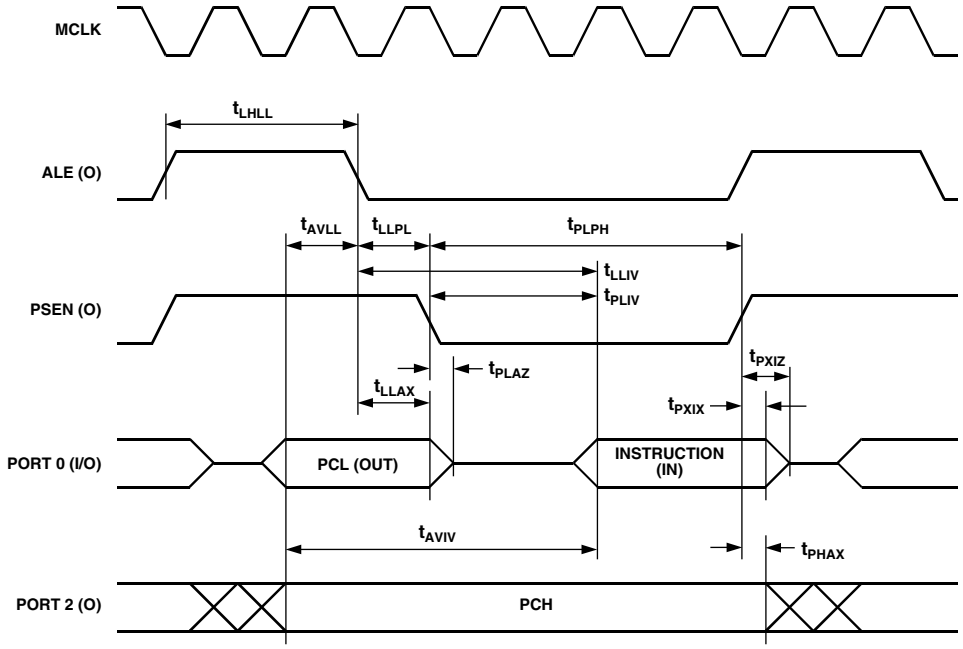


Figure 51. External Program Memory Read Cycle

Parameter		12 MHz Min	12 MHz Max	Variable Clock Min	Variable Clock Max	Unit
EXTERNAL DATA MEMORY READ CYCLE						
t_{RLRH}	$\overline{RD}$ Pulsewidth	400		$6t_{CK} - 100$		ns
t_{AVLL}	Address Valid after ALE Low	43		$t_{CK} - 40$		ns
t_{LLAX}	Address Hold after ALE Low	48		$t_{CK} - 35$		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CK} - 165$	ns
t_{RHDX}	Data and Address Hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float after $\overline{RD}$		97		$2t_{CK} - 70$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CK} - 150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CK} - 165$	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CK} - 50$	$3t_{CK} + 50$	ns
t_{AVWL}	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CK} - 130$		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	$t_{CK} - 40$	$6t_{CK} - 100$	ns

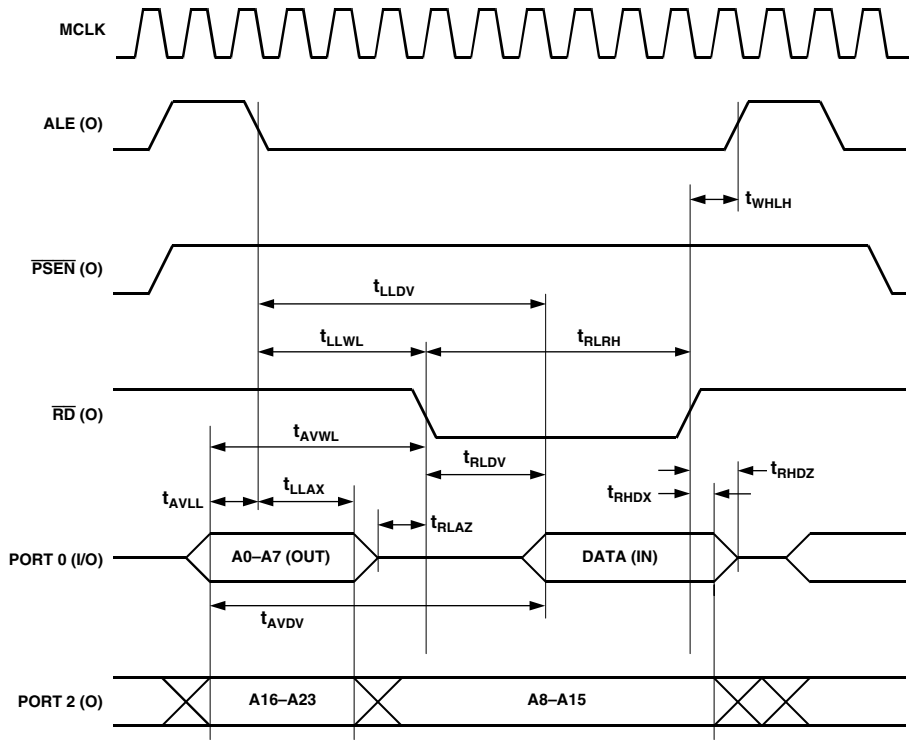


Figure 52. External Data Memory Read Cycle

Parameter		12 MHz Min	12 MHz Max	Variable Clock Min	Variable Clock Max	Unit
EXTERNAL DATA MEMORY WRITE CYCLE						
t_{WLWH}	$\overline{WR}$ Pulsewidth	400		$6t_{CK} - 100$		ns
t_{AVLL}	Address Valid after ALE Low	43		$t_{CK} - 40$		ns
t_{LLAX}	Address Hold after ALE Low	48		$t_{CK} - 35$		ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CK} - 50$	$3t_{CK} + 50$	ns
t_{AVWL}	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CK} - 130$		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	33		$t_{CK} - 50$		ns
t_{QVWH}	Data Setup before $\overline{WR}$	433		$7t_{CK} - 150$		ns
t_{WHQX}	Data and Address Hold after $\overline{WR}$	33		$t_{CK} - 50$		ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	$t_{CK} - 40$	$6t_{CK} - 100$	ns

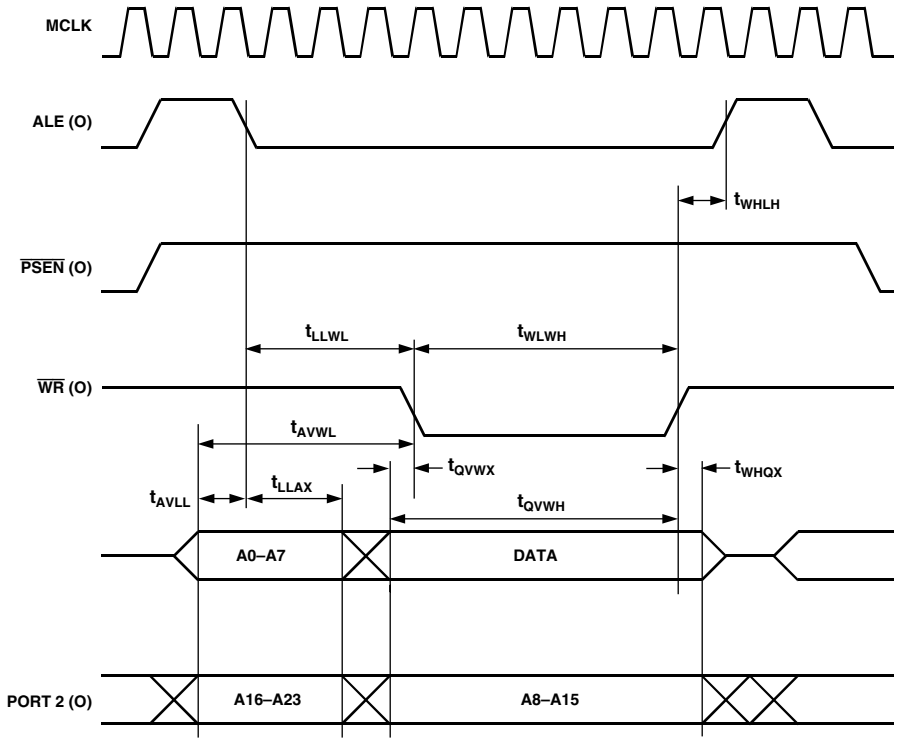


Figure 53. External Data Memory Write Cycle

Parameter		12 MHz Min	12 MHz Typ	12 MHz Max	Variable Clock Min	Variable Clock Typ	Variable Clock Max	Unit
UART TIMING (Shift Register Mode)								
t_{XLXL}	Serial Port Clock Cycle Time		1.0			$12t_{CK}$		μs
t_{QVXH}	Output Data Setup to Clock	700			$10t_{CK} - 133$			ns
t_{DVXH}	Input Data Setup to Clock	300			$2t_{CK} + 133$			ns
t_{XHDX}	Input Data Hold after Clock	0			0			ns
t_{XHQX}	Output Data Hold after Clock	50			$2t_{CK} - 117$			ns

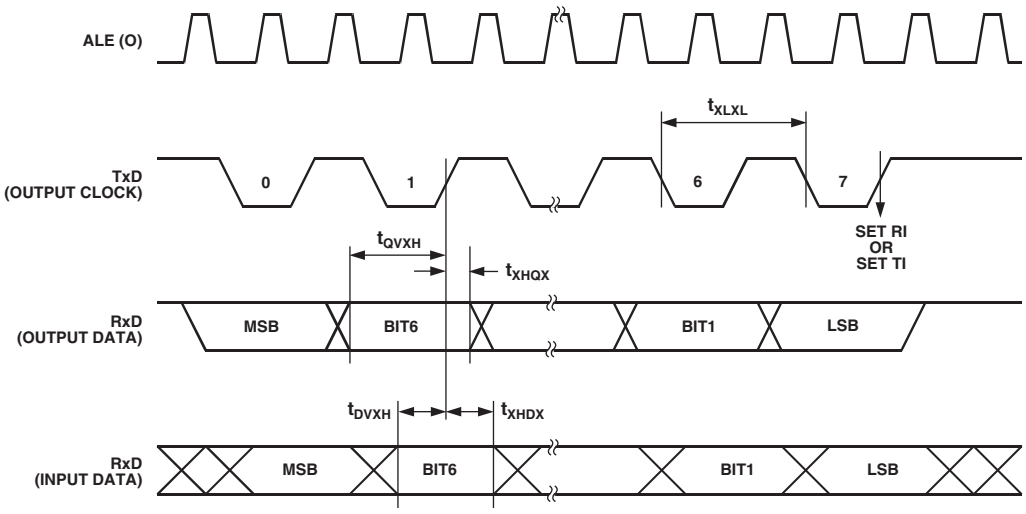


Figure 54. UART Timing in Shift Register Mode

Parameter		Min	Max	Unit
I^2C COMPATIBLE INTERFACE TIMING				
t_{LOW}	SCLOCK Low Pulsewidth	1.3		μs
t_{HIGH}	SCLOCK High Pulsewidth	0.6		μs
$t_{HD;\ STA}$	Start Condition Hold Time	0.6		μs
$t_{SU;\ DAT}$	Data Setup Time	100		μs
$t_{HD;\ DAT}$	Data Hold time	0	0.9	μs
$t_{SU;\ STA}$	Setup time for Repeated Start	0.6		μs
$t_{SU;\ STO}$	Stop Condition Setup Time	0.6		μs
t_{BUF}	Bus Free Time between a STOP Condition and a START Condition	1.3		μs
t_R	Rise Time for Both SCLOCK and SDATA		300	ns
t_F	Fall Time for Both SCLOCK and SDATA		300	ns
t_{SUP}[1]	Pulsewidth of Spike Suppressed		50	ns

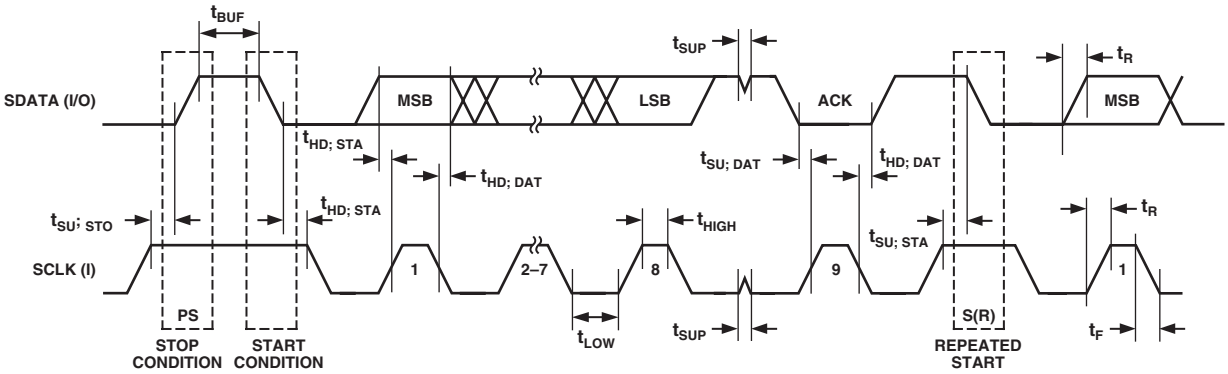


Figure 55. I^2C Compatible Interface Timing

Parameter		Min	Typ	Max	Unit
SPI MASTER MODE TIMING (CPHA = 1)					
t_{LOW}	SCLOCK Low Pulsewidth		330		ns
t_{SH}	SCLOCK High Pulsewidth		330		ns
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns
t_{DF}	Data Output Fall Time		10	25	ns
t_{DR}	Data Output Rise Time		10	25	ns
t_{SR}	SCLOCK Rise Time		10	25	ns
t_{SF}	SCLOCK Fall Time		10	25	ns

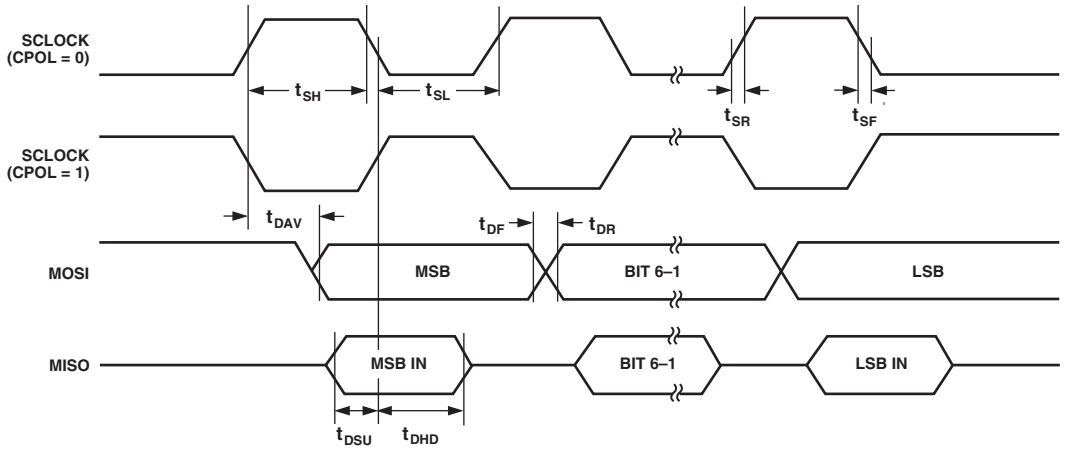


Figure 56. SPI Master Mode Timing (CPHA = 1)

Parameter		Min	Typ	Max	Unit
SPI MASTER MODE TIMING (CPHA = 0)					
t_{SL}	SCLOCK Low Pulsewidth		330		ns
t_{SH}	SCLOCK High Pulsewidth		330		ns
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns
t_{DOSU}	Data Output Setup before SCLOCK Edge			150	ns
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns
t_{DF}	Data Output Fall Time		10	25	ns
t_{DR}	Data Output Rise Time		10	25	ns
t_{SR}	SCLOCK Rise Time		10	25	ns
t_{SF}	SCLOCK Fall Time		10	25	ns

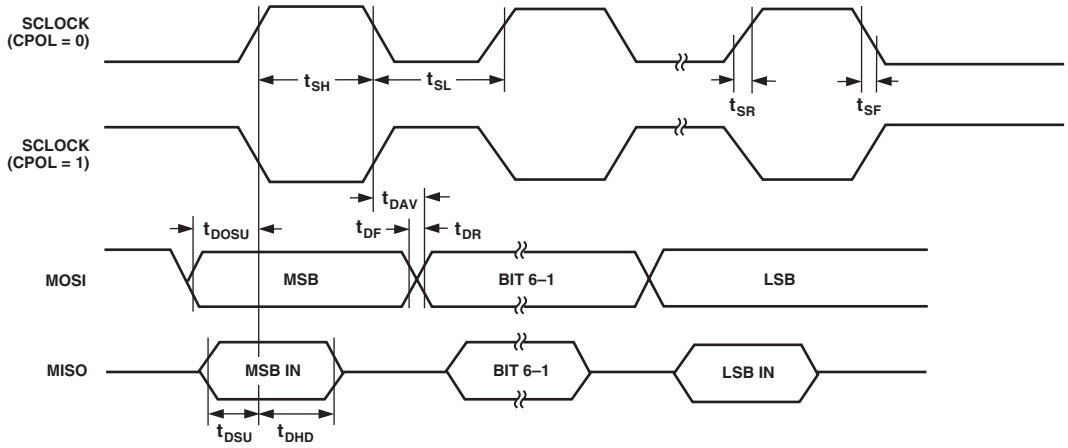


Figure 57. SPI Master Mode Timing (CPHA = 0)

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

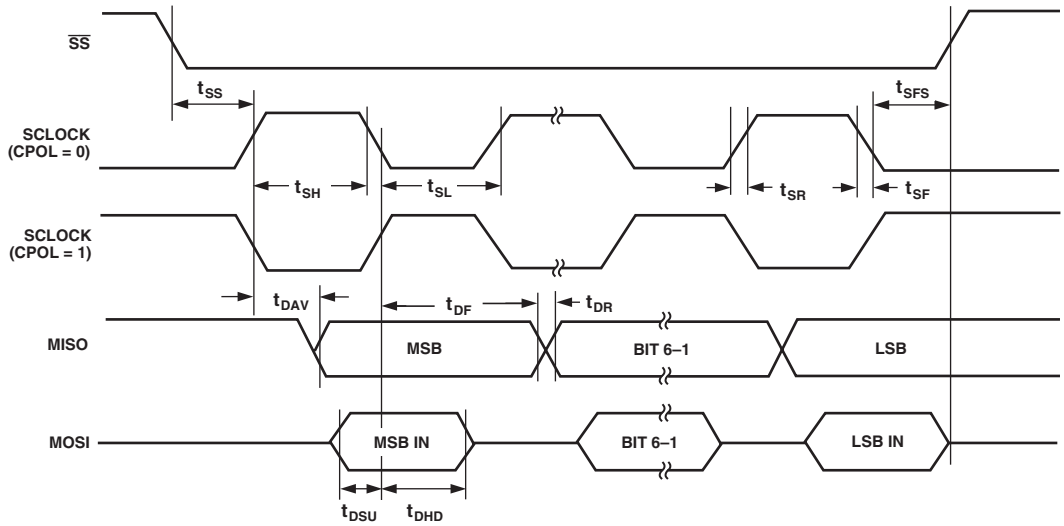


Figure 58. SPI Slave Mode Timing (CPHA = 1)

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

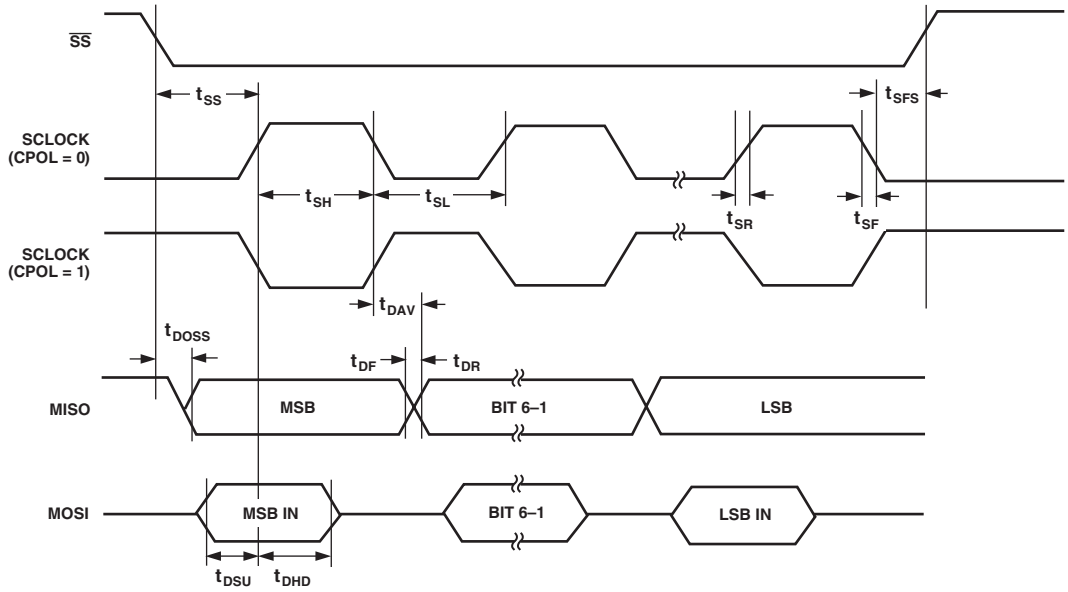
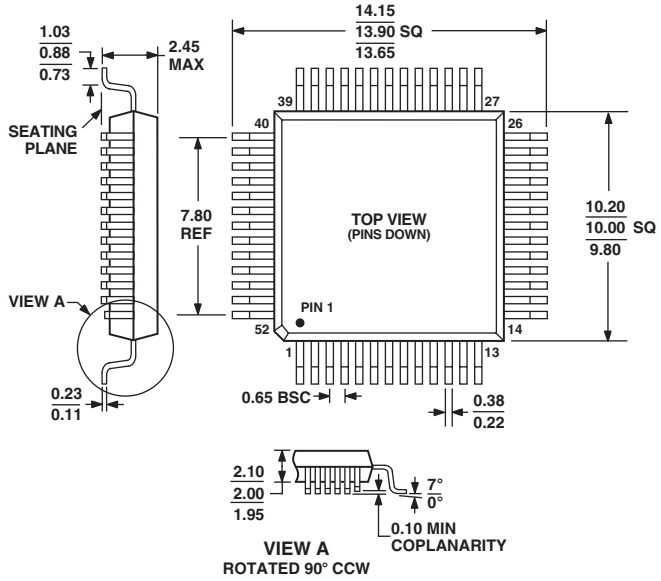


Figure 59. SPI Slave Mode Timing (CPHA = 0)

OUTLINE DIMENSIONS

52-Lead Metric Quad Flat Package [MQFP]
(S-52)

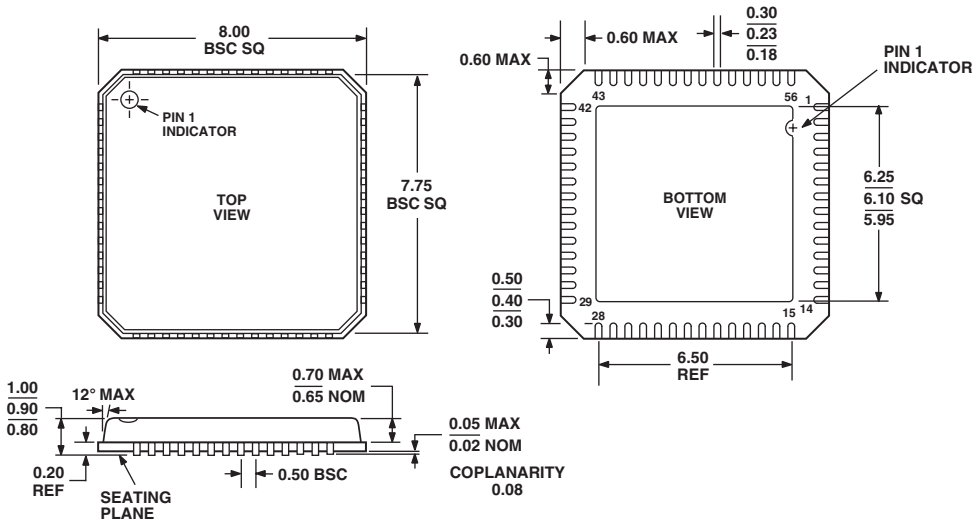
Dimensions shown in millimeters



COMPLIANT TO JEDEC STANDARDS MO-022-AC-1

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

Dimensions shown in millimeters



COMPLIANT TO JEDEC STANDARDS MO-220-VLLD-2

Revision History

Location	Page
4/03—Data Sheet changed from REV. D to REV. E.	
Updated OUTLINE DIMENSIONS	56
2/03—Data Sheet changed from REV. C to REV. D.	
Added CP-56 Package	Global
Edits to GENERAL DESCRIPTION	1
Added 56-Lead LFCSP PIN CONFIGURATION	6
Updated ORDERING GUIDE	6
Added I2C COMPATIBLE INTERFACE TIMING Table	51
Added new Figure 55	51
Updated OUTLINE DIMENSIONS	56
03/02—Data Sheet changed from REV. B to REV. C.	
Edits to FEATURES	1
Edits to GENERAL DESCRIPTION	1
Edits to FUNCTIONAL BLOCK DIAGRAM	1
Edits to SPECIFICATIONS	3
Edits to PIN CONFIGURATION	6
Edits to PIN FUNCTION DESCRIPTIONS	7
Edits to Figure 4	11
Edits to SERIAL PERIPHERAL INTERFACE Section	25
Edits to TABLE XI	26
Edits to TABLE XXIII	37
Edits to TABLES XXIV, XXV, and XXVI	38
10/01—Data Sheet changed from REV. A to REV. B.	
Entire Data Sheet Revised	All