

ADuC831**TABLE OF CONTENTS**

FEATURES	1	USER INTERFACE TO OTHER ON-CHIP	
GENERAL DESCRIPTION	1	ADuC831 PERIPHERALS	32
SPECIFICATIONS	3	Using the D/A Converter	33
ABSOLUTE MAXIMUM RATINGS	7	PWM	35
ORDERING GUIDE	7	SERIAL PERIPHERAL INTERFACE	38
BLOCK DIAGRAM	8	I2C PERIPHERAL INTERFACE	40
PIN FUNCTION DISCRIPTIONS	9	DUAL DATA POINTER	42
TERMINOLOGY	10	POWER SUPPLY MONITOR	43
TYPICAL PERFROMANCE CURVES	11	WATCHDOG TIMER	44
MEMORY ORGANIZATION	14	TIMER INTERVAL COUNTER	45
OVERVIEW OF MCU-RELATED SFRs	15	8051-COMPATIBLE ON-CHIP PERIPHERALS	47
Accumulator SFR	15	Parallel I/O Ports 0-3	47
B SFR	15	Timers/Counters	48
Stack Pointer SFR	15	UART SERIAL INTERFACE	53
Data Pointer	16	UART Operating Modes	54
Program Status Word SFR	16	Timer 1 Generated Baud Rates	55
Power Control SFR	16	Timer 2 Generated Baud Rates	55
SPECIAL FUNCTION REGISTERS	17	Timer 3 Generated Baud Rates	56
ADC CIRCUIT INFORMATION	18	INTERRUPT SYSTEM	57
General Overview	18	ADuC831 HARDWARE DESIGN CONSIDERATIONS .	59
ADC Transfer Function	18	Clock Oscillator	59
Typical Operation	18	External Memory Interface	59
ADCCON1 – (ADC Control SFR #1)	19	Power Supplies	60
ADCCON2 – (ADC Control SFR #2)	20	Power Consumption	61
ADCCON3 – (ADC Control SFR #3)	21	Power-Saving Modes	61
Driving the A/D Converter	22	Power On Reset	61
Voltage Reference Connections	23	Grounding and Board Layout Recommendations	62
Configuring the ADC	24	OTHER HARDWARE CONSIDERATIONS	63
ADC DMA Mode	24	In-Circuit Serial Download Access	63
Micro Operation during ADC DMA Mode	25	Embedded Serial Port Debugger	63
The Offset and Gain Calibration Coefficients	25	Single-Pin Emulation Mode	64
Calibration	25	Enhanced-Hooks Emulation Mode	64
NONVOLATILE FLASH MEMORY	27	Typical System Configuration	64
Flash Memory Overview	27	QUICKSTART DEVELOPMENT SYSTEM	64
Flash/EE Memory and the ADuC831	27	TIMING SPECIFICATIONS	65
ADuC831 Flash/EE Memory Reliability	27	OUTLINE DIMENSIONS	74
Using the Flash/EE Program Memory	27		
Security Bits Functionality	28		
Using the Flash/EE Data Memory	29		
ECON—Flash/EE Memory Control SFR	29		
ULOAD Mode	29		
Flash/EE Memory Timing	30		
ADuC831 Configuration Register (CFG831)	31		

ADuC831-SPECIFICATIONS1

(AV_{DD} = DV_{DD} = 2.7V to 3.3V or 4.5V to 5.5V.V_{REF} = 2.5 V Internal Reference, MCLKIN = 16 MHz, All specifications T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

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

ADuC831–SPECIFICATIONS1,2(continued)

Parameter	V _{DD} = 5 V	V _{DD} = 3 V	Unit	Test Conditions/Comments
DAC CHANNEL SPECIFICATIONS ^{12,13} Internal Buffer Disabled				DAC Load to AGND R _L = 10kΩ, C _L = 100 pF
DC ACCURACY ¹⁰ Resolution Relative Accuracy Differential Nonlinearity ¹¹ Offset Error Gain Error Gain Error Mismatch	12 ±3 -1 ±1/2 ±10 ±1 0.5	12 ±3 -1 ±1/2 ±10 ±1 0.5	Bits LSB typ LSB max LSB typ mV max % typ % typ	Guaranteed 12-Bit Monotonic V _{REF} Range VREF Range % of Full-Scale on DAC1
ANALOG OUTPUTS Voltage Range_0	0 to V _{REF}	0 to V _{REF}	V typ	DAC V _{REF} = 2.5V
REFERENCE INPUT/OUTPUT REFERENCE OUPUT ¹⁴ Output Voltage (V _{REF}) Accuracy Power Supply Rejection Reference Temperature Coefficient Internal V _{REF} Power-On Time	2.5 ±2.5 47 ±100 80	2.5 ±2.5 57 ±100 80	V % max dB typ ppm/°C typ ms typ	Of V _{REF} measured at the C _{REF} pin
EXTERNAL REFERENCE INPUT ¹⁵ Voltage Range (V _{REF}) ⁹ Input Impedance Input Leakage	0.1 V _{DD} 20 10	0.1 V _{DD} 20 10	V min V max kΩ typ μA max	Internal Band Gap Deselected via ADCCON1.6
POWER SUPPLY MONITOR (PSM) DV _{DD} Trip Point Selection Range DV _{DD} Power Supply Trip Point Accuracy	2.63 4.37 ±3.5		Vmin Vmax % max	Four Trip Points Selectable in This Range Programmed via TPD1-0 in PSMCON
WATCH DOG TIMER (WDT) ⁹ Time-out Period	0 2000	0 2000	ms min ms max.	Nine Time-out Periods Selectable in This Range
FLASH/EE MEMORY RELIABILITY CHARACTERISTICS ¹⁶ Endurance ¹⁷ Data Retention ¹⁸	100,000 100	100,000 100	Cycles min Years min	
DIGITAL INPUTS Input High Voltage (V _{INH}) Input Low Voltage (V _{INL}) Input Leakage Current (Port 0, EA) Logic 1 Input Current (All Digital Inputs) Logic 0 Input Current (Port 1, 2, 3) Logic 1-0 Transition Current (Port 2, 3)	2.4 0.8 ±10 ±1 ±10 ±1 -80 -40 -700 -400	±1 ±1 -40 -400	V min V max μA max μA typ μA max μA typ μA max μA typ μA max μA typ	V _{IN} = 0 V or V _{DD} V _{IN} = 0 V or V _{DD} V _{IN} = V _{DD} V _{IN} = V _{DD} V _{IL} = 0 V V _{IL} = 2 V V _{IL} = 2 V
CRYSTAL OSCILLATOR Logic Inputs, XTAL1 Only VINL, Input Low Voltage VINH, Input High Voltage	0.8 3.5	0.4 2.5	V typ V typ	

Parameter	V _{DD} =5V	V _{DD} =3V	Units	Test Conditions
XTAL1 Input Capacitance	18	18	pF typ	
XTAL2 Output Capacitance	18	18	pF typ	
MCU Clock Rate	16	16	MHz max	
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, 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	Vmax	I _{SINK} = 4 mA
SCLOCK/SDATA	0.4	0.4	Vmax	I _{SINK} = 8 mA
Floating State Leakage Current	±10	±10	µA max	
	±1	±1	µA typ	
Floating State Output Capacitance	10	10	pF typ	
START UP TIME				Core CLK = 16MHz
At Power-On	500	500	ms typ	
From Idle Mode	100	100	µs typ	
From Power-Down Mode				
Wakeup with INT0 Interrupt	150	400	ms typ	
Wakeup with SPI/I ² C Interrupt	150	400	ms typ	
Wakeup with External RESET	150	400	ms typ	
After External RESET in Normal Mode	3	3	ms typ	
After WDT Reset in Normal Mode	3	3	ms typ	Controlled via WDCON SFR
POWER REQUIREMENTS ^{19,20}				
Power Supply Voltages				
AV _{DD} / DV _{DD} - AGND		2.7	V min.	AV _{DD} / DV _{DD} = 3V nom.
		3.3	V max.	
	4.5		V min.	AV _{DD} / DV _{DD} = 5V nom.
	5.5		V max.	
Power Supply Currents Normal Mode				
D _{VDD} Current ⁹	6	3	mA typ	Core CLK=1 MHz
A _{VDD} Current ⁹	1.4	1.4	mA max	
D _{VDD} Current	40		mA max	Core CLK=16 MHz
	30	15	mA typ	
A _{VDD} Current	1.4	1.4	mA max	
Power Supply Currents Idle Mode				
D _{VDD} Current ⁹	5	1	mA typ	Core CLK=1 MHz
A _{VDD} Current ⁹	0.11	0.11	mA typ	
D _{VDD} Current ⁹	25		mA max	Core CLK=16 MHz
	18	17	mA typ	
A _{VDD} Current ⁹	0.11	0.11	mA typ	
Power Supply Currents Power Down Mode				Core CLK=2MHz or 16 MHz
A _{VDD} Current	2	1	µA typ	
D _{VDD} Current	35	15	µA max	TIMECON.1=0
	25	12	µA typ	
	120	120	µA typ	TIMECON.1=1
Typical Additional Power Supply Currents				AVDD = DVDD = 5V
PSM Peripheral	50		µA typ	
ADC	1.5		mA typ	
DAC	150		µA typ	

ADuC831–SPECIFICATIONS1,2(continued)

NOTES

¹ Temperature Range -40°C to +125°C.

² ADC Linearity is guaranteed during normal MicroConverter Core operation.

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

⁴ Offset and Gain Error and Offset and Gain Error Match are measured after factory calibration.

⁵ Based on external ADC system components the user may need to execute a system calibration to remove additional external channel errors and achieve these specifications.

⁶ SNR calculation includes distortion and noise components.

⁷ Channel to Channel Crosstalk is measured on adjacent channels.

⁸ The Temperature Monitor will give a measure of the die temperature directly, air temperature can be inferred from this result.

⁹ These numbers are not production tested but are guaranteed by Design and/or Characterization data on production release.

¹⁰ DAC linearity is calculated using :

reduced code range of 48 to 4095, 0 to V_{ref} range.

reduced code range of 48 to 3945, 0 to V_{DD} range.

DAC Output Load = 10K Ohms and 100 pF.

¹¹ DAC Differential NonLinearity specified on 0 to V_{ref} and 0 to V_{DD} ranges

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

¹³ DAC specifications for Isink, voltage output settling time and digital-to-analog glitch energy depend on external buffer implementation in unbuffered mode.

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

¹⁵ When using an External Reference device, the internal bandgap reference input can be bypassed by setting the ADCCON1.6 bit. In this mode the V_{ref} and C_{ref} pins need to be shorted together for correct operation.

¹⁶ Flash/EE Memory Reliability Characteristics apply to both the Flash/EE program memory and the Flash/EE data memory.

¹⁷ Endurance is qualified to 100 Kcycles as per JEDEC Std. 22 method A117 and measured at -40°C, +25°C, and +125°C, typical endurance at 25°C is 700 Kcycles.

¹⁸ Retention lifetime equivalent at junction temperature (T_j) = 55°C as per JEDEC Std. 22 method A117. Retention lifetime based on an activation energy of 0.6eV will derate with junction temperature as shown in Figure 27 in the Flash/EE Memory description section of this data sheet.

¹⁹ Power Supply current consumption is measured in Normal, Idle, and Power-Down Modes under the following conditions:

Normal Mode: Reset = 0.4 V, Digital I/O pins = open circuit, Core Executing internal software loop.

Idle Mode: Reset = 0.4 V, Digital I/O pins = open circuit, Core Execution suspended in idle mode.

Power-Down Mode: Reset = 0.4 V, All Port 0 pins = 0.4 V, All other digital I/O pins and Port 1 are open circuit, OSC off, TIC off.

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

Specifications subject to change without notice.

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

AV _{DD} to DV _{DD}	−0.3 V to +0.3 V
AGND to DGND	−0.3 V to +0.3 V
DV _{DD} to DGND, AV _{DD} to AGND	−0.3 V to +7 V
Digital Input Voltage to DGND	−0.3 V, DV _{DD} + 0.3 V
Digital Output Voltage to DGND	−0.3 V, DV _{DD} + 0.3 V
V _{REF} to AGND	−0.3 V, AV _{DD} + 0.3 V
Analog Inputs to AGND	−0.3 V, AV _{DD} + 0.3 V
Operating Temperature Range Industrial ADuC831BS	−40°C to +125°C
Operating Temperature Range Industrial ADuC831BCP	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Junction Temperature	150°C
θ _{JA} Thermal Impedance (ADuC831BS)	90°C/W
θ _{JA} Thermal Impedance (ADuC831BCP)	52°C/W
Lead Temperature, Soldering	
Vapor Phase (60 sec)	215°C
Infrared (15 sec)	220°C

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

ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option
ADuC831BS	−40°C to +125°C	52-Lead Plastic Quad Flatpack	S-52
ADuC831BCP	−40°C to +85°C	56-Lead Chip Scale Package	CP-56
EVAL-ADuC831QS		Quickstart Development System	

CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although the ADuC831 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.

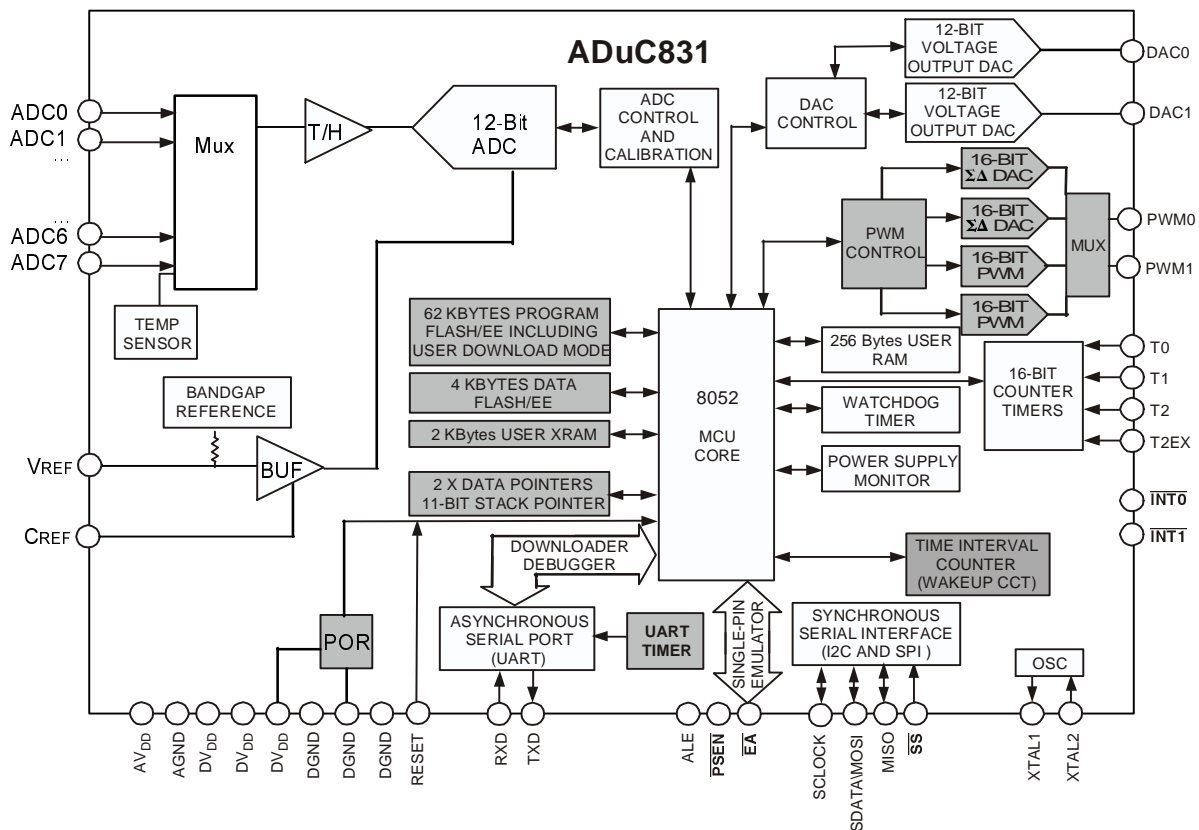


ADuC831 52 PIN PQFP TOP VIEW (Not to Scale)

Pin	Function	Pin	Function
1	P1.0/ADC0/T2	39	P2.7/PWM1/A15/A23
2	P1.1/ADC1/T2EX	38	P2.6/PWM0/A14/A22
3	P1.2/ADC2	37	P2.5/A13/A21
4	P1.3/ADC3	36	P2.4/A12/A20
5	AVDD	35	DGND
6	AGND	34	DVDD
7	CREF	33	XTAL2
8	VREF	32	XTAL1
9	DAC0	31	P2.3/A11/A19
10	DAC1	30	P2.2/A10/A18
11	P1.4/ADC4	29	P2.1/A9/A17
12	P1.5/ADC5/SS	28	P2.0/A8/A16
13	P1.6/ADC6	27	SDATA/MOSI
14	P1.7/AD7		
15	RESET		
16	P3.0/RXD		
17	P3.1/TXD		
18	P3.2/INT0		
19	P3.3/INT1/MISO/PWM0		
20	DVDD		
21	DGND		
22	P3.4/T0/PWMC/PWM1		
23	P3.5/T1/CONVST		
24	P3.6/WR		
25	P3.7/RD		
26	SCLOCK		
27	SDATA/MOSI		
28	P3.7/RD		
29	P2.0/A8/A16		
30	P2.1/A9/A17		
31	P2.2/A10/A18		
32	P2.3/A11/A19		
33	XTAL1		
34	XTAL2		
35	DVDD		
36	DGND		
37	P2.5/A13/A21		
38	P2.6/A14/A22		
39	P2.7/A15/A23		
40	PSEN		
41	ALE		
42	P0.0/AD0		
43	P0.1/AD1		
44	P0.2/AD2		
45	P0.3/AD3		
46	DGND		
47	DVDD		
48	P0.4/AD4		
49	P0.5/AD5		
50	P0.6/AD6		
51	P0.7/AD7		
52	P1.0/ADC0/T2		

ADuC831 56 PIN CSP TOP VIEW (Not to Scale)

Pin	Function	Pin	Function
1	P1.0/ADC0/T2EX	39	P2.7/A15/A23
2	P1.2/ADC2	41	P2.6/A14/A22
3	P1.3/ADC3	40	P2.5/A13/A21
4	AVDD	39	P2.4/A12/A20
5	AVDD	38	DGND
6	AGND	37	DGND
7	AGND	36	DVDD
8	AGND	35	XTAL2
9	Cref	34	XTAL1
10	Vref	33	P2.3/A11/A19
11	DAC0	32	P2.2/A10/A18
12	DAC1	31	P2.1/A9/A17
13	P1.4/ADC4	30	P2.0/A8/A16
14	P1.5/ADC5/SS	29	SDATA/MOSI
15	P1.6/ADC6		
16	P7/ADC7		
17	RESET		
18	P3.0/RXD		
19	P3.1/TXD		
20	P3.2/INT0		
21	P3.3/INT1		
22	DVDD		
23	DGND		
24	P3.4/T0		
25	P3.5/T1		
26	P3.6/WR		
27	P3.7/RD		
28	SCLOCK		
29	SDATA/MOSI		
30	P2.0/A8/A16		
31	P2.1/A9/A17		
32	P2.2/A10/A18		
33	P2.3/A11/A19		
34	XTAL1		
35	XTAL2		
36	DVDD		
37	DGND		
38	DGND		
39	P2.4/A12/A20		
40	P2.5/A13/A21		
41	P2.6/A14/A22		
42	P2.7/A15/A23		
43	PSEN		
44	ALE		
45	P0.0/AD0		
46	P0.1/AD1		
47	P0.2/AD2		
48	P0.3/AD3		
49	DGND		
50	DVDD		
51	P0.4/AD4		
52	P0.5/AD5		
53	P0.6/AD6		
54	P0.7/AD7		
55	P1.0/ADC0/T2EX		
56	P1.1/ADC1/T2EX		



REV. PrF

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 μ F between this pin and AGND.
V _{REF}	I/O	Reference Input/Output. This pin is connected to the internal reference through a series resistor and is the reference source for the analog-to-digital converter. The nominal internal reference voltage is 2.5 V and this 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 multifunction and share the following functionality.
ADC0–ADC7	I	Analog Inputs. Eight single-ended analog inputs. Channel selection is via ADCCON2 SFR.
T2	I	Timer 2 Digital Input. Input to Timer/Counter 2. When Enabled, Counter 2 is incremented in response to a 1 to 0 transition of the T2 input.
T2EX	I	Digital Input. Capture/Reload trigger for Counter 2 and also functions as an Up/Down control input for Counter 2.
SS	I	Slave Select Input for the SPI Interface
SDATA	I/O	User Selectable, I ² C-Compatible or SPI Data Input/Output Pin
SCLOCK	I/O	Serial Clock Pin for I ² C-Compatible or SPI Serial Interface Clock
MOSI	I/O	SPI Master Output/Slave Input Data I/O Pin for SPI Interface
MISO	I/O	SPI Master Input/Slave Output Data I/O Pin for SPI Serial Interface
DAC0	O	Voltage Output from DAC0
DAC1	O	Voltage Output from DAC1
RESET	I	Digital Input. A high level on this pin for 24 master clock cycles while the oscillator is running resets the device.
P3.0–P3.7	I/O	Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state 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 which are described below.
PWMC	I	PWM Clock Input
PWM0	O	PMW0 Voltage Output. PWM outputs can be configured to use ports 2.6 & 2.7 or 3.4 and 3.3
PWM1	O	PMW1 Voltage Output. See CFG831 Register for further Information.
RxD	I/O	Receiver Data Input (Asynchronous) or Data Input/Output (Synchronous) of Serial (UART) Port
TxD	O	Transmitter Data Output (Asynchronous) or Clock Output (Synchronous) of Serial (UART) Port
INT0	I	Interrupt 0, programmable edge or level triggered Interrupt input, which can be programmed to one of two priority levels. This pin can also be used as a gate control input to Timer 0.
INT1	I	Interrupt 1, programmable edge or level triggered Interrupt input, which 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
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/hold into its hold mode and starts conversion.
WR	O	Write Control Signal, Logic Output. Latches the data byte from Port 0 into the external data memory.
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 input 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, and 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.

ADuC831

PIN FUNCTION DESCRIPTION (continued)

Mnemonic	Type	Function
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. 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.
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 1FFFFH. When held low this input enables the device to fetch all instructions from external program memory. This pin should not be left float.
P0.7–P0.0 (A0–A7)	I/O	Port 0 is an 8-Bit Open Drain Bidirectional I/O port. Port 0 pins that have 1s written to them float and in that state can be used as high impedance inputs. Port 0 is also the multiplexed low order address and data bus during accesses to external program or data memory. In this application it uses strong internal pull-ups when emitting 1s.

TERMINOLOGY

ADC SPECIFICATIONS

Integral Nonlinearity

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

Differential Nonlinearity

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

Offset Error

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

Gain Error

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

Signal to (Noise + Distortion) Ratio

This is the measured ratio of signal to (noise + distortion) at the output of the A/D converter. 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:

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

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

Total Harmonic Distortion

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

DAC SPECIFICATIONS

Relative Accuracy

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

Voltage Output Settling Time

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

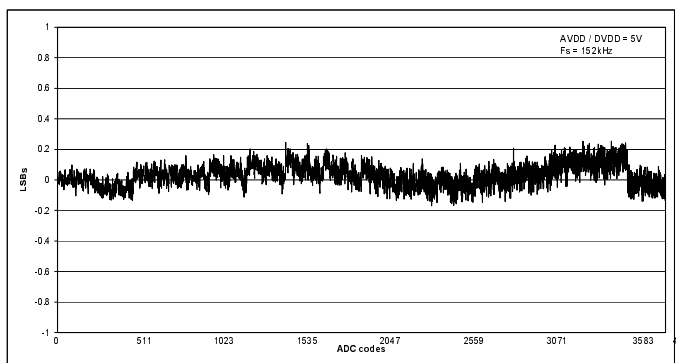
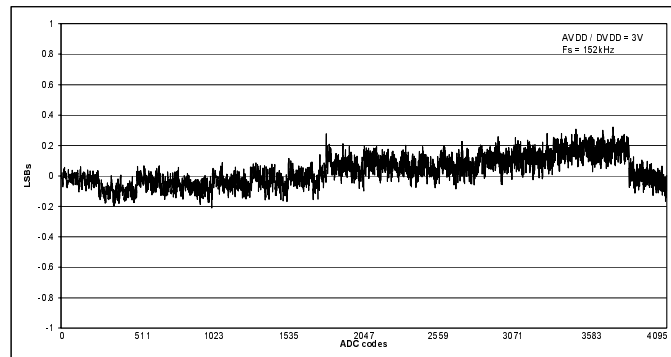
Digital-to-Analog Glitch Impulse

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

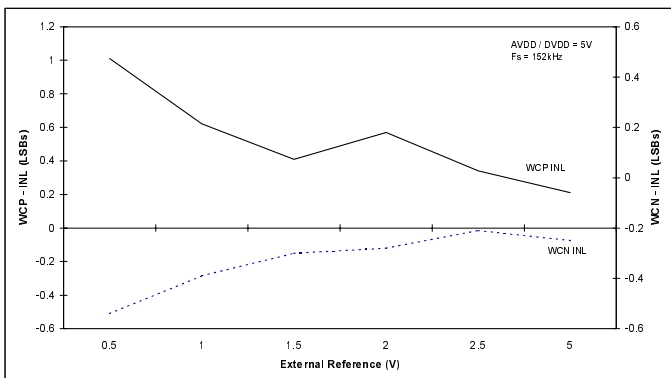
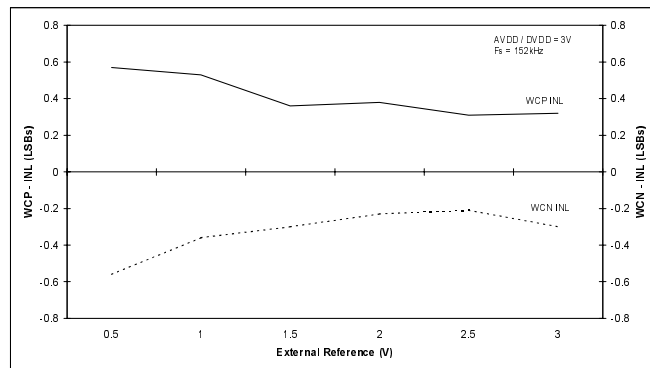
Typical Performance Curves :

The typical performance plots presented in this section illustrate typical performance of the ADuC831 under various operating conditions.

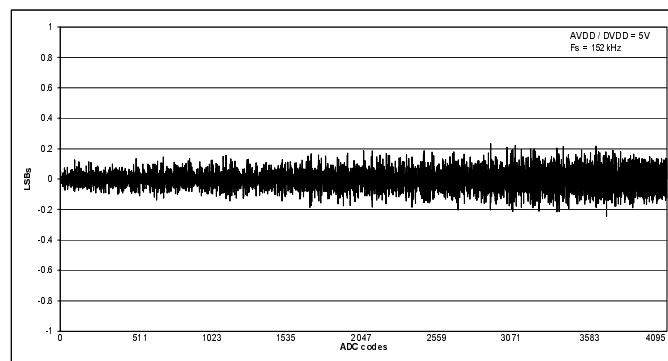
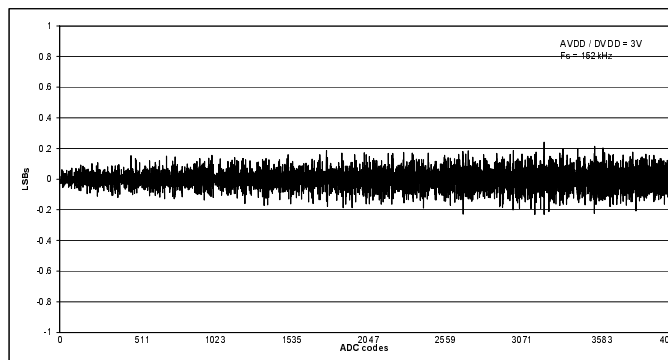
TPC1 and TPC2 below show typical ADC Integral Nonlinearity (INL) errors from ADC code 0 to code 4095 at 5V and 3V supplies respectively. The ADC is using its internal reference (2.5V) and operating at a sampling rate of 152kHz and the typically worst case errors in both plots is just less than 0.3LSBs.

TPC1: Typical INL Error, $V_{DD}=5V$ TPC2: Typical INL Error, $V_{DD}=3V$

TPC3 and TPC4 below show the variation in Worst Case Positive (WCP) INL and Worst Case Negative (WCN) INL versus external reference input voltage.

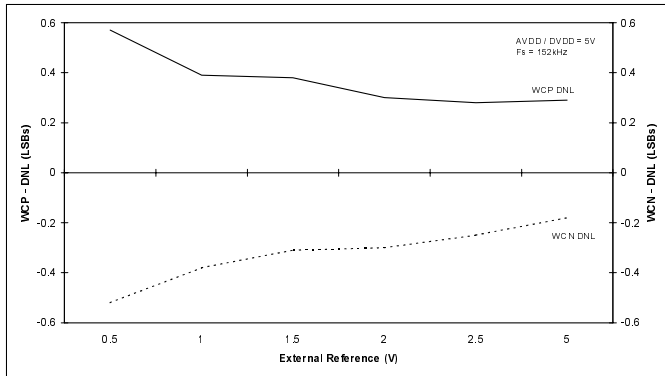
TPC3: Typical Worst Case INL Error versus V_{ref} , $V_{DD}=5V$ TPC4: Typical Worst Case INL Error versus V_{ref} , $V_{DD}=3V$

TPC5 and TPC6 show typical ADC Differential Nonlinearity (DNL) errors from ADC code 0 to code 4095 at 5V and 3V supplies respectively. The ADC is using its internal reference (2.5V) and operating at a sampling rate of 152kHz and the typically worst case errors in both plots is just less than 0.2LSBs.

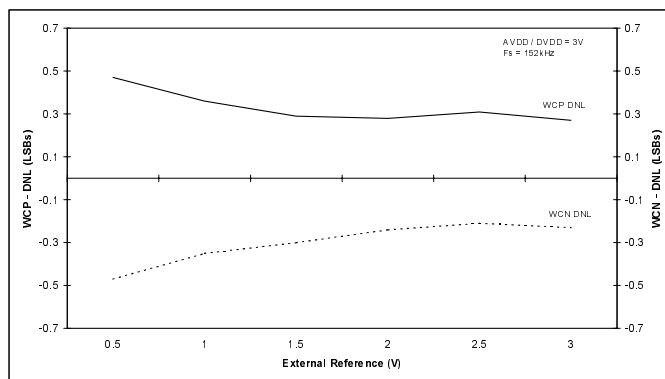
TPC5: Typical DNL Error, $V_{DD}=5V$ TPC6: Typical DNL Error, $V_{DD}=3V$

TPC7 and TPC8 show the variation in Worst Case Positive (WCP) DNL and Worst Case Negative (WCN) DNL versus external reference input voltage.

ADuC831

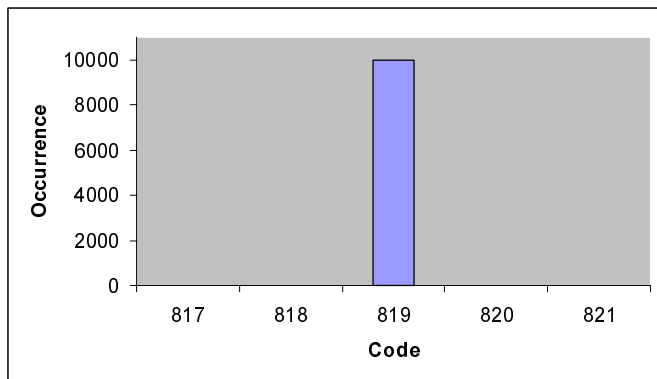


TPC7: Typical Worst Case DNL Error versus Vref, $V_{DD}=5V$



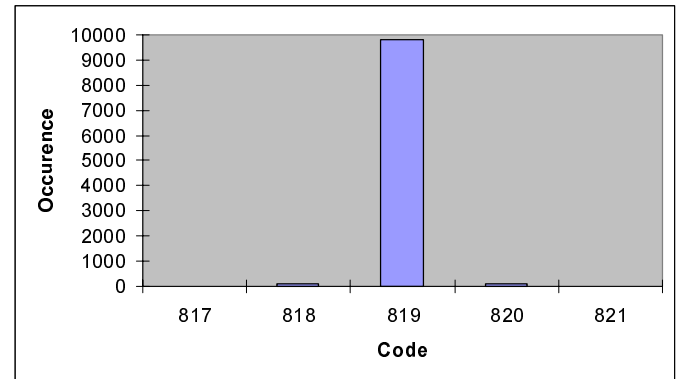
TPC8: Typical Worst Case DNL Error versus Vref, $V_{DD}=3V$

TPC9 shows a histogram plot of 10,000 ADC conversion results on a dc input with $V_{DD}=5V$. The plot illustrates an excellent code distribution pointing to the low noise performance of the on-chip precision ADC.



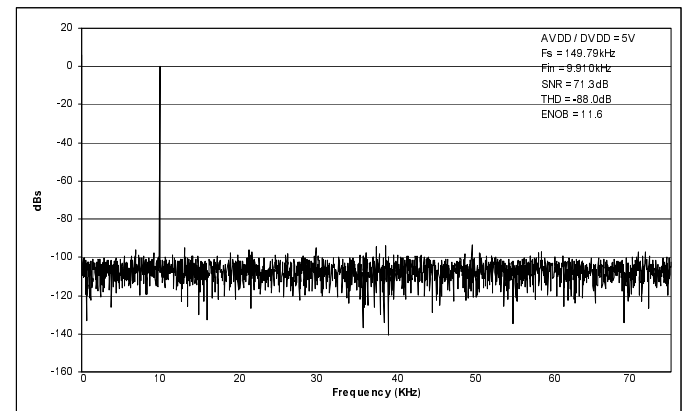
TPC9: Code Histogram plot, $V_{DD}=5V$.

TPC10 shows a histogram plot of 10,000 ADC conversion results on a dc input for $V_{DD}=3V$. The plot again illustrates a very tight code distribution of 1LSB with the majority of codes appearing in one output bin.

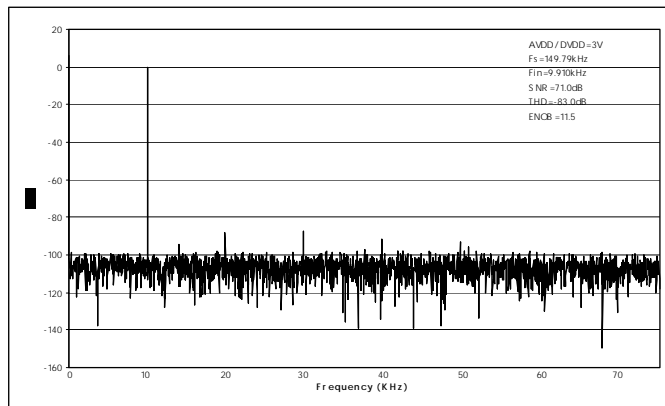


TPC10: Code Histogram Plot, $V_{DD}=3$.

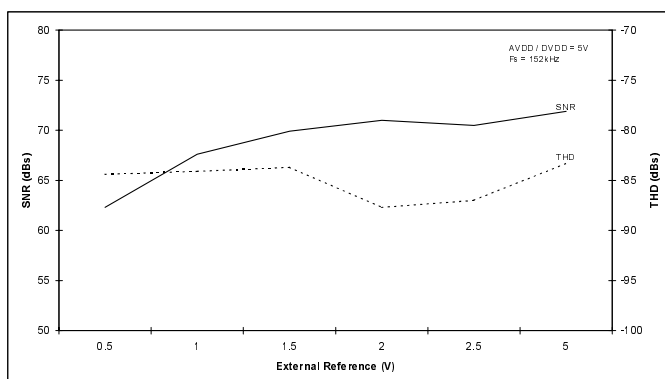
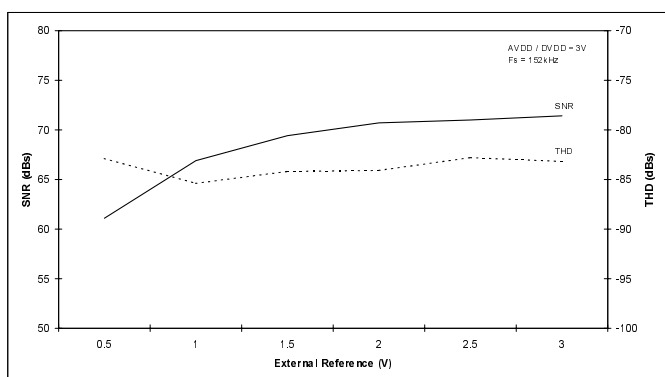
TPC11 and TPC12 show typical FFT plots for the ADuC831. These plots were generated using an external clock input. The ADC is using its internal reference (2.5V) sampling a full-scale, 10kHz sine-wave test tone input at a sampling rate of 149.79kHz. The resultant FFTs shown at 5V and 3V supplies illustrate an excellent 100dB noise floor, 71dBs Signal-to-Noise Ratio (SNR) and THD greater than -80dBs.



TPC11: ADuC831 Dynamic Performance at $V_{DD}=5V$

TPC12: ADuC831 Dynamic Performance at $V_{DD}=3V$

TPC13 and TPC14 show typical dynamic performance versus external reference voltages. Again excellent AC performance can be observed in both plots with some roll-off being observed as V_{ref} falls below 1V.

TPC13: Typical Dynamic Performance versus V_{ref} , $V_{DD}=5V$ TPC14: Typical Dynamic Performance versus V_{ref} , $V_{DD}=3V$

ADuC831

MEMORY ORGANIZATION

The ADuC831 contains 4 different memory blocks namely:

- 62kBytes of On-Chip Flash/EE Program Memory
- 4kBytes of On-Chip Flash/EE Data Memory
- 256 Bytes of General Purpose RAM
- 2kBytes of Internal XRAM

(1) Flash/EE Program Memory

The ADuC831 provides 62kBytes of Flash/EE program memory to run user code. The user can choose to run code from this internal memory or run code from an external program memory.

If the user applies power or resets the device while the EA pin is pulled low, the part will execute code from the external program space, otherwise the part defaults to code execution from its internal 62kBytes of Flash/EE program memory. Unlike the ADuC812, where code execution can overflow from the internal code space to external code space once the PC becomes greater than 1FFFh, the ADuC831 does not support the rollover from F7FFh in internal code space to F800h in external code space. Instead the 2048 bytes between F800h and FFFFh will appear as NOP instructions to user code.

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

(2) Flash/EE Data Memory

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

(3) General Purpose RAM

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

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

Reset initializes the stack pointer to location 07 hex and increments it once before loading the stack to start from locations 08 hex which is also the first register (R0) of register bank 1. Thus,

if one is going to use more than one register bank, the stack pointer should be initialized to an area of RAM not used for data storage.

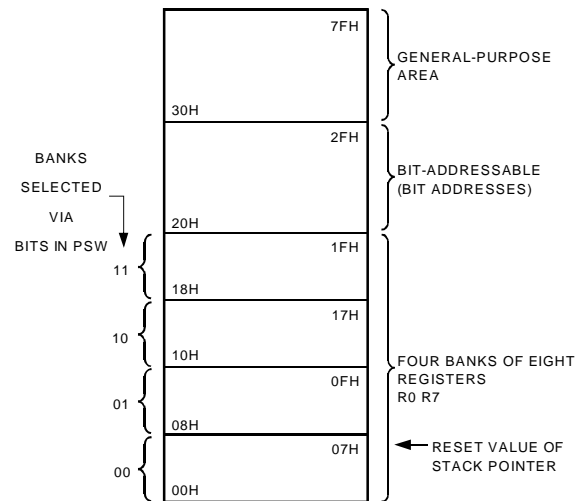


Figure 2. Lower 128 Bytes of Internal Data Memory

The ADuC831 contains 2048 bytes of internal XRAM, 1792 bytes of which can be configured to be used as an extended 11-bit stack pointer.

By default the stack will operate exactly like an 8052 in that it will rollover from FFh to 00h in the general purpose RAM. On the ADuC831 however it is possible (by setting CFG831.7) to enable the 11-bit extended stack pointer. In this case the stack will rollover from FFh in RAM to 0100h in XRAM.

The 11-bit stack pointer is visible in the SP and SPH SFRs. The SP SFR is located at 81h as with a standard 8052. The SPH SFR is located at B7h. The 3 LSBs of this SFR contain the 3 extra bits necessary to extend the 8-bit stack pointer into an 11-bit stack pointer.

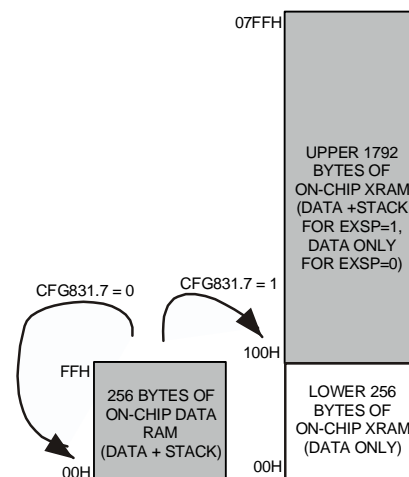


Figure 3. Extended Stack Pointer Operation

External Data Memory (External XRAM)

Just like a standard 8051 compatible core the ADuC831 can access external data memory using a MOVX instruction. The MOVX instruction automatically outputs the various control strobes required to access the data memory.

The ADuC831 however, can access up to 16MBytes of external data memory. This is an enhancement of the 64kBytes external data memory space available on a standard 8051 compatible core.

The external data memory is discussed in more detail in the ADuC831 Hardware Design Considerations section.

Internal XRAM

2kBytes of on-chip data memory exist on the ADuC831. This memory although on-chip is also accessed via the MOVX instruction. The 2kBytes of internal XRAM are mapped into the bottom 2kBytes of the external address space if the CFG831 bit is set, otherwise access to the external data memory will occur just like a standard 8051. When using the internal XRAM, ports 0 and 2 are free to be used as general purpose I/O.

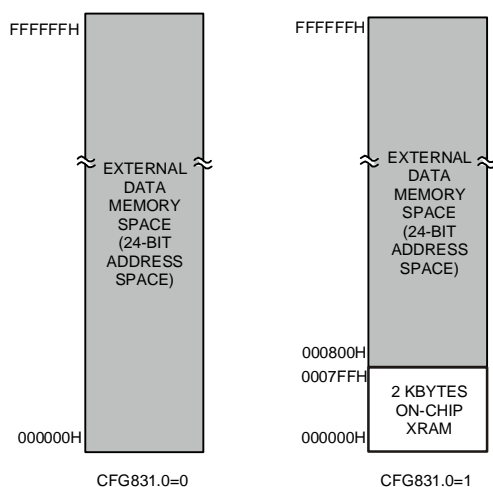


Figure 4. Internal and External XRAM

SPECIAL FUNCTION REGISTERS (SFRs)

The SFR space is mapped into the upper 128 bytes of internal data memory space and accessed by direct addressing only. It provides an interface between the CPU and all on chip peripherals. A block diagram showing the programming model of the ADuC831 via the SFR area is shown in Figure 5.

All registers except the Program Counter (PC) and the four general-purpose register banks, reside in the SFR area. The SFR registers include control, configuration, and data registers that provide an interface between the CPU and all on-chip peripherals.

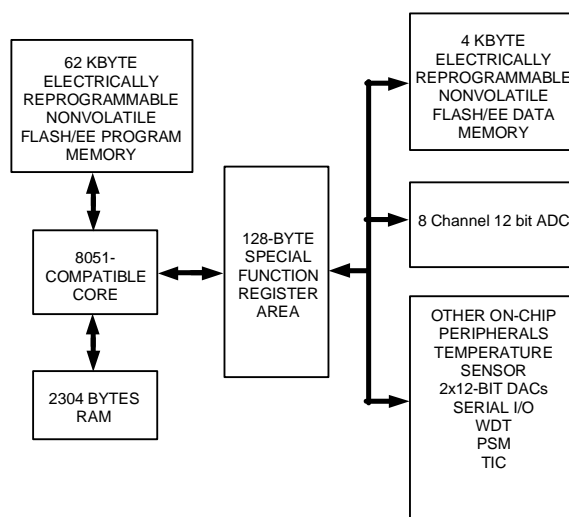


Figure 5. Programming Model

Accumulator SFR (ACC)

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 (B)

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

Stack Pointer (SP and SPH)

The SP SFR 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.

As mentioned earlier the ADuC831 offers an extended 11-bit stack pointer. The 3 extra bits to make up the 11-bit stack pointer are the 3 LSBs of the SPH byte located at B7h.

ADuC831

Data Pointer (DPTR)

The Data Pointer is made up of three 8-bit registers, named DPP (page byte), DPH (high byte) and DPL (low byte).

These are used to provide memory addresses for internal and external code access and external data access. It may be manipulated as a 16-bit register (DPTR = DPH, DPL), although INC DPTR instructions will automatically carry over to DPP, or as three independent 8-bit registers (DPP, DPH, DPL).

The ADuC831 supports dual data pointers. Refer to the Dual Data Pointer section later in this datasheet.

Program Status Word (PSW)

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

SFR Address	D0H
Power ON Default Value	00H
Bit Addressable	Yes

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

Power Control SFR (PCON)

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

SFR Address	87H
Power ON Default Value	00H
Bit Addressable	No

Bit	Name	Description
7	SMOD	Double UART Baud Rate
6	SERIPD	I2C/SPI Power-Down Interrupt Enable
5	INT0PD	INT0 Power-Down Interrupt Enable
4	ALEOFF	Disable ALE Output
3	GF1	General-Purpose Flag Bit
2	GF0	General-Purpose Flag Bit
1	PD	Power-Down Mode Enable
0	IDL	Idle Mode Enable

SPECIAL FUNCTION REGISTERS

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

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

Figure 6 Special Function Register Locations and Reset Values

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON ¹ F8H 04H	DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	RESERVED
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B ¹ F0H 00H	ADCOFSL ³ F1H 00H	ADCOFSH ³ F2H 20H	ADCGAINL ³ F3H 00H	ADCGAINH ³ F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H
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	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 00H
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC ¹ E0H 00H	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED
ADC1 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	ADCDATAL D9H 00H	ADCDATAH DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH
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	RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	RESERVED
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	RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON ¹ C0H 10H	RESERVED	CHIPID C2H 3XH	RESERVED	RESERVED	RESERVED	EDARL C8H 00H	EDARH C7H 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	ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 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	PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	NOT USED	NOT USED	SPH B7H 00H
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	IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG831 ⁴ AFH 10H
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2 ¹ A0H FFH	TIMECON A1H 00H	HTHSEC A2H 00H	SEC A3H 00H	MIN A4H 00H	HOUR A5H 00H	INTVAL A6H 00H	DPCON A7H 00H
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	SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	NOT USED
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1 ^{1,2} 90H FFH	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED	NOT USED
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	TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0 ¹ 80H FFH	SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H

SFR MAP KEY:

THESE BITS ARE CONTAINED IN THIS BYTE



SFR NOTES:

- ¹ SFRs whose address ends in 0H or 8H are bit addressable.
² THE PRIMARY FUNCTION OF PORT1 IS AS AN ANALOG INPUT PORT; THEREFORE, TO ENABLE THE DIGITAL SECONDARY FUNCTIONS ON THESE PORT PINS, WRITE A '0' TO THE CORRESPONDING PORT 1 SFR BIT.
³ CALIBRATION COEFFICIENTS ARE PRECONFIGURED ON POWER-UP TO FACTORY CALIBRATED VALUES.
⁴ VALUE DEPENDS ON EXTERNAL CRYSTAL

ADuC831

ADC CIRCUIT INFORMATION

General Overview

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

The A/D converter consists of a conventional successive-approximation converter based around a capacitor DAC. The converter accepts an analog input range of 0 to $+V_{REF}$. A high precision, low drift and factory calibrated 2.5 V reference is provided on-chip. An external reference can be connected as described later. This external reference can be in the range 0.1 V to AV_{DD} .

Single step or continuous conversion modes can be initiated in software or alternatively by applying a convert signal to the 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 ADuC831 is shipped with factory programmed calibration coefficients which are automatically downloaded to the ADC on power-up ensuring optimum ADC performance. The ADC core contains internal Offset and Gain calibration registers, which can be hardware calibrated to minimize system errors.

A voltage output from an On-Chip bandgap reference proportional to absolute temperature can also be routed through the front end ADC multiplexor (effectively a 9th ADC channel input) facilitating a temperature sensor implementation.

ADC Transfer Function

The analog input range for the ADC is 0 V to V_{REF} . For this range, the designed code transitions occur midway between

successive integer LSB values (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 $2.5\text{ V}/4096 = 0.61\text{ mV}$ when $V_{REF} = 2.5\text{ V}$. The ideal input/output transfer characteristic for the 0 to V_{REF} range is shown in Figure 7.

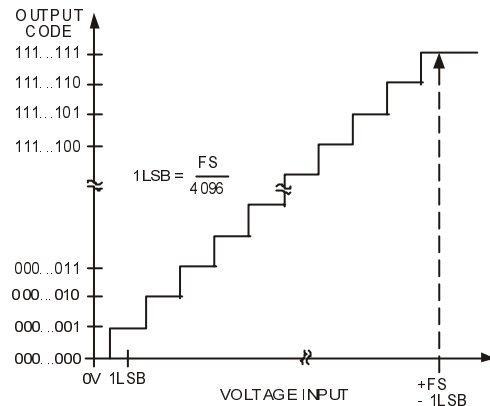


Figure 7. ADC Transfer Function

Typical Operation

Once configured via the ADCCON 1-3 SFRs (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 4 bits of the ADCDATAH SFR will be written with the channel selection bits so as to identify the channel result. The format of the ADC 12 bit result word is shown in Figure 8.

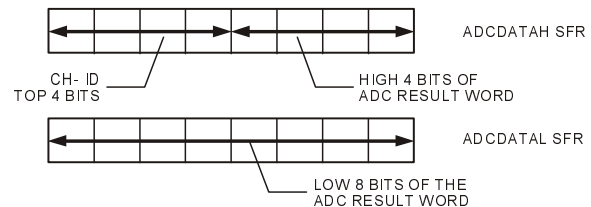


Figure 8 ADC Result Format

ADCCON1 – (ADC Control SFR #1)

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

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

Table III. ADCCON1 SFR Bit Designations

Bit	Name	Description
ADCCON1.7	MD1	The Mode bit selects the active operating mode of the ADC <i>Set by the user to Power Up the ADC</i> <i>Cleared by the user to Power Down the ADC</i>
ADCCON1.6	EXT_REF	<i>Set by the user to select an external reference</i> <i>Cleared by the user to use the internal reference</i>
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. To ensure correct ADC operation, the divider ratio must be chosen to reduce the ADC clock to 4.5MHz and below. A typical ADC conversion will require 17 ADC clocks. The divider ratio is selected as follows: CK1 CK0 MCLK Divider 0 0 16 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/hold amplifier to acquire the input signal. An acquisition of 3 or more ADC clocks is recommended and are selected as follows: AQ1 AQ0 #ADC Clks 0 0 1 0 1 2 1 0 3 1 1 4
ADCCON1.1	T2C	The Timer 2 conversion bit (T2C) is set by the user to enable the Timer 2 over flow bit be used as the ADC convert start trigger input.
ADCCON1.0	EXC	The external trigger enable bit (EXC) is set by the user to allow the external Pin 23 (CONVST) to be used as the active low convert start input. This input should be an active low pulse (minimum pulse width >100 ns) at the required sample rate.

ADuC831

ADCCON2 - (ADC Control SFR #2)

The ADCCON2 register controls ADC channel selection and conversion modes as detailed below.

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

Table IV. ADCCON2 SFR Bit Designations

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. Otherwise the ADCI bit should be cleared by user code.
ADCCON2.6	DMA	The DMA mode enable bit (DMA) is set by the user to enable a preconfigured ADC DMA mode operation. A more detailed description of this mode is given in the ADC DMA Mode section. The DMA bit is automatically set to '0' at the end of a DMA cycle. Setting this bit causes the ALE output to cease, it will start again when DMA is started and will operate correctly after DMA is complete.
ADCCON2.5	CCONV	The continuous conversion bit (CCONV) is set by the user to initiate the ADC into a continuous mode of conversion. In this mode the ADC starts converting based on the timing and channel configuration already set up in the ADCCON SFRs, the ADC automatically starts another conversion once a previous conversion has completed.
ADCCON2.4	SCONV	The single conversion bit (SCONV) is set to initiate a single conversion cycle. The SCONV bit is automatically reset to "0" on completion of the single conversion cycle.
ADCCON2.3	CS3	The channel selection bits (CS3-0) allow the user to program the ADC channel selection under software control. When a conversion is initiated the channel converted will be that pointed to by these channel selection bits. In DMA mode the channel selection is derived from the channel ID written to the external memory.
ADCCON2.2	CS2	
ADCCON2.1	CS1	
ADCCON2.0	CS0	
		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
		0 1 1 0 X No ADC Channel Selected
		0 1 1 1 X No ADC Channel Selected
		1 0 0 0 Temp Monitor Requires minimum of 1uS to acquire
		1 0 0 1 DAC0 Only use with Internal DAC o/p buffer on
		1 0 1 0 DAC1 Only use with Internal DAC o/p buffer on
		1 0 1 1 AGND
		1 1 0 0 VREF
		1 1 1 1 DMA STOP Place in XRAM location to finish DMA sequence, see p20.
		All other combinations reserved

ADCCON3 – (ADC Control SFR #3)

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

SFR Address: F5H

SFR Power On Default Value: 00H

Bit Addressable: NO

Table V. ADCCON3 SFR Bit Designations

Bit 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	GNCLD	Gain Calibration disable bit. Set to '0' to enable Gain Calibration Set to '1' to disable Gain Calibration															
ADCCON3.5	AVGS1	Number of Averages selection bits.															
ADCCON3.4	AVGS0	This bit selects the number of ADC readings averaged during a calibration cycle.															
		<table> <tr> <th>AVGS1</th><th>AVGS0</th><th>Number of Averages</th></tr> <tr> <td>0</td><td>0</td><td>15</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>31</td></tr> <tr> <td>1</td><td>1</td><td>63</td></tr> </table>	AVGS1	AVGS0	Number of Averages	0	0	15	0	1	1	1	0	31	1	1	63
AVGS1	AVGS0	Number of Averages															
0	0	15															
0	1	1															
1	0	31															
1	1	63															
ADCCON3.3	CALCLK	Calibration Clock Divide bit. This bit selects the ratio of the Calibration Clock to the ADCCLK. Set to 0 for ADCCLK / 1 Set to 1 for ADCCLK / 2															
ADCCON3.2	MODCAL	Calibration Mode select bit. This bit selects between internal (Device) and external (System) voltages when a calibration cycle is being performed. Set to 0 for Device Calibration Set to 1 for System Calibration															
ADCCON3.1	TYPECAL	Calibration Type select bit. This bit selects between Offset (zeroscale) and gain (fullscale) calibration. Set to 0 for Offset calibration Set to 1 for Gain calibration															
ADCCON3.0	SCAL	Start Calibration Cycle bit. When set, this bit starts the selected calibration cycle. It is automatically cleared when the calibration cycle is completed.															

ADuC831

Driving the A/D Converter

The ADC incorporates a successive approximation (SAR) architecture involving a charge-sampled input stage. Figure 9 shows the equivalent circuit of the analog input section. Each ADC conversion is divided into two distinct phases as defined by the position of the switches in Figure 9. During the sampling phase (with SW1 and SW2 in the “track” position) a charge proportional to the voltage on the analog input is developed across the input sampling capacitor. During the conversion phase (with both switches in the “hold” position) the capacitor DAC is adjusted via internal SAR logic until the voltage on node A is zero indicating that the sampled charge on the input capacitor is balanced out by the charge being output by the capacitor DAC. The digital value finally contained in the SAR is then latched out as the result of the ADC conversion. Control of the SAR, and timing of acquisition and sampling modes, is handled automatically by built-in ADC control logic. Acquisition and conversion times are also fully configurable under user control.

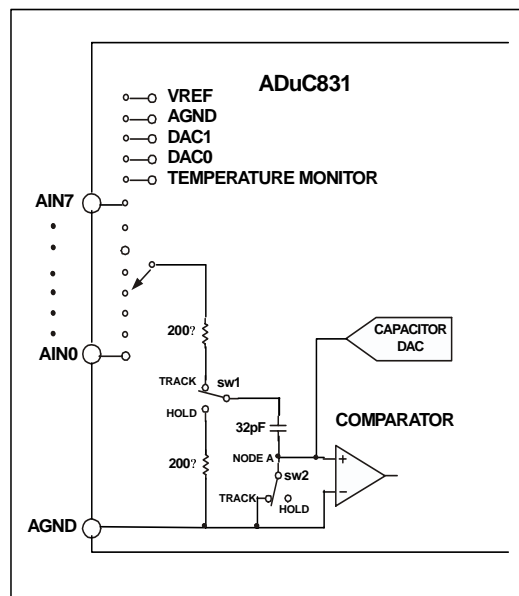


Figure 9. Internal ADC Structure

Note that whenever a new input channel is selected, a residual charge from the 32 pF sampling capacitor places a transient on the newly selected input. The signal source must be capable of recovering from this transient before the sampling switches 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 fully settle from a small signal transient in less than 300 ns in order to guarantee adequate settling under all software configurations. A better solution, recommended for use with any amplifier, is shown in Figure 10.

Though at first glance the circuit in Figure 10 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

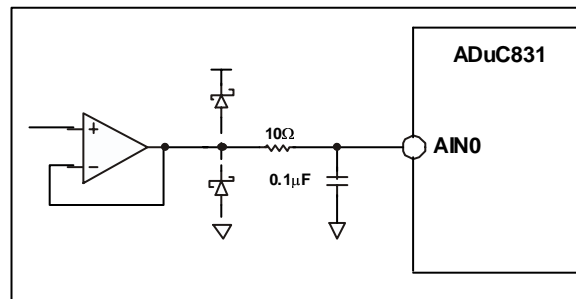


Figure 10. Buffering Analog Inputs

does so by providing a capacitive bank from which the 32 pF sampling capacitor can draw its charge. Its voltage will not change by more than one count (1/4096) of the 12-bit transfer function when the 32 pF charge from a previous channel is dumped onto it. A larger capacitor can be used if desired, but not a larger resistor (for reasons described below).

The Schottky diodes in Figure 10 may be necessary to limit the voltage applied to the analog input pin as per the data sheet absolute maximum ratings. They are not necessary if the op amp is powered from the same supply as the ADuC831 since in that case the op amp is unable to generate voltages above V_{DD} or below ground. An op amp of some kind is necessary unless the signal source is very low impedance to begin with. DC leakage currents at the ADuC831's analog inputs can cause measurable dc errors with external source impedances as little as 100Ω or so. 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	6.1 mV = 10 LSB

Although Figure 10 shows the op amp operating at a gain of 1, you can of course configure it for any gain needed. Also, you can just as easily 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 certainly not limited to, the ones given in Table VI. Check Analog Devices literature (CD ROM data book, etc.) for details on these and other op amps and instrumentation amps.

Table VI. Some Single-Supply Op Amps

Op Amp Model	Characteristics
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 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, you might consider using it to power the front-end amplifiers. If you do, however, be sure to include the Schottky diodes shown in Figure 10 (or at least the lower of the two diodes) to protect the analog input from undervoltage conditions. To summarize this section, use the circuit of Figure 10 to drive the analog input pins of the ADuC831.

Voltage Reference Connections

The on-chip 2.5V bandgap voltage reference can be used as the reference source for the ADC and DACs. In order to ensure the accuracy of the voltage reference you must decouple the V_{REF} pin to ground with a $0.1\mu\text{F}$ capacitor, and the C_{REF} pin to ground with $0.1\mu\text{F}$ capacitors as shown in Figure 11

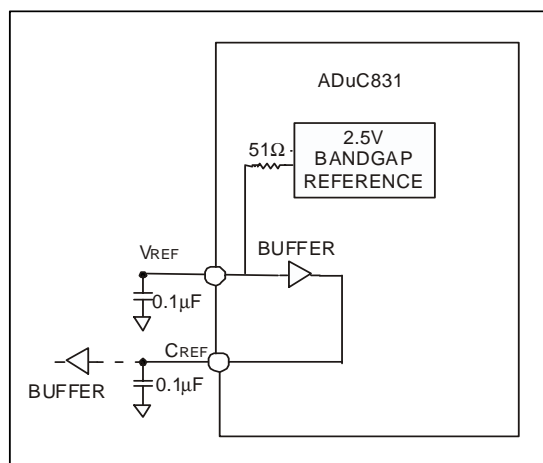


Figure 11 Decoupling V_{REF} and C_{REF} .

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

The ADuC831 powers up with its internal voltage reference in the “on” state. This is available at the V_{REF} pin, but as noted before there will be a gain error between this and that of the ADC. The C_{REF} output becomes available when the ADC is powered up.

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

To ensure accurate ADC operation, the voltage applied to V_{REF} must be between 0.1V and AV_{DD} . In situations where analog input signals are proportional to the power supply (such as some strain-gage applications) it can be desirable to connect the C_{REF} and V_{REF} pins directly to AV_{DD} .

Operation of the ADC or DACs with a reference voltage below 0.1V , however, may incur loss of accuracy eventually resulting in missing codes or non-monotonicity. For that reason, do not use a reference voltage less than 0.1V .

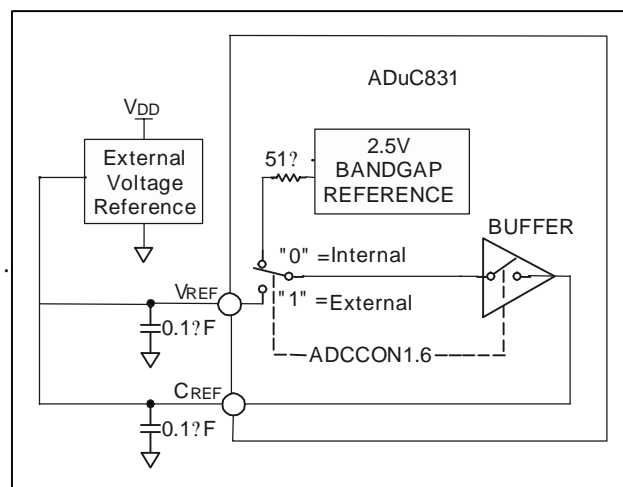


Figure 12. Using an external Voltage Reference.

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

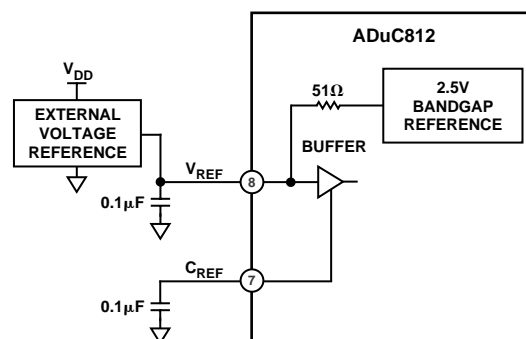


Figure 13. Using an External Voltage Reference

ADuC831

Configuring the ADC

The ADuC831's successive approximation ADC is driven by a divided down version of the master clock. To ensure adequate ADC operation, this ADC clock must be between 400 kHz and 6 MHz, and optimum performance is obtained with ADC clock between 400 kHz and 4.5 MHz. Frequencies within this range can easily be achieved with master clock frequencies from 400 kHz to well above 16 MHz with the four ADC clock divide ratios to choose from. For example, 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 1 ADC clock for synchronization, plus the selected acquisition time (1, 2, 3, or 4 ADC clocks). For the example above, with 3 clocks acquisition time, total conversion time is 19 ADC clocks (or 6.3 μ s for a 3 MHz ADC clock).

In continuous conversion mode, a new conversion begins each time the previous one finishes. The sample rate is then simply the inverse of the total conversion time described above. In the example above, the continuous conversion mode sample rate would be 157.8 kHz.

If using the temperature sensor as the ADC input, the ADC should be configured to use an $ADCCLK$ of $MCLK/16$ and 4 acquisition clocks.

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

ADC DMA Mode

The on-chip ADC has been designed to run at a maximum conversion speed of 4 μ s (247 kHz sampling rate). When converting at this rate the ADuC831 micro has 4 μ s to read the ADC result and store the result in memory for further post processing all within 4 μ s otherwise the next ADC sample could be lost. In an interrupt driven routine the micro 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 ADuC831 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 ADuC831 core. This mode allows the ADuC831 to capture a contiguous sample stream at full ADC update rates (247 kHz).

A typical DMA Mode configuration example.

To set the ADuC831 into DMA mode a number of steps must be followed.

1. The ADC must be powered down. This is done by ensuring MD1 and MD0 are both set to 0 in ADCCON1.
2. The DMA Address pointer must be set to the start address of where the ADC Results are to be written. This is done by writing to the DMA mode Address Pointers DMAL, DMAH, and DMAP. DMAL must be written to first, followed by DMAH and then by DMAP.

3. The external memory must be preconfigured. This consists of writing the required ADC channel IDs into the top four bits of every second memory location in the external SRAM starting at the first address specified by the DMA address pointer. As the ADC DMA mode operates independent from the ADuC831 core it is necessary to provide it with a stop command. This is done by duplicating the last channel ID to be converted followed by "1111" into the next channel selection field. A typical preconfiguration of external memory is as follows.

00000AH	1	1	1	1		STOP COMMAND
	0	0	1	1		REPEAT LAST CHANNEL FOR A VALID STOP CONDITION
	0	0	1	1		CONVERT ADC CH#3
	1	0	0	0		CONVERT TEMP SENSOR
	0	1	0	1		CONVERT ADC CH#5
000000H	0	0	1	0		CONVERT ADC CH#2

Figure 14. Typical DMA External Memory Preconfiguration

4. The DMA is initiated by writing to the ADC SFRs in the following sequence.
 - a. ADCCON2 is written to enable the DMA mode. i.e., `MOV ADCCON2, #40H`; DMA Mode enabled
 - b. ADCCON1 is written to configure the conversion time and power up of the ADC. It can also enable Timer 2 driven conversions or External Triggered conversions if required.
 - c. ADC conversions are initiated. This is done by starting single conversions, starting Timer 2 running for Timer 2 conversions or 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 below. It should be noted that no result is written to the last two memory locations.

When the DMA mode logic is active it takes the responsibility of storing the ADC results away from both the user and ADuC831 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 which access the external memory while DMA mode is enabled will not get access to it. The core will execute the instructions and they will take the same time to execute but they will not gain access to the external memory.

00000AH	1	1	1	1		STOP COMMAND
	0	0	1	1		NO CONVERSION RESULT WRITTEN HERE
	0	0	1	1		CONVERSION RESULT FOR ADC CH#3
	1	0	0	0		CONVERSION RESULT FOR TEMP SENSOR
	0	1	0	1		CONVERSION RESULT FOR ADC CH#5
000000H	0	0	1	0		CONVERSION RESULT FOR ADC CH#2

Figure 15. Typical External Memory Configuration Post ADC DMA Operation

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

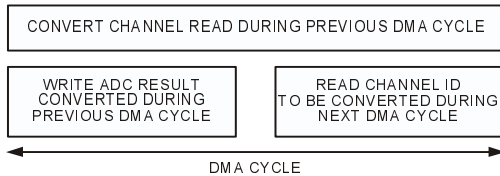


Figure 16. DMA Cycle

From the previous diagram, it can be seen that during one DMA cycle the following actions are performed by the DMA logic.

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

For the previous example the complete flow of events is shown in Figure 16. Because the DMA logic uses pipe-lining, 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 of course are being used by the DMA controller) are gated "OFF" during ADC DMA mode of operation. This means that even though the instruction that accesses the external Ports 0 or 2 will appear to execute, no data will be seen at these external Ports as a result. Note that during DMA to the internally contained XRAM Ports 0 and 2 are available for use.

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

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

ADC OFFSET AND GAIN CALIBRATION COEFFICIENTS

The ADuC831 has two ADC calibration coefficients, one for offset calibration and one for gain calibration.

Both the offset and gain calibration coefficients are 14 bit words, and are each stored in two registers located in the Special Function Register (SFR) area. The offset calibration coefficient is divided into ADCOFSH (6 bits) and ADCOFSL (8 bits) and the gain calibration coefficient is divided into ADCGAINH (6 bits) and ADCGAINL (8 bits).

The offset calibration coefficient compensates for DC offset errors in both the ADC and the input signal. Increasing the offset coefficient compensates for positive offset, and effectively pushes the ADC Transfer Function DOWN. Decreasing the offset coefficient compensates for negative offset, and effectively pushes the ADC Transfer Function UP. The maximum offset that can be compensated is typically $\pm 5\%$ of VREF, which equates to typically $\pm 125\text{mV}$ with a 2.5V 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 $1.025 * \text{VREF}$ and the minimum input range is $0.975 * \text{VREF}$ which equates to typically $\pm 2.5\%$ of the reference voltage.

CALIBRATING THE ADC

There are two hardware calibration modes provided which can be easily initiated by user software. The ADCCON3 SFR is used to calibrate the ADC. Bit 1 (CALTYP) and the CS3 to CS0 (ADCCON2) set up the calibration modes. The following table indicates the mode setup by this bit.

Device Calibration can be initiated to compensate for significant changes in operating conditions frequency, analog input range, reference voltage and supply voltages). In this calibration mode, offset calibration uses internal AGND selected via ADCCON2 register bits CS3-CS0 (1011) and gain calibration uses internal VREF select by CS3-CS0 (1100). Offset calibration should be executed first, followed by gain calibration.

System calibration can be initiated to compensate for both internal and external system errors. To perform system calibration using an external reference, tie system ground and reference to any two of the the six selectable inputs. Enable external reference mode (ADCCON1.6). Select the channel connected to AGND via CS3-CS0 and perform system offset calibration. Select the channel connected to VREF via CS3-CS0 and perform system gain calibration.

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

ADuC831

INITIATING CALIBRATION IN CODE

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

```
MOV ADCCON1,#08CH      ;ADC on; ADCCLK set
                        ;to divide by 16, 4
                        ;acquisition clock
```

To calibrate Device Offset...

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

To calibrate Device Gain...

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

To calibrate System Offset...

Connect system AGND to an ADC channel input (0).

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

To calibrate System Gain...

Connect system AGND to an ADC channel input (1).

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

The calibration cycle time Tcal is calculated by the following equation...assuming a 16MHz crystal

$$T_{cal} = 14 \times ADCCLK \times NUMAV \times (16 + T_{acq})$$

For an ADCCLK/Fcore divide ratio of 16, a T_{acq} = 4 ADCCLK, NUMAV = 15, the calibration cycle time...

$$T_{cal} = 14 \times (1 / 1000000) \times 15 \times (16 + 4)$$

$$T_{cal} = 4.2 \text{ mS}$$

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

WAIT:

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

NONVOLATILE FLASH/EE MEMORY

Flash/EE Memory Overview

The ADuC831 incorporates Flash/EE memory technology on-chip to provide the user with nonvolatile, in-circuit reprogrammable, code and data memory space. Flash/EE memory is a relatively recent type of nonvolatile memory technology and is based on a single transistor cell architecture.

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

Because Flash/EE technology is based on a single transistor cell architecture, a Flash memory array, like EPROM, can be implemented to achieve the space efficiencies or memory densities required by a given design.

Like EEPROM, Flash memory can be programmed in-system at a byte level, although it must first be erased; the erase being performed in page blocks. Thus, Flash memory is often and more correctly referred to as Flash/EE memory.

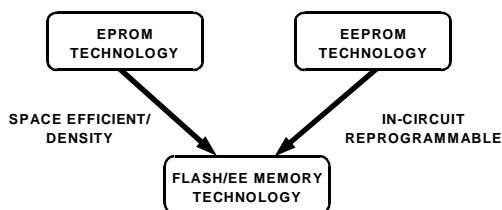


Figure 17. Flash/EE Memory Development

Overall, Flash/EE memory represents a step closer to the ideal memory device that includes nonvolatility, in-circuit programmability, high density and low cost. Incorporated in the ADuC831, Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one-time programmable (OTP) devices at remote operating nodes.

Flash/EE Memory and the ADuC831

The ADuC831 provides two arrays of Flash/EE memory for user applications. 62kBytes of Flash/EE Program space are provided on-chip to facilitate code execution without any external discrete ROM device requirements. The program memory can be programmed in-circuit, using the serial download mode provided, using conventional third party memory programmers or via a user defined protocol who can configure it as data if required.

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

ADuC831 Flash/EE Memory Reliability

The Flash/EE Program and Data Memory arrays on the ADuC831 are fully qualified for two key Flash/EE memory characteristics, namely **Flash/EE Memory Cycling Endurance** and **Flash/EE Memory Data Retention**.

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

- | | |
|--------------------------------|--|
| a. initial page erase sequence | } A single Flash/EE Memory Endurance Cycle |
| b. read/verify sequence | |
| c. byte program sequence | |
| d. second read/verify sequence | |

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

As indicated in the specification pages of this data sheet, the ADuC831 Flash/EE Memory Endurance qualification has been carried out in accordance with JEDEC Specification A117 over the industrial temperature range of -40°C , $+25^{\circ}\text{C}$, $+85^{\circ}\text{C}$ and $+125^{\circ}\text{C}$. The results allow the specification of a minimum endurance figure over supply and temperature of 100,000 cycles, with an endurance figure of 700,000 cycles being typical of operation at 25°C .

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. Again, the ADuC831 has been qualified in accordance with the formal JEDEC Retention Lifetime Specification (A117) at a specific junction temperature ($T_J = 55^{\circ}\text{C}$). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit described above, before data retention is characterized. This means that the Flash/EE memory is guaranteed to retain its data for its full specified retention lifetime every time the Flash/EE memory is reprogrammed. It should also be noted that retention lifetime, based on an activation energy of 0.6 eV, will derate with T_J as shown in Figure 18.

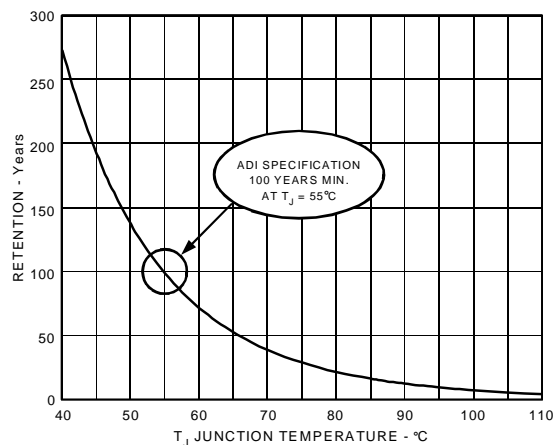


Figure 18. Flash/EE Memory Data Retention

USING THE FLASH/EE PROGRAM MEMORY

The 62kByte Flash/EE Program Memory array is mapped into the lower 62kBytes of the 64 Kbytes program space addressable by the ADuC831, and is used to hold user code in typical applications.

ADuC831

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

(1) Serial Downloading (In-Circuit Programming)

The ADuC831 facilitates code download via the standard UART serial port. The ADuC831 will enter serial download mode after a reset or power cycle if the PSEN pin is pulled low through an external 1kΩ resistor. Once in serial download mode, the user can download code to the full 62kBytes of flash/EE program memory while the device is in circuit in its target application hardware. A PC serial download executable is provided as part of the ADuC831 QuickStart development system. The Serial Download protocol is detailed in a MicroConverter Applications Note uC004.

(2) Parallel Programming

The parallel programming mode is fully compatible with conventional third party Flash or EEPROM device programmers. In this mode Ports P0, P1 and P2 operate as the external data and address bus interface, ALE operates as the Write Enable strobe and Port P3 is used a general configuration port that configures the device for various program and erase operations during parallel programming.

The high voltage (12V) supply required for Flash programming is generated using on-chip charge pumps to supply the high voltage program lines.

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

(3) USER DOWNLOAD MODE (ULOAD)

In figure 17 we can see that it was possible to use the 62kBytes of flash/EE program memory available to the user as one single block of memory. In this mode all of the flash/EE memory is read only to user code.

However the flash/EE program memory can also be written to during runtime simply by entering ULOAD mode. In ULOAD mode the lower 56kBytes of program memory can be erased and reprogrammed by user software as shown in figure 19. ULOAD mode can be used to upgrade your code in the field via any user defined download protocol. Configuring the SPI port on the ADuC831 as a slave, it is possible to completely reprogram the 56kBytes of flash/EE program memory in only 5s (see uC007).

Alternatively ULOAD mode can be used to save data to the 56kBytes of flash/EE memory. This can be extremely useful in datalogging applications where the ADuC831 can provide up to 60kBytes of NV data memory on chip (4kBytes of dedicated flash/EE data memory also exist).

The upper 6kBytes of the 62kBytes of Flash/EE program memory is only programmable via serial download or parallel programming. This means that this space appears as read only to user code. Hence it cannot be accidentally erased or reprogrammed by erroneous code execution. This makes it very suitable to use the 6kBytes as a bootloader. A Bootload Enable option exists in the serial downloader to "Always RUN from E000h after Reset". If using a bootloader this option is recommended to ensure that the bootloader always executes correct code after reset.

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

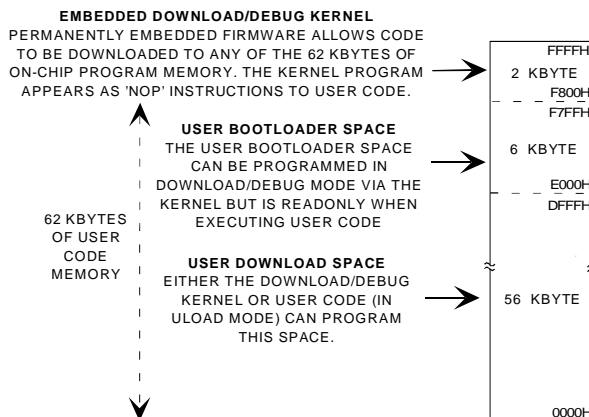


Figure 19. Flash/EE Program Memory Map in ULOAD mode

Flash/EE Program Memory Security

The ADuC831 facilitates three modes of Flash/EE program memory security. These modes can be independently activated, restricting access to the internal code space. These security modes can be enabled as part of serial download protocol as described in technote uC004 or via parallel programming. The security modes available on the ADuC831 are described as follows:

Lock Mode

This mode locks the code memory, disabling parallel programming of the program memory. However, reading the memory in parallel mode and reading the memory via a MOVC command from external memory is still allowed.

This mode is deactivated by initiating a 'code-erase' command in serial download or parallel programming modes.

Secure Mode

This mode locks code in memory, disabling parallel programming (program and verify/read commands) as well as disabling the execution of a 'MOVC' instruction from external memory, which is attempting to read the op codes from internal memory. Read/Write of internal dataflash /EE from external memory is also disabled. This mode is deactivated by initiating a "code-erase" command in serial download or parallel programming modes.

Serial Safe Mode

This mode disables serial download capability on the device. If Serial Safe mode is activated and an attempt is made to reset the part into serial download mode, i.e., RESET asserted and de-asserted with PSEN low, the part will interpret the serial download reset as a normal reset only. It will therefore not enter serial download mode but only execute a normal reset sequence. Serial Safe mode can only be disabled by initiating a code-erase command in parallel programming mode.

USING THE FLASH/EE DATA MEMORY

The 4kBytes of Flash/EE data memory is configured as 1024 pages, each of 4 bytes. As with the other ADuC831 peripherals, the interface to this memory space is via a group of registers mapped in the SFR space. A group of four data registers (EDATA1-4) are used to hold the 4 bytes of data at each page. The page is addressed via the two registers EADRH and EADRL. Finally, ECON is an 8-bit control register that may be written with one of nine Flash/EE memory access commands to trigger various read, write, erase, and verify functions.

A block diagram of the SFR interface to the flash/EE data memory array is shown in figure 20.

ECON—Flash/EE Memory Control SFR

Programming of either the flash/EE data memory or the flash/EE program memory is done through the flash/EE memory control SFR (ECON). This SFR allows the user to read, write, erase or verify the 4 KBytes of flash/EE data memory or the 56kBytes of flash/EE program memory

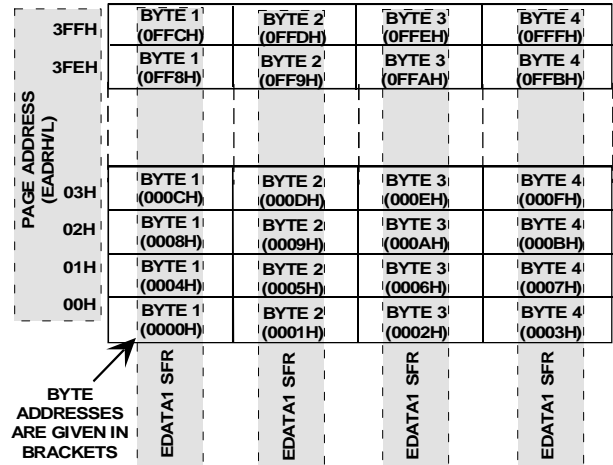


Figure 20. Flash/EE Data Memory Control and Configuration

Table VII. ECON - Flash/EE Memory Commands

ECON VALUE	COMMAND DESCRIPTION (NORMAL MODE) (power on default)	COMMAND DESCRIPTION (ULOAD MODE)
01H READ	Results in 4 bytes in the flash/EE data memory, addressed by the page address EADRH/L, being read into EDATA 1 to 4.	Not Implemented. Use the MOVC instruction
02H WRITE	Results in 4 bytes in EDATA1-4 being written to the flash/EE data memory, at the page address given by EADRH/L (0≤EADRH/L<0400h) Note: The 4 bytes in the page being addressed must be pre-erased.	Results in bytes 0-255 of internal XRAM being written to the 256 bytes of flash/EE program memory at the page address given by EADRH. (0≤EADRH<E0H) Note: The 256 bytes in the page being addressed must be pre-erased.
03H	Reserved Command	Reserved Command
04H VERIFY	Verifies if the data in EDATA1-4 is contained in the page address given by EADRH/L. A subsequent read of the ECON SFR will result in a 0 being read if the verification is valid, or a nonzero value being read to indicate an invalid verification.	Not Implemented. Use the MOVC and MOVX Instructions to verify the WRITE in software
05H ERASE PAGE	Results in the Erase of the 4 byte page of flash/EE data memory addressed by the page address EADRH/L	Results in the 64 Byte page of flash/EE program memory, addressed by the byte address EADRH/L being erased. EADRL can equal any of 64 locations within the page. A new page starts whenever EADRL is equal to 00h, 40h, 80h or C0h
06H ERASE ALL	Results in the erase of entire 4kBytes of flash/EE data memory.	Results in the Erase of the entire 56kBytes of ULOAD flash/EE program memory.
81H READBYTE	Results in the byte in the flash/EE data memory, addressed by the byte address EADRH/L , being read into EDATA1. (0≤EADRH/L≤0FFFh).	Not Implemented. Use the MOVC command.
82H WRITEBYTE	Results in the byte in EDATA1 being written into flash/EE data memory, at the byte address EADRH/L.	Results in the byte in EDATA1 being written into flash/EE program memory, at the byte address EADRH/L (0≤EADRH/L≤DFFFh).
0FH EXULOAD	Leaves the ECON instructions operate on the flash/EE data memory.	Enters NORMAL mode directing subsequent ECON instructions to operate on the flash/EE data memory.
F0H ULOAD	Enters ULOAD mode , directing subsequent ECON instructions to operate on the flash/EE program memory.	Leaves the ECON instructions operate on the flash/EE program memory.

ADuC831

Example: Programming the flash/EE data memory

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

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

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

Step 1: Set up the page address:

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

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

Step 2: Set up the EDATA registers:

We must now write the 4 values to be written into the page into the 4 SFRs EDATA1-4. Unfortunately we do not know 3 of them. Hence we must read the current page and overwrite the second byte.

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

Step 3: Program Page:

A byte in the Flash/EE array can only be programmed if it has previously been erased. To be more specific, a byte can only be programmed if it already holds the value FFH. Because of the Flash/EE architecture, this erase must happen at a page level; therefore, a minimum of four bytes (1 page) will be erased when an erase command is initiated. Once the page is erased we can program the 4 bytes in page and then perform a verification of the data.

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

Note:

Although the 4kBytes of Flash/EE data memory 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 ADuC831. An "ERASE-ALL" command consists of writing "06H" to the ECON SFR, which initiates an erase of the 4kByte Flash/EE array. This command coded in 8051 assembly would appear as:

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

Flash/EE Memory Timing

Typical program and erase times for the ADuC831 are as follows:

NORMAL MODE (operating on flash/EE data memory)

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

ULOAD MODE (operating on flash/EE program memory)

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

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 ADuC831 is idled until the requested Program/Read or Erase mode is completed.

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

ADuC831 Configuration SFR (CFG831)

The CFG831 SFR contains the necessary bits to configure the internal XRAM, EPROM controller, PWM output selection and frequency, DAC buffer and the extended SP. By default it configures the user into 8051 mode. i.e. extended SP is disabled, internal XRAM is disabled.

CFG831	ADuC831 Config SFR
SFR Address	AFH
Power-On Default Value	10*H
Bit Addressable	No

Table VIII. CFG831 SFR Bit Designations

Bit	Name	Description
7	EXSP	Extended SP Enable When set to '1' the stack will rollover from SPH/SP = 00FFh to 0100h When set to '0' the stack will roll over from SP = FFh to SP = 00h
6	PWPO	PWM pin out selection Set to '1' = PWM output pins selected as P3.4 and P3.3 Set to '0' = PWM output pins selected as P2.6 and P2.7
5	DBUF	DAC Output Buffer Set to "1" = DAC Output Buffer Bypassed. Set to "0" = DAC Output Buffer Enabled.
4	EPM2	Flash/EE controller and PWM clock frequency configuration Bits Frequency should be configured such that Fosc/Divide Factor = 32kHz \pm 50%
3	EPM1	
2	EPM0	
		EPM2 EPM1 EPM0 Divide Factor
		0 0 0 32
		0 0 1 64
		0 1 0 128
		0 1 1 256
		1 0 0 512
		1 0 1 1024
1	RSVD	Reserved - This bit should always contain 0
0	XRAMEN	XRAM Enable Bit When set to '1' the internal XRAM will be mapped into the lower 2kBytes of the external address space. When set to '0' the internal XRAM will not be accessible and the external data memory will be mapped into the lower 2kBytes of external data memory.

*The Flash/EE controller bits EPM2,EPM1,EPM0 are set to their correct values depending on the crystal frequency at powerup. The user should not modify these bits so all instructions to the CFG831 register should use the ORL,XRL, or ANL instructions. Value of 10H is for a 16MHz crystal.

ADuC831

USER INTERFACE TO OTHER ON-CHIP ADuC831 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 ADuC831 incorporates two 12-bit, voltage output DACs on-chip. Each has a rail-to-rail voltage output buffer capable of driving 10 k Ω /100 pF. Each has two selectable ranges, 0 V to V_{REF} (the internal bandgap 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

Table IX. DACCON SFR Bit Designations

Bit	Name	Description
7	MODE	The DAC MODE bit sets the overriding operating mode for both DACs. Set to "1" = 8-Bit Mode (Write 8 Bits to DACxL SFR). Set to "0" = 12-Bit Mode.
6	RNG1	DAC1 Range Select Bit. Set to "1" = DAC1 Range 0–V _{DD} . Set to "0" = DAC1 Range 0–V _{REF} .
5	RNG0	DAC0 Range Select Bit. Set to "1" = DAC0 Range 0–V _{DD} . Set to "0" = DAC0 Range 0–V _{REF} .
4	CLR1	DAC1 Clear Bit. Set to "0" = DAC1 Output Forced to 0 V. Set to "1" = DAC1 Output Normal.
3	CLR0	DAC0 Clear Bit. Set to "0" = DAC1 Output Forced to 0 V. Set to "1" = DAC1 Output Normal.
2	SYNC	DAC0/1 Update Synchronization Bit. When set to "1" the DAC outputs update as soon as DACxL SFRs are written. The user can simultaneously update both DACs by first updating the DACxL/H SFRs while SYNC is "0." Both DACs will then update simultaneously when the SYNC bit is set to "1."
1	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

Function

SFR Address

Power-On Default Value

Bit Addressable

DAC Data Registers

DAC Data Registers, written by user to update the DAC output.

DAC0L (DAC0 Data Low Byte) ->F9H; DAC1L (DAC1 Data Low Byte)->FBH

DAC0H (DAC0 Data High Byte) ->FAH; DAC1H(DAC1 Data High Byte)->FCH

->All four Registers

->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 D/A Converter

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

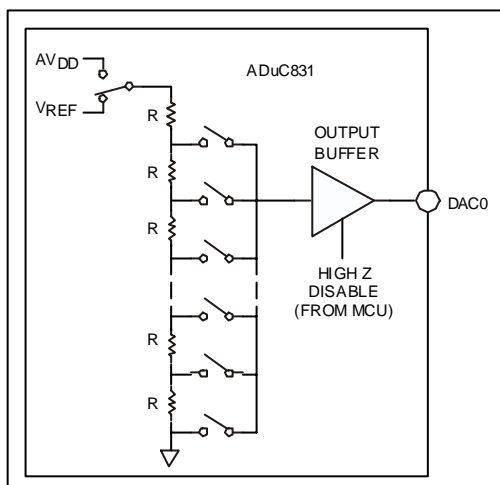


Figure 21. Resistor String DAC Functional Equivalent

As illustrated in Figure 21, the reference source for each DAC is user selectable in software. It can be either AV_{DD} or V_{REF} . In 0-to- AV_{DD} mode, the DAC output transfer function spans from 0 V to the voltage at the AV_{DD} pin. In 0-to- V_{REF} mode, the DAC output transfer function spans from 0 V to the internal V_{REF} or if an external reference is applied the voltage at the V_{REF} pin. The DAC output buffer amplifier features a true rail-to-rail output stage implementation. This means that, unloaded, each output is capable of swinging to within less than 100 mV of both AV_{DD} and ground. Moreover, the DAC's linearity specification (when driving a 10 k Ω 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 3945 to 4095. Linearity degradation near ground and V_{DD} is caused by saturation of the output amplifier, and a general representation of its effects (neglecting offset and gain error) is illustrated in Figure 22. The dotted line in Figure 22 indicates the *ideal* transfer function, and the solid line represents what the transfer function might look like with endpoint nonlinearities due to saturation of the output amplifier. Note that Figure 22 represents a transfer function in 0-to- V_{DD} mode only. In 0-to- V_{REF} mode (with $V_{REF} < V_{DD}$) the lower nonlinearity would be similar, but the upper portion of the transfer function would follow the "ideal" line right to the end (V_{REF} in this case, not V_{DD}), showing no signs of endpoint linearity errors.

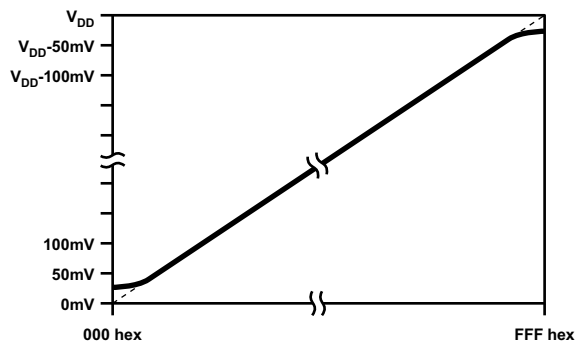


Figure 22. Endpoint Nonlinearities Due to Amplifier Saturation

The endpoint nonlinearities conceptually illustrated in Figure 22 get worse as a function of output loading. Most of the ADuC831's data sheet specifications assume a 10 k Ω resistive load to ground at the DAC output. As the output is forced to source or sink more current, the nonlinear regions at the top or bottom (respectively) of Figure 22 become larger. With larger current demands, this can significantly limit output voltage swing. Figure 23 and Figure 24 illustrate this behavior. It should be noted that the upper trace in each of these figures is only valid for an output range selection of 0-to- AV_{DD} . In 0-to- V_{REF} mode, DAC loading will not cause 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 24 drops below 2.5 V (V_{REF}) indicating that at these higher currents the output will not be capable of reaching V_{REF} .

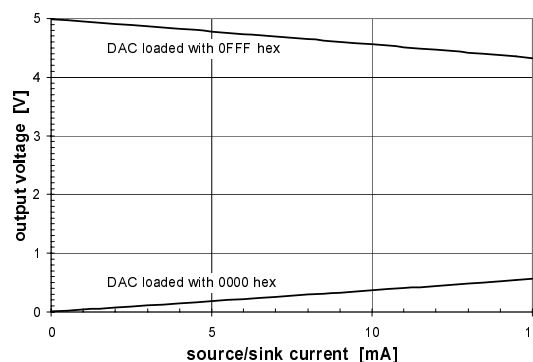


Figure 23. Source and Sink Current Capability with $V_{REF} = V_{DD} = 5$ V

ADuC831

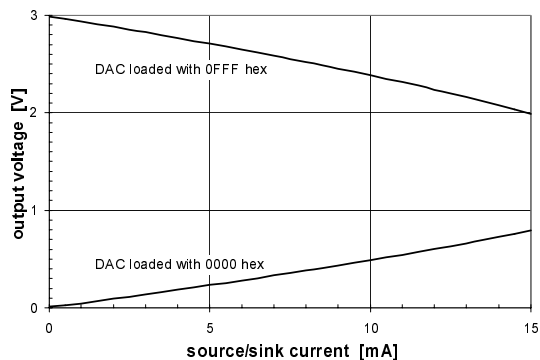


Figure 24. Source and Sink Current Capability with $V_{REF} = V_{DD} = 3\text{ V}$

In order to reduce the effects of the saturation of the output amplifier at values close to ground and to give reduced offset and gain errors the internal buffer can be bypassed. This is done by setting the DBUF bit in the CFG831 register. This allows a full rail to rail output from the DAC which should then be buffered externally using a dual supply op-amp in order to get a rail to rail output. This external buffer should be located as near as physically possible to the DAC output pin on the pcb. Note the unbuffered mode only works in the 0 to Vref range

To drive significant loads with the DAC outputs, external buffering may be required (even with the internal buffer enabled), as illustrated in Figure 25. A list of recommended op-amps is on page 22.

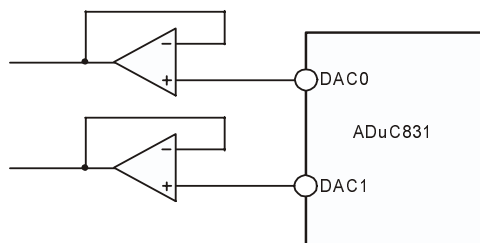


Figure 25. Buffering the DAC Outputs

The DAC output buffer also features a high-impedance disable function. In the chip's default power-on state, both DACs are disabled, and their outputs are in a high-impedance state (or "three-state") where they remain inactive until enabled in software. This means that if a zero output is desired during power-up or power-down transient conditions, then a pull-down resistor must be added to each DAC output. Assuming this resistor is in place, the DAC outputs will remain at ground potential whenever the DAC is disabled. However, each DAC output will still spike briefly when you first apply power to the chip, and again when each DAC is first enabled in software. Typical scope shots of these spikes are given in Figure 26 and Figure 27 respectively.

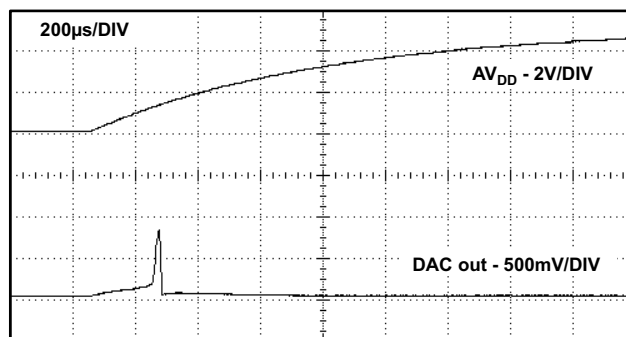


Figure 26. DAC Output Spike at Chip Power-Up

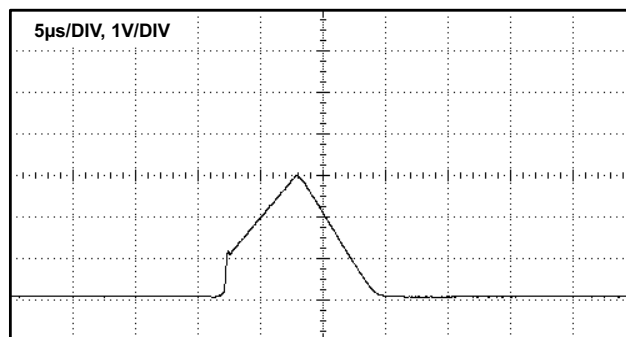


Figure 27. DAC Output Spike at DAC Enable

PULSE WIDTH MODULATOR (PWM)

The PWM on the ADuC831 is highly flexible PWM offering programmable resolution, input clock and can be configured for any one of six different modes of operation. Two of these modes allow the PWM to be configured as a $\Sigma\Delta$ DAC with up to 16-bits of resolution. A block diagram of the PWM is shown in figure 28.

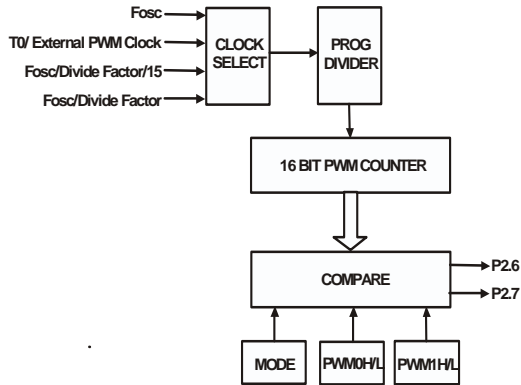


Figure 28. PWM Block Diagram

The PWM uses 5 SFRs; the control SFR, PWMCON, and 4 data SFRs PWM0H, PWM0L, PWM1H and PWM1L. PWMCON (as described below) controls the different modes of operation of the PWM as well as the PWM clock frequency. PWM0H/L and PWM1H/L are the data registers that determine the duty cycles of the PWM outputs. The output pins that the PWM uses are determined by the CFG831 register, they can be either P2.6 and P2.7 or P3.4 and P3.3. In this section of the datasheet it is assumed that P2.6 and P2.7 are selected as the PWM outputs.

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

PWMCON	PWM Control SFR
SFR Address	A Eh
Power-On Default Value	00h
Bit Addressable	No

Table X. PWMCON SFR Bit Designations

Bit	Name	Description																																				
7	SNGL	Turns off PWM output at P2.6 or P3.4 leaving port pin free for digital I/O																																				
6	MD2	PWM Mode Bits The MD2/1/0 bits choose the PWM mode as follows <table><tr><th>MD2</th><th>MD1</th><th>MD0</th><th>Mode</th></tr><tr><td>0</td><td>0</td><td>0</td><td>Mode 0: PWM Disabled</td></tr><tr><td>0</td><td>0</td><td>1</td><td>Mode 1: Single variable resolution PWM on P2.7 or P3.3</td></tr><tr><td>0</td><td>1</td><td>0</td><td>Mode 2: Twin 8-bit PWM</td></tr><tr><td>0</td><td>1</td><td>1</td><td>Mode 3: Twin 16-bit PWM</td></tr><tr><td>1</td><td>0</td><td>0</td><td>Mode 4: Dual NRZ 16-bit $\Sigma\Delta$ DAC</td></tr><tr><td>1</td><td>0</td><td>1</td><td>Mode 5: Dual 8-bit PWM</td></tr><tr><td>1</td><td>1</td><td>0</td><td>Mode 6: Dual RZ 16-bit $\Sigma\Delta$ DAC</td></tr><tr><td>1</td><td>1</td><td>1</td><td>Reserved for future use</td></tr></table>	MD2	MD1	MD0	Mode	0	0	0	Mode 0: PWM Disabled	0	0	1	Mode 1: Single variable resolution PWM on P2.7 or P3.3	0	1	0	Mode 2: Twin 8-bit PWM	0	1	1	Mode 3: Twin 16-bit PWM	1	0	0	Mode 4: Dual NRZ 16-bit $\Sigma\Delta$ DAC	1	0	1	Mode 5: Dual 8-bit PWM	1	1	0	Mode 6: Dual RZ 16-bit $\Sigma\Delta$ DAC	1	1	1	Reserved for future use
MD2	MD1		MD0	Mode																																		
0	0		0	Mode 0: PWM Disabled																																		
0	0		1	Mode 1: Single variable resolution PWM on P2.7 or P3.3																																		
0	1		0	Mode 2: Twin 8-bit PWM																																		
0	1		1	Mode 3: Twin 16-bit PWM																																		
1	0		0	Mode 4: Dual NRZ 16-bit $\Sigma\Delta$ DAC																																		
1	0		1	Mode 5: Dual 8-bit PWM																																		
1	1		0	Mode 6: Dual RZ 16-bit $\Sigma\Delta$ DAC																																		
1	1		1	Reserved for future use																																		
5	MD1																																					
4	MD0																																					
3	CDIV1	PWM Clock Divider																																				
2	CDIV0	Scale the clock source for the PWM counter as shown below. <table><tr><th>CDIV1</th><th>CDIV0</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>PWM Counter = Selected Clock /1</td></tr><tr><td>0</td><td>1</td><td>PWM Counter = Selected Clock /4</td></tr><tr><td>1</td><td>0</td><td>PWM Counter = Selected Clock /16</td></tr><tr><td>1</td><td>1</td><td>PWM Counter = Selected Clock /64</td></tr></table>	CDIV1	CDIV0	Description	0	0	PWM Counter = Selected Clock /1	0	1	PWM Counter = Selected Clock /4	1	0	PWM Counter = Selected Clock /16	1	1	PWM Counter = Selected Clock /64																					
CDIV1	CDIV0	Description																																				
0	0	PWM Counter = Selected Clock /1																																				
0	1	PWM Counter = Selected Clock /4																																				
1	0	PWM Counter = Selected Clock /16																																				
1	1	PWM Counter = Selected Clock /64																																				
1	CSEL1	PWM Clock Divider																																				
0	CSEL0	Select the clock source for the PWM as shown below. <table><tr><th>CSEL1</th><th>CSEL0</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>PWM Clock = $f_{OCS}/DIVIDE\ FACTOR/15$ (see CFG831 register)</td></tr><tr><td>0</td><td>1</td><td>PWM Clock = $f_{OCS}/DIVIDE\ FACTOR$ (see CFG831 register)</td></tr><tr><td>1</td><td>0</td><td>PWM Clock = External input at P3.4/T0</td></tr><tr><td>1</td><td>1</td><td>PWM Clock = f_{OSC}</td></tr></table>	CSEL1	CSEL0	Description	0	0	PWM Clock = $f_{OCS}/DIVIDE\ FACTOR/15$ (see CFG831 register)	0	1	PWM Clock = $f_{OCS}/DIVIDE\ FACTOR$ (see CFG831 register)	1	0	PWM Clock = External input at P3.4/T0	1	1	PWM Clock = f_{OSC}																					
CSEL1	CSEL0	Description																																				
0	0	PWM Clock = $f_{OCS}/DIVIDE\ FACTOR/15$ (see CFG831 register)																																				
0	1	PWM Clock = $f_{OCS}/DIVIDE\ FACTOR$ (see CFG831 register)																																				
1	0	PWM Clock = External input at P3.4/T0																																				
1	1	PWM Clock = f_{OSC}																																				

ADuC831

PWM MODES OF OPERATION

MODE 0: PWM disabled

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

MODE 1: Single Variable resolution PWM

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

PWM1H/L sets the period of the output waveform. Reducing PWM1H/L reduces the resolution of the PWM output but increases the maximum output rate of the PWM. (e.g. Setting PWM1H/L to 65536 gives a 16-bit PWM with a maximum output rate of 244Hz (16MHz/65536). Setting PWM1H/L to 4096 gives a 12-bit PWM with a maximum output rate of 3906Hz (16MHz/4096)).

PWM0H/L sets the duty cycle of the PWM output waveform as shown in the diagram below.

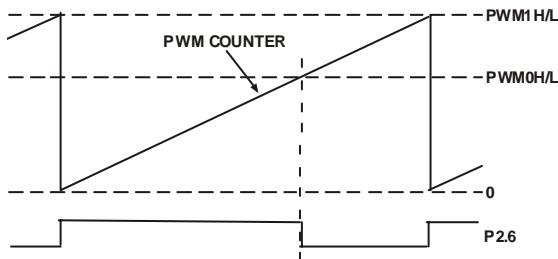


Figure 29. ADuC831 PWM in Mode 1

MODE 2: Twin 8-bit PWM

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

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

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

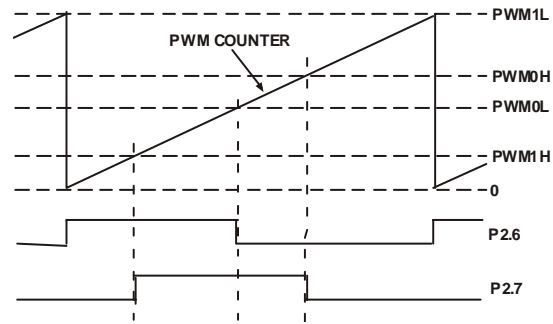


Figure 30. PWM Mode 2

MODE 3: Twin 16-bit PWM

In mode 3 the PWM counter is fixed to count from 0 to 65536 giving a fixed 16-bit PWM. Operating from the 16 MHz core clock results in a PWM output rate of 244Hz. The duty cycle of the PWM outputs at P2.6 and P2.7 are independently programmable.

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

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

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

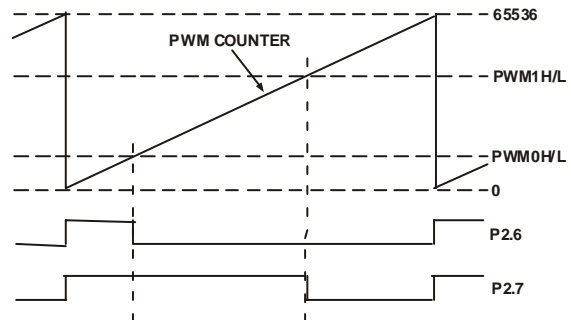


Figure 31. PWM Mode 3

MODE 4: Dual NRZ 16-bit $\Sigma\Delta$ DAC

Mode 4 provides a high speed PWM output similar to that of a $\Sigma\Delta$ DAC. Typically this mode will be used with the PWM clock equal to 16 MHz.

In this mode P2.6 and P2.7 are updated every PWM clock (62ns in the case of 16MHz). Over any 65536 cycles (16 bit PWM) PWM0 (P2.6) is high for PWM0H/L cycles and low for (65536 - PWM0H/L) cycles. Similarly PWM1 (P2.7) is high for PWM1H/L cycles and low for (65536 - PWM1H/L) cycles.

e.g. if PWM1H was set to 4010H (slightly above one quarter of FS) then typically P2.7 will be low for three clocks and high for one clock (each clock is 80ns approx). Over every 65536 clocks the PWM will compromise for the fact that the output should be slightly above one quarter of fullscale by having a high cycle followed by only two low cycles.

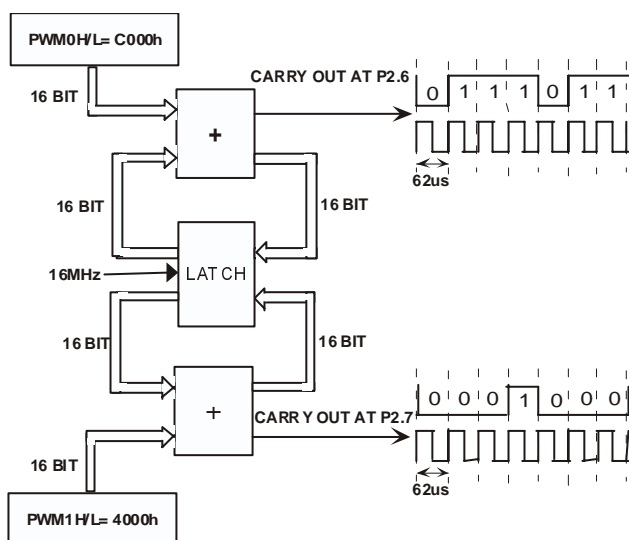


Figure 32. PWM Mode 4

For faster DAC outputs (at lower resolution) write 0's to the LSBs that are not required. If for example only 12 bit performance is required then write 0's to the 4LSBs. This means that a 12-bit accurate $\Sigma\Delta$ DAC output can occur at 3.906kHz. Similarly writing 0's to the 8LSBs gives an 8-bit accurate $\Sigma\Delta$ DAC output at 62kHz

MODE 5: Dual 8-bit PWM

In mode 5 the duty cycle of the PWM outputs and the resolution of the PWM outputs are individually programmable. The maximum resolution of the PWM output is 8-bits.

The output resolution is set by the PWM1L and PWM1H SFRs for the P2.6 and P2.7 outputs respectively. PWM0L and PWM0H sets the duty cycles of the PWM outputs at P2.6 and P2.7 respectively. Both PWMs have same clock source and clock divider.

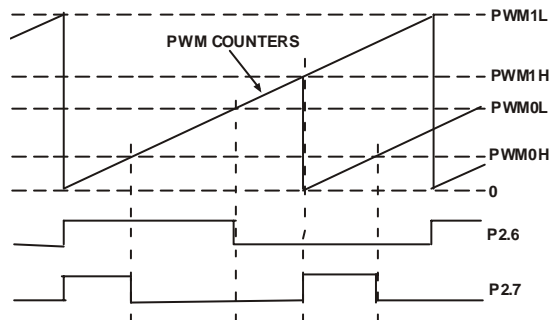


Figure 33. PWM Mode 5

MODE 6: Dual RZ 16-bit $\Sigma\Delta$ DAC

Mode 6 provides a high speed PWM output similar to that of a $\Sigma\Delta$ DAC. Mode 6 operates very similarly to mode 4. However the key difference is that mode 6 provides return to zero (RZ) $\Sigma\Delta$ DAC output. Mode 4 provides non return to zero $\Sigma\Delta$ DAC outputs. The RZ mode ensures that any difference in the rise and fall times will not effect the $\Sigma\Delta$ DAC INL. However, the RZ mode halves the dynamic range of the $\Sigma\Delta$ DAC outputs from $0 \rightarrow AV_{DD}$ to $0 \rightarrow AV_{DD}/2$. For best results this mode should be used with a PWM clock divider of 4.

If PWM1H was set to 4010H (slightly above one quarter of FS) then typically P2.7 will be low for three full clocks (3×62 ns), high for half a clock (31ns) and then low again for half a clock (31ns) before repeating itself. Over every 65536 clocks the PWM will compromise for the fact that the output should be slightly above one quarter of fullscale by leaving the output high for two half clocks in four every so often.

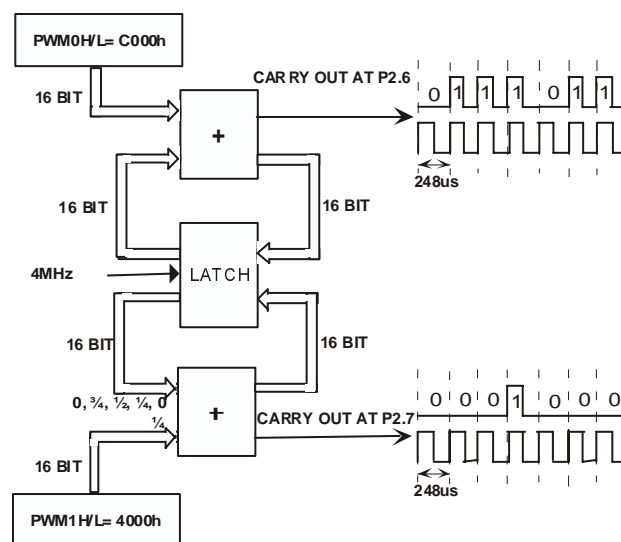


Figure 34. PWM Mode 6

ADuC831

SERIAL PERIPHERAL INTERFACE

The ADuC831 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 I2C interface pins therefore the user can only enable one or the other interface at any given time (see SPE in SPICON below). The SPI Port can be configured for Master or Slave operation and typically consists of four pins, namely:

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

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

MOSI (Master Out, Slave In Pin)

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

SCLOCK (Serial Clock I/O Pin)

The master serial clock (SCLOCK) is used to synchronize the data being transmitted and received through the MOSI and MISO data lines. A single data bit is transmitted and received in each SCLOCK period. Therefore, a byte is transmitted/received after eight SCLOCK periods. The

SCLOCK pin is configured as an output in master mode and as an input in slave mode. In master mode the bit-rate, polarity and phase of the clock are controlled by the CPOL, CPHA, SPR0 and SPR1 bits in the SPICON SFR (see Table XII). In slave mode the SPICON register will have to be configured with the phase and polarity (CPHA and CPOL) of the expected input clock. In both master and slave mode 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.

SS (Slave Select Input Pin)

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

This line is active low. Data is only received or transmitted in slave mode when the SS pin is low, allowing the ADuC831 to be used in single master, multislave SPI configurations. If CPHA = 1 then the SS input may be permanently pulled low. With CPHA = 0 then the 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 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

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 the I2C 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.

Table XII. SPICON SFR Bit Designations (continued)

Bit	Name	Description
1	SPR1	SPI Bit-Rate Select Bits. These bits select the SCLOCK rate (bit-rate) in Master Mode as follows: SPR1 SPR0 Selected Bit Rate 0 0 $f_{OSC}/2$ 0 1 $f_{OSC}/4$ 1 0 $f_{OSC}/8$ 1 1 $f_{OSC}/16$
0	SPR0	
		In SPI Slave Mode, i.e., SPIM = 0, the logic level on the external SS pin can be read via the SPR0 bit.

NOTE

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

SPIDAT

Function

SPI Data Register

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 ADuC831 SPI interface will transmit or receive data in a number of possible modes. Figure 35 shows all possible ADuC831 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.

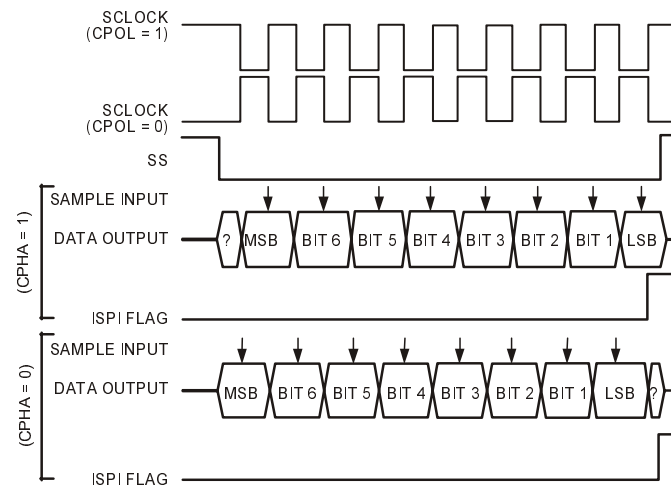


Figure 35. SPI Timing, All Modes

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 SS pin is not used in master mode. If the ADuC831 needs to assert the 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 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 SS returns high if CPHA = 0.

ADuC831

I²C-COMPATIBLE INTERFACE

The ADuC831 supports a fully licensed* I²C serial interface. The I²C interface is implemented as a full hardware slave and software master. SDATA is the data I/O pin and SCLOCK is the serial clock. These two pins are shared with the MOSI and SCLOCK pins of the on-chip SPI interface. Therefore the user

can only enable one or the other interface at any given time (see SPE in SPICON previously). Application Note uC001 describes the operation of this interface as implemented is available from the MicroConverter Website at www.analog.com/microconverter.

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

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

Table XIII. I2CCON SFR Bit Designations

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

I2CADD	I²C Address Register
Function	Holds the I ² C peripheral address for the part. It may be overwritten by user code. Technical Note uC001 at www.analog.com/microconverter describes the format of the I ² C standard 7-bit address in detail.
SFR Address	9BH
Power-On Default Value	55H
Bit Addressable	No

I2CDAT	I²C Data Register
Function	The I2CDAT SFR is written by the user to transmit data over the I ² C interface or read by user code to read data just received by the I ² C interface. Accessing I2CDAT automatically clears any pending I ² C interrupt and the I2CI bit in the I2CCON SFR. User software should only access I2CDAT once per interrupt cycle.
SFR Address	9AH
Power-On Default Value	00H
Bit Addressable	No

* Purchase of licensed 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 the ADuC831 in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

The main features of the MicroConverter I²C interface are:

- Only two bus lines are required; a serial data line (SDATA) and a serial clock line (SCLOCK).
- An I²C master can communicate with multiple slave devices. Because each slave device has a unique 7-bit address then single master/slave relationships can exist at all times even in a multi slave environment (figure 36)
- On-Chip filtering rejects <50ns spikes on the SDATA and the SCLOCK lines to preserve data integrity.

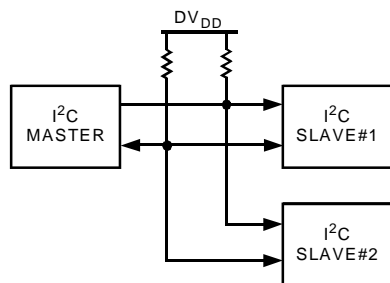


Figure 36. Typical I²C System

Software Master Mode

The ADuC831 can be used as a I²C master device by configuring the I²C peripheral in master mode and writing software to output the data bit by bit. This is referred to as a software master. Master mode is enabled by setting the I2CM bit in the I2CCON register.

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

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

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

Hardware Slave Mode

After reset the ADuC831 defaults to hardware slave mode. The I²C interface is enabled by clearing the SPE bit in SPICON. Slave mode is enabled by clearing the I2CM bit in I2CCON. The ADuC831 has a full hardware slave. In slave mode the I²C address is stored in the I2CADD register. Data received or to be transmitted is stored in the I2CDAT register.

Once enabled in I²C slave mode the slave controller waits for a START condition. If the ADuC831 detects a valid start condition, followed by a valid address, followed by the R/W bit the I2CI interrupt bit will get automatically set by hardware.

The I²C peripheral will only generate a core interrupt if the user has pre-configured the I²C interrupt enable bit in the IEIP2 SFR as well as the global interrupt bit EA in the IE SFR. i.e.

```
; Enabling I2C Interrupts for the ADuC831
MOV IEIP2,#01h      ; enable I2C interrupt
SETB EA
```

On the ADuC831 an auto-clear of the I2CI bit is implemented so this bit is cleared automatically on a read or write access to the I2CDAT SFR.

```
MOV I2CDAT, A      ; I2CI auto-cleared
MOV A, I2CDAT      ; I2CI auto-cleared
```

If for any reason the user tries to clear the interrupt more than once i.e. access the data SFR more than once per interrupt then the I²C controller will halt. The interface will then have to be reset using the I2CRS bit.

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

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

Once the ADuC831 has received a valid address, hardware will hold SCLOCK low until the I2CI bit is cleared by software. This allows the master to wait for the slave to be ready before transmitting the clocks for the next byte.

The I2CI interrupt bit will be set every time a complete data byte is received or transmitted provided it is followed by a valid ACK. If the byte is followed by a NACK an interrupt is NOT generated. The ADuC831 will continue to issue interrupts for each complete data byte transferred until a STOP condition is received or the interface is reset.

When a STOP condition is received, the interface will reset to a state where it is waiting to be addressed (idle). Similarly, if the interface receives a NACK at the end of a sequence it also returns to the default idle state. The I2CRS bit can be used to reset the I²C interface. This bit can be used to force the interface back to the default idle state.

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

ADuC831

DUAL DATA POINTER

The ADuC831 incorporates two data pointers. The second data pointer is a shadow data pointer and is selected via the data pointer control SFR (DPCON). DPCON also includes some nice features such as automatic hardware post increment and post decrement as well as automatic data pointer toggle. DPCON is described below.

DPCON

SFR Address
Power-On Default Value
Bit Addressable

Data Pointer Control SFR

A7h
00h
No

Table XIV. DPCON SFR Bit Designations

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

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

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

To illustrate the operation of DPCON, the following code will copy 256 bytes of code memory at address D000h into XRAM starting from address 0000h.

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

```

MOV DPTR,#0           ; Main DPTR = 0
MOV DPCON,#55h        ; Select shadow DPTR
                        ; DPTR1 increment mode,
                        ; DPTR0 increment mode
                        ; DPTR auto toggling ON
MOV DPTR,#0D000h      ; Shadow DPTR = D000h
MOVELOOP:
  CLR A
  MOVC A,@A+DPTR       ; Get data
                        ; Post Inc DPTR
                        ; Swap to Main DPTR (Data)
  MOVX @DPTR,A         ; Put ACC in XRAM
                        ; Increment main DPTR
                        ; Swap Shadow DPTR (Code)
  MOV A, DPL
  JNZ MOVELOOP

```

POWERSUPPLYMONITOR

As its name suggests, the Power Supply Monitor, once enabled, monitors the DVDD supply on the ADuC831. It will indicate when any of the supply pins drop below one of four user-selectable voltage trip points from 2.63 V to 4.63 V. For correct operation of the Power Supply Monitor function, AV_{DD} must be equal to or greater than 2.7 V. Monitor function is controlled via the PSMCON SFR. If enabled via the IEIP2 SFR, the monitor will interrupt the core using the PSMI bit in the PSMCON SFR. This bit will not be cleared until the failing power supply has returned above the trip point for at least 250 ms. This monitor function allows the user to save working registers to avoid possible data loss due to the low supply condition, and also ensures that normal code execution will not resume until a safe supply level has been well established. The supply monitor is also protected against spurious glitches triggering the interrupt circuit.

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

Table XV PSMCON SFR Bit Designations

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

ADuC831

WATCHDOGTIMER

The purpose of the watchdog timer is to generate a device reset or interrupt within a reasonable amount of time if the ADuC831 enters an erroneous state, possibly due to a programming error or electrical noise. The Watchdog function can be disabled by clearing the WDE (Watchdog Enable) bit in the Watchdog Control (WDCON) SFR. When enabled; the watchdog circuit will generate a system reset or interrupt (WDS) if the user program fails to set the watchdog (WDE) bit within a predetermined amount of time (see PRE3-0 bits in WDCON). The watchdog timer itself is a 16-bit counter that is clocked at 32kHz by the internal R/C Oscillator. The watchdog time-out interval can be adjusted via the PRE3-0 bits in WDCON. Full Control and Status of the watchdog timer function can be controlled via the watchdog timer control SFR (WDCON). The WDCON SFR can only be written by user software if the double write sequence described in WDWR below is initiated on every write access to the WDCON SFR.

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

Table XVI WDCON SFR Bit Designations

Bit	Name	Description																																																																		
7	PRE3	<p>Watchdog Timer Prescale Bits.</p> <p>The Watchdog timeout period is given by the equation: $t_{WD} = (2^{PRE} * (2^9/f_{R/C OSC}))$</p> <p>(0 - PRE - 7; $f_{R/C OSC} = 32kHz \pm 10\%$)</p> <table><thead><tr><th>PRE3</th><th>PRE2</th><th>PRE1</th><th>PRE0</th><th>Timeout Period (ms)</th><th>Action</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>15.6</td><td>Reset or Interrupt</td></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td><td>31.2</td><td>Reset or Interrupt</td></tr><tr><td>0</td><td>0</td><td>1</td><td>0</td><td>62.5</td><td>Reset or Interrupt</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td><td>125</td><td>Reset or Interrupt</td></tr><tr><td>0</td><td>1</td><td>0</td><td>0</td><td>250</td><td>Reset or Interrupt</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td><td>500</td><td>Reset or Interrupt</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1000</td><td>Reset or Interrupt</td></tr><tr><td>0</td><td>1</td><td>1</td><td>1</td><td>2000</td><td>Reset or Interrupt</td></tr><tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0.0</td><td>Immediate Reset</td></tr><tr><td colspan="5">PRE3-0 > 1001</td><td>Reserved</td></tr></tbody></table>	PRE3	PRE2	PRE1	PRE0	Timeout Period (ms)	Action	0	0	0	0	15.6	Reset or Interrupt	0	0	0	1	31.2	Reset or Interrupt	0	0	1	0	62.5	Reset or Interrupt	0	0	1	1	125	Reset or Interrupt	0	1	0	0	250	Reset or Interrupt	0	1	0	1	500	Reset or Interrupt	0	1	1	0	1000	Reset or Interrupt	0	1	1	1	2000	Reset or Interrupt	1	0	0	0	0.0	Immediate Reset	PRE3-0 > 1001					Reserved
PRE3	PRE2		PRE1	PRE0	Timeout Period (ms)	Action																																																														
0	0		0	0	15.6	Reset or Interrupt																																																														
0	0		0	1	31.2	Reset or Interrupt																																																														
0	0		1	0	62.5	Reset or Interrupt																																																														
0	0		1	1	125	Reset or Interrupt																																																														
0	1		0	0	250	Reset or Interrupt																																																														
0	1		0	1	500	Reset or Interrupt																																																														
0	1		1	0	1000	Reset or Interrupt																																																														
0	1		1	1	2000	Reset or Interrupt																																																														
1	0		0	0	0.0	Immediate Reset																																																														
PRE3-0 > 1001					Reserved																																																															
3	WDIR		<p>Watchdog Interrupt Response Enable Bit.</p> <p>If this bit is set by the user, the watchdog will generate an interrupt response instead of a system reset when the watchdog timeout period has expired. This interrupt is not disabled by the CLR EA instruction and it is also a fixed, high-priority interrupt. If the watchdog is not being used to monitor the system, it can alternatively be used as a timer. The prescaler is used to set the timeout period in which an interrupt will be generated.</p>																																																																	
2	WDS	<p>Watchdog Status Bit.</p> <p>Set by the Watchdog Controller to indicate that a watchdog timeout has occurred.</p> <p>Cleared by writing a '0' or by an external hardware reset. It is not cleared by a watchdog reset.</p>																																																																		
1	WDE	<p>Watchdog Enable Bit.</p> <p>Set by user to enable the watchdog and clear its counters. If this bit is not set by the user within the watchdog timeout period, the watchdog will generate a reset or interrupt, depending on WDIR. Cleared under the following conditions, User writes '0,' Watchdog Reset (WDIR = '0'); Hardware Reset; PSM Interrupt.</p>																																																																		
0	WDWR	<p>Watchdog Write Enable Bit.</p> <p>To write data into the WDCON SFR involves a double instruction sequence. The WDWR bit must be set and the very next instruction must be a write instruction to the WDCON SFR.</p> <pre>e.g., CLR EA ;disable interrupts while writing ;to WDT SETB WDWR ;allow write to WDCON MOV WDCON, #72h ;enable WDT for 2.0s timeout SETB EA ;enable interrupts again (if rqd)</pre>																																																																		

TIME INTERVAL COUNTER (TIC)

A time interval counter is provided on-chip for counting longer intervals than the standard 8051-compatible timers are capable of. The TIC is capable of timeout intervals ranging from 1/128th second to 255 hours. Furthermore, this counter is clocked by an internal R/C oscillator rather than the external crystal and has the ability to remain active in power-down mode and time long power-down intervals. This has obvious applications for remote battery-powered sensors where regular widely spaced readings are required. The R/C oscillator is accurate to $\pm 10\%$. Note: Instructions to the TIC SFR's are also clocked at 32kHz, sufficient time must be allowed for in user code for these instructions to execute.

Six SFRs are associated with the time interval counter, TIMECON being its control register. Depending on the configuration of the IT0 and IT1 bits in TIMECON, the selected time counter register overflow will clock the interval counter. When this counter is equal to the time interval value loaded in the INTVAL SFR, the TII bit (TIMECON.2) is set and generates an interrupt if enabled. If the ADuC831 is in power-down mode, again with TIC interrupt enabled, the TII bit will wake up the device and resume code execution by vectoring directly to the TIC interrupt service vector address at 0053 hex. The TIC-related SFRs are described below. Note also that the timebase SFRs can be written initially with the current time, the TIC can then be controlled and accessed by user software. In effect, this facilitates the implementation of a real-time clock. A block diagram of the TIC is shown opposite.

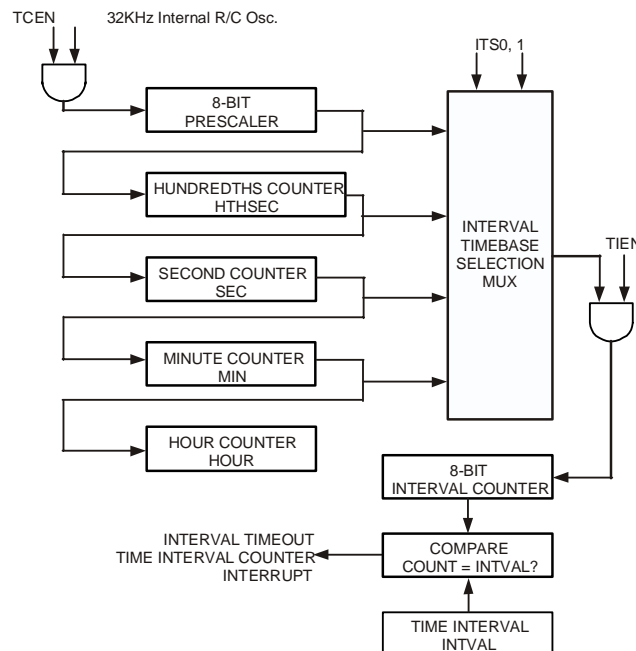


Figure 37. TIC, Simplified Block Diagram

TIMECON

SFR Address
Power-On Default Value
Bit Addressable

TIC CONTROL REGISTER

A1H
00H
No

Table XVII TIMECON SFR Bit Designations

Bit	Name	Description															
7	---	Reserved for Future Use.															
6	TFH	Twenty four hour select bit. Set by the user to enable the Hour counter to count from 0 to 23 Cleared by the user to enable the Hour counter to count from 0 to 255															
5	ITS1	Interval Timebase Selection Bits.															
4	ITS0	Written by user to determine the interval counter update rate. <table><tr><td>ITS1</td><td>ITS0</td><td>Interval Timebase</td></tr><tr><td>0</td><td>0</td><td>1/128 Second</td></tr><tr><td>0</td><td>1</td><td>Seconds</td></tr><tr><td>1</td><td>0</td><td>Minutes</td></tr><tr><td>1</td><td>1</td><td>Hours</td></tr></table>	ITS1	ITS0	Interval Timebase	0	0	1/128 Second	0	1	Seconds	1	0	Minutes	1	1	Hours
ITS1	ITS0	Interval Timebase															
0	0	1/128 Second															
0	1	Seconds															
1	0	Minutes															
1	1	Hours															
3	STI	Single Time Interval Bit. Set by user to generate a single interval timeout. If set, a timeout will clear the TIEN bit. Cleared by user to allow the interval counter to be automatically reloaded and start counting again at each interval timeout.															
2	TII	TIC Interrupt Bit. Set when the 8-bit Interval Counter matches the value in the INTVAL SFR. Cleared by user software.															
1	TIEN	Time Interval Enable Bit. Set by user to enable the 8-bit time interval counter. Cleared by user to disable the interval counter.															
0	TCEN	Time Clock Enable Bit. Set by user to enable the time clock to the time interval counters. Cleared by user to disable the clock to the time interval counters and reset the time interval SFRs to the last value written to them by the user. The time registers (HTHSEC, SEC, MIN and HOUR) can be written while TCEN is low.															

ADuC831**INTVAL**

Function

SFR Address

Power-On Default Value

Bit Addressable

Valid Value

User Time Interval Select Register

User code writes the required time interval to this register. When the 8-bit interval counter is equal to the time interval value loaded in the INTVAL SFR, the TII bit (TIMECON.2) bit is set and generates an interrupt if enabled.

A6H

00H

No

0 to 255 decimal

HTHSEC

Function

SFR Address

Power-On Default Value

Bit Addressable

Valid Value

Hundredths Seconds Time Register

This register is incremented in (1/128) second intervals once TCEN in TIMECON is active. The HTHSEC SFR counts from 0 to 127 before rolling over to increment the SEC time register.

A2H

00H

No

0 to 127 decimal

SEC

Function

SFR Address

Power-On Default Value

Bit Addressable

Valid Value

Seconds Time Register

This register is incremented in 1-second intervals once TCEN in TIMECON is active. The SEC SFR counts from 0 to 59 before rolling over to increment the MIN timeregister.

A3H

00H

No

0 to 59 decimal

MIN

Function

SFR Address

Power-On Default Value

Bit Addressable

Valid Value

Minutes Time Register

This register is incremented in 1-minute intervals once TCEN in TIMECON is active. The MIN counts from 0 to 59 before rolling over to increment the HOUR time register

A4H

00H

No

0 to 59 decimal

HOUR

Function

SFR Address

Power-On Default Value

Bit Addressable

Valid Value

Hours Time Register

This register is incremented in 1-hour intervals once TCEN in TIMECON is active. The HOUR SFR counts from 0 to 23 before rolling over to 0.

A5H

00H

No

0 to 23 decimal

8052-COMPATIBLE ON CHIP PERIPHERALS

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

Parallel I/O

The ADuC831 uses four input/output ports to exchange data with external devices. In addition to performing general-purpose I/O, some ports are capable of external memory operations while others are multiplexed with alternate functions for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Port 0

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

Figure 38 shows a typical bit latch and I/O buffer for a Port 0 port pin. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which will clock in a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal. See the following "Read-Modify-Write" section for more details.

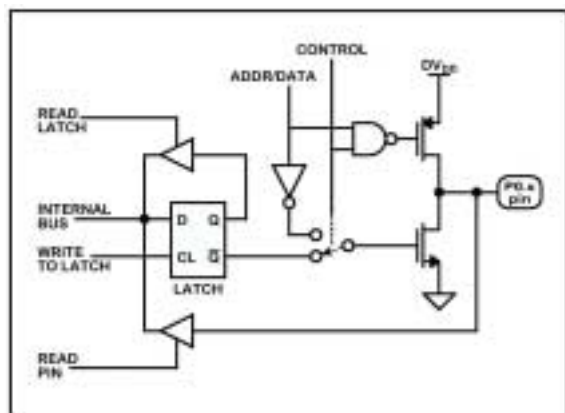


Figure 38 – Port 0 Bit Latch and I/O Buffer

As shown in figure 38 the output drivers of Port 0 pins are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses the P0 SFR gets 1s written to it (i.e. all of its bit latches become 1). When accessing external memory, the CONTROL signal in figure 1 goes high, enabling push-pull operation of the output pin from the internal address or data bus (ADDR/DATA line). Therefore, no external pull-ups are required on Port 0 in order for it to access external memory.

In general purpose I/O port mode, Port 0 pins that have 1s written to them via the Port 0 SFR will be configured as "open drain" and will therefore float. In this state, Port 0 pins can be used as high

impedance inputs. This is represented in figure 1 by the NAND gate whose output remains high as long as the CONTROL signal is low, thereby disabling the top FET. External pull-up resistors are therefore required when Port 0 pins are used as general-purpose outputs. Port 0 pins with 0's written to them will drive a logic low output voltage (V_{OL}) and will be capable of sinking 1.6mA.

Port 1

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

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

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

Table XVIII. Port 1, Alternate Pin Functions

Pin	Alternate Function
P1.0	T2 (Timer/Counter 2 External Input)
P1.1	T2EX (Timer/Counter 2 Capture/Reload Trigger)
P1.5	SS (Slave Select for the SPI interface)

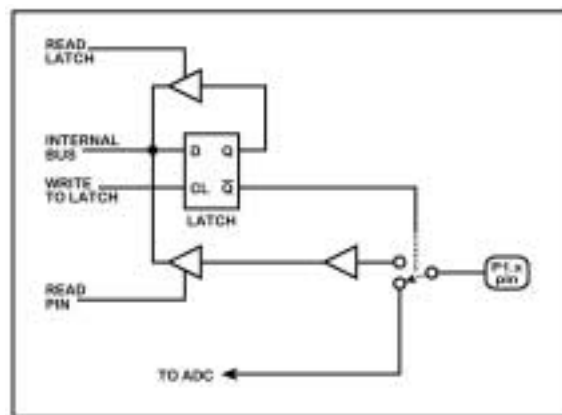


Figure 39 – Port 1 Bit Latch and I/O Buffer

Port 2

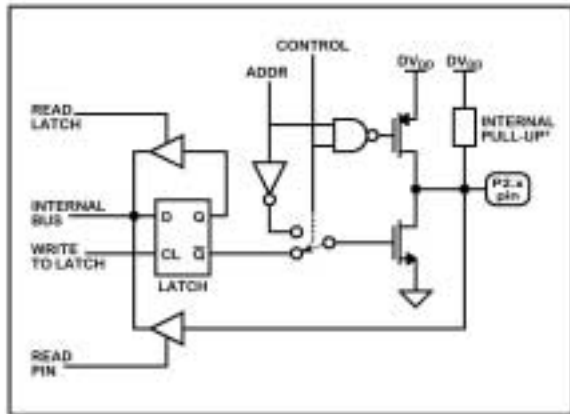
Port 2 is a bi-directional port with internal pull-up resistors directly controlled via the P2 SFR. Port 2 also emits the high order address bytes during fetches from external program memory and middle and high order address bytes during accesses to the 24-bit external data memory space.

As shown in figure 40 the output drivers of Ports 2 are switchable to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses (as for Port 0). In external memory addressing mode (CONTROL=1) the port pins feature push/pull operation controlled by the internal address bus (ADDR line). However unlike the P0 SFR during external memory accesses, the P2 SFR remains unchanged.

ADuC831

In general purpose I/O port mode, Port 2 pins that have 1s written to them are pulled high by the internal pull-ups (figure 41) and, 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 pins with 0's written to them will drive a logic low output voltage (V_{OL}) and will be capable of sinking 1.6mA.

P2.6 and P2.7 can also be used as PWM outputs. In the case that they are selected as the PWM outputs via the CFG831 SFR, the PWM outputs will overwrite anything written to P2.6 or P2.7



*See Figure 3 for details of internal pull-up.

Figure 40– Port 2 Bit Latch and I/O Buffer

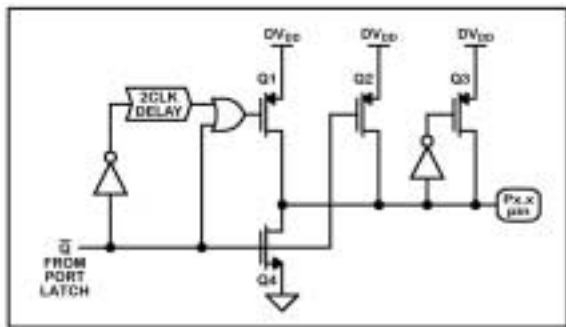


Figure 41 – Internal Pull-Up Configuration

Port 3

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

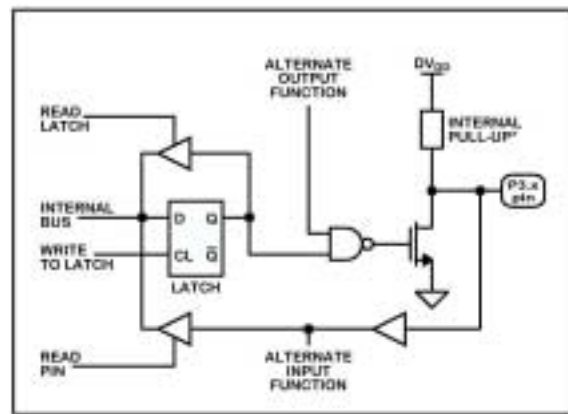
Port 3 pins also have various secondary functions described in Table XIX. The alternate functions of Port 3 pins can only be activated if the corresponding bit latch in the P3 SFR contains a 1. Otherwise, the port pin is stuck at 0.

Table XIX. Port 3, Alternate Pin Functions

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

P3.4 and P2.3 can also be used as PWM outputs. In the case that they are selected as the PWM outputs via the CFG831 SFR, the PWM outputs will overwrite anything written to P3.4 or P3.3

The internal pullup configuration is also defined by that in figure 41.



*See Figure 3 for details of internal pull-up.

Figure 42 – Port 3 Bit Latch and I/O Buffer

Additional Digital I/O

In addition to the port pins, the dedicated SPI/I²C pins (SCLOCK and SDATA/MOSI) also feature both input and output functions. Their equivalent I/O architectures are illustrated in figure 43 and figure 45 respectively for SPI operation and in figure 44 and figure 46 for I²C operation.

Notice that in I²C mode (SPE=0) the strong pull-up FET (Q1) is disabled leaving only a weak pull-up (Q2) present. By contrast, in SPI mode (SPE=1) the strong pull-up FET (Q1) is controlled directly by SPI hardware, giving the pin push/pull capability.

In I²C mode (SPE=0) two pull-down FETs (Q3 and Q4) operate in parallel in order to provide an extra 60% or 70% of current sinking capability. In SPI mode, however, (SPE=1) only one of the pull-down FETs (Q3) operates on each pin resulting in sink capabilities identical to that of port 0 and port 2 pins.

On the input path of SCLOCK, notice that a Schmitt trigger conditions the signal going to the SPI hardware to prevent false triggers (double triggers) on slow incoming edges. For incoming

signals from the SCLOCK and SDATA pins going to I²C hardware, a filter conditions the signals in order to reject glitches of up to 50ns in duration.

Notice also that direct access to the SCLOCK and SDATA/MOSI pins is afforded through the SFR interface in I²C master mode. Therefore, if you are not using the SPI or I²C functions, you can use these two pins to give additional high current digital outputs.

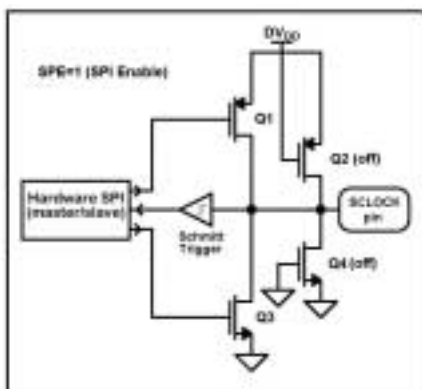


Figure 43 – SCLOCK pin I/O functional equivalent in SPI mode

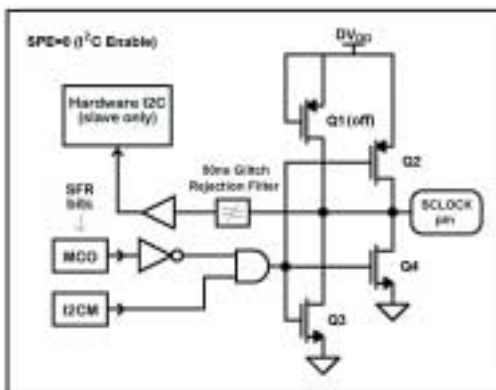


Figure 44 – SCLOCK pin I/O functional equivalent in I²C mode

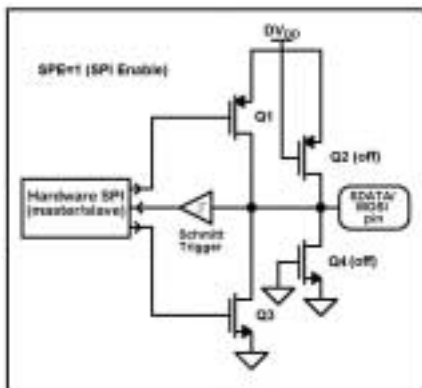


Figure 45 – SDATA/MOSI pin I/O functional equivalent in SPI mode

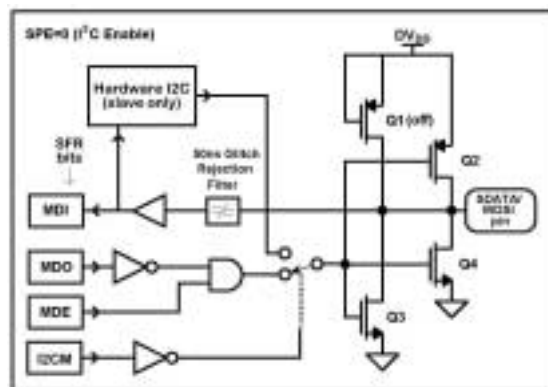


Figure 46 – SDATA/MOSI pin I/O functional equivalent in I²C mode

MISO is shared with P3.3 and as such as the same configuration as shown in figure 42.

Read-Modify-Write Instructions

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

ANL	(logical AND, e.g. ANL P1, A)
ORL	(logical OR, e.g. ORL P2, A)
XRL	(logical EX-OR, e.g. XRL P3, A)
JBC	(jump if bit = 1 and clear bit, e.g. JBC P1.1, LABEL)
CPL	(complement bit, e.g. CPL P3.0)
INC	(increment, e.g. INC P2)
DEC	(decrement, e.g. DEC P2)
DJNZ	(decrement and jump if not zero, e.g. DJNZ P3, LABEL)
MOV PX.Y, C*	(move carry to bit Y of Port X)
CLR PX.Y*	(clear bit Y of Port X)
SETB PX.Y*	(set bit Y of port X)

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

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

ADuC831

Timers/Counters

The ADuC831 has three 16-bit Timer/Counters: Timer 0, Timer 1, and Timer 2. The Timer/Counter hardware has been included on-chip to relieve the processor core of the overhead inherent in implementing timer/counter functionality in software. Each Timer/Counter consists of two 8-bit registers THx and TLx (x = 0, 1 and 2). All three can be configured to operate either as timers or event counters.

In “Timer” function, the TLx register is incremented every machine cycle. Thus one can think of it as counting machine cycles. Since a machine cycle consists of 12 core clock periods, the maximum count rate is 1/12 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, namely:

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

Table XX. TMOD SFR Bit Designations

Bit	Name	Description
7	Gate	Timer 1 Gating Control. Set by software to enable timer/counter 1 only while INT1 pin is high and TR1 control bit is set. Cleared by software to enable timer 1 whenever TR1 control bit is set.
6	C/T	Timer 1 Timer or Counter Select Bit. Set by software to select counter operation (input from T1 pin). Cleared by software to select timer operation (input from internal system clock).
5	M1	Timer 1 Mode Select Bit 1 (Used with M0 Bit).
4	M0	Timer 1 Mode Select Bit 0.
	M1 M0	
	0 0	TH1 operates as an 8-bit timer/counter. TL1 serves as 5-bit prescaler.
	0 1	16-Bit Timer/Counter. TH1 and TL1 are cascaded; there is no prescaler.
	1 0	8-Bit Auto-Reload Timer/Counter. TH1 holds a value which 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 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 Auto-Reload Timer/Counter. TH0 holds a value which 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

NOTE

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

Table XXI. TCON SFR Bit Designations

Bit	Name	Description
7	TF1	Timer 1 Overflow Flag. Set by hardware on a timer/counter 1 overflow. Cleared by hardware when the Program Counter (PC) vectors to the interrupt service routine.
6	TR1	Timer 1 Run Control Bit. Set by 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 ¹	External Interrupt 1 (INT1) Flag. Set by hardware by a falling edge or zero level being applied to external interrupt pin INT1, depending on bit IT1 state. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
2	IT1 ¹	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 ¹	External Interrupt 0 (INT0) Flag. Set by hardware by a falling edge or zero level being applied to external interrupt pin 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 ¹	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/Counter 0 and 1 Data Registers

Each timer consists of two 8-bit registers. These can be used as independent registers or combined to be a single 16-bit register depending on the timer mode configuration.

TH0 and TL0

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

TH1 and TL1

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

ADuC831

TIMER/COUNTER 0 AND 1 OPERATING MODES

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

Mode 0 (13-Bit Timer/Counter)

Mode 0 configures an 8-bit timer/counter with a divide-by-32 prescaler. Figure 47 shows mode 0 operation.

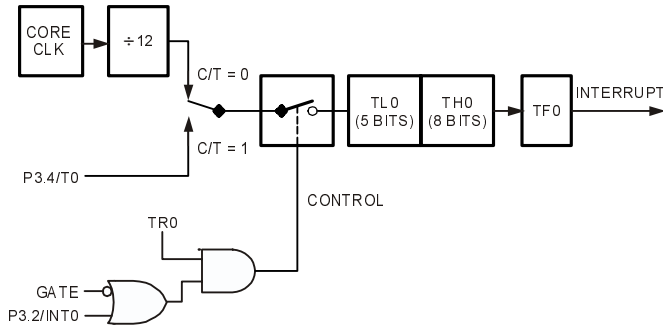


Figure 47. Timer/Counter 0, Mode 0

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 INT0 = 1. Setting Gate = 1 allows the timer to be controlled by external input 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 48.

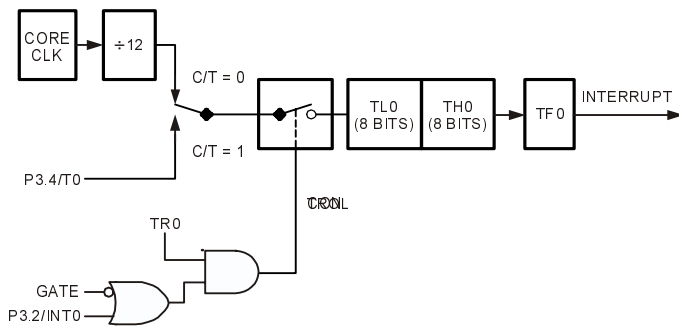


Figure 48. Timer/Counter 0, Mode 1

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 49. 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.

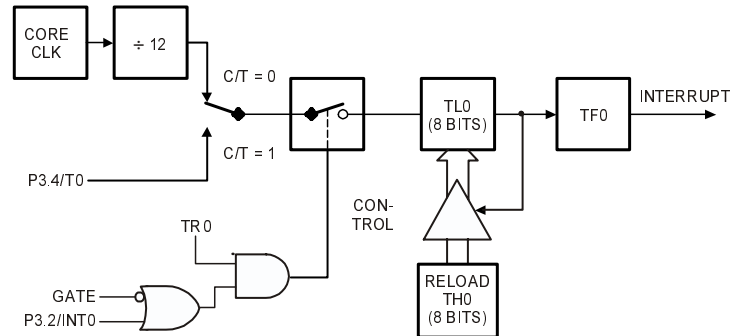


Figure 49. Timer/Counter 0, Mode 2

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 50. TL0 uses the timer 0 control bits: C/T, Gate, TR0, 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.

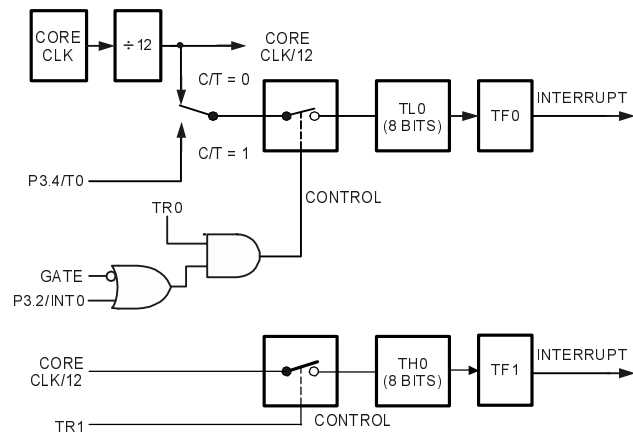


Figure 50. Timer/Counter 0, Mode 3

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

Table XXII. T2CON SFR Bit Designations

Bit	Name	Description
7	TF2	Timer 2 Overflow Flag. Set by hardware on a timer 2 overflow. TF2 will not be set when either RCLK 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 user to select counter function (input from external T2 pin). Cleared by 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 auto-reloads with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to autoreload on Timer 2 overflow.

Timer/Counter 2 Data Registers

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

TH2 and TL2

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

RCAP2H and RCAP2L

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

ADuC831

Timer/Counter Operation Modes

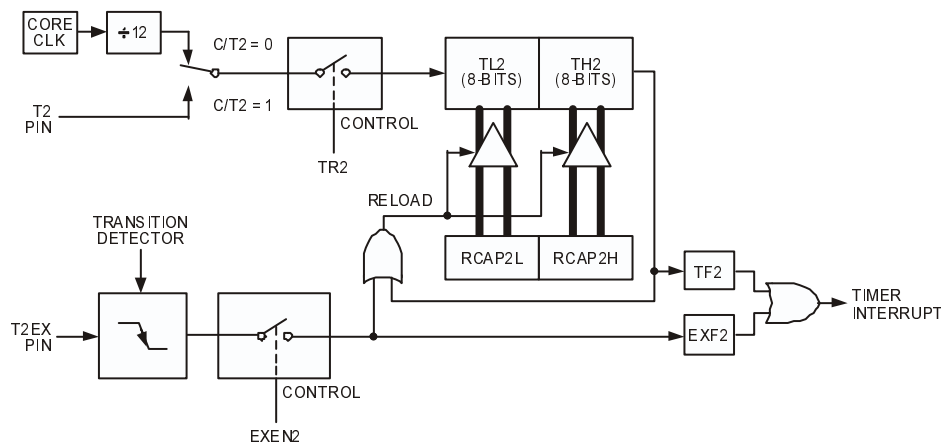
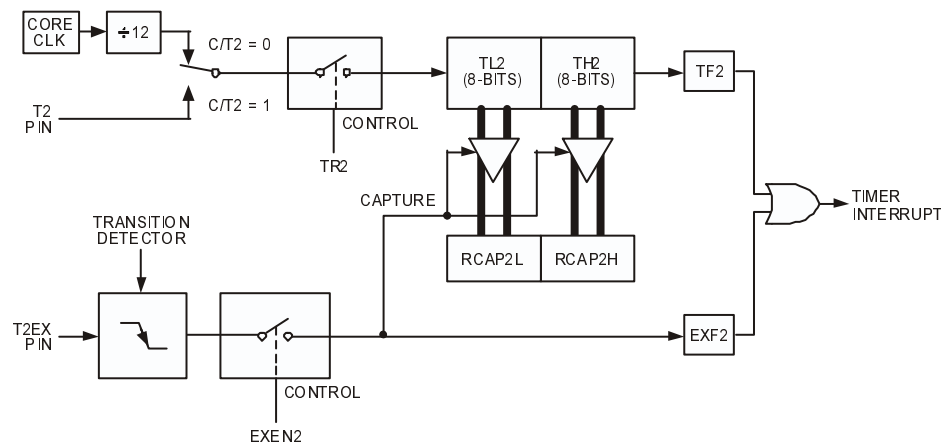
The following paragraphs describe the operating modes for timer/counter 2. The operating modes are selected by bits in the T2CON SFR as shown in Table XXIII.

Table XXIII. T2CON OPERATING MODES

RCLK (or) TCLK	CAP2	TR2	MODE
0	0	1	16-Bit Autoreload
0	1	1	16-Bit Capture
1	X	1	Baud Rate
X	X	0	OFF

16-Bit Autoreload Mode

In “Autoreload” mode, there are two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still performs the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2. The Autoreload mode is illustrated in Figure 51 below.

**Figure 51. Timer/Counter 2, 16-Bit Autoreload Mode****Figure 52. Timer/Counter 2, 16-Bit Capture Mode****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 which, upon overflowing, sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still performs the above, but a 1-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 52

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. Hence Timer 2 interrupts will not occur so they do not have to be disabled. In this mode the EXF2 flag, however, can still cause interrupts and this can be used as a third external interrupt.

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

UART SERIAL INTERFACE

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

TXD(P3.1) while the SFR interface to the UART is comprised of SBUF and SCON, as described below.

SBUF

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

SCON

SFR Address
Power-On Default Value
Bit Addressable

UART Serial Port Control Register

98H
00H
Yes

Table XXIV. SCON SFR Bit Designations

Bit	Name	Description															
7	SM0	UART Serial Mode Select Bits.															
6	SM1	These bits select the Serial Port operating mode as follows: <table> <tr> <td>SM0</td><td>SM1</td><td>Selected Operating Mode</td></tr> <tr> <td>0</td><td>0</td><td>Mode 0: Shift Register, fixed baud rate (Core_Clk/2)</td></tr> <tr> <td>0</td><td>1</td><td>Mode 1: 8-bit UART, variable baud rate</td></tr> <tr> <td>1</td><td>0</td><td>Mode 2: 9-bit UART, fixed baud rate (Core_Clk/64) or (Core_Clk/32)</td></tr> <tr> <td>1</td><td>1</td><td>Mode 3: 9-bit UART, variable baud rate</td></tr> </table>	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
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.															

ADuC831

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 53.

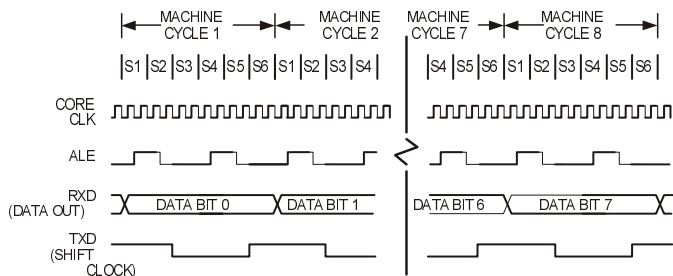


Figure 53. UART Serial Port Transmission, Mode 0

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 54.

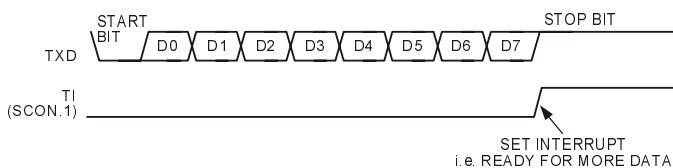


Figure 54. UART Serial Port Transmission, Mode 0

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 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:

$$\text{Mode 0 Baud Rate} = (\text{Core Clock Frequency}/12)$$

Mode 2 Baud Rate Generation

The baud rate in Mode 2 depends on the value of the SMOD bit in the PCON SFR. If SMOD = 0, the baud rate is 1/64 of the core clock. If SMOD = 1, the baud rate is 1/32 of the core clock:

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

ADuC831

BAUD RATE GENERATION USING TIMER 3

The high integer dividers in a UART block means that high speed baud rates are not always possible using some particular crystals. e.g. using a 12MHz crystal a baud rate of 115200 is not possible. To address this problem the ADuC831 has added a dedicated baud rate timer (timer 3) specifically for generating highly accurate baudrates.

Timer 3 can be used instead of timer1 or timer2 for generating very accurate high speed UART baudrates including 115200 and 230400. Timer 3 also allows a much wider range of baud rates to be obtained. In fact, every desired bit rate from 12bit/s to 393216 bit/s can be generated to within an error of $\pm 0.8\%$. Timer 3 also frees up the other three timers allowing them to be used for different applications. A block diagram of timer 3 is shown in figure 56 below.

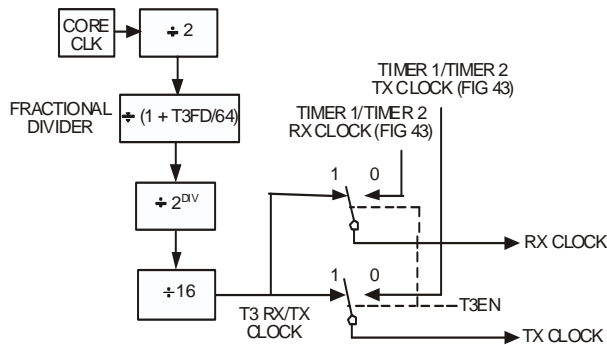


Figure 56. Timer 3, UART Baud Rates

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

Table XXVII. T3CON SFR Bit Designations

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

The appropriate value to write to the DIV2-1-0 bits can be calculated using the following formula where fcore is the crystal frequency.

Note: The DIV value must be rounded down.

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

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

Note: T3FD should be rounded to the nearest integer.

$$T3FD = \frac{2 \times f_{CORE}}{2^{DIV} \times BaudRate} - 64$$

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

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

e.g. to get a baud rate of 115200 while operating at 11.0592MHz

$$DIV = \log(11059200 / (32 \times 115200)) / \log 2 = 1.58 = 1$$

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

therefore the actual baud rate is 115200bit/s

Table XXVIII. Commonly used Baud Rates using Timer 3

Ideal Baud	Crystal	DIV	T3CON	T3FD	Error
230400	11.0592	0	80h	20h	0.0%
115200	11.0592	1	81h	20h	0.0%
57600	11.0592	2	82h	20h	0.0%
38400	11.0592	3	83h	08h	0.0%
19200	11.0592	4	84h	08h	0.0%
9600	11.0592	5	85h	08h	0.0%
230400	12	0	80h	28h	0.16%
115200	12	1	81h	28h	0.16%
57600	12	2	82h	28h	0.16%
38400	12	3	83h	0Eh	0.16%
19200	12	4	84h	0Eh	0.16%
9600	12	5	85h	0Eh	0.16%
230400	14	0	80h	3Ah	0.39%
115200	14	1	81h	3Ah	0.39%
57600	14	2	82h	3Ah	0.39%
38400	14	3	83h	1Bh	0.16%
19200	14	4	84h	1Bh	0.16%
9600	14	5	85h	1Bh	0.16%
230400	16	1	81h	05h	0.64%
115200	16	2	82h	05h	0.64%
57600	16	3	83h	05h	0.64%
38400	16	3	83h	28h	0.16%
19200	16	4	84h	28h	0.16%
9600	16	5	85h	28h	0.16%

INTERRUPT SYSTEM

The ADuC831 provides a total of nine interrupt sources with two priority levels. The control and configuration of the interrupt system is carried out through three Interrupt-related SFRs.

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

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

Table XXIX. IE SFR Bit Designations

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

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

Table XXX. IP SFR Bit Designations

Bit	Name	Description
7	PSI	Written by User to Select SPI/I2C 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)

ADuC831

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

Table XXXI. IEIP2 SFR Bit Designations

Bit	Name	Description
7	—	Reserved for Future Use.
6	PTI	Priority for time interval interrupt
5	PPSM	Priority for Power Supply monitor interrupt
4	PSI	Priority for SPI interrupt
3	—	This bit must contain zero
2	ETI	Written by User to Enable “1” or Disable “0” Time Interval Counter Interrupt.
1	EPSMI	Written by User to Enable “1” or Disable “0” Power Supply Monitor Interrupt.
0	ESI	Written by User to Enable “1” or Disable “0” SPI/I2C Serial Port Interrupt.

Interrupt Priority

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

Table XXXII. Priority within an Interrupt Level

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

Interrupt Vectors

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

Table XXXIII. Interrupt Vector Addresses

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

ADuC831 HARDWARE DESIGN CONSIDERATIONS

This section outlines some of the key hardware design considerations that must be addressed when integrating the ADuC831 into any hardware system.

Clock Oscillator

The clock source for the ADuC831 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 XTAL1 and XTAL2, and connect a capacitor from each pin to ground as shown below.

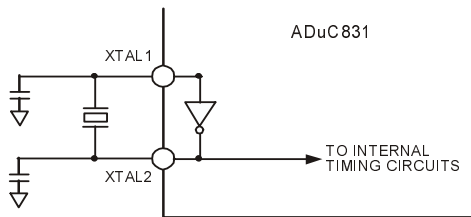


Figure 57. External Parallel Resonant Crystal Connections

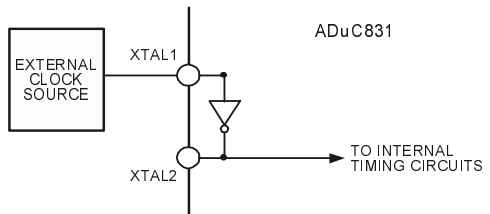


Figure 58. Connecting an External Clock Source

Whether using the internal oscillator or an external clock source, the ADuC831's specified operational clock speed range is 300 kHz to 16 MHz. The core itself is static, and will function all the way down to dc. But at clock speeds slower than 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 ADuC831 can access up to 64K bytes of external program memory (ROM/PROM/etc.) and up to 16M bytes of external data memory (SRAM).

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

Note that a second very important function of the EA pin is described in the Single Pin Emulation Mode section of this data sheet.

External program memory (if used) must be connected to the ADuC831 as illustrated in Figure 59. 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 PSEN strobes the EPROM and the code byte is read into the ADuC831.

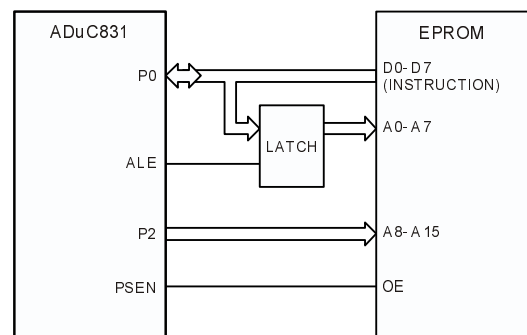


Figure 59. External Program Memory Interface

Note that program memory addresses are always 16 bits wide, even in cases where the actual amount of program memory used is less than 64K 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 60 shows a hardware configuration for accessing up to 64K bytes of external RAM. This interface is standard to any 8051-compatible MCU.

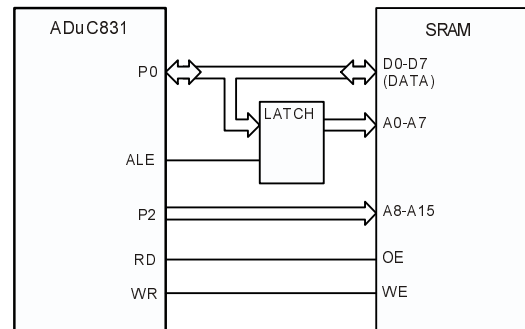


Figure 60. External Data Memory Interface (64 K Address Space)

ADuC831

If access to more than 64K bytes of RAM is desired, a feature unique to the ADuC831 allows addressing up to 16M bytes of external RAM simply by adding an additional latch as illustrated in Figure 61.

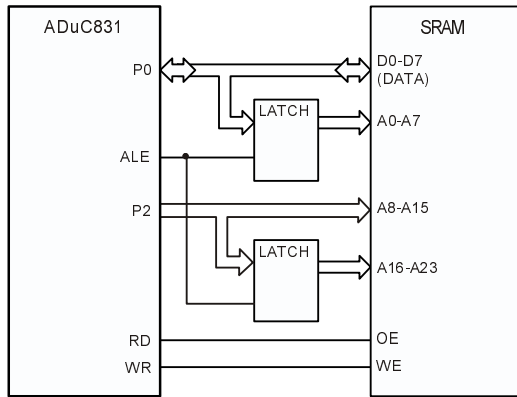


Figure 61. External Data Memory Interface (16 M Bytes Address Space)

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 ADuC831 (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.

Power Supplies

The ADuC831'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 $\pm 10\%$ of the nominal 5 V level, the chip will function equally well at any power supply level between 2.7 V and 5.5 V.

Note: The figures below refer to the PQFP package, for the CSP package connect the extra DVdd, DGnd, AVdd and AGnd in the same manner

Separate analog and digital power supply pins (AV_{DD} and DV_{DD} respectively) allow AV_{DD} to be kept relatively free of noisy digital signals often present on the system DV_{DD} line. However, though you can power AV_{DD} and DV_{DD} from two separate supplies if desired, you must ensure that they remain within ± 0.3 V of one another at all times in order to avoid damaging the chip (as per the Absolute Maximum Ratings section of this data sheet). 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 62.

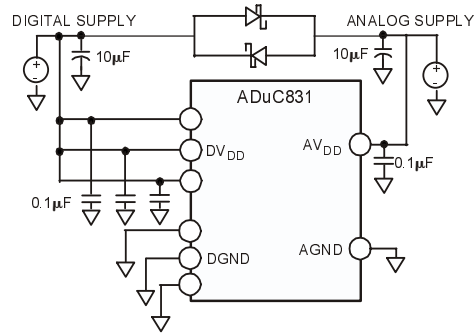


Figure 62. External Dual-Supply Connections

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 63. With this configuration other analog circuitry (such as op amps, voltage reference, etc.) 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.

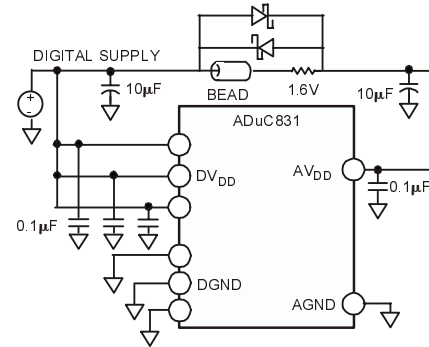


Figure 63. External Single-Supply Connections

Notice that in both Figure 62 and Figure 63, a large value (10 μ F) reservoir capacitor sits on DV_{DD} and a separate 10 μ F capacitor sits on AV_{DD} . Also, local small-value (0.1 μ F) 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 ADuC831 must be referenced to the same system ground reference point.

Power Consumption

The currents consumed by the various sections of the ADuC831 are shown in Table XXXIV. The “CORE” values given represent the current drawn by DV_{DD}, while the rest (“ADC,” “DAC,” “voltage ref”) are pulled by the AV_{DD} pin and can be disabled in software when not in use. The other on-chip peripherals (watchdog timer, power supply monitor, etc.) consume negligible current and are therefore lumped in with the “CORE” operating current here. Of course, the user must add any currents sourced by the parallel and serial I/O pins, and that sourced by the DAC, in order to determine the *total* current needed at the ADuC831’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 XXXIV. Typical I_{DD} of Core and Peripherals

	VDD = 5 V	VDD = 3 V
Core: (Normal Mode)	(1.6 nAs x MCLK) + 6 mA	(0.8 nAs x MCLK) + 3 mA
Core: (Idle Mode)	(0.75 nAs x MCLK) + 5 mA	(0.25 nAs x 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 XXXIV are given as functions of M_{CLK}, the oscillator frequency. Plug in a value for M_{CLK} in hertz to determine the current consumed by the core at that oscillator frequency. Since the ADC and DACs can be enabled or disabled in software, add only the currents from the peripherals you expect to use. 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, etc., 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.

Power-Saving 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 on 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). During full power-down mode, the ADuC831 consumes a total of approximately 15 µA. There are five ways of terminating power-down mode:

Asserting the RESET pin (#15)

Returns to normal mode all registers are set to their default state and program execution starts at the reset vector once the Reset pin is de-asserted.

Cycling Power

All registers are set to their default state and program execution starts at the reset vector approximately 128ms later.

Time Interval Counter (TIC) Interrupt

Power-down mode is terminated and the CPU services the TIC interrupt. The RETI at the end of the TIC ISR will return the core to the instruction after that which enabled power down.

I²C or SPI Interrupt

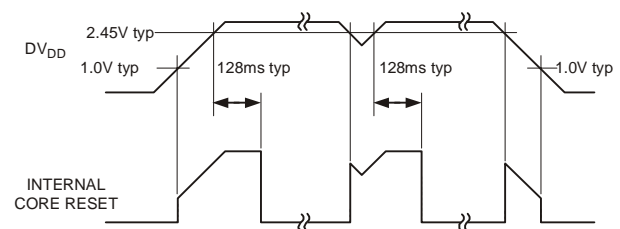
Power-down mode is terminated and the CPU services the I²C/SPI interrupt. The RETI at the end of the ISR will return the core to the instruction after that which enabled power down. It should be noted that the I²C/SPI power down interrupt enable bit (SERIPD) in the PCON SFR must first be set to allow this mode of operation.

INT0 Interrupt

Power-down mode is terminated and the CPU services the INT0 interrupt. The RETI at the end of the ISR will return the core to the instruction after that which enabled power-down. It should be noted that the INT0 power-down interrupt enable bit (INT0PD) in the PCON SFR must first be set to allow this mode of operation.

Power-On Reset

An internal POR (Power-On-Reset) is implemented on the ADuC831. For DV_{DD} below 2.45V, the internal POR will hold the ADuC831 in reset. As DV_{DD} rises above 2.45V an internal timer will timeout for 128ms approx. before the part is released from reset with a 16MHz crystal. With other crystal values the timeout will increase. The user must ensure that the power supply must have reached a stable 2.7V minimum level by this time. Likewise on powerdown the internal POR will hold the ADuC831 in reset until the power supply has dropped below 1V. Figure 64 illustrates the operation of the internal POR in detail.

**Figure 64. Internal POR Operation**

ADuC831

Grounding and Board Layout Recommendations

As with all high resolution data converters, special attention must be paid to grounding and PC board layout of ADuC831-based designs in order to achieve optimum performance from the ADCs and DAC.

Although the ADuC831 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 ADuC831, as illustrated in the simplified example of Figure 65a. In systems where digital and analog ground planes are connected together somewhere else (at the system's power supply for example), they cannot be connected again near the ADuC831 since a ground loop would result. In these cases, tie the ADuC831's AGND and DGND pins all to the analog ground plane, as illustrated in Figure 65b. 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 ADuC831 can then be placed between the digital and analog sections, as illustrated in Figure 65c.

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 65b 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 65c. Whenever possible, avoid large discontinuities in the ground plane(s) (such as are formed by a long trace on the same layer), since they force return signals to travel a longer path. And of course, make all connections to the ground plane directly, with little or no trace separating the pin from its via to ground.

If the user plans to connect fast logic signals (rise/fall time < 5 ns) to any of the ADuC831's digital inputs, add a series resistor to each relevant line to keep rise and fall times longer than 5 ns at the ADuC831 input pins. A value of 100Ω or 200Ω is usually sufficient to prevent high-speed signals from coupling capacitively into the ADuC831 and affecting the accuracy of ADC conversions.

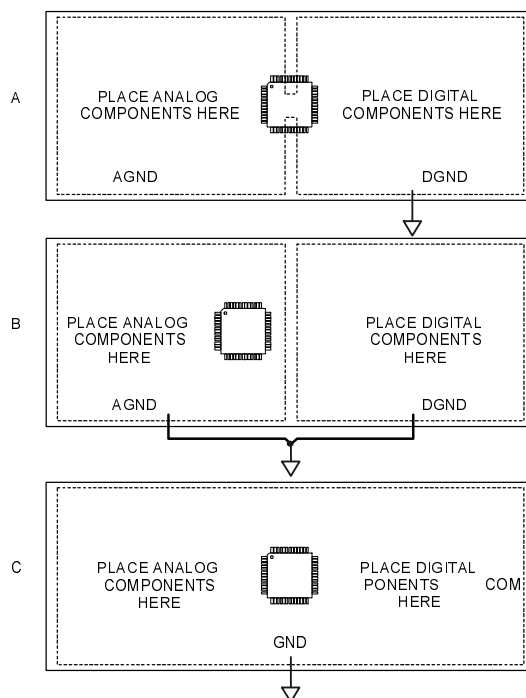


Figure 65. System Grounding Schemes

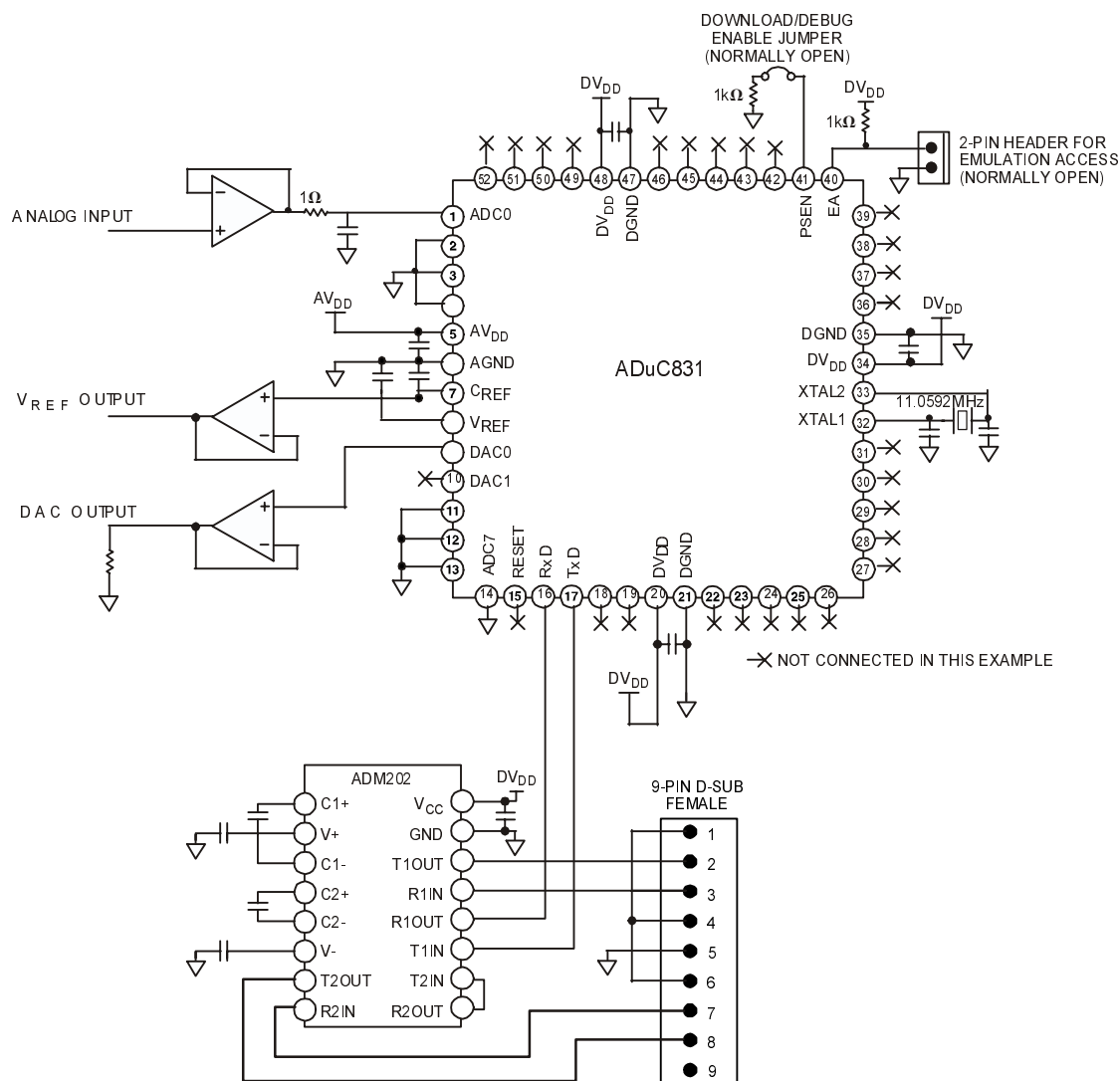


Figure 66 Example ADuC831 System (PQFP Package)

OTHER HARDWARE CONSIDERATIONS

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

In-Circuit Serial Download Access

Nearly all ADuC831 designs will want to take advantage of the in-circuit reprogrammability of the chip. This is accomplished by a connection to the ADuC831'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 57 with a simple ADM202-based circuit. If users would rather not design an RS-232 chip onto a board, refer to the application note "uC006-A 4-Wire UART-to-PC Interface"* for a simple (and zero-cost-per-board) method of gaining in-circuit serial download access to the ADuC831.

In addition to the basic UART connections, users will also need a way to trigger the chip into download mode. This is accomplished via a 1 kΩ pull-down resistor that can be jumpered onto

the PSEN pin, as shown in Figure 66. To get the ADuC831 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 PSEN is normally an output (as described in the External Memory Interface section) and it 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 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 PSEN pin low, except for the external 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.

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

ADuC831

Note that the serial port debugger is fully contained on the ADuC831 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 ADuC831 is a dedicated controller for single-pin in-circuit emulation (ICE) using standard production ADuC831 devices. In this mode, emulation access is gained by connection to a single pin, the EA pin. Normally, this pin is hard-wired 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 EA pin high through a 1 k Ω resistor as shown in Figure 66. The emulator will then connect to the 2-pin header also shown in Figure 54. 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 66, 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

ADuC831 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 ADuC831 configuration is shown in Figure 66. 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 ADuC831. The system consists of the following PC-based (Windows-compatible) hardware and software development tools.

Hardware:	ADuC831 Evaluation Board, Plug-In Power Supply and Serial Port Cable
Code Development:	8051 Assembler
In-Circuit Code Download:	Serial Downloader
In-Circuit Debugger:	Serial Port Debugger
Miscellaneous Other:	CD-ROM Documentation and Two Additional Prototype Devices

Figure 67 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 below.



Figure 67. Components of the QuickStart Development System



Figure 68. 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. An Application Note (uC004) detailing this serial download protocol is available from 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 break-point code execution control.

TIMING SPECIFICATIONS^{1, 2, 3}

($AV_{DD} = DV_{DD} = 3.0\text{ V}$ or $5.0\text{ V} \pm 10\%$. All specifications $T_A = T_{MIN}$ to T_{MAX} unless otherwise noted.)

Parameter	12 MHz			Variable Clock			Unit	Figure
	Min	Typ	Max	Min	Typ	Max		
CLOCK INPUT (External Clock Driven XTAL1)								
t _{CK}		83.33		62.5		1000	ns	69
t _{CKL}	20			20			ns	69
t _{CKH}	20			20			ns	69
t _{CKR}			20			20	ns	69
t _{CKF}			20			20	ns	69
t _{CYC} ⁴		1			12t _{CK}		µs	

NOTES

¹AC inputs during testing are driven at $DV_{DD} - 0.5\text{ V}$ 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.

²For 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 Port0, ALE, PSEN outputs = 100 pF; C_{LOAD} for all other outputs = 80 pF unless otherwise noted.

⁴ADuC831 Machine Cycle Time is nominally defined as $MCLKIN/12$.

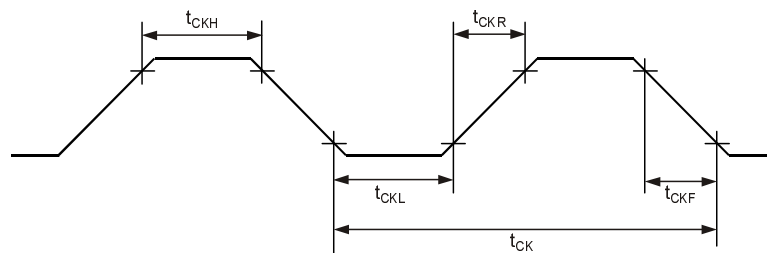


Figure 69. XTAL 1 Input

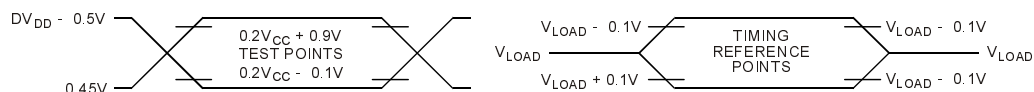


Figure 70. Timing Waveform Characteristics

ADuC831

Parameter		12 MHz		Variable Clock		Unit	Figure
		Min	Max	Min	Max		
EXTERNAL PROGRAM MEMORY							
t _{LHLL}	ALE Pulsewidth	127		2t _{CK} – 40		ns	71
t _{AVLL}	Address Valid to ALE Low	43		t _{CK} – 40		ns	71
t _{LLAX}	Address Hold after ALE Low	53		t _{CK} – 30		ns	71
t _{LLIV}	ALE Low to Valid Instruction In		234		4t _{CK} – 100	ns	71
t _{LLPL}	ALE Low to PSEN Low	53		t _{CK} – 30		ns	71
t _{PLPH}	PSEN Pulsewidth	205		3t _{CK} – 45		ns	71
t _{PLIV}	PSEN Low to Valid Instruction In		145		3t _{CK} – 105	ns	71
t _{PXIX}	Input Instruction Hold after PSEN	0		0		ns	71
t _{PXIZ}	Input Instruction Float after PSEN		59		t _{CK} – 25	ns	71
t _{AVIV}	Address to Valid Instruction In		312		5t _{CK} – 105	ns	71
t _{PLAZ}	PSEN Low to Address Float		25		25	ns	71
t _{PHAX}	Address Hold after PSEN High	0		0		ns	71

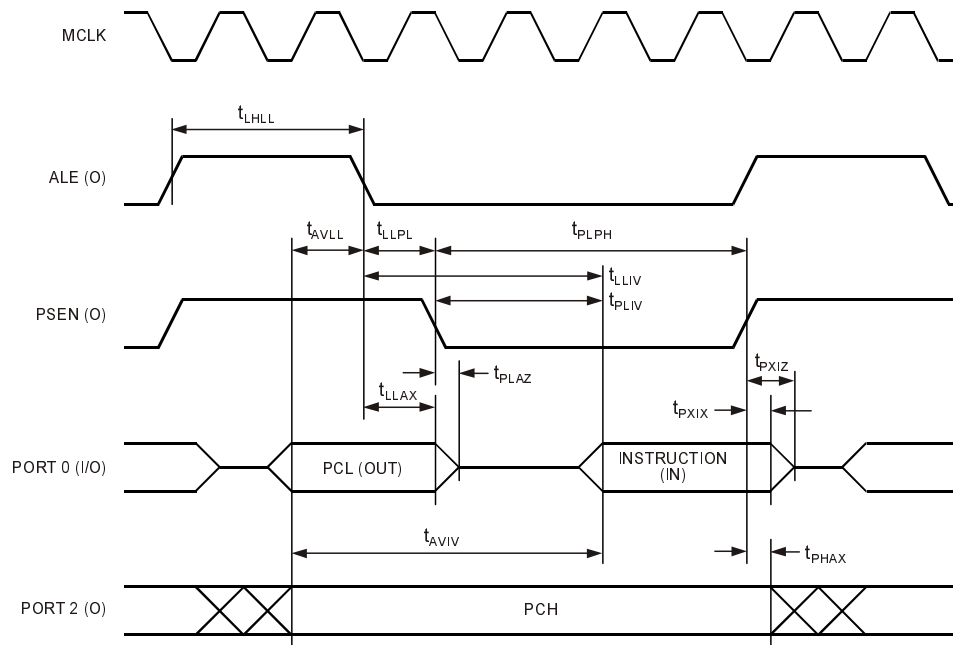


Figure 71. External Program Memory Read Cycle

Parameter		12 MHz		Variable Clock		Unit	Figure
		Min	Max	Min	Max		
EXTERNAL DATA MEMORY READ CYCLE							
t _{RLRH}	RD Pulsewidth	400		6t _{CK} – 100		ns	72
t _{AVLL}	Address Valid after ALE Low	43		t _{CK} – 40		ns	72
t _{LLAX}	Address Hold after ALE Low	48		t _{CK} – 35		ns	72
t _{RLDV}	RD Low to Valid Data In		252		5t _{CK} – 165	ns	72
t _{RHDX}	Data and Address Hold after RD	0		0		ns	72
t _{RHDZ}	Data Float after RD		97		2t _{CK} – 70	ns	72
t _{LLDV}	ALE Low to Valid Data In		517		8t _{CK} – 150	ns	72
t _{AVDV}	Address to Valid Data In		585		9t _{CK} – 165	ns	72
t _{LLWL}	ALE Low to RD or WR Low	200	300	3t _{CK} – 50	3t _{CK} + 50	ns	72
t _{AVWL}	Address Valid to RD or WR Low	203		4t _{CK} – 130		ns	72
t _{RLAZ}	RD Low to Address Float		0		0	ns	72
t _{WHLH}	RD or WR High to ALE High	43	123	t _{CK} – 40	6t _{CK} – 100	ns	72

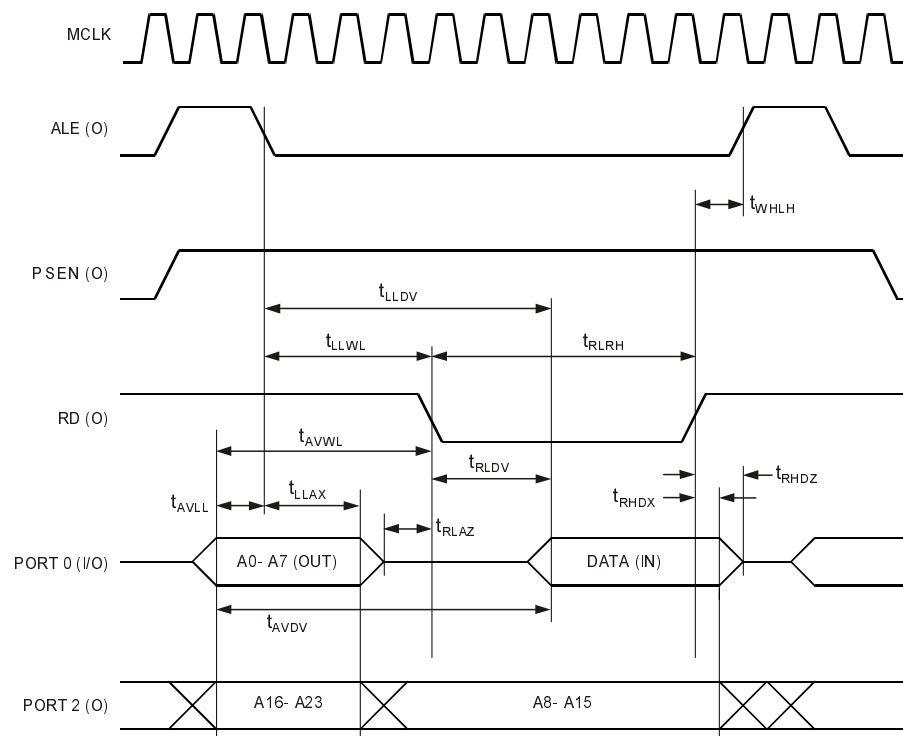


Figure 72. External Data Memory Read Cycle

ADuC831

Parameter		12 MHz		Variable Clock		Unit	Figure
		Min	Max	Min	Max		
EXTERNAL DATA MEMORY WRITE CYCLE							
tWLWH	WR Pulsewidth	400		6tCK – 100		ns	73
tAVLL	Address Valid after ALE Low	43		tCK – 40		ns	73
tLLAX	Address Hold after ALE Low	48		tCK – 35		ns	73
tLLWL	ALE Low to RD or WR Low	200	300	3tCK – 50	3tCK + 50	ns	73
tAVWL	Address Valid to RD or WR Low	203		4tCK – 130		ns	73
tQVWX	Data Valid to WR Transition	33		tCK – 50		ns	73
tQVWH	Data Setup Before WR	433		7tCK – 150		ns	73
tWHQX	Data and Address Hold after WR	33		tCK – 50		ns	73
tWHLH	RD or WR High to ALE High	43	123	tCK – 40	6tCK – 100	ns	73

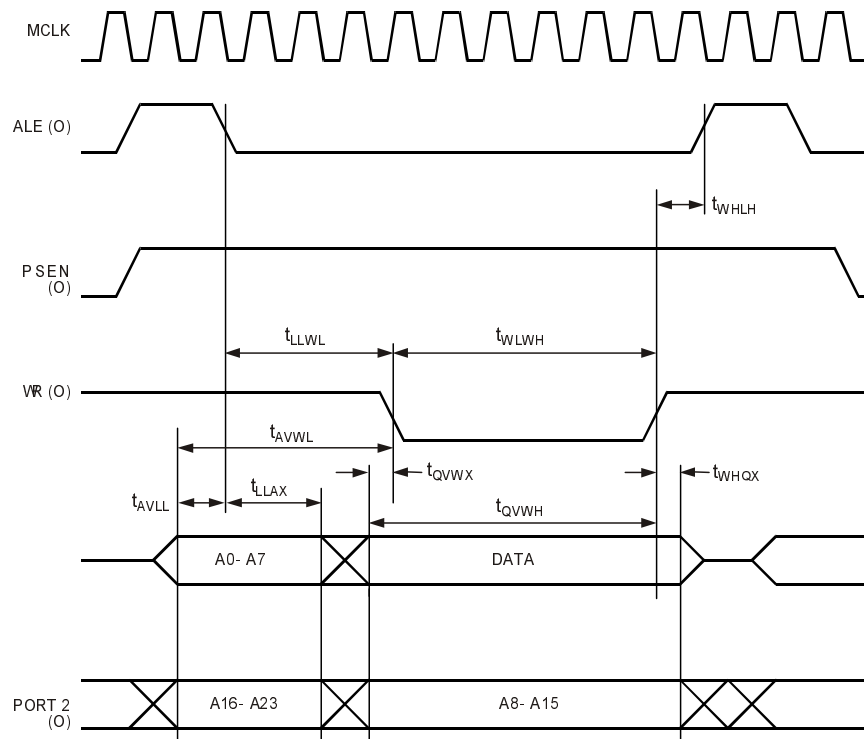


Figure 73. External Data Memory Write Cycle

Parameter	12 MHz			Variable Clock			Unit	Figure		
	Min	Typ	Max	Min	Typ	Max				
UART TIMING (Shift Register Mode)										
t _{XLXL}	Serial Port Clock Cycle Time			1.0			12t _{CK}	µs	74	
t _{QVXH}	Output Data Setup to Clock			700				10t _{CK} – 133	ns	74
t _{DVXH}	Input Data Setup to Clock			300				2t _{CK} + 133	ns	74
t _{XHDX}	Input Data Hold after Clock			0				0	ns	74
t _{XHQX}	Output Data Hold after Clock			50				2t _{CK} – 117	ns	74

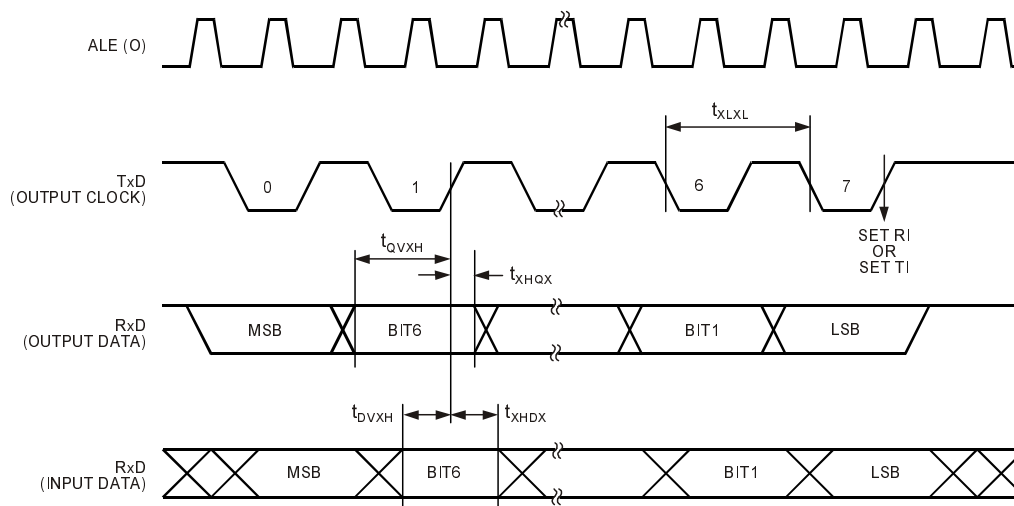
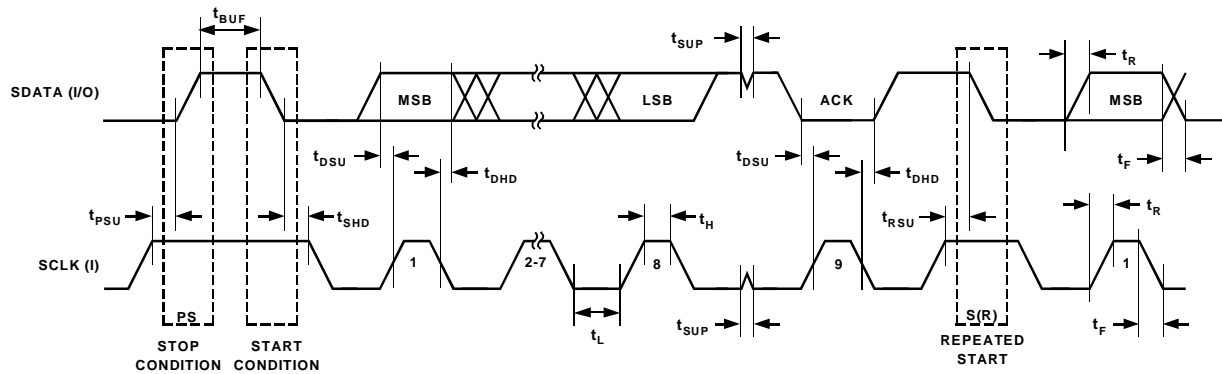


Figure 74. UART Timing in Shift Register Mode

ADuC831

Parameter		Min	Max	Unit	Figure
I²C-COMPATIBLE INTERFACE TIMING					
t_L	SCLOCK Low Pulsewidth	4.7		μ s	75
t_H	SCLOCK High Pulsewidth	4.0		μ s	75
t_{SHD}	Start Condition Hold Time	0.6		μ s	75
t_{DSU}	Data Setup Time	100		μ s	75
t_{DHD}	Data Hold Time		0.9	μ s	75
t_{RSU}	Setup Time for Repeated Start	0.6		μ s	75
t_{PSU}	Stop Condition Setup Time	0.6		μ s	75
t_{BUF}	Bus Free Time Between a STOP Condition and a START Condition	1.3		μ s	75
t_R	Rise Time of Both SCLOCK and SDATA		300	ns	75
t_F	Fall Time of Both SCLOCK and SDATA		300	ns	75
t_{SUP}^1	Pulsewidth of Spike Suppressed		50	ns	75

NOTE

¹Input filtering on both the SCLOCK and SDATA inputs suppresses noise spikes less than 50 ns.Figure 75 I²C-Compatible Interface Timing

Parameter		Min	Typ	Max	Unit	Figure
SPI MASTER MODE TIMING (CPHA = 1)						
t_{SL}	SCLOCK Low Pulsewidth		330		ns	76
t_{SH}	SCLOCK High Pulsewidth		330		ns	76
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns	76
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns	76
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns	76
t_{DF}	Data Output Fall Time		10	25	ns	76
t_{DR}	Data Output Rise Time		10	25	ns	76
t_{SR}	SCLOCK Rise Time		10	25	ns	76
t_{SF}	SCLOCK Fall Time		10	25	ns	76

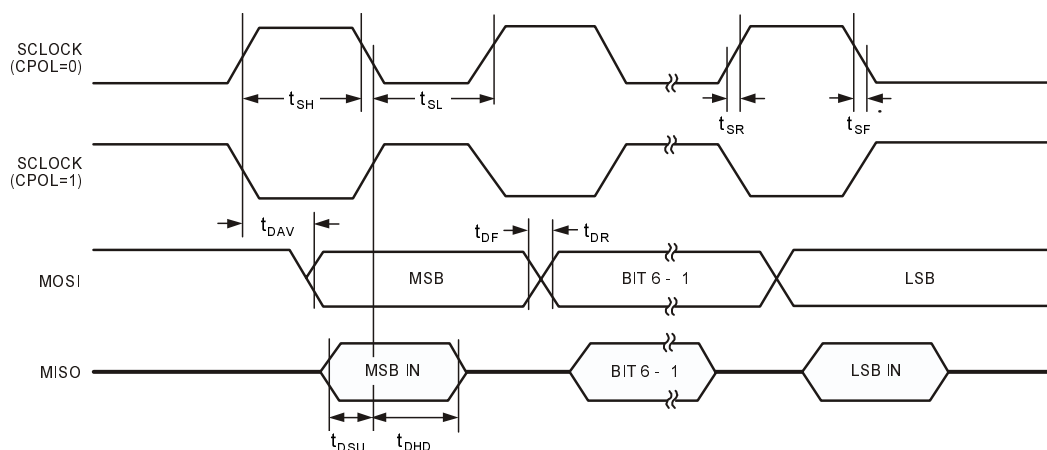


Figure 76. SPI Master Mode Timing (CPHA = 1)

ADuC831

Parameter		Min	Typ	Max	Unit	Figure
SPI MASTER MODE TIMING (CPHA = 0)						
t_{SL}	SCLOCK Low Pulsewidth		330		ns	77
t_{SH}	SCLOCK High Pulsewidth		330		ns	77
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns	77
t_{DOSU}	Data Output Setup before SCLOCK Edge			150	ns	77
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns	77
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns	77
t_{DF}	Data Output Fall Time		10	25	ns	77
t_{DR}	Data Output Rise Time		10	25	ns	77
t_{SR}	SCLOCK Rise Time		10	25	ns	77
t_{SF}	SCLOCK Fall Time		10	25	ns	77

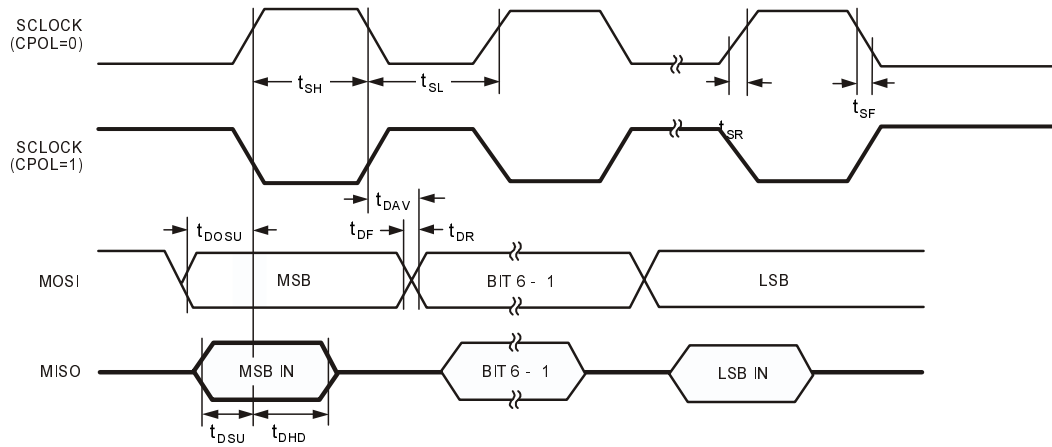


Figure 77. SPI Master Mode Timing (CPHA = 0)

Parameter		Min	Typ	Max	Unit	Figure
SPI SLAVE MODE TIMING (CPHA = 1)						
t_{SS}	SS to SCLOCK Edge	0			ns	78
t_{SL}	SCLOCK Low Pulsewidth		330		ns	78
t_{SH}	SCLOCK High Pulsewidth		330		ns	78
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns	78
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns	78
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns	78
t_{DF}	Data Output Fall Time		10	25	ns	78
t_{DR}	Data Output Rise Time		10	25	ns	78
t_{SR}	SCLOCK Rise Time		10	25	ns	78
t_{SF}	SCLOCK Fall Time		10	25	ns	78
t_{SFS}	SS High after SCLOCK Edge	0			ns	78

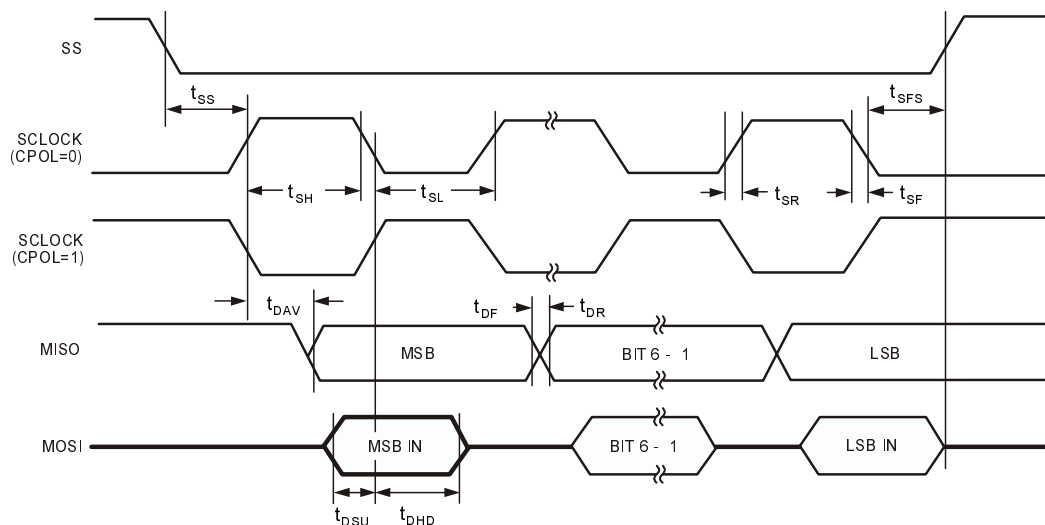


Figure 78. SPI Slave Mode Timing (CPHA = 1)

ADuC831

Parameter		Min	Typ	Max	Unit	Figure
SPI SLAVE MODE TIMING (CPHA = 0)						
t_{SS}	SS to SCLOCK Edge	0			ns	79
t_{SL}	SCLOCK Low Pulsewidth		330		ns	79
t_{SH}	SCLOCK High Pulsewidth		330		ns	79
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns	79
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns	79
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns	79
t_{DF}	Data Output Fall Time		10	25	ns	79
t_{DR}	Data Output Rise Time		10	25	ns	79
t_{SR}	SCLOCK Rise Time		10	25	ns	79
t_{SF}	SCLOCK Fall Time		10	25	ns	79
t_{DOSS}	Data Output Valid after SS Edge			20	ns	79
t_{SFS}	SS High after SCLOCK Edge	0			ns	79

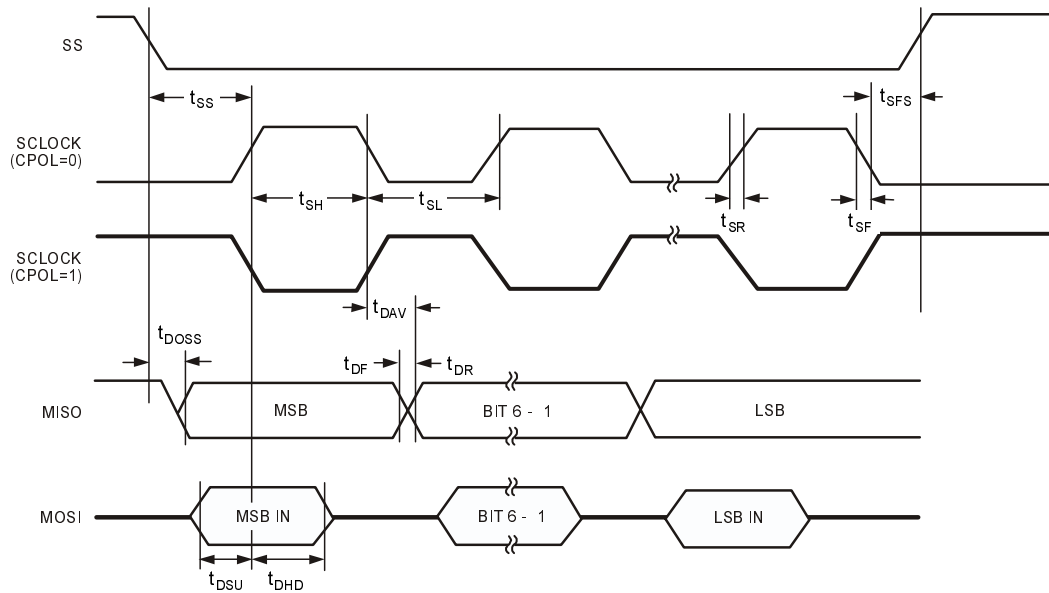
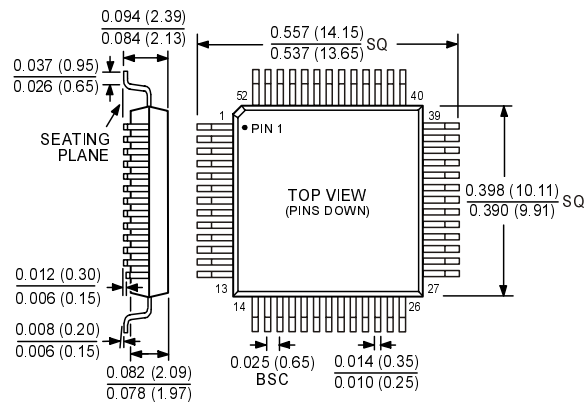


Figure 79. SPI Slave Mode Timing (CPHA = 0)

OUTLINE DIMENSIONS
Dimensions shown in inches and (mm).

52-Lead Plastic Quad Flatpack
(S-52)



56Lead Chip Scale Package(CP-56)

