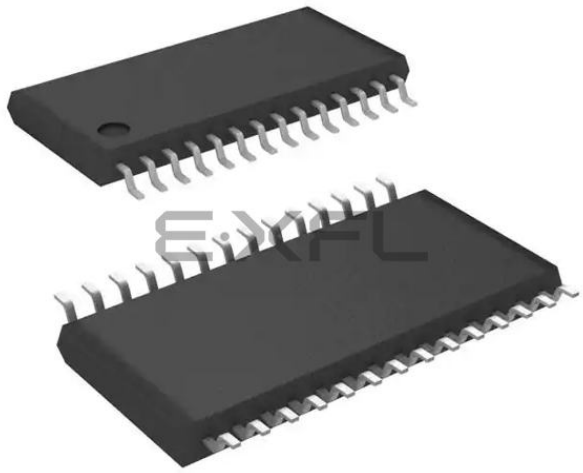

Welcome to E-XFL.COM

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	16.78MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	POR, PSM, Temp Sensor, WDT
Number of I/O	17
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	640 x 8
RAM Size	256 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 6x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	28-TSSOP (0.173", 4.40mm Width)
Supplier Device Package	28-TSSOP
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc814aruz-reel7

[1]Temperature range –40ºC to +125ºC.
[2]ADC linearity is guaranteed when operating in nonpipelined mode, i.e., ADC conversion followed sequentially by a read of the ADC result. ADC linearity is also guaranteed during normal MicroConverter core operation.
[3]ADC LSB size = V_{REF} /2^{12}, i.e., for internal V_{REF} = 2.5 V, 1 LSB = 610 µV, and for external V_{REF} = 1 V, 1 LSB = 244 µV.
[4]Offset and gain error and offset and gain error match are measured after factory calibration.
[5]Based on external ADC system components the user may need to execute a system calibration to remove additional external channel errors and achieve these specifications.
[6]Measured with coherent sampling system using external 16.77 MHz clock via P3.5 (Pin 22).
[7]SNR calculation includes distortion and noise components.
[8]Channel-to-channel crosstalk is measured on adjacent channels.
[9]The temperature monitor gives a measure of the die temperature directly; air temperature can be inferred from this result.
[10]DAC linearity is calculated using a reduced code range of 48 to 4095, 0 V to V_{REF} range; a reduced code range of 48 to 3950, 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 to V_{DD} ranges.
[12]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.
[13]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.
[14]These numbers are not production tested but are guaranteed by design and/or characterization data on production release.
[15]Pins configured in I^2C compatible mode or SPI mode; pins configured as digital inputs during this test.
[16]These typical specifications assume no loading on the XTAL2 pin. Any additional loading on the XTAL2 pin increases the power-on times.
[17]Flash/EE memory reliability characteristics apply to both the Flash/EE program memory and the Flash/EE data memory.
[18]Endurance is qualified to 100 kcycles as per JEDEC Std. 22, Method A117 and measured at –40ºC, +25°C, and +125°C; typical endurance at +25°C is 700 kcycles.
[19]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 33 in the Flash/EE memory description section.
[20]Power supply current consumption is measured in normal, idle, and power-down modes under the following conditions:
Normal Mode: Reset and all digital I/O pins = open circuit, core Clk changed via CD bits in PLLCON, core executing internal software loop.
Idle Mode: Reset and all digital I/O pins = open circuit, core Clk changed via CD bits in PLLCON, PCON.0 = 1, core execution suspended in idle mode.
Power-Down Mode: Reset and all P1.2–P1.7 pins = 0.4 V; all other digital I/O pins are open circuit, Core Clk changed via CD bits in PLLCON, PCON.1 = 1, Core execution suspended in power-down mode, OSC turned on or off via OSC_PD bit (PLLCON.7) in PLLCON SFR.
[21]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.

DRIVING THE ADC

The ADC incorporates a successive approximation architecture (SAR) involving a charge-sampled input stage. Each ADC conversion is divided into two distinct phases as defined by the position of the switches in Figure 25. During the sampling phase (with SW1 and SW2 in the track position), a charge proportional to the voltage on the analog input is developed across the input sampling capacitor. During the conversion phase (with both switches in the hold position), the capacitor DAC is adjusted via internal SAR logic until the voltage on Node A is zero, indicating that the sampled charge on the input capacitor is balanced out by the charge being output by the capacitor DAC. The digital value finally contained in the SAR is then latched out as the result of the ADC conversion. Control of the SAR, and timing of acquisition and sampling modes, is handled automatically by built-in ADC control logic. Acquisition and conversion times are also fully configurable under user control.

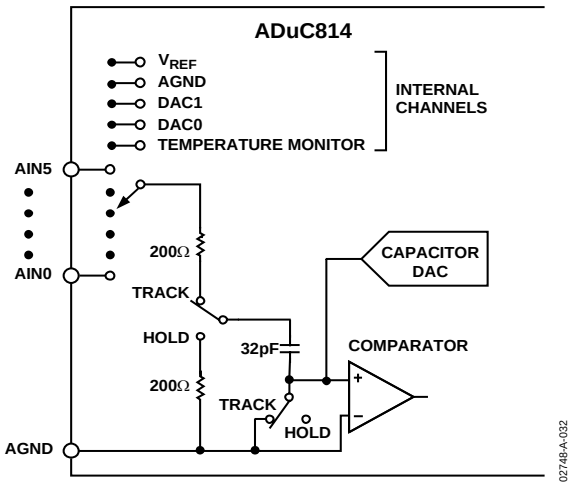

Figure 25. Internal ADC Structure

Note that whenever a new input channel is selected, a residual charge from the 32 pF sampling capacitor places a transient on the newly selected input. The signal source must be capable of recovering from this transient before the sampling switches click into hold mode. Delays can be inserted in software (between channel selection and conversion request) to account for input stage settling, but a hardware solution alleviates this burden from the software design task and ultimately results in a cleaner system implementation.

One hardware solution is to choose a very fast settling op amp to drive each analog input. Such an op amp would need to fully settle from a small signal transient in less than 300 ns in order to guarantee adequate settling under all software configurations. A better solution, recommended for use with any amplifier, is shown in Figure 26.

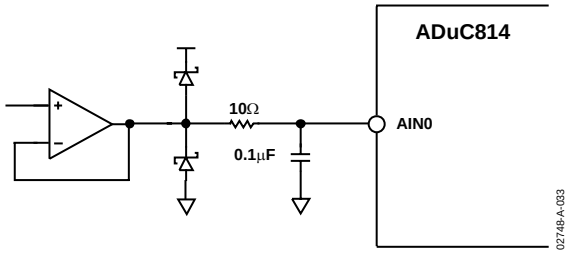

Figure 26. Buffering Analog Inputs

At first glance the circuit in Figure 26 may look like a simple anti-aliasing filter, it actually serves no such purpose. Though the R/C does help to reject some incoming high frequency noise, its primary function is to ensure that the transient demands of the ADC input stage are met. It does so by providing a capacitive bank from which the 32 pF sampling capacitor can draw its charge. Since the 0.1 μF capacitor in Figure 26 is more than 3000 times the size of the 32 pF sampling capacitor, its voltage does not change by more than one count of the 12-bit transfer function when the 32 pF charge from a previous channel is dumped onto it. A larger capacitor can be used if desired, but care needs to be taken if choosing a larger resistor (see Table 9).

The Schottky diodes in Figure 26 may be necessary to limit the voltage applied to the analog input pin as per the Absolute Maximum Ratings. They are not necessary if the op amp is powered from the same supply as the ADuC814 because, in that case, the op amp is unable to generate voltages above V_{DD} or below ground. An op amp of some kind is necessary unless the signal source is very low impedance to begin with. DC leakage currents at the ADuC814 analog inputs can cause measurable dc errors with external source impedances as little as 100 Ω or so. To ensure accurate ADC operation, keep the total source impedance at each analog input less than 61 Ω. Table 9 illustrates examples of how source impedance can affect dc accuracy.

Table 9. Source Impedance Errors

Source Impedance	Error from 1 μA Leakage Current	Error from 10 μA Leakage Current
61 Ω	61 μV = 0.1 LSB	610 μV = 1 LSB
610 Ω	610 μV = 1 LSB	6.1 mV = 10 LSB

Although Figure 26 shows the op amp operating at a gain of 1, you can configure it for any gain needed. Also, you can just as easily use an instrumentation amplifier in its place to condition differential signals. Use any modern amplifier that is capable of delivering the signal (0 V to V_{REF}) with minimal saturation. Some single-supply, rail-to-rail op-amps that are useful for this purpose include, but are certainly not limited to, the ones given in Table 10. Check the Analog Devices literature (CD ROM data book, etc.) for details on these and other op amps and instrumentation amps.

Both the ADCCLK frequency and the acquisition time are used in determining the ADC conversion time. Two other parameters are also used in this calculation. To convert the acquired signal into its corresponding digital output word takes 15 ADCCLK periods (T_{CONV}). When a conversion is initiated, the start of conversion signal is synchronized to the ADCCLK. This synchronization (T_{SYNC}) can take from 0.5 to 1.5 ADCCLKs to occur. The total ADC conversion time T_{ADC} is calculated using the following formula:

$$T_{ADC} = T_{SYNC} + T_{ACQ} + T_{CONV}$$

Assuming T_{SYNC} = 1, T_{ACQ} = 1 and F_{CORE}/ADCCLK divider of 4. The total conversion time is calculated by

$$T_{ADC} = (1 + 1 + 15) \times (1 / 4194304)$$

$$T_{ADC} = 4.05\ \mu s$$

These settings allow a maximum conversion speed or sampling rate of 246.7 kHz.

When converting on the temperature monitor channel, the conversion time is not controlled via the ADCCON registers. It is controlled in hardware and sets the ADCCLK to F_{CORE} /32 and uses four acquisition clocks, giving a total ADC conversion time of

$$T_{ADC} = (1 + 4 + 15) \times (1 / 524288) = 38.14\ \mu s$$

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

INITIATING ADC CONVERSIONS

After the ADC has been turned on and configured, there are four methods of initiating ADC conversions.

Single conversions can be initiated in software by setting the SCONV bit in the ADCCON2 register via user code. This causes the ADC to perform a single conversion and puts the result into the ADCDATAH/L SFRs. The SCONV bit is cleared as soon as the ADCDATA SFRs have been updated.

Continuous conversion mode can be initiated by setting the CCONV bit in ADCCON2 via user code. This performs back-to-back conversions at the configured rate (246.7 kHz for the settings detailed previously). In continuous mode, the ADC results must be read from the ADCDATA SFRs before the next conversion is completed to avoid loss of data. Continuous mode can be stopped by clearing the CCONV bit.

An external signal can also be used to initiate ADC conversions. Setting Bit 0 in ADCCON1 enables the logic to allow an external start-of-conversion signal on Pin 7 ($\overline{CONVST}$). This active low pulse should be at least 100 ns wide. The rising edge of this signal initiates the conversion.

Timer 2 can also be used to initiate conversions. Setting Bit 1 of ADCCON1 enables the Timer 2 overflow signal to start a conversion. For Timer 2 configuration information, see the Timers/Counters section.

For both external $\overline{CONVST}$ and Timer 2 overflow, the conversion rate must be equal to or greater than the conversion time (T_{ADC}) to avoid incorrect ADC results.

When initiating conversions, the user must ensure that only one of the trigger modes is active at any one time. Initiating conversions with more than one of the trigger modes active results in erratic ADC behavior.

ADC HIGH SPEED DATA CAPTURE MODE

The on-chip ADC has been designed to run at a maximum conversion speed of 4.05 µs (247 kHz sampling rate). When converting at this rate, the ADuC814 MCU has 4.05 µs to read the ADC result and store it in memory for further post processing; otherwise the next ADC sample could be lost. The time to complete a conversion and store the ADC results without errors is known as the throughput rate. In an interrupt driven routine, the MCU also has to jump to the ADC interrupt service routine, which decreases the throughput rate of the ADuC814. In applications where the ADuC814 standard operating mode throughput is not fast enough, an ADC high speed data capture (HSDC) mode is provided.

In HSDC mode, ADC results are transferred to the SPI logic without intervention from the ADuC814 core logic. In applications where the ADC throughput is slow, the HSDC logic operates in non-pipelined mode (Figure 29). In this mode, there is adequate time for the ADC conversion and the ADC-to-SPI data transfer to complete before the next start of conversion. As the ADC throughput increases, the HSDC logic begins to operate in pipelined mode as shown in Figure 30.

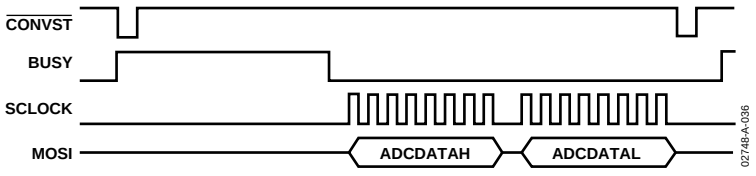

Figure 29. High Speed Data Capture Logic Timing (Non-Pipelined Mode)

USING FLASH/EE DATA MEMORY

The user Flash/EE data memory array consists of 640 bytes that are configured into 160 (00H to 9FH) 4-byte pages as shown in Figure 36.

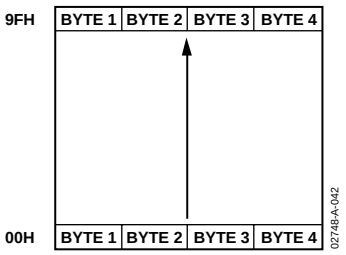

Figure 36. Flash/EE Data Memory Configuration

As with other ADuC814 user-peripheral circuits, the interface to this memory space is via a group of registers mapped in the SFR space. EADRL is used to hold the 8-bit address of the page to be accessed. A group of four data registers (EDATA1–4) is used to hold 4-byte page data just accessed. Finally, ECON is an 8-bit control register that may be written with one of five Flash/EE memory access commands to trigger various read, write, erase, and verify functions. These registers can be summarized as follows:

ECON

SFR Address	B9H
Function	Controls access to 640 bytes Flash/EE data space.
Default	00H

EADRL

SFR Address	C6H
Function	Holds the Flash/EE data page address. (640 bytes = > 160 page addresses)
Default	00H

EDATA1–4

SFR Address	BCH to BFH, respectively
Function	Holds Flash/EE data memory page write or page read data bytes.
Default	EDATA1–4 > 00H

A block diagram of the SFR interface to the Flash/EE data memory array is shown in Figure 37.

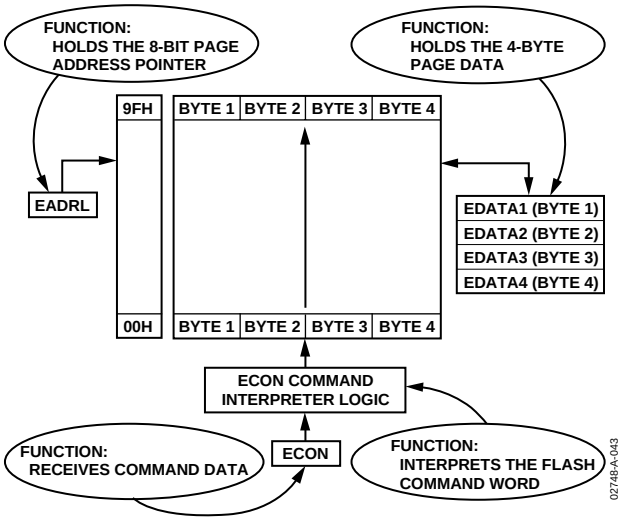

Figure 37. Flash/EE Data Memory Control and Configuration

ECON—Flash/EE Memory Control SFR

This SFR acts as a command interpreter and may be written with one of five command modes to enable various read, program, and erase cycles as detailed in Table 11.

Table 11. ECON–Flash/EE Memory Control Register Command Modes

Command Byte	Command Mode	Description
01H	READ	Results in 4 bytes being read into EDATA1–4 from memory page address contained in EADRL.
02H	PROGRAM	Results in 4 bytes (EDATA1–4) being written to memory page address in EADRL. This write command assumes the designated write page has been erased.
03H	Reserved	For internal use. 03H should not be written to the ECON SFR.
04H	VERIFY	Allows the user to verify if data in EDATA1–4 is contained in page address designated by EADRL. A subsequent read of the ECON SFR results in a zero being read if the verification is valid, a nonzero value is read to indicate an invalid verification.
05H	ERASE	Results in an erase of the 4-byte page designated in EADRL.
06H	ERASE-ALL	Results in an erase of the full Flash/EE aata memory, 160-page (640 bytes) array.
07H to FFH	Reserved	For future use.

USER INTERFACE TO OTHER ON-CHIP ADuC814 PERIPHERALS

This section gives a brief overview of the various peripherals also available on-chip. A summary of the SFRs used to control and configure these peripherals is also given.

DACS

The ADuC814 incorporates two 12-bit, voltage output DACs on-chip. Each DAC has a rail-to-rail voltage output buffer capable of driving 10 kΩ/100 pF. They have two selectable ranges, 0 V to V_{REF} (an external or the internal band gap 2.5 V reference) and 0 V to AV_{DD}, and can operate in 12-bit or 8-bit modes. DAC operation is controlled by a single special function (SFR) register, DACCON. Each DAC has two data registers, DACxH/L. The DAC0 and DAC1 outputs share pins with ADC inputs ADC4 and ADC5, respectively. When both DACs are on, the number of analog inputs is reduced to four. Note that in 12-bit mode, the DAC voltage output is updated as soon as the DACL data SFR has been written; therefore, the DAC data registers should be updated as DACH first, followed by DACL.

When using the DACs on the V_{REF} range it is necessary to power up the ADC to enable the reference to the DAC section. See Note 1.

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

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

Table 12. DACCON SFR Bit Designations

Bit No.	Name	Description
7	MODE	Mode Select Bit. Selects either 12-bit or 8-bit mode for both DACs. Set to 1 by the user to enable 8-bit mode (DACxL is the active data register). Set to 0 by the user to enable 12-bit mode.
6	RNG1	DAC1 Output Voltage Range Select Bit. Set to 1 by the user to configure DAC1 range of 0 V to AV_{DD}. Set to 0 by the user to configure DAC1 range of 0 V to 2.5 V (V_{REF} range) [1].
5	RNG0	DAC0 Output Voltage Range Select Bit. Set to 1 by he user to configure DAC0 range of 0 V to AV_{DD}. Set to 0 by the user to configure DAC0 range of 0 V to 2.5 V (V_{REF} range) [1].
4	$\overline{CLR1}$	DAC1 Clear Bit. Set to 1 by the user to enable normal DAC1 operation. Set to 0 by the user to force DAC1 output voltage to 0 V.
3	$\overline{CLR0}$	DAC0 Clear Bit. Set to 1 by the user to enable normal DAC0 operation. Set to 0 by the user to force DAC0 output voltage to 0 V.
2	SYNC	DAC0/1 Update Synchronization Bit. Set to 1 by the user to enable asynchronous update mode. The DAC outputs update as soon as the DACxL SFRs are written. Set to 0 by the user to enable synchronous update mode. The user can simultaneously update both DACs by first updating the DACxH/L SFRs while SYNC is 0. Both DACs then update simultaneously when the SYNC bit is set to 1.
1	$\overline{PD1}$	DAC1 Power-Down Bit. Set to 1 by the user to power up DAC1. Set to 0 by the user to power down DAC1.
0	$\overline{PD0}$	DAC0 Power-Down Bit. Set to 1 by the user to power up DAC0. Set to 0 by the user to power down DAC0.

[1]For correct DAC operation on the 2.5 V to V_{REF} range, the ADC must be powered on.

DACxH/L	DAC0 and DAC1 Data Registers
Function	DAC Data Registers, written by the user to update the DAC outputs.
SFR Address	DAC0L (DAC0 data low byte) –> F9H DAC0H (DAC0 data high byte) –> FAH; DAC1L (DAC1 data low byte) –> FBH DAC1H (DAC1 data high byte) –> FCH
Power-On Default	00H –> Both DAC0 and DAC1 data registers.
Bit Addressable	No –> Both DAC0 and DAC1 data registers.

The 12-bit DAC data should be written into DACxH/L right-justified such that DACL contains the lower eight bits, and the lower nibble of DACH contains the upper four bits.

Using the DACs

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 38. Features of this architecture include inherent guaranteed monotonicity and excellent differential linearity.

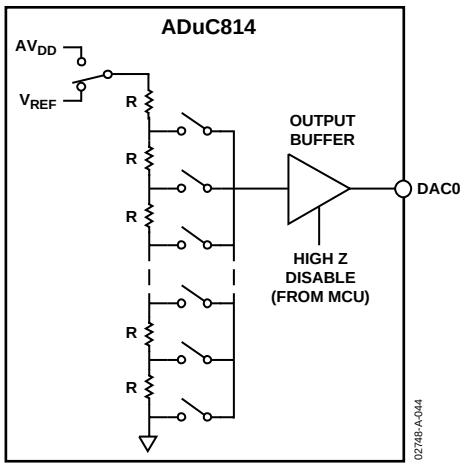

Figure 38. Resistor String DAC Functional Equivalent

As illustrated in Figure 38, 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 features a true rail-to-rail output stage implementation. This means that, unloaded, each output is capable of swinging to within less than 100 mV of both AV_{DD} and ground. Moreover, the DAC's linearity specification (when driving a 10 kΩ resistive load to ground) is guaranteed through the full transfer function except Codes 0 to 48, and, in 0 V-to-AV_{DD} mode only, Codes 3945 to 4095. Linearity degradation near ground and V_{DD} is caused by saturation of the output buffer, and a general representation of its effects (neglecting offset and gain error) is illustrated in Figure 39. The dotted line in Figure 39 indicates the ideal transfer function, and the solid line represents what the transfer function might look like with endpoint nonlinearities due to saturation of the output buffer. Note that Figure 39 represents a transfer function in 0 V-to-V_{DD} mode only. In 0 V-to-V_{REF} mode (with $V_{REF} < V_{DD}$), the lower nonlinearity would be similar, but the upper portion of the transfer function would follow the ideal line right to the end (V_{REF} in this case, not V_{DD}), showing no signs of upper endpoint linearity error.

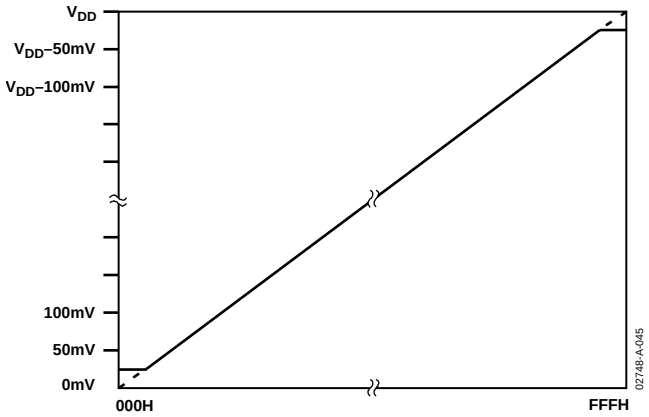

Figure 39. Endpoint Nonlinearities Due to Amplifier Saturation

The endpoint nonlinearities conceptually illustrated in Figure 39 get worse as a function of output loading. Most ADuC814 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 39 become larger. With larger current demands, this can significantly limit output voltage swing. Figure 40 and Figure 41 illustrate this behavior. Note that the upper trace in each of these figures is valid only 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. But around 7 mA, the upper curve in Figure 41 drops below 2.5 V (V_{REF}), indicating that at these higher currents the output cannot reach V_{REF}.

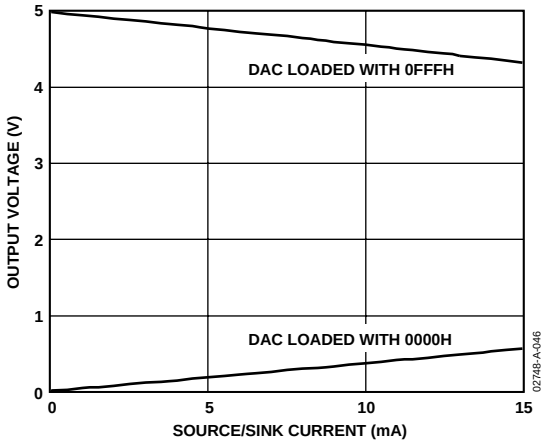

Figure 40. Source and Sink Current Capability with $V_{REF} = V_{DD} = 5$ V

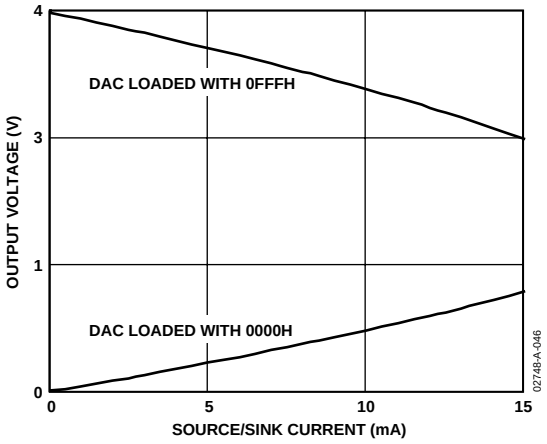

Figure 41. Source and Sink Current Capability with $V_{REF} = V_{DD} = 3$ V

For larger loads, the current drive capability may not be sufficient. To increase the source and sink current capability of the DACs, an external buffer should be added, as shown in Figure 42.

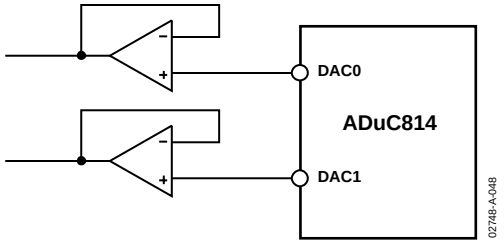

Figure 42. Buffering the DAC Outputs

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.

ON-CHIP PLL

The ADuC814 is intended for use with a 32.768 kHz watch crystal. An on-board PLL locks onto a multiple (512) of this 32.768kHz frequency to provide a stable 16.777216 MHz clock for the system. The core can operate at this frequency or at binary submultiples of it to allow power saving in cases where maximum core performance is not required. The default core clock is the PLL clock divided by 8 ($2^{CD} = 2^3$) or 2.097152 MHz. The PLL is controlled via the PLLCON special function register.

PLLCON	**PLL Control Register**
SFR Address	D7H
Power-On Default	03H
Bit Addressable	No

OSC_PD	LOCK	---	---	FINT	CD2	CD1	CD0

Table 13. PLLCON SFR Bit Designations

Bit No.	Name	Description			
7	OSC_PD	Oscillator Power-Down Bit.			
		Set by the user to halt the 32 kHz oscillator in power-down mode.			
		Cleared by the user to enable the 32 kHz oscillator in power-down mode. This feature allows the oscillator to continue clocking the TIC even in power-down mode.			
6	LOCK	PLL Lock Bit. This is a read-only bit.			
		Set automatically at power-on to indicate that the PLL loop is correctly tracking the crystal clock. If the external crystal becomes subsequently disconnected, the PLL rails and the core halts.			
		Cleared automatically at power-on to indicate that the PLL is not correctly tracking the crystal clock. This may be due to the absence of a crystal clock or an external crystal at power-on. In this mode, the PLL output is 16.78 MHz ± 20%.			
5	---	Reserved. Should be written with 0.			
4	---	Reserved. Should be written with 0.			
3	FINT	Fast Interrupt Response Bit.			
		Set by the user to enable the response to any interrupt to be executed at the fastest core clock frequency, regardless of the configuration of the CD2–CD0 bits (see below). Once user code has returned from an interrupt, the core resumes code execution at the core clock selected by the CD2–CD0 bits.			
		Cleared by the user to disable the fast interrupt response feature.			
2	CD2	CPU (Core Clock) Divider Bits.			
1	CD1	This number determines the frequency at which the microcontroller core operates.			
0	CD0	CD2	CD1	CD0	Core Clock Frequency (MHz)
		0	0	0	16.777216
		0	0	1	8.388608
		0	1	0	4.194304
		0	1	1	2.097152 (Default Core Clock Frequency)
		1	0	0	1.048576
		1	0	1	0.524288
		1	1	0	0.262144
		1	1	1	0.131072

8051 COMPATIBLE ON-CHIP PERIPHERALS

This section gives a brief overview of the various secondary peripheral circuits that are available to the user on-chip. These functions are fully 8051 compatible and are controlled via standard 8051 SFR bit definitions.

Parallel I/O Ports 1 and 3

The ADuC814 has two input/output ports. In addition to performing general-purpose I/O, some ports are multiplexed with an alternate function for the peripheral features on the device. In general, when a peripheral is enabled, that pin may not be used as a general purpose I/O pin.

Port 1 is an 8-bit port directly controlled via the P1 SFR (SFR address = 90H). The Port 1 pins are divided into two distinct pin groupings.

P1.0 and P1.1 pins on Port 1 are bidirectional digital I/O pins with internal pull-ups. If P1.0 and P1.1 have 1s written to them via the P1 SFR, these pins are pulled high by the internal pull-up resistors. In this state they can also be used as inputs. As input pins being externally pulled low, they source current because of the internal pull-ups. With 0s written to them, both of these pins drive a logic low output voltage (VOL) and are capable of sinking 10 mA compared to the standard 1.6 mA sink capability on the other port pins. These pins also have various secondary functions described in Table 20.

Table 20. Port 1, Alternate Pin Functions

Pin No.	Alternate Function
P1.0	T2 (Timer/Counter 2 External Input)
P1.1	T2EX (Timer/Counter 2 Capture/Reload Trigger)

The remaining Port 1 pins (P1.2–P1.7) can be configured only as analog input (ADC), analog output (DAC) or digital input pins. By default (power-on) 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, the user should write a 0 to these port bits to configure the corresponding pin as a high impedance digital input.

Port 3 is a bidirectional port with internal pull-ups directly controlled via the P3 SFR (SFR address = B0H). Port 3 pins that have 1s written to them are pulled high by the internal pull-ups and in that state, they can be used as inputs. As inputs, Port 3 pins being pulled externally low sources current because of the internal pull-ups. Port 3 pins also have various secondary functions described in Table 21.

Table 21. Port 3, Alternate Pin Functions

Pin No.	Alternate Function
P3.0	RxD (UART Input Pin)
	(or Serial Data I/O in Mode 0)
P3.1	TxD (UART Output Pin)
	(or Serial Clock Output in Mode 0)
P3.2	INT0 (External Interrupt 0)
P3.3	INT1 (External Interrupt 1)
P3.4	T0 (Timer/Counter 0 External Input)
P3.5	T1 (Timer/Counter 1 External Input)
	$\overline{SS}$ (Slave Select in SPI Slave Mode)
P3.6	MISO (Master in Slave Out in SPI Mode)
P3.7	MOSI (Master Out Slave In in SPI Mode)

Additional Digital Outputs Pins

Pins P1.0 and P1.1 can be used to provide high current (10 mA sink) general-purpose I/O.

TIMERS/COUNTERS

The ADuC814 has three 16-bit timer/counters: Timer 0, Timer 1, and Timer 2. The timer/counter hardware has been included on-chip to relieve the processor core of the overhead inherent in implementing timer/counter functionality in software. Each timer/counter consists of two 8-bit registers, THx and TLx (x = 0, 1 and 2). All three can be configured to operate either as timers or event counters.

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

In counter function, the TLx register is incremented by a 1-to-0 transition at its corresponding external input pin, T0, T1, or T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since it takes two machine cycles (16 core clock periods) to recognize a 1-to-0 transition, the maximum count rate is 1/16 of the core clock frequency. There are no restrictions on the duty cycle of the external input signal, but to ensure that a given level is sampled at least once before it changes, it must be held for a minimum of one full machine cycle. Remember that the core clock frequency is programmed via the CD0–CD2 selection bits in the PLLCON SFR. User configuration and control of all timer operating modes is achieved via three SFRs: TMOD, TCON, and T2CON.

TMOD	**Timer/Counter 0 and 1 Mode Register**
SFR Address	89H
Power-On Default	00H
Bit Addressable	No

GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

Table 22. TMOD SFR Bit Designations

Bit	Name	Description
7	GATE	Timer 1 Gating Control. Set by software to enable Timer/Counter 1 only while $\overline{INT1}$ pin is high and TR1 control bit is set. Cleared by software to enable Timer 1 whenever TR1 control bit is set.
6	C/$\overline{T}$	Timer 1 Timer or Counter Select Bit. Set by software to select counter operation (input from T1 pin). Cleared by software to select timer operation (input from internal system clock).
5	M1	Timer 1 Mode Select Bit 1 (Used with M0 Bit).
4	M0	Timer 1 Mode Select Bit 0. M1 M0 0 0 TH1 operates as an 8-bit timer/counter. TL1 serves as 5-bit prescaler. 0 1 16-Bit Timer/Counter. TH1 and TL1 are cascaded; there is no prescaler. 1 0 8-Bit Auto-Reload Timer/Counter. TH1 holds a value which is to be reloaded into TL1 each time it overflows. 1 1 Timer/Counter 1 Stopped.
3	Gate	Timer 0 Gating Control. Set by software to enable Timer/Counter 0 only while $\overline{INT0}$ pin is high and the TR0 control bit is set. Cleared by software to enable Timer 0 whenever the TR0 control bit is set.
2	C/$\overline{T}$	Timer 0 Timer or Counter Select Bit. Set by software to select counter operation (input from T0 pin). Cleared by software to select timer operation (input from internal system clock).
1	M1	Timer 0 Mode Select Bit 1.
0	M0	Timer 0 Mode Select Bit 0. M1 M0 0 0 TH0 operates as an 8-bit timer/counter. TL0 serves as 5-bit prescaler. 0 1 16-Bit Timer/Counter. TH0 and TL0 are cascaded; there is no prescaler. 1 0 8-Bit Autoreload Timer/Counter. TH0 holds a value which is to be reloaded into TL0 each time it overflows. 1 1 TL0 is an 8-bit timer/counter controlled by the standard Timer 0 control bits.

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

TF1	TR1	TF0	TR0	IE1[1]	IT1[1]	IE0	IT0[1]

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

Table 23. TCON SFR Bit Designations

Bit No.	Name	Description
7	TF1	Timer 1 Overflow Flag. Set by hardware on a Timer/Counter 1 overflow. Cleared by hardware when the program counter (PC) vectors to the interrupt service routine.
6	TR1	Timer 1 Run Control Bit. Set by the user to turn on Timer/Counter 1. Cleared by the user to turn off Timer/Counter 1.
5	TF0	Timer 0 Overflow Flag. Set by hardware on a Timer/Counter 0 overflow. Cleared by hardware when the PC vectors to the interrupt service routine.
4	TR0	Timer 0 Run Control Bit. Set by the user to turn on Timer/Counter 0. Cleared by the user to turn off Timer/Counter 0.
3	IE1	External Interrupt 1 ($\overline{INT1}$) Flag. Set by hardware by a falling edge or zero level being applied to external interrupt pin $\overline{INT1}$, depending on bit IT1 state. Cleared by hardware when the when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
2	IT1	External Interrupt 1 (IE1) Trigger Type. Set by software to specify edge-sensitive detection, that is, 1-to-0 transition. Cleared by software to specify level-sensitive detection, that is, zero level.
1	IE0	External Interrupt 0 ($\overline{INT0}$) Flag. Set by hardware by a falling edge or zero level being applied to external interrupt pin $\overline{INT0}$, depending on bit IT0 state. Cleared by hardware when the PC vectors to the interrupt service routine, but only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
0	IT0	External Interrupt 0 (IE0) Trigger Type. Set by software to specify edge-sensitive detection, that is, 1-to-0 transition. Cleared by software to specify level-sensitive detection, that is, zero level.

Timer/Counter 0 and 1 Data Registers

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

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

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

TIMER/COUNTER 0 AND 1 OPERATING MODES

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

Mode 0 (13-Bit Timer/Counter)

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

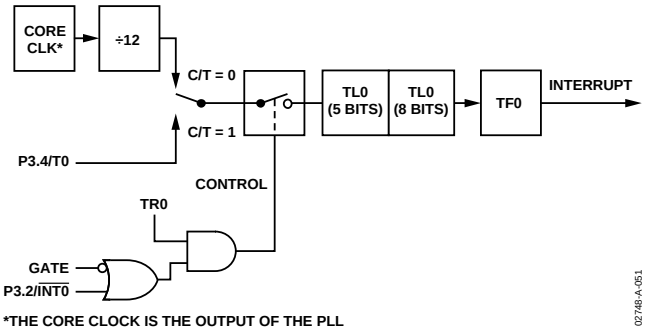

Figure 45. Timer/Counter 0, Mode 0

In this mode, the timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the timer overflow flag TF0. The overflow flag, TF0, can then be used to request an interrupt. The counted input is enabled to the timer when TR0 = 1 and either Gate = 0 or $\overline{INT0}$ = 1.

Setting Gate = 1 allows the timer to be controlled by external input $\overline{INT0}$ to facilitate pulse width measurements. TR0 is a control bit in the special function register TCON; Gate is in TMOD. The 13-bit register consists of all eight bits of TH0 and the lower five bits of TL0. The upper three bits of TL0 are indeterminate and should be ignored. Setting the run flag (TR0) does not clear the registers.

Mode 1 (16-Bit Timer/Counter)

Mode 1 is the same as Mode 0, except that the timer register is running with all 16 bits. Mode 1 is shown in Figure 46.

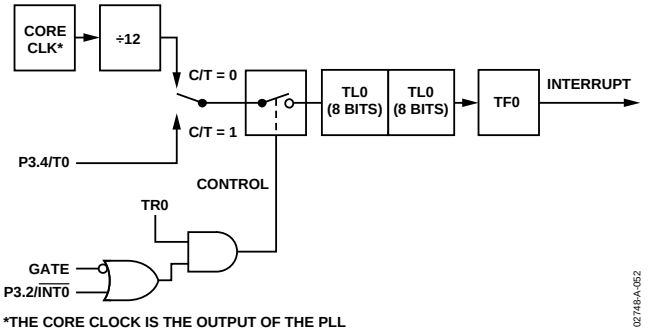

Figure 46. Timer/Counter 0, Mode 1

Mode 2 (8-Bit Timer/Counter with Autoreload)

Mode 2 configures the timer register as an 8-bit counter (TL0) with automatic reload, as shown in Figure 47. Overflow from TL0 not only sets TF0, but also reloads TL0 with the contents of TH0, which is preset by software. The reload leaves TH0 unchanged.

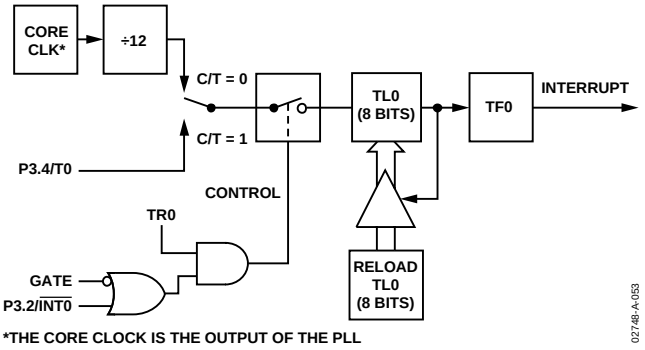

Figure 47. Timer/Counter 0, Mode 2

Mode 3 (Two 8-Bit Timer/Counters)

Mode 3 has different effects on Timer 0 and Timer 1. Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0. Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. This configuration is shown in Figure 48. TL0 uses the Timer 0 control bits: C/T, GATE, TR0, $\overline{INT0}$, and TF0. TH0 is locked into a timer function (counting machine cycles) and takes over the use of TR1 and TF1 from Timer 1. Thus, TH0 controls the Timer 1 interrupt. Mode 3 is provided for applications requiring an extra 8-bit timer or counter. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of, and into, its own Mode 3, or can still be used by the serial interface as a baud rate generator. In fact, it can be used in any application not requiring an interrupt from Timer 1 itself.

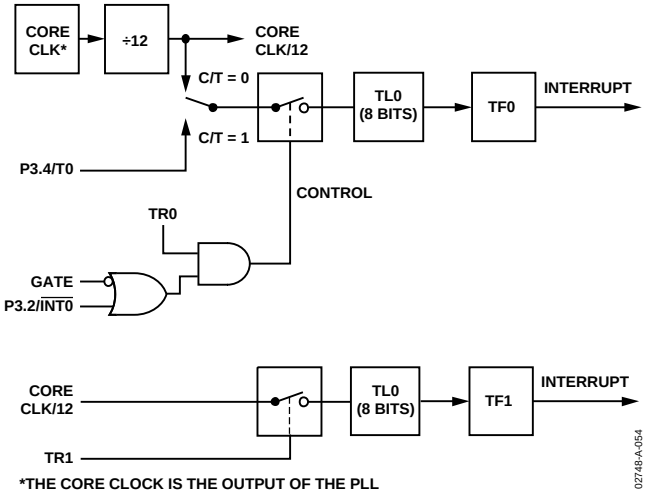

Figure 48. Timer/Counter 0, Mode 3

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

---	PADC	PT2	PS	PT1	PX1	PT0	PX0

Table 30. IP SFR Bit Designations

Bit No.	Name	Description
7	---	Reserved.
6	PADC	ADC Interrupt Priority. Written to by user to set interrupt priority level (1 = High; 0 = Low).
5	PT2	Timer 2 Interrupt Priority. Written to by the user to set interrupt priority level (1 = High; 0 = Low).
4	PS	UART Serial Port Interrupt Priority. Written to by the user to set interrupt priority level (1 = High; 0 = Low).
3	PT1	Timer 1 Interrupt Priority. Written to by the user to set interrupt priority level (1 = High; 0 = Low).
2	PX1	External Interrupt 1 Priority (INT1). Written to by the user to set interrupt priority level (1 = High; 0 = Low).
1	PT0	Timer 0 Interrupt Priority. Written to by the user to set interrupt priority level (1 = High; 0 = Low).
0	PX0	External Interrupt 0 Priority (INT0). Written to by the user to set interrupt priority level (1 = High; 0 = Low).

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

---	PT1	PPSM	PSI	---	ETI	EPSM	ESI

Table 31. IEIP2 SFR Bit Designations

Bit No.	Name	Description
7	---	Reserved.
6	PTI	Time Interval Counter Interrupt Priority. Written to by the user to set TIC interrupt priority (1 = High; 0 = Low).
5	PPSM	PSM Interrupt Priority. Written to by the user to select power supply monitor interrupt priority (1 = High; 0 = Low).
4	PSI	SPI Serial Port Interrupt Priority. Written to by the user to select SPI serial port interrupt priority (1 = High; 0 = Low).
3	---	Reserved. This bit must be 0.
2	ETI	TIC Interrupt. Set by the user to enable the TIC interrupt. Cleared by the user to disable the TIC interrupt.
1	EPSM	Power Supply Monitor Interrupt. Set by the user to enable the power supply monitor interrupt. Cleared by the user to disable the power supply monitor interrupt.
0	ESI	SPI Serial Port Interrupt. Set by the user to enable the SPI serial port interrupt. Cleared by the user to disable the SPI serial port interrupt.

Interrupt Priority

The interrupt enable registers are written by the user to enable individual interrupt sources, while the interrupt priority registers allow the user to select one of two priority levels for each interrupt. An interrupt of a high priority may interrupt the service routine of a low priority interrupt, and if two interrupts of different priority occur at the same time, the higher level interrupt is serviced first. An interrupt cannot be interrupted by another interrupt of the same priority level. If two interrupts of the same priority level occur simultaneously, a polling sequence is observed as shown in Table 32.

Table 32. Priority within an Interrupt Level

Source	Priority	Description
PSMI	1 (Highest)	Power Supply Monitor Interrupt
WDS	2	Watchdog Interrupt
IE0	3	External Interrupt 0
RDY0/RDY1	4	ADC Interrupt
TF0	5	Timer/Counter 0 Interrupt
IE1	6	External Interrupt 1
TF1	7	Timer/Counter 1 Interrupt
ISPI	8	SPI Interrupt
RI + TI	9	Serial Interrupt
TF2 + EXF2	10	Timer/Counter 2 Interrupt
TII	11 (Lowest)	Time Interval Counter Interrupt

Interrupt Vectors

When an interrupt occurs, the program counter is pushed onto the stack, and the corresponding interrupt vector address is loaded into the program counter. The interrupt vector addresses are shown in Table 33.

Table 33. Interrupt Vector Addresses

Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
RI + TI	0023H
TF2 + EXF2	002BH
RDY0/RDY1 (ADC)	0033H
ISPI	003BH
PSMI	0043H
TII	0053H
WDS (WDIR = 1)[1]	005BH

[1] The watchdog can be configured to generate an interrupt instead of a reset when it times out. This is used for logging errors or to examine the internal status of the microcontroller core to understand, from a software debug point of view, why a watchdog timeout occurred. The watchdog interrupt is slightly different from normal interrupts in that its priority level is always set to 1, and it is not possible to disable the interrupt via the global disable bit (EA) in the IE SFR. This is done to ensure that the interrupt is always responded to if a watchdog timeout occurs. The watchdog produces an interrupt only if the watchdog timeout is greater than zero.

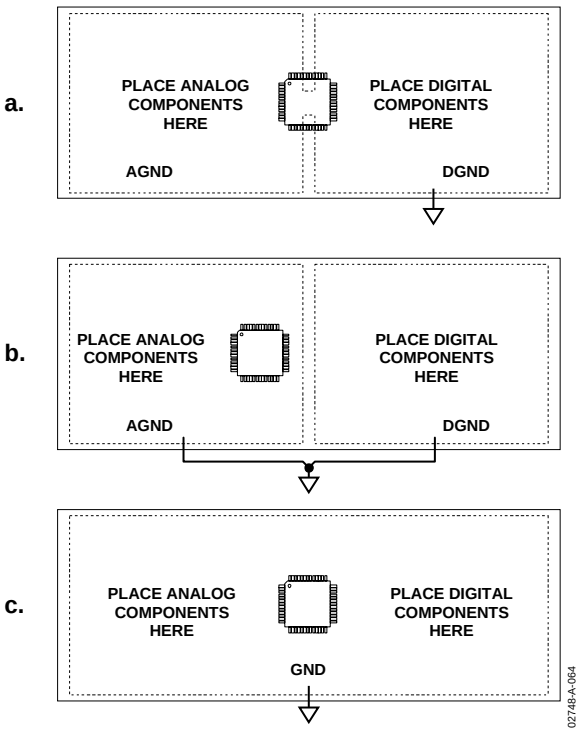

Figure 58. System Grounding Schemes

In all of these scenarios, and in more complicated real-life applications, keep in mind the flow of current from the supplies and back to ground. Make sure the return paths for all currents are as close as possible to the paths the currents took to reach their destinations. For example, do not put power components on the analog side of Figure 58b with DV_{DD} because that would force return currents from DV_{DD} to flow through AGND. Also, try to avoid digital currents flowing under analog circuitry, which could happen if a noisy digital chip is placed on the left half of the board in Figure 58c. Whenever possible, avoid large discontinuities in the ground plane(s) (such as are formed by a long trace on the same layer), because they force return signals to travel a longer path. And of course, make all connections to the ground plane directly, with little or no trace separating the pin from its via to ground.

If the user plans to connect fast logic signals (rise/fall time < 5 ns) to any of the ADuC814's digital inputs, add a series resistor to each relevant line to keep rise and fall times longer than 5 ns at the ADuC814 input pins. A value of 100 Ω or 200 Ω is usually sufficient to prevent high speed signals from coupling capacitively into the ADuC814 and affecting the accuracy of ADC conversions.

OTHER HARDWARE CONSIDERATIONS

To facilitate in-circuit programming, in-circuit debug, and emulation options, users should implement some simple connection points in their hardware. A typical ADuC814 connection diagram is shown in Figure 59.

In-Circuit Serial Download Access

Nearly all ADuC814 designs will want to take advantage of the in-circuit reprogrammability of the chip. This is accomplished by a connection from the ADuC814's UART to a PC, which requires an external RS-232 chip for level translation. If users would rather not design an RS-232 chip onto a board, refer to the Application Note uC006, *A 4-Wire UART-to-PC Interface* (available at www.analog.com/microconverter) for a simple (and zero-cost-per-board) method of gaining in-circuit serial download access to the ADuC814.

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-up resistor that can be jumpered onto the DLOAD pin. To get the ADuC814 into download mode, simply connect this jumper and power-cycle the device (or manually reset the device, if a manual reset button is available), and it will be ready to receive a new program serially. To enable the device to enter normal mode (and run the program) whenever power is cycled or RESET is toggled, the DLOAD pin must be pulled low through a 1 kΩ resistor.

Embedded Serial Port Debugger

From a hardware perspective, entry to serial port debug mode is identical to the serial download entry sequence described in the preceding 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 ADuC814 device, unlike ROM monitor type debuggers.

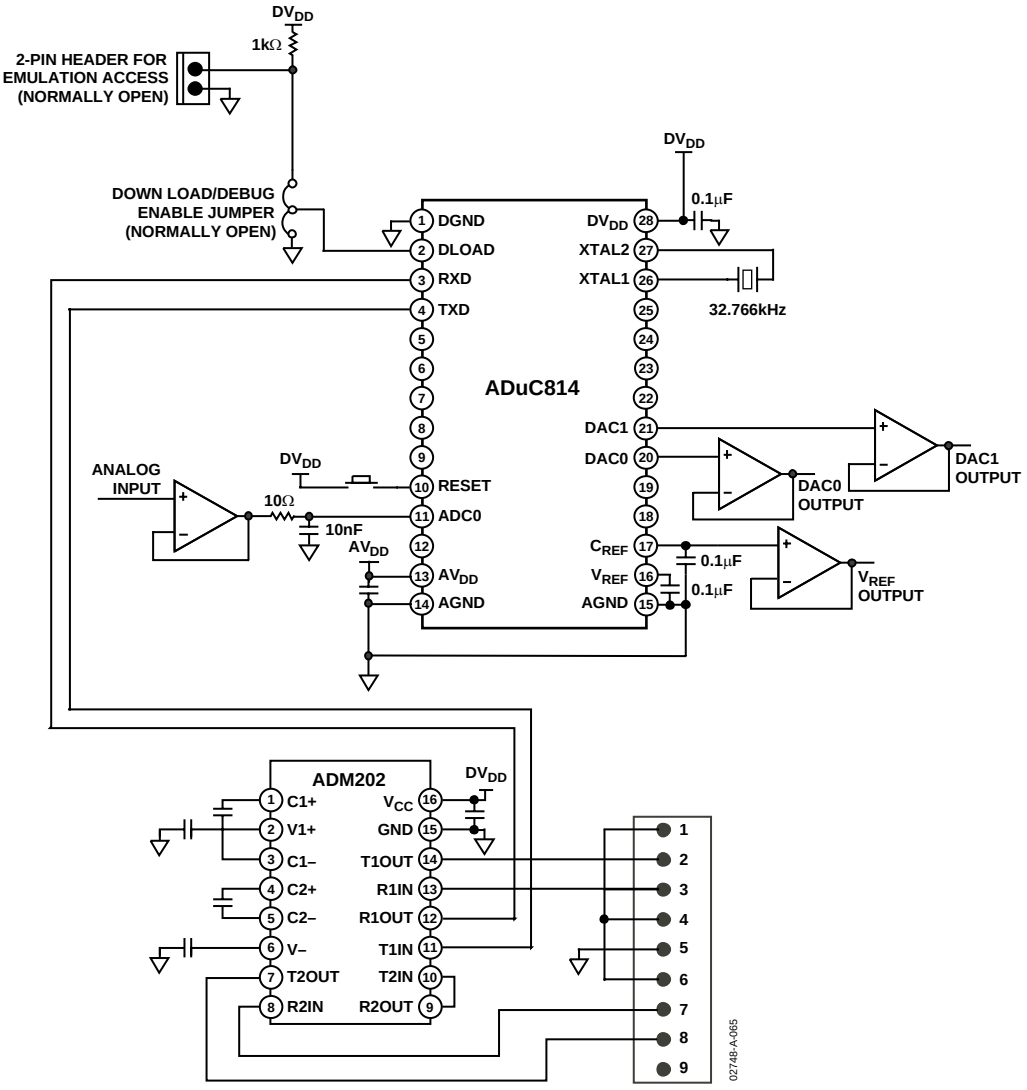

Figure 59. Typical ADuC814 System Connection Diagram

Single-Pin Emulation Mode

Also built into the ADuC814 is a dedicated controller for single-pin in-circuit emulation (ICE) using standard production ADuC814 devices. In this mode, emulation access is gained by connection to a single pin, again the DLOAD pin is used for this function. As described previously, this pin is either high to enable entry into serial download and serial debug modes or low to select normal code execution. To enable single-pin emulation mode, however, users need to pull the DLOAD pin high through a 1 kΩ resistor. The emulator then connects to the 2-pin header. To be compatible with the standard connector that comes with the single-pin emulator available from Accutron Limited (www.accutron.com), use a 2-pin 0.1-inch pitch friction lock header from Molex (www.molex.com) such as their part number 22-27-2021. Be sure to observe the polarity of this header. When the friction lock tab is at the right, the ground pin should be the lower of the two pins (when viewed from the top).

TIMING SPECIFICATIONS[1,2,3]

Table 34. Clock Input (External Clock Driven XTAL1)

AV_{DD} = 2.7 V to 3.3 V or 4.75 V to 5.25 V, DV_{DD} = 2.7 V to 3.3 V or 4.75 V to 5.25 V; all specifications T_{MIN} to T_{MAX}, unless otherwise noted

		32.768 kHz External Crystal			
Parameter		**Min**	**Typ**	**Max**	**Unit**
t_{CK}	XTAL1 Period		30.52		µs
t_{CKL}	XTAL1 Width Low		15.16		µs
t_{CKH}	XTAL1 Width High		15.16		µs
t_{CKR}	XTAL1 Rise Time		20		ns
t_{CKF}	XTAL1 Fall Time		20		ns
$1/t_{CORE}$	ADuC814 Core Clock Frequency[4]	0.131		16.78	MHz
t_{CORE}	ADuC814 Core Clock Period[5]		0.476		µs
t_{CYC}	ADuC814 Machine Cycle Time[6]	0.72	5.7	91.55	µs

[1] AC inputs during testing are driven at DV_{DD} – 0.5 V for a Logic 1, and at 0.45 V for a Logic 0. Timing measurements are made at V_{IH} min for a Logic 1, and at V_{IL} max for a Logic 0 as shown in Figure 61.
[2] For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs as shown in Figure 61.
[3] C_{LOAD} for all outputs = 80 pF, unless otherwise noted.
[4] ADuC814 internal PLL locks onto a multiple (512 times) the external crystal frequency of 32.768 kHz to provide a stable 16.777216 MHz internal clock for the system. The core can operate at this frequency or at a binary submultiple called Core_Clk, selected via the PLLCON SFR.
[5] This number is measured at the default Core_Clk operating frequency of 2.09 MHz.
[6] ADuC814 Machine Cycle Time is nominally defined as 12/Core_CLK.

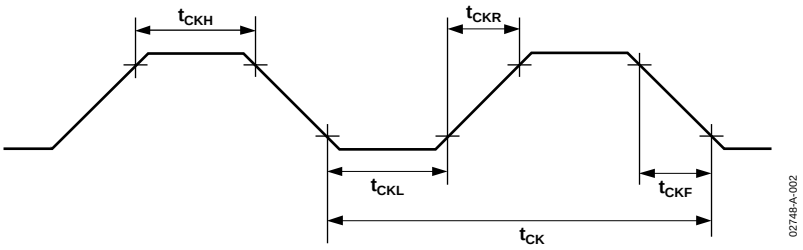

Figure 60. XTAL1 Input

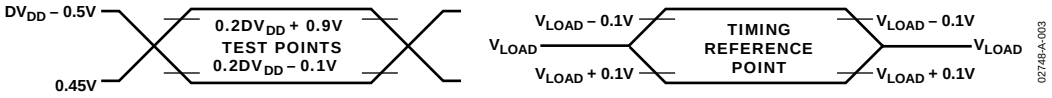

Figure 61. Timing Waveform Characteristics

Table 35. UART Timing (Shift Register Mode)

Parameter		16.78 MHz Core_Clk			Variable Core_Clk			Unit
		Min	Typ	Max	Min	Typ	Max	
t_{XLXL}	Serial Port Clock Cycle Time		715			12 t_{CORE}		µs
t_{QVXH}	Output Data Setup to Clock	463			10 t_{CORE}	−133		ns
t_{DVXH}	Input Data Setup to Clock	252			2 t_{CORE}	+133		ns
t_{XHDX}	Input Data Hold after Clock	0			0			ns
t_{XHQX}	Output Data Hold after Clock	22			2 t_{CORE}	−117		ns

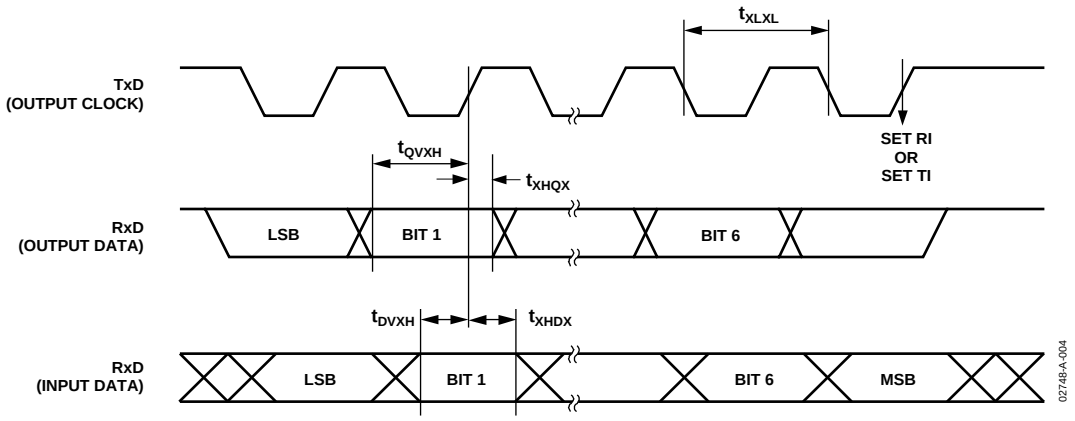

Figure 62. UART Timing in Shift Register Mode

Table 37. SPI Master Mode Timing (CPHA = 0)

Parameter		Min	Typ	Max	Unit
t_{SL}	SCLOCK Low Pulse Width[1]		630		ns
t_{SH}	SCLOCK High Pulse Width[1]		630		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] Characterized under the following conditions:
a. Core clock divider bits CD2, CD1, and CD0 bits in PLLCON SFR set to 0, 1, and 1, respectively, i.e., core clock frequency = 2.09 MHz.
b. SPI bit-rate selection bits SPR1 and SPR0 bits in SPICON SFR set to 0 and 0, respectively.

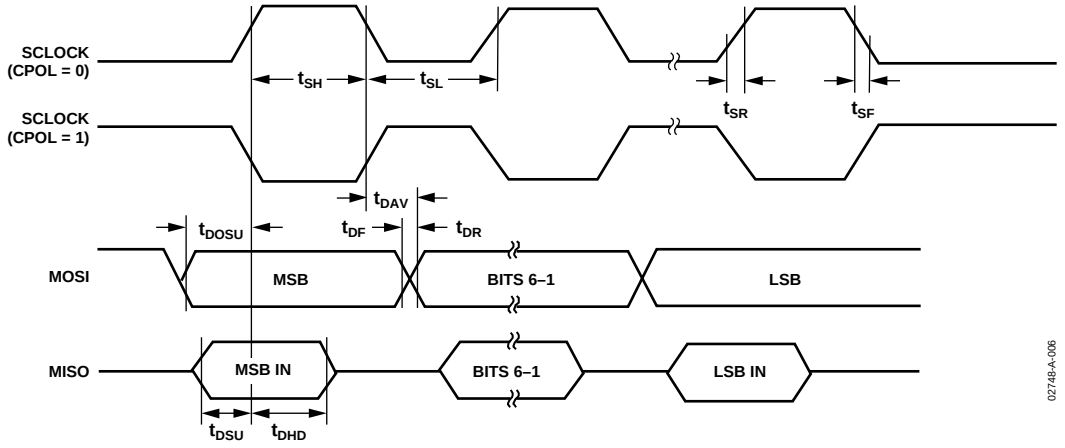

Figure 64. SPI Master Mode Timing (CPHA = 0)

Table 39. SPI Slave Mode Timing (CPHA = 0)

Parameter		Min	Typ	Max	Unit
t_{SS}	$\overline{SS}$ to SCLOCK Edge	0			
t_{SL}	SCLOCK Low Pulse Width		330		ns
t_{SH}	SCLOCK High Pulse Width		330		ns
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns
t_{DSU}	Data Input Setup Time before SCLOCK Edge		100		ns
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns
t_{DF}	Data Output Fall Time		10	25	ns
t_{DR}	Data Output Rise Time		10	25	ns
t_{SR}	SCLOCK Rise Time		10	25	ns
t_{SF}	SCLOCK Fall Time		10	25	ns
t_{SSR}	$\overline{SS}$ to SCLOCK Edge			50	ns
t_{DOSS}	Data Output Valid after $\overline{SS}$ Edge			20	ns
t_{SFS}	$\overline{SS}$ High after SCLOCK Edge	0			ns

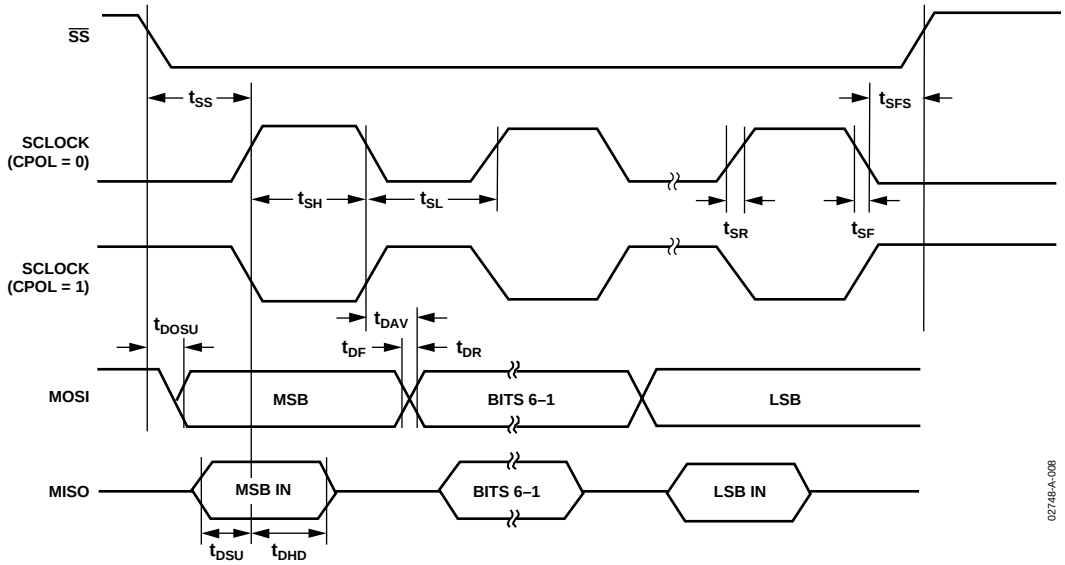

Figure 66. SPI Slave Mode Timing (CPHA = 0)