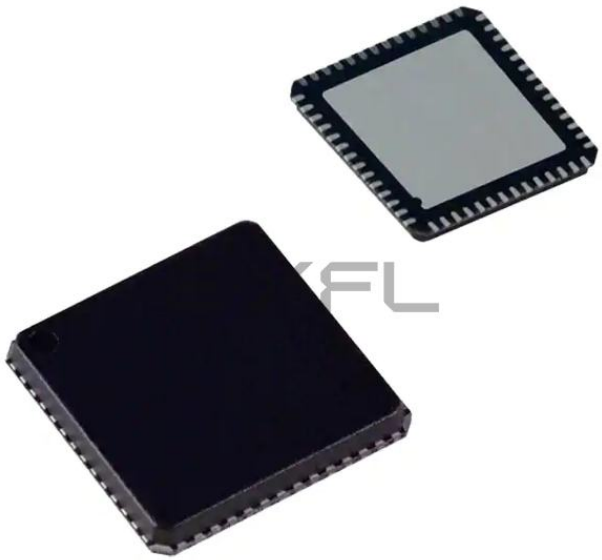

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	8052
Core Size	8-Bit
Speed	16MHz
Connectivity	EBI/EMI, I²C, SPI, UART/USART
Peripherals	PSM, Temp Sensor, WDT
Number of I/O	34
Program Memory Size	62KB (62K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	56-VFQFN Exposed Pad, CSP
Supplier Device Package	56-LFCSP-VQ (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc832bcpz-reel

Table 4. External Data Memory Read Cycle

Parameter[1]	Description	16.78 MHz Core_CLK		Variable Clock		Unit
		Min	Max	Min	Max	
t_{RLRH}	$\overline{RD}$ pulse width	257		$6t_{CK} - 100$		ns
t_{AVLL}	Address valid before ALE low	19		$t_{CK} - 40$		ns
t_{LLAX}	Address hold after ALE low	24		$t_{CK} - 35$		ns
t_{RLDV}	$\overline{RD}$ low to valid data in		133		$5t_{CK} - 165$	ns
t_{RHDX}	Data and address hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data float after $\overline{RD}$		49		$2t_{CK} - 70$	ns
t_{LLDV}	ALE low to valid data in		326		$8t_{CK} - 150$	ns
t_{AVDV}	Address to valid data in		371		$9t_{CK} - 165$	ns
t_{LLWL}	ALE low to $\overline{RD}$ low	128	228	$3t_{CK} - 50$	$3t_{CK} + 50$	ns
t_{AVWL}	Address valid to $\overline{RD}$ low	108		$4t_{CK} - 130$		ns
t_{RLAZ}	$\overline{RD}$ low to address float	0		0		ns
t_{WHLH}	$\overline{RD}$ high to ALE high	19	257	$t_{CK} - 40$	$6t_{CK} - 100$	ns

[1] See Figure 6.

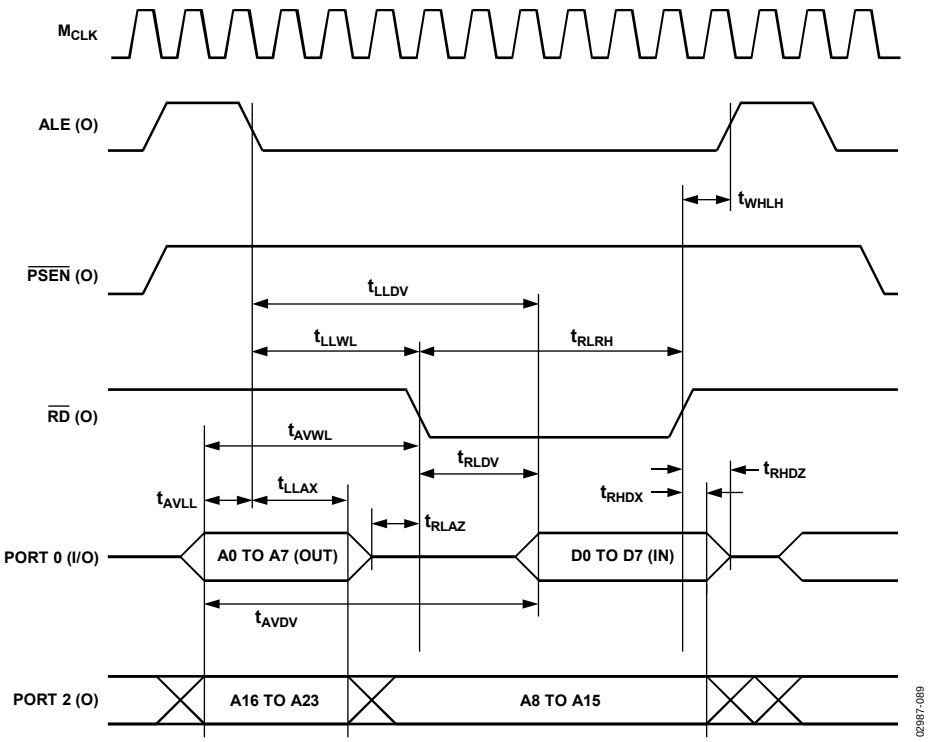

Figure 6. External Data Memory Read Cycle

Table 5. External Data Memory Write Cycle

Parameter[1]	Description	16.78 MHz Core_CLK Min	16.78 MHz Core_CLK Max	Variable Clock Min	Variable Clock Max	Unit
t_{WLWH}	$\overline{WR}$ pulse width	257		$6t_{CK} - 100$		ns
t_{AVLL}	Address valid before ALE low	19		$t_{CK} - 40$		ns
t_{LLAX}	Address hold after ALE low	24		$t_{CK} - 35$		ns
t_{LLWL}	ALE low to $\overline{WR}$ low	128	228	$3t_{CK} - 50$	$3t_{CK} + 50$	ns
t_{AVWL}	Address valid to $\overline{WR}$ Low	108		$4t_{CK} - 130$		ns
t_{QVWX}	Data valid to $\overline{WR}$ transition	9		$t_{CK} - 50$		ns
t_{QVWH}	Data setup before $\overline{WR}$	267		$7t_{CK} - 150$		ns
t_{WHQX}	Data and address hold after $\overline{WR}$	9		$t_{CK} - 50$		ns
t_{WHLH}	$\overline{WR}$ high to ALE high	19	257	$t_{CK} - 40$	$6t_{CK} - 100$	ns

[1] See Figure 7.

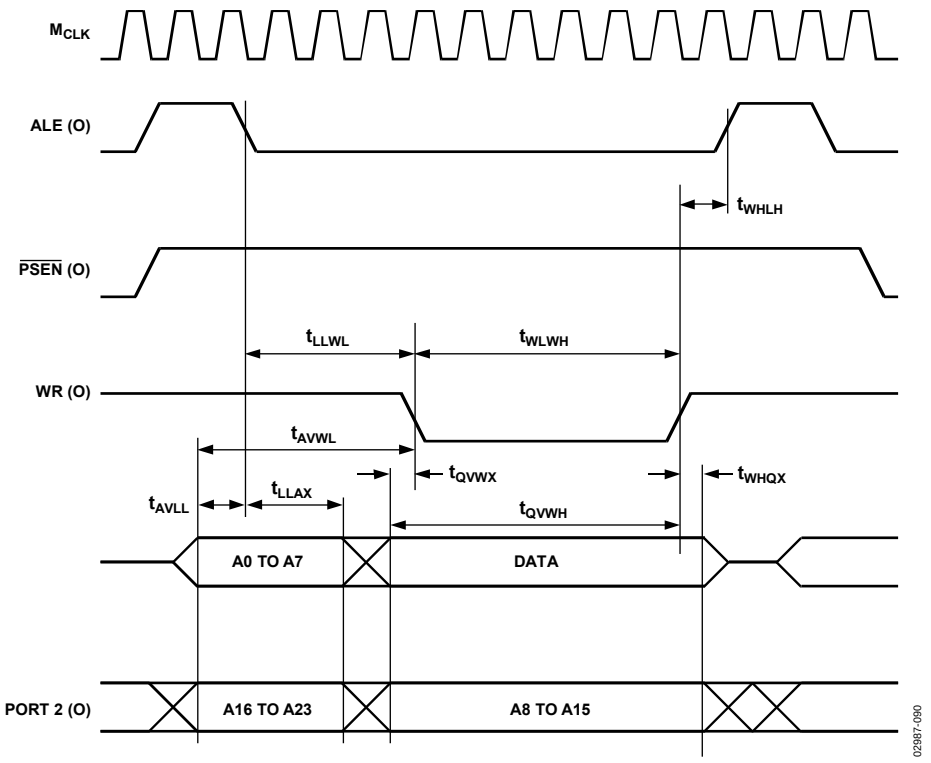

Figure 7. External Data Memory Write Cycle

Table 8. SPI Master Mode Timing (CPHA = 1)

Parameter[1]	Description	Min	Typ	Max	Unit
t_{SL}	SCLOCK low pulse width[2]		476		ns
t_{SH}	SCLOCK high pulse width[2]		476		ns
t_{DAV}	Data output valid after SCLOCK edge			50	ns
t_{DSU}	Data input setup time before SCLOCK edge	100			ns
t_{DHD}	Data input hold time after SCLOCK edge	100			ns
t_{DF}	Data output fall time		10	25	ns
t_{DR}	Data output rise time		10	25	ns
t_{SR}	SCLOCK rise time		10	25	ns
t_{SF}	SCLOCK fall time		10	25	ns

[1] See Figure 10.
[2] Characterized under the following conditions:
a. Core clock divider bits (CD2, CD1, and CD0 bits in PLLCON SFR) set to 0, 1, and 1, respectively, that is, core clock frequency = 2.09 MHz
b. SPI bit rate selection bits (SPR1 and SPR0 bits in SPICON SFR) set to 0 and 0, respectively.

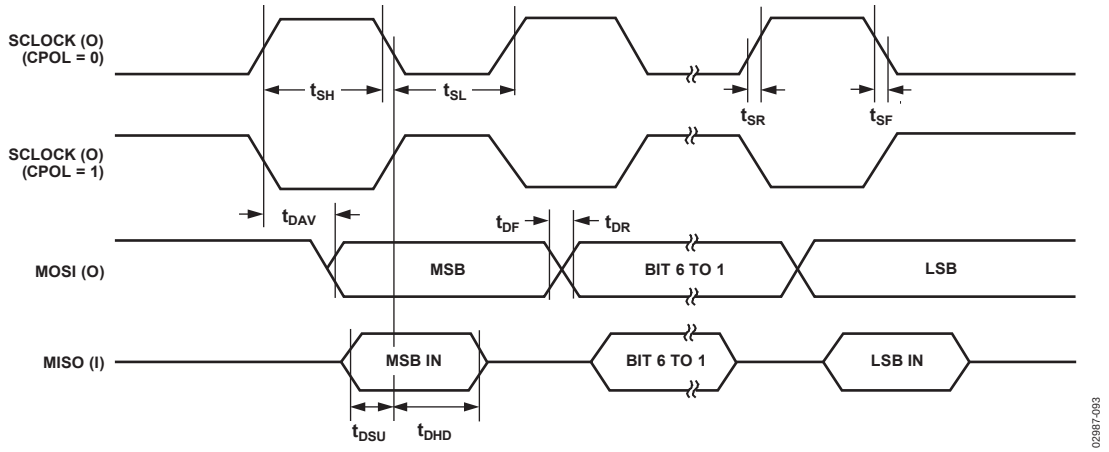

Figure 10. SPI Master Mode Timing (CPHA = 1)

ABSOLUTE MAXIMUM RATINGS

T_A = 25°C, unless otherwise noted.

Table 12.

Parameter	Rating
AV_{DD} to DV_{DD}	−0.3 V to +0.3 V
AGND to DGND	−0.3 V to +0.3 V
DV_{DD} to DGND, AV_{DD} to AGND	−0.3 V to +7 V
Digital Input Voltage to DGND	−0.3 V to DV_{DD} + 0.3 V
Digital Output Voltage to DGND	−0.3 V to DV_{DD} + 0.3 V
V_{REF} to AGND	−0.3 V to AV_{DD} + 0.3 V
Analog Inputs to AGND	−0.3 V to AV_{DD} + 0.3 V
Operating Temperature Range Industrial	
ADuC832BSZ	−40°C to +125°C
Operating Temperature Range Industrial	
ADuC832BCPZ	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Junction Temperature	150°C
θ_{JA} Thermal Impedance (ADuC832BSZ)	90°C/W
θ_{JA} Thermal Impedance (ADuC832BCPZ)	52°C/W
Lead Temperature, Soldering	
Vapor Phase (60 sec)	215°C
Infrared (15 sec)	220°C

Stresses at or above those listed under Absolute Maximum Ratings may cause permanent damage to the product. This is a stress rating only; functional operation of the product at these or any other conditions above those indicated in the operational section of this specification is not implied. Operation beyond the maximum operating conditions for extended periods may affect product reliability.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

Mnemonic	Pin No. MQFP	Pin No. LFCSP	Type	Description
P0.2/AD2	45	48	I/O	Input/Output Port 0 (P0.2). Port 0 is an 8-Bit Open-Drain Bidirectional I/O Port. Port 0 pins that have 1s written to them float and in that state can be used as high impedance inputs.
				External Memory Address and Data (AD2). 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.
P0.3/AD3	46	49	I/O	Input/Output Port 0 (P0.3). Port 0 is an 8-Bit Open-Drain Bidirectional I/O Port. Port 0 pins that have 1s written to them float and in that state can be used as high impedance inputs.
			I/O	External Memory Address and Data (AD3). 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.
P0.4/AD4	49	52	I/O	Input/Output Port 0 (P0.4). Port 0 is an 8-Bit Open-Drain Bidirectional I/O Port. Port 0 pins that have 1s written to them float and in that state can be used as high impedance inputs.
			I/O	External Memory Address and Data (AD4). 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.
P0.5/AD5	50	53	I/O	Input/Output Port 0 (P0.5). Port 0 is an 8-Bit Open-Drain Bidirectional I/O Port. Port 0 pins that have 1s written to them float and in that state can be used as high impedance inputs.
			I/O	External Memory Address and Data (AD5). 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.
P0.6/AD6	51	54	I/O	Input/Output Port 0 (P0.6). Port 0 is an 8-Bit Open-Drain Bidirectional I/O Port. Port 0 pins that have 1s written to them float and in that state can be used as high impedance inputs.
			I/O	External Memory Address and Data (AD6). 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.
P0.7/AD7	52	56	I/O	Input/Output Port 0 (P0.7). Port 0 is an 8-Bit Open-Drain Bidirectional I/O Port. Port 0 pins that have 1s written to them float and in that state can be used as high impedance inputs.
				External Memory Address and Data (AD7). 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.

[1] N/A means not applicable.

MEMORY ORGANIZATION

The ADuC832 contains four different memory blocks:

- 62 kB of on-chip Flash/EE program memory
- 4 kB of on-chip Flash/EE data memory
- 256 bytes of general-purpose RAM
- 2 kB of internal XRAM

FLASH/EE PROGRAM MEMORY

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

If the user applies power or resets the device while the $\overline{EA}$ pin is pulled low, the part executes code from the external program space; otherwise, the part defaults to code execution from its internal 62 kB of Flash/EE program memory. Unlike the ADuC812, where code execution can overflow from the internal code space to external code space once the PC becomes greater than 1FFFH, the ADuC832 does not support the rollover from F7FFH in internal code space to F800H in external code space. Instead, the 2048 bytes between F800H and FFFFH 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. During runtime, 56 kB of the program memory can be reprogrammed; thus the code space can be upgraded in the field using a user defined protocol, or it can be used as a data memory (for more details, see the Using the Flash/EE Program Memory section).

FLASH/EE DATA MEMORY

4 kB 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 in the Using the Flash/EE Data Memory section.

GENERAL-PURPOSE RAM

The general-purpose RAM is divided into two separate memories, the upper and the lower 128 bytes of RAM. The lower 128 bytes of RAM can be accessed through direct or indirect addressing. The upper 128 bytes of RAM can only be accessed through indirect addressing because 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 32. The lowest 32 bytes are grouped into four banks of eight registers addressed as R0 through R7. The next 16 bytes (128 bits), above the register banks, form a block of bit addressable memory space at Address 20H through Address 2FH. The stack can be located anywhere in the internal memory address space, and the stack depth can be expanded up to 2048 bytes.

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

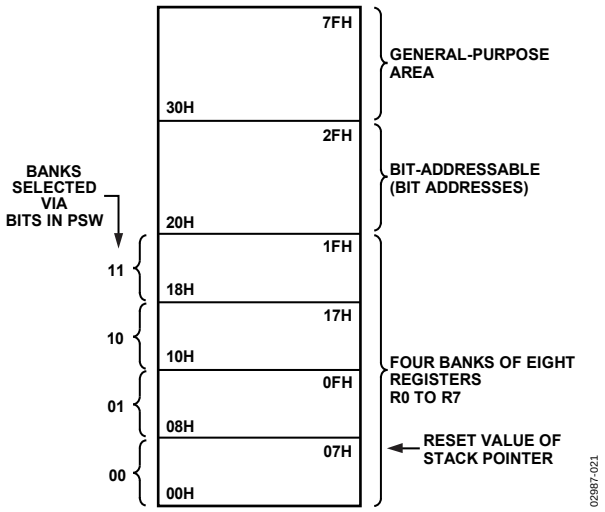

Figure 32. Lower 128 Bytes of Internal Data Memory

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

By default, the stack operates exactly like an 8052 in that it rolls over from FFH to 00H in the general-purpose RAM. On the ADuC832, however, it is possible (by setting CFG832[7]) to enable the 11-bit extended stack pointer. In this case, the stack rolls over from 00FFH 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 three LSBs of this SFR contain the three extra bits necessary to extend the 8-bit stack pointer into an 11-bit stack pointer.

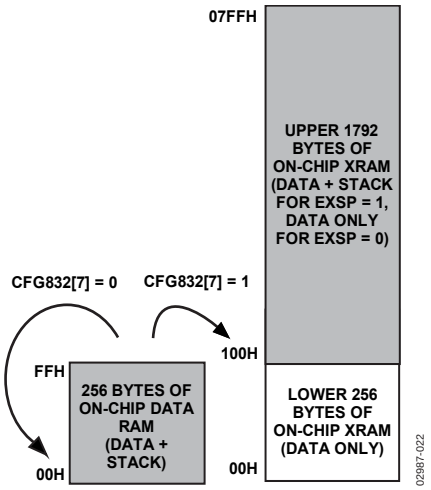

Figure 33. Extended Stack Pointer Operation

ADCCON3 (ADC Control SFR 3)

SFR Address: F5H

SFR Power-On Default Value: 00H

Bit Addressable: No

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

Table 18. ADCCON3 SFR Bit Designations

Bit	Name	Description		
[7]	Busy	The ADC busy status bit 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.		
[6]	GNCLD	Gain calibration disable bit. Set to 0 to enable gain calibration. Set to 1 to disable gain calibration.		
[5:4]	AVGS[1:0]	Number of averages selection bits. These bits select the number of ADC readings averaged during a calibration cycle.		
		AVGS1	**AVGS0**	**Number of Averages**
		0	0	15
		0	1	1
		1	0	31
		1	1	63
[3]	RSVD	Reserved. This bit should always be written as 0.		
[2]	RSVD	This bit should always be written as 1 by the user when performing calibration.		
[1]	Typical	Calibration type select bit. This bit selects between offset (zero-scale) and gain (full-scale) calibration. Set to 0 for offset calibration. Set to 1 for gain calibration.		
[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.		

Table 20. 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
ADA4610-1, OP249	High gain-bandwidth product (GBP)
OP162/OP262/OP462	High GBP, micro package
AD820/AD822/AD824	FET input, low cost
AD823	FET input, high GBP

Keep in mind that the ADC's transfer function is 0 V to V_{REF}, and any signal range lost to amplifier saturation near ground impacts dynamic range. Though the op amps in Table 20 are capable of delivering output signals very closely approaching ground, no amplifier can deliver signals all the way to ground when powered by a single supply. Therefore, if a negative supply is available, consider using it to power the front-end amplifiers. However, be sure to include the Schottky diodes shown in Figure 40 (or at least the lower of the two diodes) to protect the analog input from undervoltage conditions. In summary, use the circuit of Figure 40 to drive the analog input ADCx pins of the ADuC832.

VOLTAGE REFERENCE CONNECTIONS

The on-chip 2.5 V band gap voltage reference can be used as the reference source for the ADC and DACs. To ensure the accuracy of the voltage reference, the user must decouple the V_{REF} pin to ground with a 0.1 µF capacitor, and the C_{REF} pin to ground with a 0.1 µF capacitor, as shown in Figure 41.

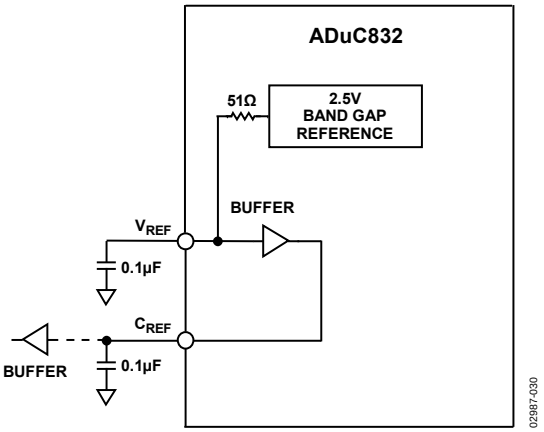

Figure 41. Decoupling V_{REF} and C_{REF}

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} is the recommended output for the external reference; this can be used but note that there is a gain error between this reference and that of the ADC.

The ADuC832 powers up with its internal voltage reference in the on state. This is available at the V_{REF} pin, but as noted previously, there is 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 42. Bit 6 of the ADCCON1 SFR must be set to 1 to switch in the external reference voltage.

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

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

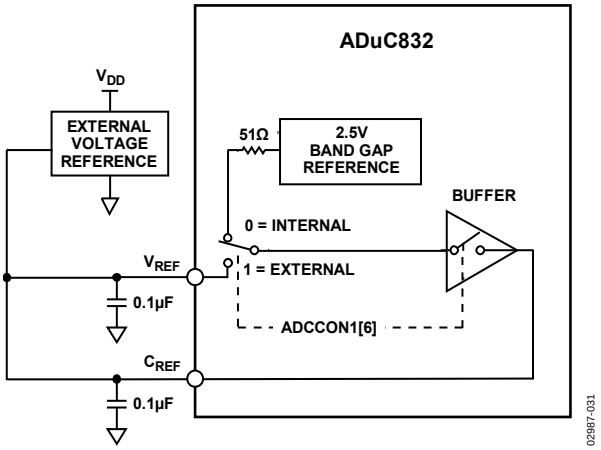

Figure 42. Using an External Voltage Reference

To maintain compatibility with the ADuC812, the external reference can also be connected to the V_{REF} pin, as shown in Figure 43, to overdrive the internal reference. Note that this introduces a gain error for the ADC that has to be calibrated out; thus the previous method is the recommended one for most users. For this method to work, ADCCON1[6] should be configured to use the internal reference. The external reference then overdrives this.

CALIBRATING THE ADC

There are two hardware calibration modes provided that can be easily initiated by user software. The ADCCON3 SFR is used to calibrate the ADC. The typical bit (ADCCON3[1]) and the CS3 to CS0 bits (ADCCON2[3:0]) set up the calibration modes.

Device calibration can be initiated to compensate for significant changes in operating conditions frequency, analog input range, reference voltage, and supply voltages. In this calibration mode, offset calibration uses the internal AGND selected via ADCCON2 register bits CS[3:0] = 1011, and gain calibration uses the internal V_{REF} selected by CS[3:0] = 1100. Offset calibration should be executed first, followed by gain calibration.

System calibration can be initiated to compensate for both internal and external system errors. To perform system calibration using an external reference, tie system ground and reference to any two of the six selectable inputs. Enable external reference mode (ADCCON1[6]). Select the channel connected to AGND via CS[3:0] and perform system offset calibration. Select the channel connected to V_{REF} via CS[3:0] and perform system gain calibration.

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

USING THE DAC

The on-chip DAC architecture consists of a resistor string DAC followed by an output buffer amplifier, the functional equivalent of which is illustrated in Figure 51. Details of the actual DAC architecture can be found in U.S. Patent Number 5,969,657. Features of this architecture include inherent guaranteed monotonicity and excellent differential linearity.

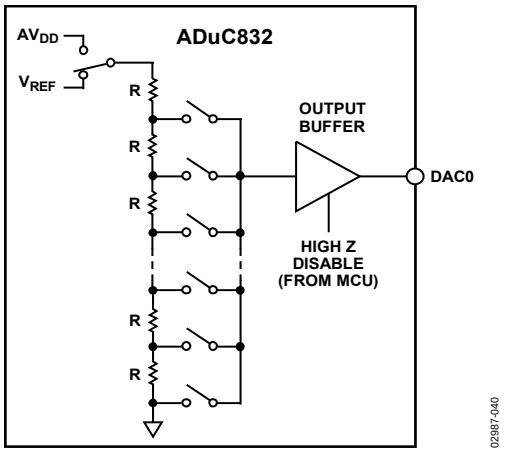

Figure 51. Resistor String DAC Functional Equivalent

As illustrated in Figure 51, the reference source for each DAC is user selectable in software. It can be either AV_{DD} or V_{REF}. In 0 V to AV_{DD} mode, the DAC output transfer function spans from 0 V to the voltage at the AV_{DD} pin. In 0 V to V_{REF} mode, the DAC output transfer function spans from 0 V to the internal V_{REF} or, if an external reference is applied, the voltage at the V_{REF} pin. The DAC output buffer amplifier features a true rail-to-rail output stage implementation. This means that, unloaded, each output is capable of swinging to within less than 100 mV of both AV_{DD} and ground. Moreover, the DAC's linearity specification (when driving a 10 kΩ resistive load to ground) is guaranteed through the full transfer function except Code 0 to Code 100, and, in 0 V to AV_{DD} mode only, Code 3995 to Code 4095. Linearity degradation near ground and AV_{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 52. The dotted line in Figure 52 indicates the ideal transfer function, and the solid line represents what the transfer function may look like with endpoint nonlinearities due to saturation of the output amplifier. Note that Figure 52 represents a transfer function in 0 V to AV_{DD} mode only. In 0 V to V_{REF} mode (with $V_{REF} < AV_{DD}$), the lower nonlinearity is similar, but the upper portion of the transfer function follows the ideal line to the end (V_{REF} in this case, not AV_{DD}), showing no signs of endpoint linearity errors.

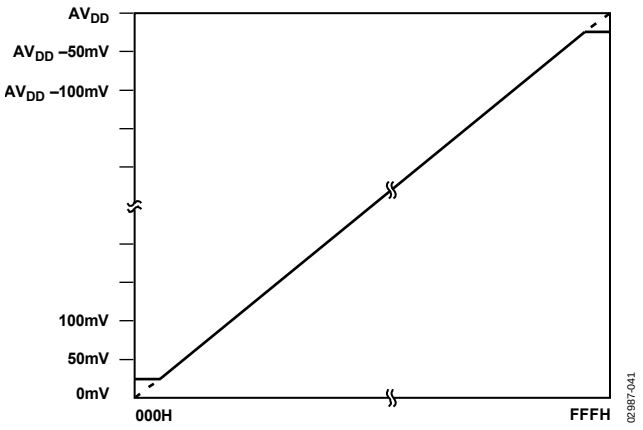

Figure 52. Endpoint Nonlinearities Due to Amplifier Saturation

The endpoint nonlinearities conceptually illustrated in Figure 52 become worse as a function of output loading. Most of the ADuC832 specifications assume a 10 kΩ resistive load to ground at the DAC output. As the output is forced to source or sink more current, the nonlinear regions at the top or bottom (respectively) of Figure 52 become larger. With larger current demands, this can significantly limit output voltage swing. Figure 53 and Figure 54 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 V to AV_{DD}. In 0 V to V_{REF} mode, DAC loading does 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 is not affected by loads less than 5 mA. However, around 7 mA, the upper curve in Figure 54 drops below 2.5 V (V_{REF}), indicating that, at these higher currents, the output is not capable of reaching V_{REF}.

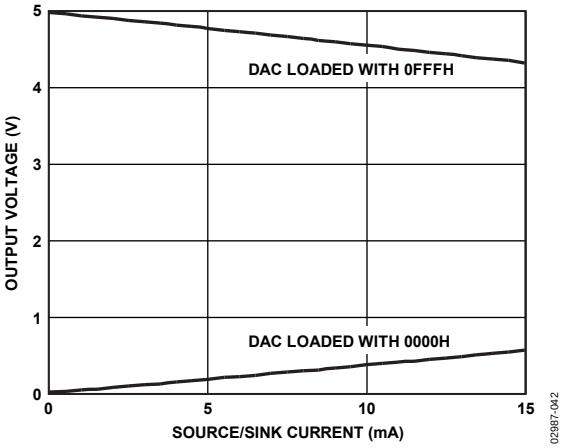

Figure 53. Source and Sink Current Capability with $V_{REF} = AV_{DD} = 5$ V

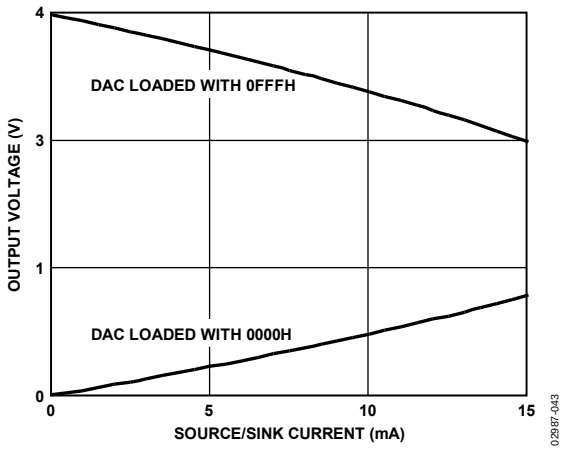

Figure 54. Source and Sink Current Capability with $V_{REF} = AV_{DD} = 3$ V

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

To drive significant loads with the DAC outputs, external buffering may be required (even with the internal buffer enabled), as illustrated in Figure 55. A list of recommended op amps is shown in Table 20.

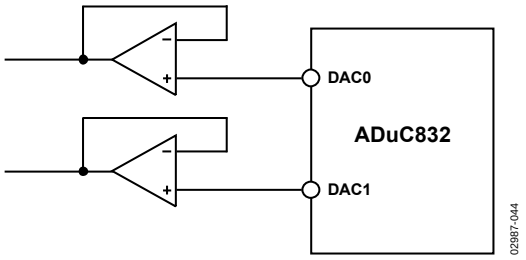

Figure 55. Buffering the DAC Outputs

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 remain at ground potential whenever the DAC is disabled.

PWM MODES OF OPERATION

MODE 0: PWM DISABLED

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

MODE 1: SINGLE VARIABLE RESOLUTION PWM

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

PWM1H/PWM1L sets the period of the output waveform. Reducing PWM1H/PWM1L reduces the resolution of the PWM output but increases the maximum output rate of the PWM. (for example, setting PWM1H/PWM1L to 65,536 gives a 16-bit PWM with a maximum output rate of 266 Hz (16.78 MHz/65,536). Setting PWM1H/PWM1L to 4096 gives a 12-bit PWM with a maximum output rate of 4096 Hz (16.78 MHz/4096).

PWM0H/PWM0L sets the duty cycle of the PWM output waveform, as shown in Figure 57.

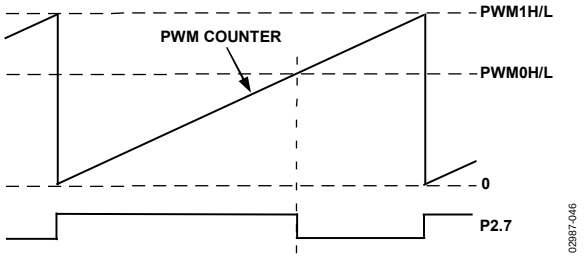

Figure 57. PWM Mode 1

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 is set to 255 (FFH) to give an 8-bit PWM although it is possible to reduce this as necessary. A value of 100 can be loaded here to give a percentage PWM (that is, the PWM is accurate to 1%).

The outputs of the PWM at P2.6 and P2.7 are shown in Figure 58.

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.

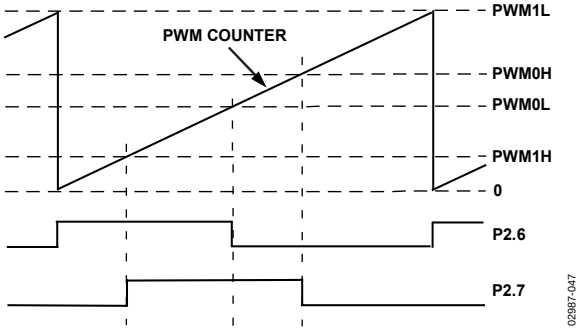

Figure 58. PWM Mode 2

MODE 3: TWIN 16-BIT PWM

In Mode 3, the PWM counter is fixed to count from 0 to 65,536, giving a fixed 16-bit PWM. Operating from the 16.78 MHz core clock results in a PWM output rate of 256 Hz. The duty cycle of the PWM outputs at P2.6 and P2.7 is independently programmable.

As shown in Figure 59, while the PWM counter is less than PWM0H/PWM0L, the output of PWM0 (P2.6) is high. Once the PWM counter equals PWM0H/PWM0L, PWM0 (P2.6) goes low and remains low until the PWM counter rolls over.

Similarly, while the PWM counter is less than PWM1H/PWM1L, the output of PWM1 (P2.7) is high. Once the PWM counter equals PWM1H/PWM1L, PWM1 (P2.7) goes low and remains low until the PWM counter rolls over.

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

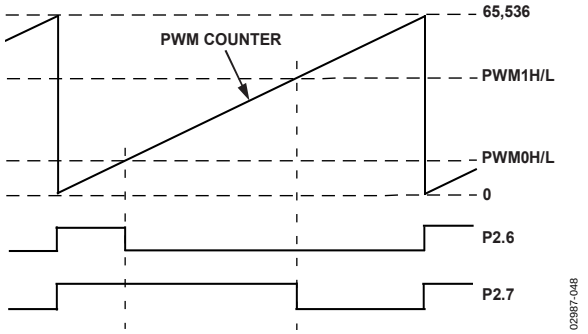

Figure 59. PWM Mode 3

I²C-COMPATIBLE INTERFACE

The ADuC832 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 interface or the other at any given time (see SPE in SPICON in Table 28). The uC001 Technical Note, *MicroConverter® I²C® Compatible Interface*, describes the operation of this interface as implemented.

I²C INTERFACE SFRs

Three SFRs are used to control the I²C interface. These are described in the following sections.

I2CCON (I²Control Register)

SFR Address: E8H

Power-On Default Value: 00H

Bit Addressable: Yes

I2CADD (I²C Address Register)

SFR Address: 9BH

Power-On Default Value: 55H

Bit Addressable: No

The I2CADD SFR holds the I²C peripheral address for the part. It can be overwritten by user code. Technical Note uC001 describes the format of the I²C standard 7-bit address in detail.

I2CDAT (I²C Data Register)

SFR Address: 9AH

Power-On Default Value: 00H

Bit Addressable: No

The I2CDAT SFR is written by the user to transmit data over the I²C interface or read by user code to read data just received by the I²C interface. Accessing I2CDAT automatically clears any pending I²C interrupt and the I2CI bit in the I2CCON SFR. User software should only access I2CDAT once per interrupt cycle.

Table 29. I2CCON SFR Bit Designations

Bit	Name	Description
[7]	MDO	I²C software master data output bit (master mode only). This data bit is used to implement a master I²C transmitter interface in software. Data written to this bit is output on the SDATA pin if the data output enable (MDE) bit is set.
[6]	MDE	I²C software master data output enable bit (master mode only). Set by user to enable the SDATA pin as an output (Tx). Cleared by the user to enable SDATA pin as an input (Rx).
[5]	MCO	I²C software master clock output bit (master mode only). This data bit is used to implement a master I²C transmitter interface in software. Data written to this bit is output on the SCLOCK pin.
[4]	MDI	I²C software master data input bit (master mode only). This data bit is used to implement a master I²C receiver interface in software. Data on the SDATA pin is latched into this bit on SCLOCK if the data output enable (MDE) bit is 0.
[3]	I2CM	I²C master/slave mode bit set by user to enable I²C software master mode. Cleared by user to enable I²C hardware slave mode.
[2]	I2CRS	I²C reset bit (slave mode only). Set by user to reset the I²C interface. Cleared by user code for normal I²C operation.
[1]	I2CTX	I²C direction transfer bit (slave mode only). Set by the MicroConverter if the interface is transmitting. Cleared by the MicroConverter if the interface is receiving.
[0]	I2CI	I²C interrupt bit (slave mode only). Set by the MicroConverter after a byte has been transmitted or received. Cleared automatically when user code reads the I2CDAT SFR (see the I2CADD (I2C Address Register) section).

8052-COMPATIBLE ON-CHIP PERIPHERALS

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

PARALLEL I/O

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

PORT 0

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

Figure 66 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 clocks 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 Read-Modify-Write Instructions section for more details.

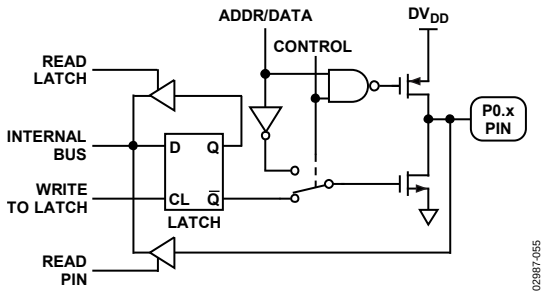

Figure 66. Port 0 Bit Latch and I/O Buffer

As shown in Figure 66, 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, 1s are written to the P0 SFR (that is, all of its bit latches become 1). When accessing external memory, the control signal in Figure 66 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 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 are configured as open drain and therefore float. In this state, Port 0 pins can be used as high impedance inputs. This is represented in Figure 66 by the NAND gate whose output remains high as long as the control signal is low, thereby disabling the top FET. External pull-up resistors are therefore required when Port 0 pins are used as general-purpose outputs. Port 0 pins with 0s written to them drive a logic low output voltage (V_{OL}) and are capable of sinking 1.6 mA.

PORT 1

Port 1 is also an 8-bit port directly controlled via the P1 SFR.

Port 1 digital output capability is not supported on this device.

Port 1 pins can be configured as digital inputs or analog inputs.

By (power-on) default, these pins are configured as analog inputs, that is, 1 written in the corresponding Port 1 register bit. To configure any of these pins as digital inputs, 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 34.

Table 34. Port 1, Alternate Pin Functions

Pin	Alternate Function
P1.0	T2 (Timer/Counter 2 external input) or ADC0 (single-ended analog input)
P1.1	T2EX (Timer/Counter 2 capture/reload trigger) or ADC1
P1.5	$\overline{SS}$ (slave select for the SPI interface) or ADC5

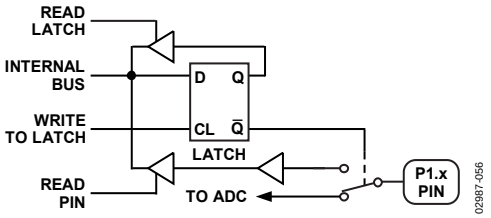

Figure 67. Port 1 Bit Latch and I/O Buffer

PORT 2

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

As shown in Figure 68, the output drivers of Port 2 are switchable to an internal ADDR and ADDR/data bus by an internal control signal for use in external memory accesses (as for Port 0). In external memory addressing mode (control = 1), the port pins feature push-pull operation controlled by the internal address bus (ADDR line). However, unlike the P0 SFR during external memory accesses, the P2 SFR remains unchanged.

In general-purpose I/O port mode, Port 2 pins that have 1s written to them are pulled high by the internal pull-ups (see Figure 69) and, in that state, can be used as inputs. As inputs, Port 2 pins pulled externally low source current because of the internal pull-up resistors. Port 2 pins with 0s written to

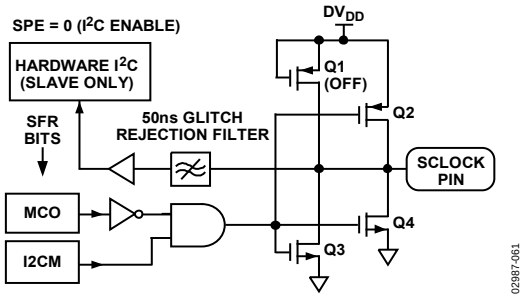

Figure 72. SCLOCK Pin I/O Functional Equivalent in I²C Mode

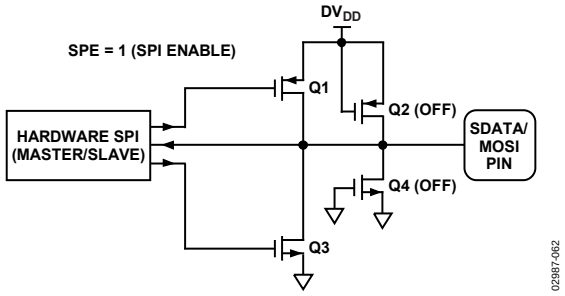

Figure 73. SDATA/MOSI Pin I/O Functional Equivalent in SPI Mode

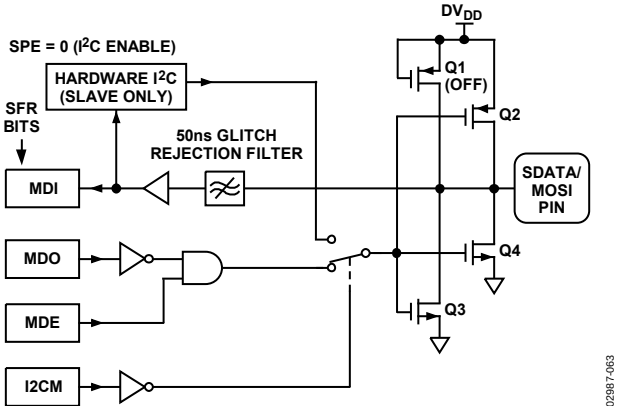

Figure 74. SDATA/MOSI Pin I/O Functional Equivalent in I²C Mode

MISO is shared with P3.3 and as such has the same configuration as that shown in Figure 70.

READ-MODIFY-WRITE INSTRUCTIONS

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

Instruction	Description
ANL	Logical AND, for example, ANL P1, A
ORL	Logical OR, for example, ORL P2, A
XRL	Logical EX-OR, for example., XRL P3, A
JBC	Jump if bit = 1 and clear bit, for example, JBC P1.1, LABEL
CPL	Complement bit, for example, CPL P3.0
INC	Increment, for example, INC P2
DEC	Decrement, for example, DEC P2
DJNZ	Decrement and jump if not zero, for example, DJNZ P3, LABEL
MOV PX.Y, C[1]	Move carry to Bit Y of Port X
CLR PX.Y[1]	Clear Bit Y of Port X
SETB PX.Y[1]	Set Bit Y of Port X

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

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.

TCON (Timer/Counter 0 and Timer/Counter 1 Control Register)

SFR Address: 88H

Power-On Default Value: 00H

Bit Addressable: Yes

TIMER/COUNTER 0 AND TIMER/COUNTER 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

TH0 is the Timer 0 high byte and TL0 is the low byte. The SFR addresses for TH0 and TL0 are 8CH and 8AH, respectively.

TH1 and TL1

TH1 is the Timer 1 high byte and TH0 is the low byte. The SFR addresses for TH1 and TL1 are 8DH and 8BH, respectively.

Table 37. TCON SFR Bit Designations

Bit	Name	Description
[7]	TF1	Timer 1 overflow flag. Set by hardware on a Timer/Counter 1 overflow. Cleared by hardware when the program counter (PC) vectors to the interrupt service routine.
[6]	TR1	Timer 1 run control bit. Set by the user to turn on Timer/Counter 1. Cleared by the user to turn off Timer/Counter 1.
[5]	TF0	Timer 0 overflow flag. Set by hardware on a Timer/Counter 0 overflow. Cleared by hardware when the PC vectors to the interrupt service routine.
[4]	TR0	Timer 0 run control bit. Set by the user to turn on Timer/Counter 0. Cleared by the user to turn off Timer/Counter 0.
[3]	IE1[1]	External Interrupt 1 ($\overline{INT1}$) flag. Set by hardware by a falling edge or zero level being applied to external interrupt Pin $\overline{INT1}$, depending on the state of Bit IT1. 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 (that is, a 1-to-0 transition). Cleared by software to specify level-sensitive detection (that is, zero level).
[1]	IE0[1]	External Interrupt 0 ($\overline{INT0}$) flag. Set by hardware by a falling edge or zero level being applied to external interrupt Pin $\overline{INT0}$, depending on the state of Bit IT0. 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 (that is, 1-to-0 transition). Cleared by software to specify level-sensitive detection (that is, zero level).

[1] These bits are not used in the control of Timer/Counter 0 and Timer/Counter 1, but are used instead in the control and monitoring of the external INT0 and INT1 interrupt pins.

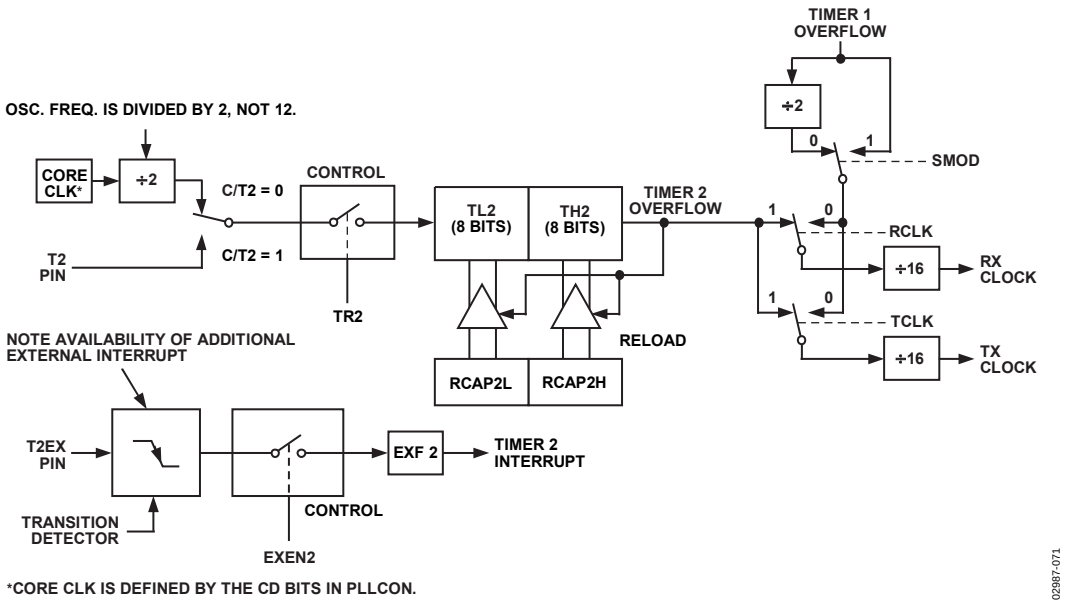

Figure 83. Timer 2, UART Baud Rates

TIMER 3 GENERATED BAUD RATES

The high integer dividers in a UART block mean that high speed baud rates are not always possible using some particular crystals. For example, using a 12 MHz crystal, a baud rate of 115,200 is not possible. To address this problem, the ADuC832 has a dedicated baud rate timer (Timer 3) specifically for generating highly accurate baud rates.

Timer 3 can be used instead of Timer 1 or Timer 2 for generating very accurate high speed UART baud rates including 115,200 and 230,400. Timer 3 also allows a much wider range of baud rates to be obtained. Every desired bit rate from 12 bits/sec to 393,216 bits/sec can be generated to within an error of ±0.8%. Timer 3 also frees up the other three timers, allowing them to be used for different applications. A block diagram of Timer 3 is shown in Figure 84.

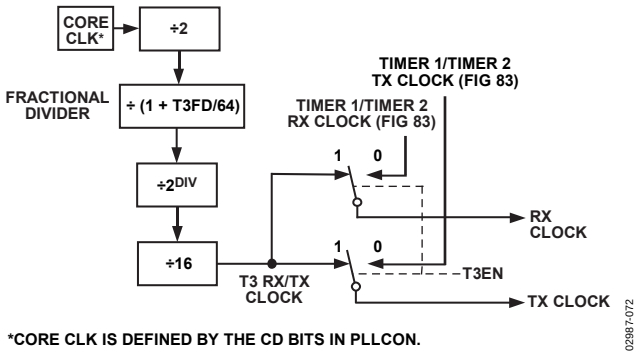

Figure 84. Timer 3, UART Baud Rates

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 43. T3CON SFR Bit Designations

Bit	Name	Description			
[7]	T3BAUDEN	T3 UART baud rate 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.			
[6:4]	Reserved				
[2:0]	DIV[2:0]	Binary divider factor			
		DIV2	**DIV1**	**DIV0**	**Binary Divider**
		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 DIV[2:0] bits can be calculated using the following formula

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

where f_{CORE} is defined in the PLLCON SFR, PLLCON[2:0].

Note that the DIV value must be rounded down.

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

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

Note that T3FD should be rounded to the nearest integer.

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

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

For example, to obtain a baud rate of 115,200 while operating at 16.78 MHz,

$DIV = \log(11{,}059{,}200/(32 \times 115{,}200))/\log(2) = 1.58 = 1$

$T3FD = (2 \times 11{,}059{,}200)/(21 \times 115{,}200) - 64 = 32 = 20H$

Therefore, the actual baud rate is 114,912 bits/sec.

Table 44. Commonly Used Baud Rates Using Timer 3

Ideal Baud	CD	DIV	T3CON	T3FD	% Error
230,400	0	1	81H	09H	0.25
115,200	0	2	82H	09H	0.25
115,200	1	1	81H	09H	0.25
115,200	2	0	80H	09H	0.25
57,600	0	3	83H	09H	0.25
57,600	1	2	82H	09H	0.25
57,600	2	1	81H	09H	0.25
57,600	3	0	80H	09H	0.25
38,400	0	3	83H	2DH	0.2
38,400	1	2	82H	2DH	0.2
38,400	2	1	81H	2DH	0.2
38,400	3	0	80H	2DH	0.2
19,200	0	4	84H	2DH	0.2
19,200	1	3	83H	2DH	0.2
19,200	2	2	82H	2DH	0.2
19,200	3	1	81H	2DH	0.2
19,200	4	0	80H	2DH	0.2
9600	0	5	85H	2DH	0.2
9600	1	4	84H	2DH	0.2
9600	2	3	83H	2DH	0.2
9600	3	2	82H	2DH	0.2
9600	4	1	81H	2DH	0.2
9600	5	0	80H	2DH	0.2

POWER CONSUMPTION

The currents consumed by the various sections of the ADuC832 are shown in Table 50. The core values given represent the current drawn by DV_{DD}, and the rest (ADC, DAC, voltage reference) are pulled by the AV_{DD} pin and can be disabled in software when not in use. The other on-chip peripherals (for example, watchdog timer and power supply monitor) consume negligible current and are therefore included with the core operating current. The user must add any currents sourced by the parallel and serial I/O pins and by the DAC to determine the total current needed at the ADuC832 supply pins. Also, current drawn from the DV_{DD} supply increases by approximately 10 mA during Flash/EE erase and program cycles.

Table 50. Typical I_{DD} of Core and Peripherals

Core/Peripherals	V_{DD} = 5 V	V_{DD} = 3 V
Core, Normal Mode	(1.6 nA × M_{CLK}) + 6 mA	(0.8 nA × M_{CLK}) + 3 mA
Core, Idle Mode	(0.75 nA × M_{CLK}) + 5 mA	(0.25 nA × M_{CLK}) + 3 mA
ADC	1.3 mA	1.0 mA
DAC (Each)	250 µA	200 µA
Voltage Reference	200 µA	150 µA

Because the operating DV_{DD} current is primarily a function of clock speed, the expressions for core supply current in Table 50 are given as functions of M_{CLK}, the core clock frequency. Use a value for M_{CLK} in hertz to determine the current consumed by the core at that oscillator frequency. Because the ADC and DACs can be enabled or disabled in software, add only the currents from the peripherals that are expected to be used. Do not forget to include current sourced by I/O pins, serial port pins, and DAC outputs, 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 into full power-down mode. The following sections provide brief descriptions of power-down and idle modes.

POWER SAVING MODES

In idle mode, the oscillator continues to run but the core clock generated from the PLL is halted. The on-chip peripherals continue to receive the clock, and remain functional. The CPU status is preserved with the stack pointer and program counter, and all other internal registers maintain their data during idle mode. Port pins and DAC output pins retain their states in this mode. The chip recovers from idle mode upon receiving any enabled interrupt, or upon receiving a hardware reset.

In full power-down mode, both the PLL and the clock to the core are stopped. The on-chip oscillator can be halted or can continue to oscillate depending on the state of the oscillator power-down bit (OSC_PD) in the PLLCON SFR. The TIC, being driven directly from the oscillator, can also be enabled during power-down. All other on-chip peripherals, however, are shut down. Port pins retain their logic levels in this mode, but the DAC output goes to a high impedance state (three-state).

During full power-down mode, the ADuC832 consumes a total of approximately 20 µA. There are five ways of terminating power-down mode.

Asserting the RESET Pin

Asserting the RESET pin returns the part to normal mode. All registers are set to their default state and program execution starts at the reset vector when the RESET pin is deasserted.

Cycling Power

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

Time Interval Counter (TIC) Interrupt

Power-down mode is terminated and the CPU services the TIC interrupt. The RETI at the end of the TIC ISR returns the core to the instruction following the one that enabled power-down.

I^2C or SPI Interrupt

Power-down mode is terminated and the CPU services the I^2C/SPI interrupt. The RETI at the end of the ISR returns the core to the instruction following the one that enabled power-down. It should be noted that the I^2C/SPI power-down interrupt enable bit (SERIPD) in the PCON SFR must first be set to allow this mode of operation.

$\overline{INT0}$ Interrupt

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

POWER-ON RESET

An internal power-on reset (POR) is implemented on the ADuC832. For DV_{DD} below 2.45 V, the internal POR holds the ADuC832 in reset. As DV_{DD} rises above 2.45 V, an internal timer times out for approximately 128 ms before the part is released from reset. The user must ensure that the power supply has reached a stable 2.7 V minimum level by this time. Likewise upon power-down, the internal POR holds the ADuC832 in reset until the power supply drops below 1 V. Figure 92 illustrates the operation of the internal POR in detail.

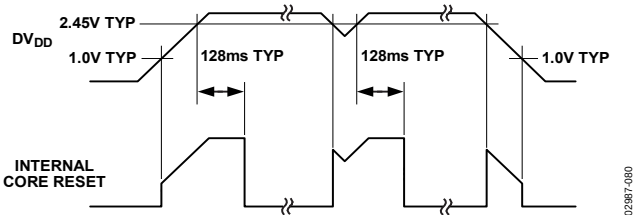

Figure 92. Internal POR Operation

NOTES