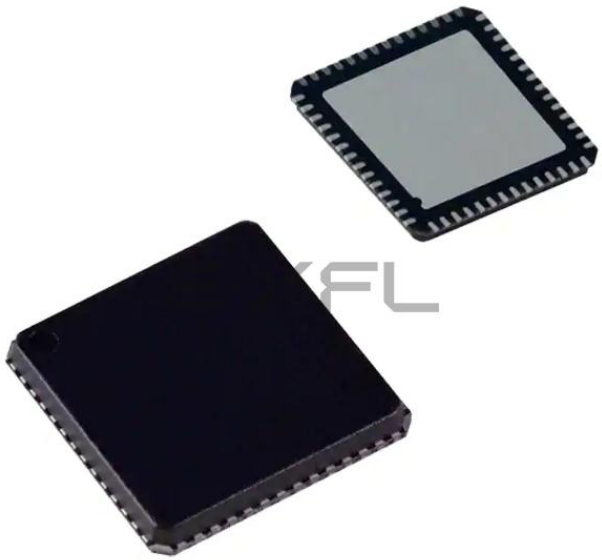

Welcome to E-XFL.COM

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

On-chip factory firmware supports in-circuit serial download and debug modes (via UART) as well as single-pin emulation mode via the $\overline{EA}$ pin. The ADuC832 is supported by QuickStart™ and QuickStart Plus development systems featuring low cost software and hardware development tools. A functional block diagram of the ADuC832 is shown in Figure 1 with a more detailed block diagram shown in Figure 2.

The part is specified for 3 V and 5 V operation over the extended industrial temperature range and is available in a 52-lead metric quad flat package (MQFP) and a 56-lead lead frame chip scale package (LFCSP).

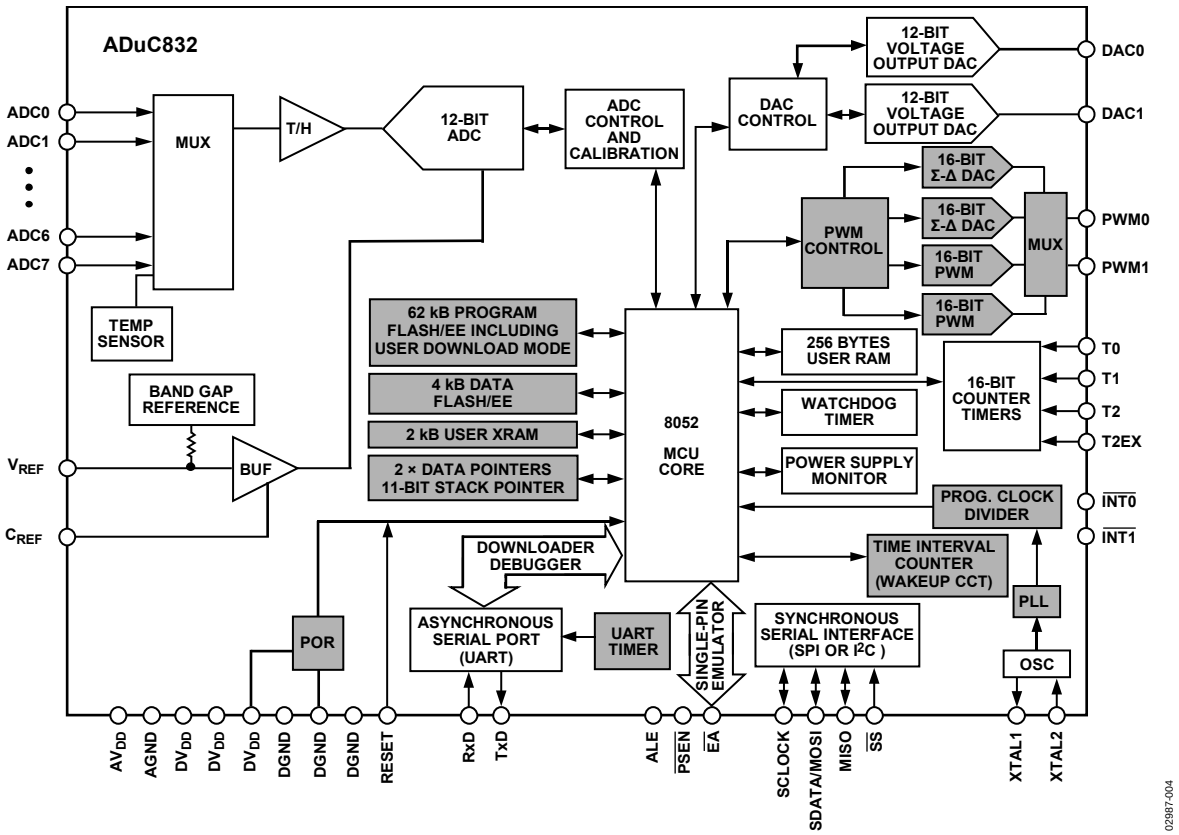

Figure 2. ADuC832 Block Diagram (Shaded Areas are Features Not Present on the ADuC812)

Parameter[1]	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
Power Supply Currents Power-Down Mode				Core_CLK = 2.097 MHz or 16.78 MHz
DV_{DD} Current[4]	80	25	µA max	Oscillator on
	38	14	µA typ	
AV_{DD} Current	2	1	µA typ	
DV_{DD} Current	35	20	µA max	Oscillator off
	25	12	µA typ	
Typical Additional Power Supply Currents				AV_{DD} = DV_{DD} = 5 V
PSM Peripheral	50		µA typ	
ADC	1.5		mA typ	
DAC	150		µA typ	

[1] Temperature range: −40°C to +125°C.
[2] ADC linearity is guaranteed during normal MicroConverter core operation.
[3] ADC LSB size = $V_{REF}/2^{12}$, that is, for internal V_{REF} = 2.5 V, 1 LSB = 610 µV and for external V_{REF} = 1 V, 1 LSB = 244 µV.
[4] Not production tested, but are guaranteed by design and/or characterization data on production release.
[5] Offset error, gain error, offset error match, and gain error match are measured after factory calibration.
[6] Based on external ADC system components, the user may need to execute a system calibration to remove additional external channel errors and achieve these specifications.
[7] SNR calculation includes distortion and noise components.
[8] Channel-to-channel crosstalk is measured on adjacent channels.
[9] The temperature sensor gives a measure of the die temperature directly; air temperature can be inferred from this result.
[10] DAC linearity is calculated using:
Reduced code range of 100 to 4095, 0 V to V_{REF} range.
Reduced code range of 100 to 3945, 0 V to V_{DD} range.
DAC output load = 10 kΩ and 100 pF.
[11] DAC differential nonlinearity specified on 0 V to V_{REF} and 0 V to V_{DD} ranges.
[12] DAC specification for output impedance in the unbuffered case depends on DAC code.
[13] DAC specifications for I_{SINK}, voltage output settling time, and digital-to-analog glitch energy depend on external buffer implementation in unbuffered mode. DAC in unbuffered mode tested with OP270 external buffer, which has a low input leakage current.
[14] Measured with V_{REF} and C_{REF} pins decoupled with 0.1 µF capacitors to ground. Power-up time for the internal reference is determined by the value of the decoupling capacitor chosen for both the V_{REF} and C_{REF} pins.
[15] When using an external reference device, the internal band gap reference input can be bypassed by setting the ADCCON1[6] bit. In this mode, the V_{REF} and C_{REF} pins need to be shorted together for correct operation.
[16] Flash/EE Memory reliability characteristics apply to both the Flash/EE program memory and the Flash/EE data memory.
[17] Endurance is qualified to 100,000 cycles as per JEDEC Std. 22 method A117 and measured at −40°C, +25°C, and +125°C. Typical endurance at 25°C is 700,000 cycles.
[18] Retention lifetime equivalent at junction temperature (T_J) = 55°C as per JEDEC Std. 22 Method A117. Retention lifetime based on an activation energy of 0.6 eV derates with junction temperature as shown in Figure 48 in the ADuC832 Flash/EE Memory Reliability section.
[19] Power supply current consumption is measured in normal, idle, and power-down modes under the following conditions:
Normal mode: RESET = 0.4 V, digital I/O pins = open circuit, Core_CLK changed via the CD bits in PLLCON[2:0], core executing internal software loop.
Idle mode: RESET = 0.4 V, digital I/O pins = open circuit, Core_CLK changed via the CD bits in PLLCON, PCON[0] = 1, core execution suspended in idle mode.
Power-down mode: RESET = 0.4 V, all Port 0 pins = 0.4 V, all other digital I/O and Port 1 pins are open circuit, Core_CLK changed via the CD bits in PLLCON, PCON[1] = 1, core execution suspended in power-down mode, oscillator turned on or off via OSC_PD bit (PLLCON[7]).
[20] DV_{DD} power supply current increases typically by 3 mA (3 V operation) and 10 mA (5 V operation) during a Flash/EE memory program or erase cycle.

Table 9. SPI Master Mode Timing (CPHA = 0)

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_{DOSU}	Data output setup before SCLOCK edge			150	ns
t_{DSU}	Data input setup time before SCLOCK edge	100			ns
t_{DHD}	Data input hold time after SCLOCK edge	100			ns
t_{DF}	Data output fall time		10	25	ns
t_{DR}	Data output rise time		10	25	ns
t_{SR}	SCLOCK rise time		10	25	ns
t_{SF}	SCLOCK fall time		10	25	ns

[1] See Figure 11.
[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.

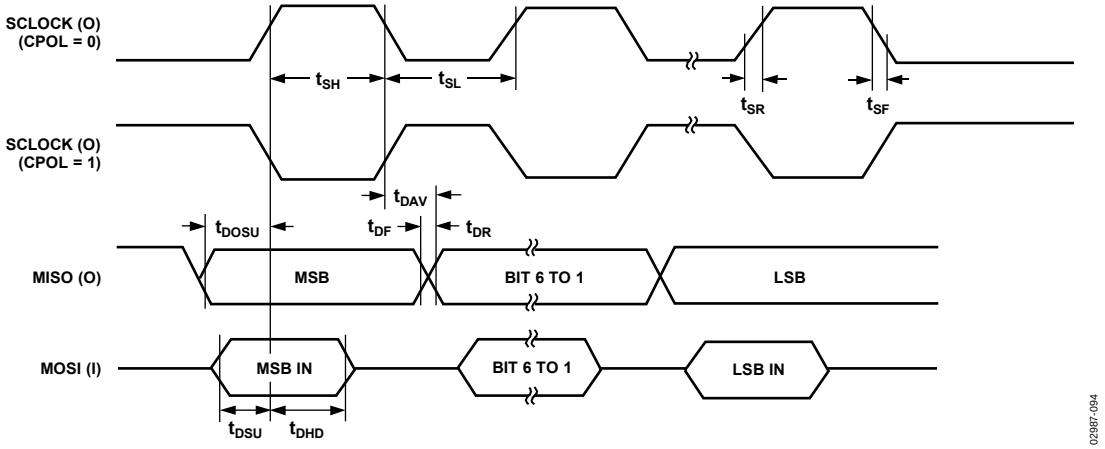

Figure 11. SPI Master Mode Timing (CPHA = 0)

Mnemonic	Pin No. MQFP	Pin No. LFCSP	Type	Description
P3.4/T0/PWMC/PWM0/EXTCLK	22	24	I/O	Input/Output Port 3 (P3.4). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low sources current because of the internal pull-up resistors.
			I	Timer/Counter 0 Input (T0).
			I	PWM Clock Input (PWMC).
			O	PWM0 Voltage Output (PWM0). PWM outputs can be configured to uses Port 2.6 and Port 2.7, or Port 3.4 and Port 3.3.
			I	Input for External Clock Signal (EXTCLK). This pin must be enabled via the CFG832 register.
P3.5/T1/$\overline{\text{CONVST}}$	23	25	I/O	Input/Output Port 3 (P3.5). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low sources current because of the internal pull-up resistors.
			I	Timer/Counter 1 Input (T1).
			I	Active Low Convert Start Logic Input for the ADC Block When the External Convert Start Function is Enabled ($\overline{\text{CONVST}}$). A low-to-high transition on this input puts the track-and-hold into its hold mode and starts conversion.
P3.6/$\overline{\text{WR}}$	24	26	I/O	Input/Output Port 3 (P3.6). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low sources current because of the internal pull-up resistors.
			O	Write Control Signal, Logic Output ($\overline{\text{WR}}$). Latches the data byte from Port 0 into the external data memory.
P3.7/$\overline{\text{RD}}$	25	27	O	Input/Output Port 3 (P3.7). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low sources current because of the internal pull-up resistors.
			O	Read Control Signal, Logic Output ($\overline{\text{RD}}$). Enables the external data memory to Port 0.
SCLOCK	26	28	I/O	Serial Clock Pin for I^2C-Compatible or SPI Serial Interface Clock.
SDATA/MOSI	27	29	I/O	User Selectable, I^2C-Compatible or SPI Data Input/Output Pin (SDATA).
			I/O	SPI Master Output/Slave Input Data I/O Pin for SPI Interface (MOSI).
P2.0/A8/A16	28	30	I/O	Input/Output Port 2 (P2.0). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low sources current because of the internal pull-up resistors.
			I/O	External Memory Addresses (A8/A16). 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.
P2.1/A9/A17	29	31	I/O	Input/Output Port 2 (P2.1). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low sources current because of the internal pull-up resistors.
			I/O	External Memory Addresses (A9/A17). 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.
P2.2/A10/A18	30	32	I/O	Input/Output Port 2 (P2.2). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low sources current because of the internal pull-up resistors.
			I/O	External Memory Addresses (A10/A18). 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.
P2.3/A11/A19	31	33	I/O	Input/Output Port 2 (P2.3). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low sources current because of the internal pull-up resistors.
			I/O	External Memory Addresses (A11/A19). 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.
XTAL1	32	34	I	Input to the Inverting Oscillator Amplifier.

Mnemonic	Pin No. MQFP	Pin No. LFCSP	Type	Description
XTAL2	33	35	O	Output of the Inverting Oscillator Amplifier.
P2.4/A12/A20	36	39	I/O	Input/Output Port 2 (P2.4). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low sources current because of the internal pull-up resistors.
			I/O	External Memory Addresses (A12/A20). 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.
P2.5/A13/A21	37	40	I/O	Input/Output Port 2 (P2.5). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low sources current because of the internal pull-up resistors.
			I/O	External Memory Addresses (A13/A21). 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.
P2.6/PWM0/A14/A22	38	41	I/O	Input/Output Port 2 (P2.6). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low sources current because of the internal pull-up resistors.
			O	PWM0 Voltage Output (PWM0). PWM outputs can be configured to use Port 2.6 and Port 2.7 or Port 3.4 and Port 3.3
			I/O	External Memory Addresses (A14/A22). 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.
P2.7/PWM1/A15/A23	39	42	I/O	Input/Output Port 2 (P2.7). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low sources current because of the internal pull-up resistors.
			O	PWM1 Voltage Output (PWM1). See the ADuC832 Configuration SFR (CFG832) section for further information.
			I/O	External Memory Addresses (A15/A23). 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.
$\overline{EA}$	40	43	I	External Access Enable, Logic Input. When held high, this input enables the device to fetch code from internal program memory locations 0000H to 1FFFH. When held low, this input enables the device to fetch all instructions from external program memory. This pin should not be left floating.
$\overline{PSEN}$	41	44	O	Program Store Enable, Logic Output. This output is a control signal that enables the external program memory to the bus during external fetch operations. It is active every six oscillator periods except during external data memory accesses. This pin remains high during internal program execution. $\overline{PSEN}$ can also be used to enable serial download mode when pulled low through a resistor on power-up or reset.
ALE	42	45	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.
P0.0/AD0	43	46	I/O	Input/Output Port 0 (P0.0). 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 (AD0). 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.
P0.1/AD1	44	47	I/O	Input/Output Port 0 (P0.1). 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 (AD1). 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.

EXPLANATION OF TYPICAL PERFORMANCE PLOTS

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

Figure 16 and Figure 17 show typical ADC integral nonlinearity (INL) errors from ADC Code 0 to Code 4095 at 5 V and 3 V supplies, respectively. The ADC is using its internal reference (2.5 V) and operating at a sampling rate of 152 kHz, and the typically worst-case errors in both plots are slightly less than 0.3 LSBs.

Figure 18 and Figure 19 show the variation in worst-case positive (WCP) INL and worst-case negative (WCN) INL vs. external reference input voltage.

Figure 20 and Figure 21 show typical ADC differential nonlinearity (DNL) errors from ADC Code 0 to Code 4095 at 5 V and 3 V supplies, respectively. The ADC is using its internal reference (2.5 V) and operating at a sampling rate of 152 kHz, and the typically worst-case errors in both plots is slightly less than 0.2 LSBs.

Figure 22 and Figure 23 show the variation in worst-case positive (WCP) DNL and worst-case negative (WCN) DNL vs. external reference input voltage.

Figure 24 shows a histogram plot of 10,000 ADC conversion results on a dc input with V_{DD} = 5 V. The plot illustrates an excellent code distribution pointing to the low noise performance of the on-chip precision ADC.

Figure 25 shows a histogram plot of 10,000 ADC conversion results on a dc input for V_{DD} = 3 V. The plot again illustrates a very tight code distribution of 1 LSB with the majority of codes appearing in one output pin.

Figure 26 and Figure 27 show typical FFT plots for the ADuC832. These plots were generated using an external clock input. The ADC is using its internal reference (2.5 V) sampling a full-scale, 10 kHz sine wave test tone input at a sampling rate of 149.79 kHz. The resultant FFTs shown at 5 V and 3 V supplies illustrate an excellent 100 dB noise floor, 71 dB or greater signal-to-noise ratio (SNR), and THD greater than −80 dB.

Figure 28 and Figure 29 show typical dynamic performance vs. external reference voltages. Again, excellent ac performance can be observed in both plots with some roll-off being observed as V_{REF} falls below 1 V.

Figure 30 shows typical dynamic performance vs. sampling frequency. SNR levels of 71 dB are obtained across the sampling range of the ADuC832.

Figure 31 shows the voltage output of the on-chip temperature sensor vs. temperature. Although the initial voltage output at 25°C can vary from part to part, the resulting slope of −2 mV/°C is constant across all parts.

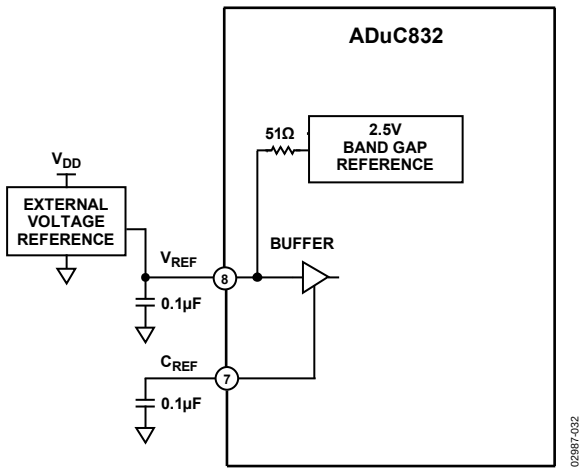

Figure 43. Using an External Voltage Reference

CONFIGURING THE ADC

The successive approximation ADC of the ADuC832 is driven by a divided down version of the master clock. To ensure adequate ADC operation, this ADC clock must be between 400 kHz and 6 MHz, and optimum performance is obtained with ADC clock between 400 kHz and 4.5 MHz. Frequencies within this range can easily be achieved with master clock frequencies from 400 kHz to well above 16 MHz with the four ADC clock divide ratios to choose from. For example, set the ADC clock divide ratio to 4 (that is, ADCCLK = 16.78 MHz/8 = 2 MHz) by setting the appropriate bits in ADCCON1 (ADCCON1[5:4] = 00).

The total ADC conversion time is 15 ADC clocks, plus 1 ADC clock for synchronization, plus the selected acquisition time (one, two, three, or four ADC clocks). For the preceding example, with a three-clock acquisition time, total conversion time is 19 ADC clocks (or 9.05 sec for a 2 MHz ADC clock).

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

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

Increasing the conversion time on the temperature sensor 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 is designed to run at a maximum conversion speed of 4 µs (247 kSPS sampling rate). When converting at this rate, the ADuC832 MicroConverter® has 4 µs to read the ADC result and store the result in memory for further postprocessing; otherwise, the next ADC sample may be lost. In an interrupt driven routine, the MicroConverter also has to jump to the ADC interrupt service routine, which also increases the time required to store the ADC results. In applications where the ADuC832 cannot sustain the interrupt rate, an ADC DMA mode is provided.

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

A Typical DMA Mode Configuration Example

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

1. The ADC must be powered down. This is done by ensuring MD1 is 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. Because the ADC DMA mode operates independent from the ADuC832 core, it is necessary to provide it with a stop command. This is done by duplicating the last channel ID to be converted, followed by 1111 into the next channel selection field. A typical preconfiguration of external memory is as follows:

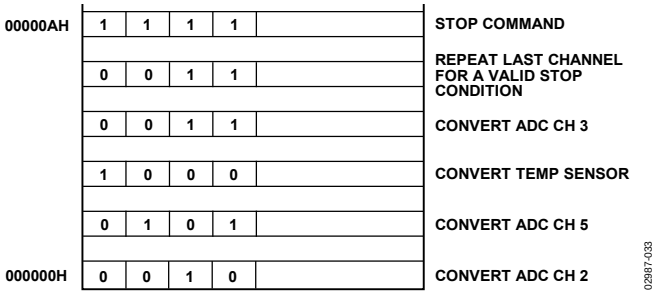

Figure 44. Typical DMA External Memory Preconfiguration

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

NONVOLATILE FLASH/EE MEMORY

FLASH/EE MEMORY OVERVIEW

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

This technology is basically an outgrowth of EPROM technology and was developed through the late 1980s. Flash/EE memory takes the flexible in-circuit reprogrammable features of EEPROM and combines them with the space efficient/density features of EPROM (see Figure 47).

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.

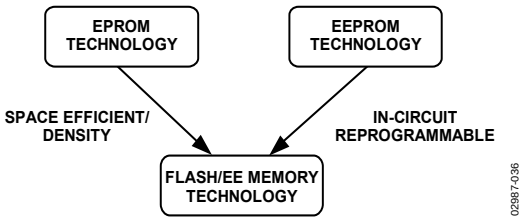

Figure 47. Flash/EE Memory Development

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

FLASH/EE MEMORY AND THE ADUC832

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

A 4 kB Flash/EE data memory space is also provided on chip. This can 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 the byte level, although it must first be erased in 4-byte pages.

ADUC832 FLASH/EE MEMORY RELIABILITY

The Flash/EE program and data memory arrays on the ADuC832 are fully qualified for two key Flash/EE memory characteristics, namely Flash/EE memory cycling endurance and Flash/EE memory data retention.

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

- Initial page erase sequence
- Read/verify sequence
- Byte program sequence
- Second read/verify sequence

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

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

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. Again, the ADuC832 has been qualified in accordance with the formal JEDEC retention lifetime specification (A117) at a specific junction temperature (T_J = 55°C). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit described previously, 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, derates with T_J, as shown in Figure 48.

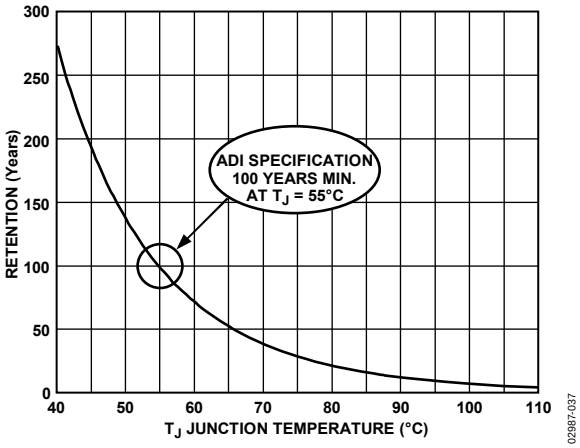

Figure 48. Flash/EE Memory Data Retention

ADUC832 CONFIGURATION SFR (CFG832)

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

CFG832 (ADuC832 Configuration SFR)

SFR Address: AFH

Power-On Default Value: 00H

Bit Addressable: No

Table 24. CFG832 SFR Bit Designations

Bit	Name	Description
[7]	EXSP	Extended SP enable. When set to 1 by the user, the stack rolls over from SPH/SP = 00FFH to SPH/SP = 0100H. When set to 0 by the user, the stack rolls over from SP = FFH to SP = 00H.
[6]	PWPO	PWM pinout selection. When set to 1 by the user, the PWM output pins are selected as P3.4 and P3.3. When set to 0 by the user, the PWM output pins are selected as P2.6 and P2.7.
[5]	DBUF	DAC output buffer. When set to 1 by the user, the DAC output buffer is bypassed. When set to 0 by the user, the DAC output buffer is enabled.
[4]	EXTCLK	Set by the user to 1 to select an external clock input on P3.4. Set by the user to 0 to use the internal PLL clock.
[3]	RSVD	Reserved. This bit should always contain 0.
[2]	RSVD	Reserved. This bit should always contain 0.
[1]	RSVD	Reserved. This bit should always contain 0.
[0]	XRAMEN	XRAM enable bit. When set to 1 by the user, the internal XRAM is mapped into the lower 2 kB of the external address space. When set to 0 by the user, the internal XRAM is not accessible and the external data memory is mapped into the lower 2 kB of external data memory.

SERIAL PERIPHERAL INTERFACE

The ADuC832 integrates a complete hardware serial peripheral interface (SPI) on chip. SPI is an industry standard synchronous serial interface that allows eight bits of data to be synchronously transmitted and received simultaneously, that is, full duplex. It should be noted that the SPI pins are shared with the I²C pins. Therefore, the user can only enable one or the other interface at any given time (see SPE in Table 28). The SPI port can be configured for master or slave operation and typically consists of four pins: MISO, MOSI, SCLOCK, and $\overline{SS}$.

MISO (MASTER INPUT, SLAVE OUTPUT DATA PIN)

The MISO (master input, slave output) 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 OUTPUT, SLAVE INPUT PIN)

The MOSI (master output, slave input) 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 28). In slave mode, the SPICON register must be configured with the phase and polarity (CPHA and CPOL) of the expected input clock. In both master and slave modes, the data is transmitted on one edge of the SCLOCK signal and sampled on the other. It is important therefore that the CPHA and CPOL are configured the same for the master and slave devices.

$\overline{SS}$ (SLAVE SELECT INPUT PIN)

The slave select ($\overline{SS}$) input pin is shared with the ADC5 input. To configure this pin as a digital input, the bit must be cleared, for example, CLR P1.5.

This line is active low. Data is only received or transmitted in slave mode when the $\overline{SS}$ pin is low, allowing the ADuC832 to be used in single master, multislave SPI configurations. If CPHA = 1, then the $\overline{SS}$ input may be permanently pulled low. With CPHA = 0, the $\overline{SS}$ input must be driven low before the first bit in a byte-wide transmission or reception, and return high again after the last bit in that byte-wide transmission or reception. In SPI slave mode, the logic level on the external $\overline{SS}$ pin can be read via the SPR0 bit in the SPICON SFR.

The following SFR registers (SPICON and SPIDAT) are used to control the SPI interface.

SPICON (SPI Control Register)

SFR Address: F8H

Power-On Default Value: 04H

Bit Addressable: Yes

Table 28. 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 pins.		
[4]	SPIM	SPI master/slave mode select bit. Set by user to enable Master Mode operation (SCLOCK is an output). Cleared by user to enable slave mode operation (SCLOCK is an input).		
[3]	CPOL[1]	Clock polarity select bit. Set by user if SCLOCK idles high. Cleared by user if SCLOCK idles low.		
[2]	CPHA[1]	Clock phase select bit. Set by user if leading SCLOCK edge is to transmit data. Cleared by user if trailing SCLOCK edge is to transmit data.		
[1:0]	SPR[1:0]	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	$f_{OSC}/16$
		In SPI slave mode, that is, SPIM = 0, the logic level on the external $\overline{SS}$ pin can be read via the SPR0 bit.		

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

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.

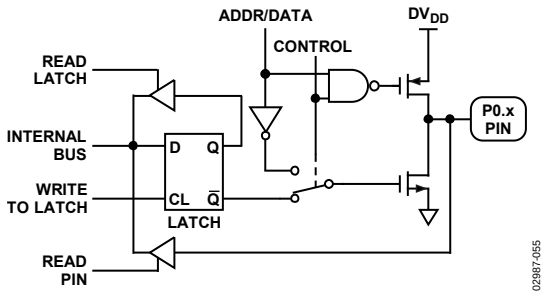

Figure 66. Port 0 Bit Latch and I/O Buffer

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

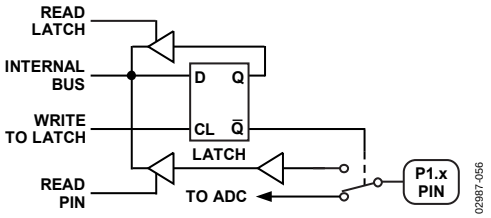

Figure 67. Port 1 Bit Latch and I/O Buffer

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

them drive a logic low output voltage (V_{OL}) and are capable of sinking 1.6 mA.

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

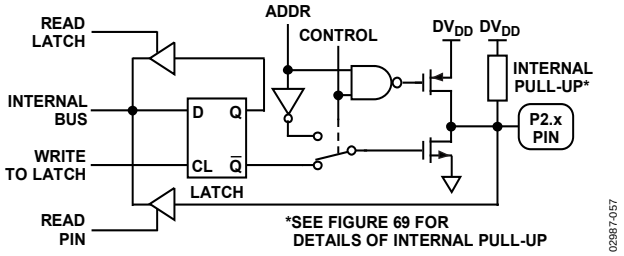

Figure 68. Port 2 Bit Latch and I/O Buffer

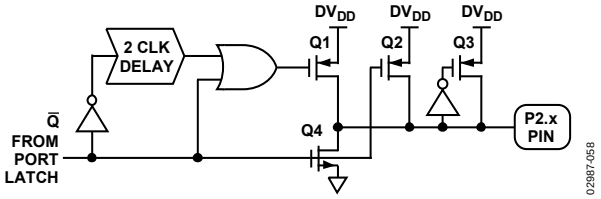

Figure 69. Internal Pull-Up Configuration

PORT 3

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

Port 3 pins also have various secondary functions described in Table 35. 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 35. Port 3, Alternate Pin Functions

Pin	Alternate Function
P3.0	RxD (UART input pin or serial data I/O in Mode 0)
P3.1	TxD (UART output pin or serial clock output in Mode 0)
P3.2	$\overline{INT0}$ (External Interrupt 0)
P3.3	$\overline{INT1}$ (External Interrupt 1) or PWM1/MISO
P3.4	T0 (Timer/Counter 0 external input), PWMC, PWM0, or EXTCLK
P3.5	T1 (Timer/Counter 1 external input) or $\overline{CONVST}$
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

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

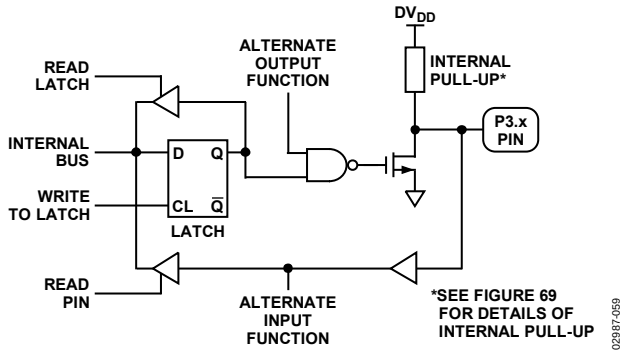

Figure 70. Port 3 Bit Latch and I/O Buffer

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 71 and Figure 73, respectively, for SPI operation and in Figure 72 and Figure 74 for I²C operation.

Notice that in I²C mode (SPE, SPICON[5] = 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 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 50 ns 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 the SPI or I²C functions are not used, these two pins can be used to give additional high current digital outputs.

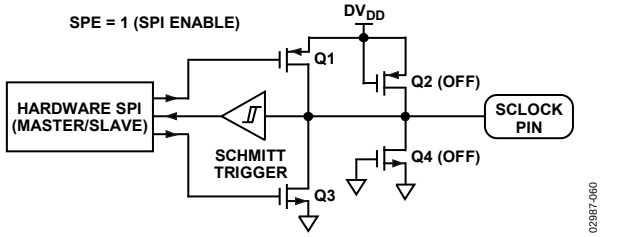

Figure 71. SCLOCK Pin I/O Functional Equivalent in SPI Mode

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.

TIMER/COUNTER 2

T2CON (TIMER/COUNTER 2 CONTROL REGISTER)

SFR Address: C8H

Power-On Default Value: 00H

Bit Addressable: Yes

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

TH2 is the Timer 2 data high byte and TL2 is the low byte. The SFR addresses for TH2 and TL2 are CDH and CCH, respectively.

RCAP2H and RCAP2L

RCAP2H is the Timer 2 capture/reload high byte and RCAP2L is the low byte. The SFR addresses for RCAP2H and RCAP2L are CBH and CAH, respectively.

Table 38. T2CON SFR Bit Designations

Bit	Name	Description
[7]	TF2	Timer 2 overflow flag. Set by hardware on a Timer 2 overflow. TF2 is not set when either RCLK = 1 or TCLK = 1. Cleared by user software.
[6]	EXF2	Timer 2 external flag. Set by hardware when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. Cleared by user software.
[5]	RCLK	Receive clock enable bit. Set by the user to enable the serial port to use Timer 2 overflow pulses for its receive clock in serial port Mode 1 and Mode 3. Cleared by the user to enable Timer 1 overflow to be used for the receive clock.
[4]	TCLK	Transmit clock enable bit. Set by the user to enable the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Mode 1 and Mode 3. Cleared by the user to enable Timer 1 overflow to be used for the transmit clock.
[3]	EXEN2	Timer 2 external enable flag. Set by the user to enable a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. Cleared by the user for Timer 2 to ignore events at T2EX.
[2]	TR2	Timer 2 start/stop control bit. Set by the user to start Timer 2. Cleared by the user to stop Timer 2.
[1]	CNT2	Timer 2 timer or counter function select bit. Set by the user to select counter function (input from external T2 pin). Cleared by the user to select timer function (input from on-chip core clock).
[0]	CAP2	Timer 2 capture/reload select bit. Set by the user to enable captures on negative transitions at T2EX if EXEN2 = 1. Cleared by the user to enable autoreloads with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to autoreload on Timer 2 overflow.

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 is lost. The physical interface to the serial data network is via the RXD and TXD pins, and the SFR interface to the UART is comprised of SBUF and SCON.

SBUF

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

SCON (UART SERIAL PORT CONTROL REGISTER)

SFR Address: 98H

Power-On Default Value: 00H

Bit Addressable: Yes

Table 40. SCON SFR Bit Designations

Bit	Name	Description		
[7:6]	SM[0:1]	UART serial mode select bits. These bits select the serial port operating mode as follows:		
		SM0	**SM1**	**Selected Operating Mode**
		0	0	Mode 0: shift register, fixed baud rate (Core_CLK/2)
		0	1	Mode 1: 8-bit UART, variable baud rate
		1	0	Mode 2: 9-bit UART, fixed baud rate (Core_CLK/64) or (Core_CLK/32)
		1	1	Mode 3: 9-bit UART, variable baud rate
[5]	SM2	Multiprocessor communication enable bit. Enables multiprocessor communication in Mode 2 and Mode 3. In Mode 0, SM2 should be cleared. In Mode 1, if SM2 is set, RI is not activated if a valid stop bit was not received. If SM2 is cleared, RI is set as soon as the byte of data has been received. In Mode 2 or Mode 3, if SM2 is set, RI is not activated if the received ninth data bit in RB8 is 0. If SM2 is cleared, RI is 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 is the ninth data bit that is transmitted in Mode 2 and Mode 3.		
[2]	RB8	Serial Port Receiver Bit 9. The ninth data bit received in Mode 2 and Mode 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 Mode 1, Mode 2, and Mode 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 Mode 1, Mode 2, and Mode 3. RI must be cleared by software.		

MODE 0: 8-BIT SHIFT REGISTER MODE

Mode 0 is selected by clearing both the SM0 and SM1 bits in the SCON SFR. Serial data enter and exit 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 81.

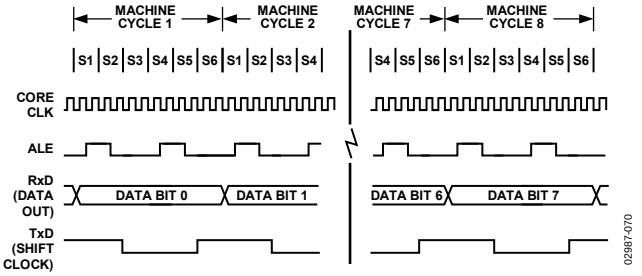

Figure 81. UART Serial Port Transmission, Mode 0

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), 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 82.

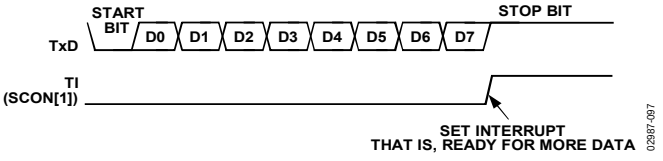

Figure 82. UART Serial Port Transmission, Mode 0

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.

These events occur 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
- 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 starts 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.

These events occur 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
- The received stop bit = 1

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set.

MODE 3: 9-BIT UART WITH VARIABLE BAUD RATE

Mode 3 is selected by setting both SM0 and SM1. In this mode, the 8051 UART serial port operates in 9-bit mode with a variable baud rate determined by either Timer 1 or Timer 2. The operation of the 9-bit UART is the same as for Mode 2 but the baud rate can be varied as for Mode 1.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

UART SERIAL PORT BAUD RATE GENERATION

Mode 0 Baud Rate Generation

The baud rate in Mode 0 is fixed.

Mode 0 Baud Rate = (Core_CLK 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:

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

Mode 1 and Mode 3 Baud Rate Generation

The baud rates in Mode 1 and Mode 3 are determined by the overflow rate in Timer 1 or Timer 2, or both (one for transmit and the other for receive).

TIMER 1 GENERATED BAUD RATES

When Timer 1 is used as the baud rate generator, the baud rates in Mode 1 and Mode 3 are determined by the Timer 1 overflow rate and the value of SMOD as follows:

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

The Timer 1 interrupt should be disabled in this application. The timer itself can be configured for either timer or counter operation, and in any of its three running modes. In the most typical application, it is configured for timer operation in the autoreload mode (high nibble of TMOD = 0010 binary). In that case, the baud rate is given by the formula:

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

Table 41 shows some commonly used baud rates and how they can be calculated from a core clock frequency of 16.78 MHz and 2.0971 MHz. A 5% error is tolerable using asynchronous (start/stop) communications.

Table 41. Commonly Used Baud Rates, Timer 1

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

TIMER 2 GENERATED BAUD RATES

Baud rates can also be generated using Timer 2. Using Timer 2 is similar to using Timer 1 in that the timer must overflow 16 times before a bit is transmitted/received. Because Timer 2 has a 16-bit autoreload mode, a wider range of baud rates is possible using Timer 2.

Mode 1 and Mode 3 Baud Rate =
(1/16) × (Timer 2 Overflow Rate)

Therefore, when Timer 2 is used to generate baud rates, the timer increments every two clock cycles and not every core machine cycle. Thus, it increments six times faster than Timer 1, and therefore baud rates six times faster are possible. Because Timer 2 has 16-bit autoreload capability, very low baud rates are still possible.

Timer 2 is selected as the baud rate generator by setting the TCLK and/or RCLK bit in T2CON. The baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 83.

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

$$\text{Modes 1 and 3 Baud Rate} = (\text{Core\_CLK})/(32 \times [6556 - (RCAP2H, RCAP2L)])$$

Table 42 shows some commonly used baud rates and how they can be calculated from a core clock frequency of 16.78 MHz and 2.10 MHz.

Table 42. Commonly Used Baud Rates, Timer 2

Ideal Baud	Core_CLK (MHz)	RCAP2H Value	RCAP2L Value	Actual Baud	% Error
19,200	16.78	−1 (FFH)	−27 (E5H)	19418	1.14
9600	16.78	−1 (FFH)	−55 (C9H)	9532	0.7
2400	16.78	−1 (FFH)	−218 (26H)	2405	0.21
1200	16.78	−2 (FEH)	−181 (4BH)	1199	0.02
9600	2.10	−1 (FFH)	−7 (FBH)	9362	2.4
2400	2.10	−1 (FFH)	−27 (ECH)	2427	1.14
1200	2.10	−1 (FFH)	−55 (C9H)	1191	0.7

INTERRUPT SYSTEM

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

- IE—interrupt enable register
- IP—interrupt priority register
- IEIP2—secondary interrupt enable register

IE (INTERRUPT ENABLE REGISTER)

SFR Address: A8H

Power-On Default Value: 00H

Bit Addressable: Yes

Table 45. IE SFR Bit Designations

Bit	Name	Description
[7]	EA	Written by user to enable or disable all interrupt sources (1 = enable; 0 = disable)
[6]	EADC	Written by user to enable or disable ADC interrupt (1 = enable; 0 = disable)
[5]	ET2	Written by user to enable or disable Timer 2 interrupt (1 = enable; 0 = disable)
[4]	ES	Written by user to enable or disable UART serial port interrupt (1 = enable; 0 = disable)
[3]	ET1	Written by user to enable or disable Timer 1 interrupt (1 = enable; 0 = disable)
[2]	EX1	Written by user to enable or disable External Interrupt 1 (1 = enable; 0 = disable)
[1]	ET0	Written by user to enable or disable Timer 0 interrupt (1 = enable; 0 = disable)
[0]	EX0	Written by user to enable or disable External Interrupt 0 (1 = enable; 0 = disable)

IP (INTERRUPT PRIORITY REGISTER)

SFR Address: B8H

Power-On Default Value: 00H

Bit Addressable: Yes

Table 46. IP SFR Bit Designations

Bit	Name	Description
[7]	Reserved	Reserved for future use.
[6]	PADC	Written by user to select ADC interrupt priority (1 = high; 0 = low)
[5]	PT2	Written by user to select Timer 2 interrupt priority (1 = high; 0 = low)
[4]	PS	Written by user to select UART serial port interrupt priority (1 = high; 0 = low)
[3]	PT1	Written by user to select Timer 1 interrupt priority (1 = high; 0 = low)
[2]	PX1	Written by user to select External Interrupt 1 priority (1 = high; 0 = low)
[1]	PT0	Written by user to select Timer 0 interrupt priority (1 = high; 0 = low)
[0]	PX0	Written by user to select External Interrupt 0 priority (1 = high; 0 = low)

IEIP2 (SECONDARY INTERRUPT ENABLE REGISTER)

SFR Address A9H

Power-On Default Value A0H

Bit Addressable No

Table 47. IEIP2 SFR Bit Designations

Bit	Name	Description
[7]	Reserved	Reserved for future use
[6]	PTI	Priority for time interval interrupt
[5]	PPSM	Priority for power supply monitor interrupt
[4]	PSI	Priority for SPI/I^2C interrupt
[3]	Reserved	This bit must contain 0.
[2]	ETI	Written by user to enable or disable time interval counter interrupt. (1 = enable; 0 = disable)
[1]	EPSMI	Written by user to enable or disable power supply monitor interrupt. (1 = enable; 0 = disable)
[0]	ESI	Written by user to enable or disable SPI or I^2C serial port interrupt. (1 = enable; 0 = disable)

ADUC832 HARDWARE DESIGN CONSIDERATIONS

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

CLOCK OSCILLATOR

The clock source for the ADuC832 can be generated by the internal PLL or by an external clock input. To use the internal PLL, connect a 32.768 kHz parallel resonant crystal between XTAL1 and XTAL2, and connect a capacitor from each pin to ground as shown in Figure 85. This crystal allows the PLL to lock correctly to give a f_{VCO} of 16.78 MHz. If no crystal is present, the PLL free runs, giving a f_{VCO} of 16.7 MHz ± 20%. This is useful if an external clock input is required. The part powers up and the PLL free runs; the user can then write to the CFG832 SFR in the software to enable the external clock input on P3.4.

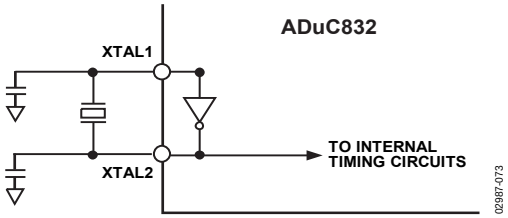

Figure 85. External Parallel Resonant Crystal Connections

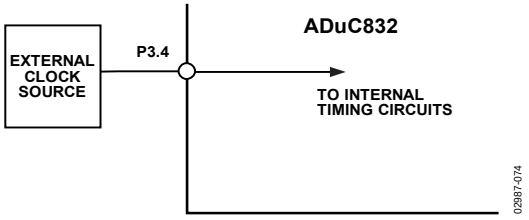

Figure 86. Connecting an External Clock Source

Whether using the internal PLL or an external clock source, the ADuC832 specified operational clock speed range is 400 kHz to 16.78 MHz. The core itself is static, and functions all the way down to dc. However, at clock speeds slower than 400 kHz, the ADC no longer functions correctly. Therefore, to ensure specified operation, use a clock frequency of at least 400 kHz and no more than 16.78 MHz.

EXTERNAL MEMORY INTERFACE

In addition to its internal program and data memories, the ADuC832 can access up to 64 kB of external program memory (such as ROM and PROM) and up to 16 MB of external data memory (SRAM).

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

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

External program memory (if used) must be connected to the ADuC832 as illustrated in Figure 87. Note that 16 I/O lines (Port 0 and Port 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 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 address latch enable (ALE) clocks this byte into an address latch. Meanwhile, Port 2 (P2) emits the high byte of the program counter, then $\overline{PSEN}$ strobes the EPROM and the code byte is read into the ADuC832.

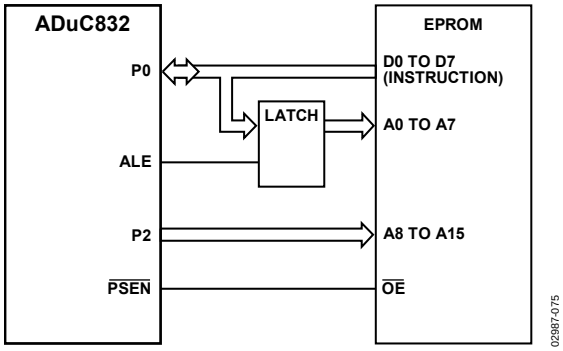

Figure 87. External Program Memory Interface

Note that program memory addresses are always 16 bits wide, even in cases where the actual amount of program memory used is less than 64 kB. 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, Port 0 and Port 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 88 shows a hardware configuration for accessing up to 64 kB of external RAM. This interface is standard to any 8051-compatible MCU.

OTHER HARDWARE CONSIDERATIONS

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

IN-CIRCUIT SERIAL DOWNLOAD ACCESS

Nearly all ADuC832 designs can take advantage of the in-circuit reprogrammability of the chip. This is accomplished by a connection to the ADuC832 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 94 with a simple ADM202-based circuit. To avoid designing an RS-232 chip onto a board, refer to the uC006 Technical Note, *A 4-Wire UART-to-PC Interface,* for a simple (and zero-cost-per-board) method of gaining in-circuit serial download access to the ADuC832.

In addition to the basic UART connections, users also need a way to trigger the chip into download mode. This is accomplished via a 1 kΩ pull-down resistor that can be jumpered onto the $\overline{\text{PSEN}}$ pin, as shown in Figure 94. To put the ADuC832 into download mode, connect this jumper and power-cycle the device (or manually reset the device, if a manual reset button is available) so that it can receive a new program serially. With the jumper removed, the device comes up in normal mode (and runs the program) whenever power is cycled or RESET is toggled.

Note that $\overline{\text{PSEN}}$ is normally an output (as described in the External Memory Interface section) and is sampled as an input only on the falling edge of RESET (that is, at power-up or upon an external manual reset). Note also that if any external circuitry unintentionally pulls $\overline{\text{PSEN}}$ low during power-up or reset events, it may cause the chip to enter download mode and therefore fail to begin user code execution as it should. To prevent this, ensure that no external signals are capable of pulling the $\overline{\text{PSEN}}$ pin low, except for the external $\overline{\text{PSEN}}$ jumper itself.

EMBEDDED SERIAL PORT DEBUGGER

From a hardware perspective, entry into serial port debug mode is identical to the serial download entry sequence described in the In-Circuit Serial Download Access section. In fact, both serial download and serial port debug modes can be thought of as essentially one mode of operation used in two different ways.

Note that the serial port debugger is fully contained on the ADuC832 device (unlike ROM monitor type debuggers) and therefore no external memory is needed to enable in-system debug sessions.

SINGLE-PIN EMULATION MODE

Also built into the ADuC832 is a dedicated controller for single-pin in-circuit emulation (ICE) using standard production ADuC832 devices. In this mode, emulation access is gained by connection to a single pin, the $\overline{\text{EA}}$ pin. Normally, this pin is hardwired either high or low to select execution from internal or external program memory space. To enable single-pin emulation mode, however, users need to pull the $\overline{\text{EA}}$ pin high through a 1 kΩ resistor, as shown in Figure 94. The emulator then connects to the 2-pin header, also shown in Figure 94. To be compatible with the standard connector that comes with the single-pin emulator, use a 2-pin 0.1 inch pitch friction lock header from Molex such as Part Number 22-27-2021. Be sure to observe the polarity of this header. As represented in Figure 94, when the friction lock tab is located on the right, the ground pin should be the lower of the two pins (when viewed from the top).

TYPICAL SYSTEM CONFIGURATION

A typical ADuC832 configuration is shown in Figure 94. It summarizes some of the hardware considerations discussed in the Single-Pin Emulation Mode section.