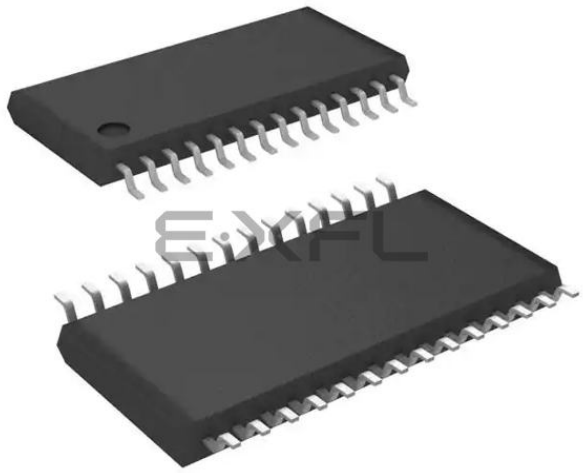

Welcome to E-XFL.COM

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

TABLE OF CONTENTS

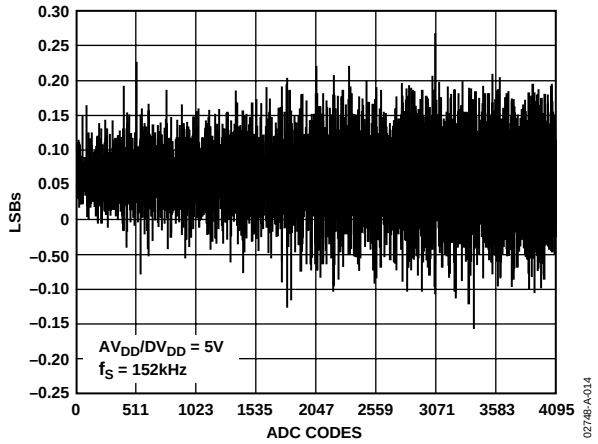

Figure 7. Typical DNL Error, V_{DD} = 5 V

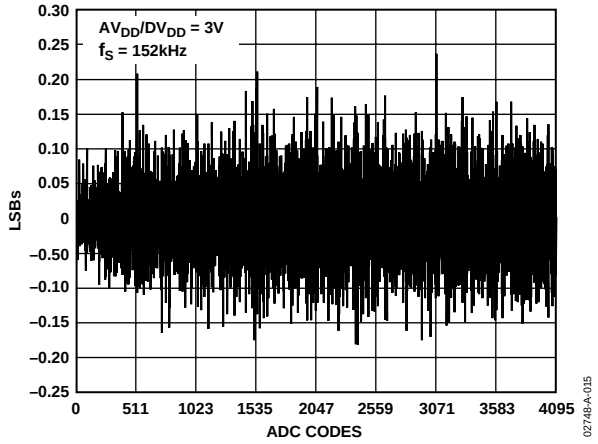

Figure 8. Typical DNL Error, V_{DD} = 3 V

Figure 9 and Figure 10 show the variation in worst-case positive (WCP) DNL and worst-case negative (WCN) DNL versus external reference input voltage.

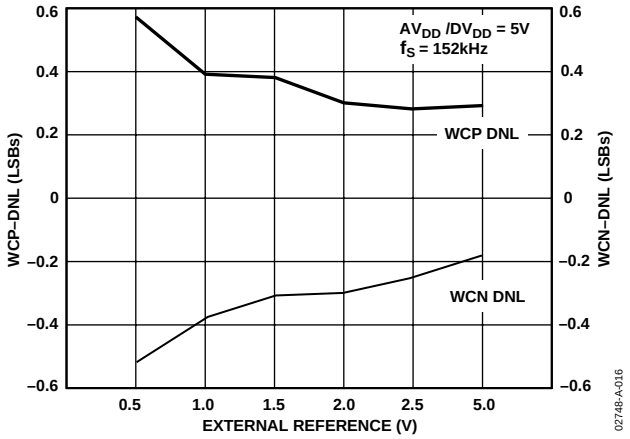

Figure 9. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 5 V

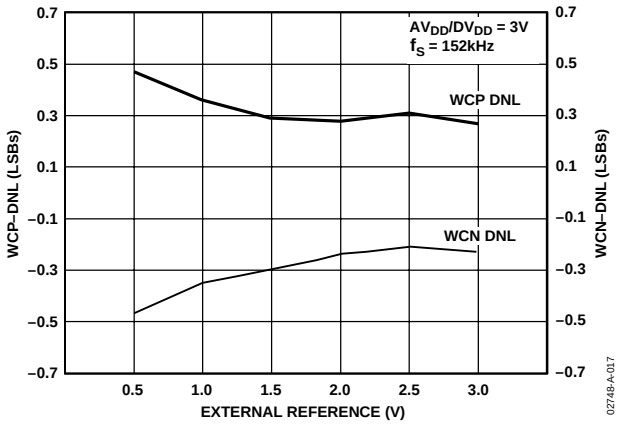

Figure 10. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 3 V

Figure 11 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.

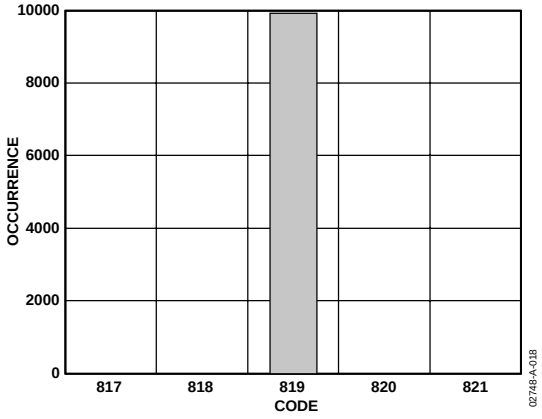

Figure 11. Code Histogram plot, V_{DD} = 5 V

Figure 12 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 bin.

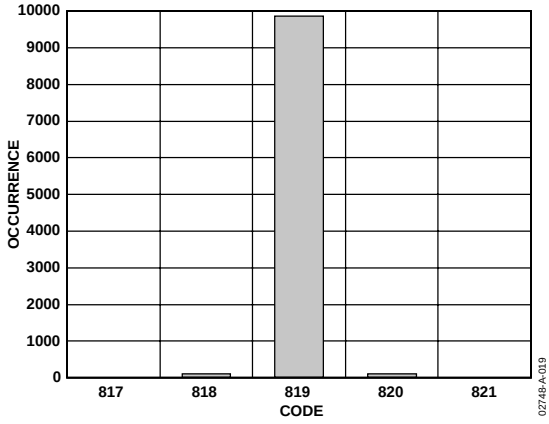

Figure 12. Code Histogram Plot, V_{DD} = 3 V

MEMORY ORGANIZATION

The ADuC814 does not have Port 0 and Port 2 pins and therefore does not support external program or data memory interfaces. The device executes code from the internal 8-kByte Flash/EE program memory. This internal code space can be programmed via the UART serial port interface while the device is in-circuit. The program memory space of the ADuC814 is shown in Figure 18.

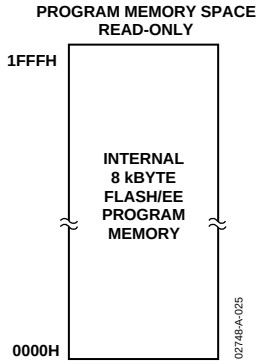

Figure 18. Program Memory Map

The data memory address space consists of internal memory only. The internal memory space is divided into four physically separate and distinct blocks, namely the lower 128 bytes of RAM, the upper 128 bytes of RAM, the 128 bytes of special function register (SFR) area, and a 640-byte Flash/EE data memory. While the upper 128 bytes of RAM and the SFR area share the same address locations, they are accessed through different addressing modes.

The lower 128 bytes of data memory can be accessed through direct or indirect addressing, the upper 128 bytes of RAM can be accessed through indirect addressing, and the SFR area is accessed through direct addressing.

Also, as shown in Figure 19, an additional 640 bytes of Flash/EE data memory are available to the user and can be accessed indirectly via a group of control registers mapped into the SFR area. Access to the Flash/EE data memory is discussed in detail later as part of the Flash/EE Memory section.

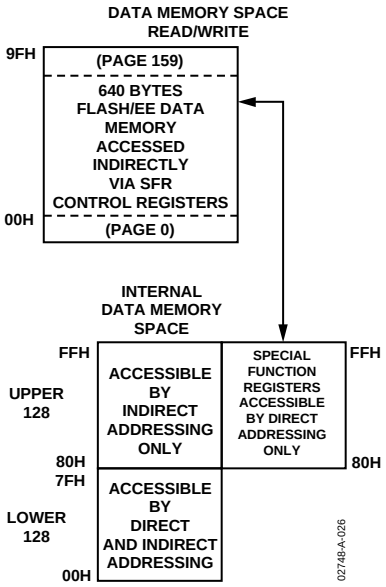

Figure 19. Data Memory Map

The lower 128 bytes of internal data memory are mapped as shown in Figure 20. The lowest 32 bytes are grouped into four banks of eight registers addressed as R0 to R7. The next 16 bytes (128 bits), locations 20H to 2FH above the register banks, form a block of directly addressable bit locations at bit addresses 00H through 7FH. The stack can be located anywhere in the internal memory address space, and the stack depth can be expanded up to 256 bytes.

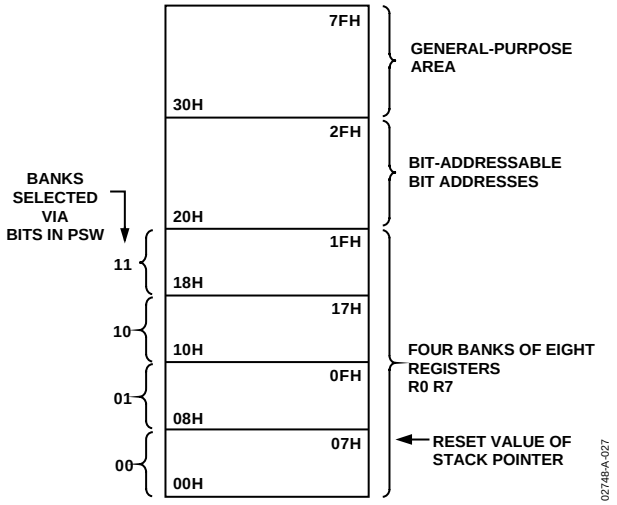

Figure 20. Lower 128 Bytes of Internal Data Memory

RESET initializes the stack pointer to location 07H and increments it once to start from location 08H, which is also the first register (R0) of Register Bank 1. If more than one register bank is being used, the stack pointer should be initialized to an area of RAM not used for data storage.

Power Control SFR

The power control (PCON) register contains bits for power-saving options and general-purpose status flags as shown in Table 5.

SFR Address 87H
Power-On Default 00H
Bit Addressable No

SMOD	SERIPD	INT0PD	---	GF1	GF0	PD	IDL

Table 5. PCON SFR Bit Designations

Bit No.	Name	Description
7	SMOD	Double UART Baud Rate.
6	SERIPD	SPI Power-Down Interrupt Enable.
5	INT0PD	$\overline{\text{INT0}}$ Power-Down Interrupt Enable.
4	RSVD	Reserved.
3	GF1	General-Purpose Flag Bit.
2	GF0	General-Purpose Flag Bit.
1	PD	Power-Down Mode Enable.
0	IDL	Idle Mode Enable.

ADCCON3 (ADC CONTROL SFR 3)

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

SFR Address F5H
SFR Power-On Default 00H

BUSY	GNCLD	AVGS1	AVGS0	OFCLD	MODCAL	TYPECAL	SCAL

Table 8. ADCCON3 SFR Bit Designations

Bit No.	Name	Description
7	BUSY	ADC Busy Status Bit. BUSY is a read-only status bit that is set during a valid ADC conversion or calibration cycle. Busy is automatically cleared by the core at the end of a conversion or calibration cycle.
6	GNCLD	Gain Calibration Disable Bit. This bit enables/disables the gain calibration coefficients from affecting the ADC results. Set to 0 to enable gain calibration coefficient Set to 1 to disable gain calibration coefficient.
5	AVGS1	Number of Averages Selection Bits.
4	AVGS0	This bit selects the number of ADC readings averaged for each bit decision during a calibration cycle. AVGS1 AVGS0 Number of Averages 0 0 15 0 1 1 1 0 31 1 1 63
3	OFCLD	Offset Calibration Disable Bit. This bit enables/disables the offset calibration coefficients from affecting the ADC results. Set to 0 to enable offset calibration coefficient. Set to 1 to disable the offset calibration coefficient
2	MODCAL	Calibration Mode Select Bit. This bit should be set to 1 for all calibration cycles.
1	TYPECAL	Calibration Type Select Bit. This bit selects between offset (zero-scale) and gain (full-scale) calibration. Set to 0 for offset calibration. Set to 1 for gain calibration.
0	SCAL	Start Calibration Cycle Bit. When set, this bit starts the selected calibration cycle. It is automatically cleared when the calibration cycle is completed.

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.

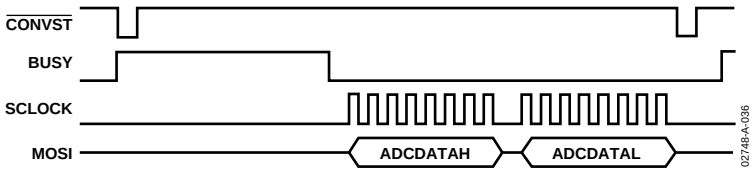

Figure 29. High Speed Data Capture Logic Timing (Non-Pipelined Mode)

gain coefficient compensates for a larger analog input signal range and scales down the ADC transfer function, effectively decreasing the slope of the transfer function. The maximum analog input signal range for which the gain coefficient can compensate is 1.035 × V_{REF}, and the minimum input range is 0.965 × V_{REF}, which equates to typically ±3.5% of the reference voltage.

CALIBRATING THE ADC

The ADuC814 has two hardware calibration modes, device calibration and system calibration, that can be easily initiated by the user software. The ADCCON3 SFR is used to calibrate the ADC. See Table 8.

Device calibration is so called because the relevant signals used for the calibration are available internally to the ADC. This calibration method can be used to compensate for significant changes in operating conditions, such as core frequency, analog input range, reference voltage and supply voltages. In this calibration mode, offset calibration uses internal AGND selected via the ADCCON2 register bits CS3–CS0 (1011), and gain calibration uses internal V_{REF} selected by CS3–CS0 (1100). Offset calibration should be executed first, followed by gain calibration.

System calibration is so called because the AGND and V_{REF} required for calibration must be the system AGND and V_{REF} signals. These must be supplied in turn, externally, to the ADC inputs. This calibration method can be used to compensate for both internal and external system errors. To perform system calibration using an external reference, tie system ground and reference to any two of the six selectable inputs. Enable external reference mode (ADCCON1.6). Select the channel connected to AGND via CS3–CS0 and perform system offset calibration. Select the channel connected to V_{REF} via CS3–CS0 and perform system gain calibration.

INITIATING CALIBRATION IN CODE

When calibrating the ADC, ADCCON1 should be set to the configuration in which the ADC is used. The ADCCON3 register can then be used to configure and execute the offset and gain calibration required in sequence.

Configure the ADC as required. In this case, ADCCLK = /4, acquisition time is set to 1 clock (T_{ACQ}), and ADC is enabled.

```
MOV ADCCON1,#0D0H     ;ADC on, ADCCLK set to
                      ;divide by 4, 1 acquisition
                      ;clock (Tacq)
```

To perform device offset calibration:

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

To perform device gain calibration:

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

To perform system offset calibration:

Connect system AGND to an ADC input (Channel 0 in this case).

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

To perform system gain calibration:

Connect system V_{REF} to an ADC input (Channel 1 in this case).

```
MOV ADCCON2,#01H      ;select external VREF MOV
ADCCON3,#27H          ;select offset calibration,
                      ;31 averages / bit (NUMAV),
                      ;offset calibration
```

The calibration cycle time T_{CAL} is calculated by

$$T_{CAL} = 14 \times ADCCLK \times NUMAV \times (16 + T_{ACQ})$$

For an ADCCLK/F_{CORE}, divide ratio of 4, a T_{ACQ} = 1 ADCCLK, NUMAV = 31, the calibration cycle time is

$$T_{CAL} = 14 \times (1 / 4194304) \times 31 \times (16 + 1)$$

$$T_{CAL} = 1.76\ mS$$

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

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

NONVOLITILE FLASH/EE MEMORY

FLASH/EE MEMORY OVERVIEW

The ADuC814 incorporates Flash/EE memory technology on-chip to provide the user with nonvolatile, in-circuit reprogrammable code and data memory space.

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 32).

Because Flash/EE technology is based on a single transistor cell architecture, a Flash memory array such as 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.

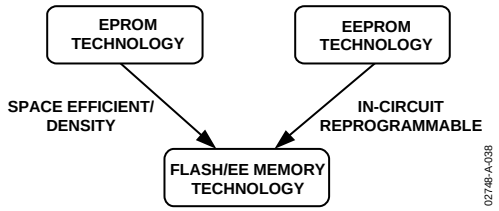

Figure 32. Flash/EE Memory Development

Incorporated in the ADuC814, 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 ADUC814

The ADuC814 provides two arrays of Flash/EE memory for user applications. There are 8 kbytes of Flash/EE program space provided on-chip to facilitate code execution, therefore removing the requirement for an external discrete ROM device. The program memory can be programmed using conventional third-party memory programmers. This array can also be programmed in-circuit, using the serial download mode provided.

A 640-byte Flash/EE data memory space is also provided on-chip. This may be used as a general-purpose nonvolatile scratchpad area. User access to this area is via a group of six SFRs. This space can be programmed at a byte level, although it must first be erased in 4-byte pages.

ADUC814 FLASH/EE MEMORY RELIABILITY

The Flash/EE program and data memory arrays on the ADuC814 are fully qualified for two key Flash/EE memory characteristics: Flash/EE memory cycling endurance and Flash/EE memory data retention.

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

1. Initial page erase sequence
2. Read/verify sequence
3. Byte program sequence
4. 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 tables, the ADuC814 Flash/EE memory endurance qualification has been carried out in accordance with JEDEC Specification A117 over the industrial temperature range of –40°C to +125°C. The results allow the specification of a minimum endurance figure over supply and a temperature of 100,000 cycles, with an endurance figure of 700,000 cycles being typical of operation at 25°C.

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. Again, the ADuC814 has been qualified in accordance with the formal JEDEC Retention Lifetime Specification (A117) at a specific junction temperature (T_J = 55°C). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit described above, before data retention is characterized. This means that the Flash/EE memory is guaranteed to retain its data for its full specified retention lifetime every time the Flash/EE memory is reprogrammed. It should be noted that retention lifetime, based on an activation energy of 0.6 eV, derates with T_J as shown in Figure 33.

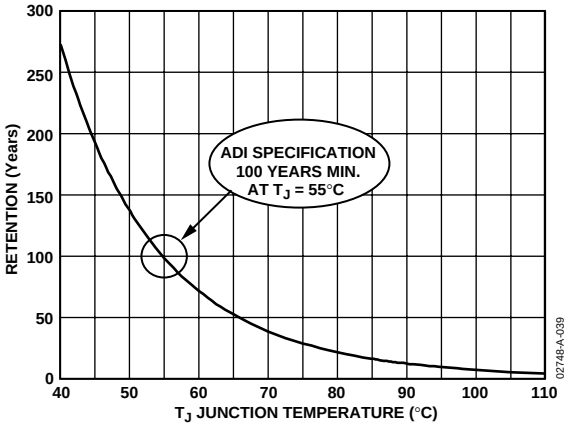

Figure 33. Flash/EE Memory Data Retention

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.

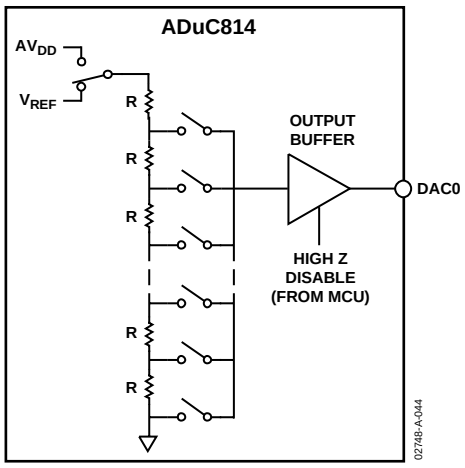

Figure 38. Resistor String DAC Functional Equivalent

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.

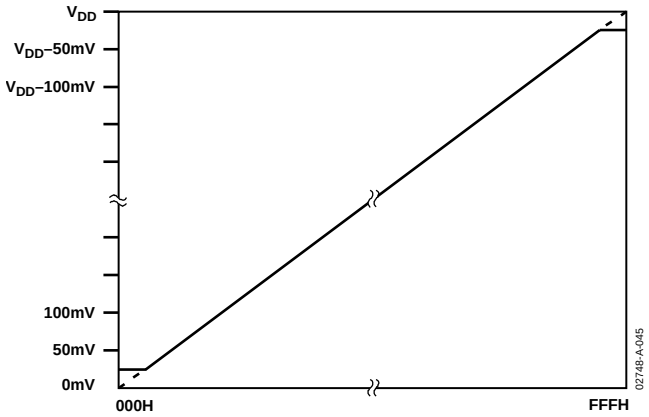

Figure 39. Endpoint Nonlinearities Due to Amplifier Saturation

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}.

ADuC814 CONFIGURATION REGISTER (CFG814)

The ADuC814 is housed in a 28-lead TSSOP package. To maintain as much functional compatibility with other MicroConverter products, some pins share multiple I/O functionality. Switching between these functions is controlled via the ADuC814 configuration SFR, CFG814, located at SFR address 9CH. A summary of these functions is described and a detailed bit designation for the CFG814 SFR is given in Table 17.

Serial Peripheral Interface

The SPI interface on the ADuC814 shares the same pins as digital outputs P3.5, P3.6, and P3.7. The SPE bit in SPICON is used to select which interface is active at any one time. This is described in greater detail in the next section. By default, these pins operate as standard Port 3 pins. Bit 0 of the CFG814 SFR must be set to 1 to enable the SPI interface on these Port 3 pins.

External Clock

The ADuC814 is intended for use with a 32.768 kHz watch crystal. The on-chip PLL locks onto a multiple of this to provide a stable 16.777216 MHz clock for the device. On the ADuC814, P3.5 alternate functions include T1 input and slave select in SPI master mode. P3.5 also functions as external clock input, EXTCLK, selected via Bit 1 of the CFG814 SFR. When selected, this external clock bypasses the PLL and is used as the clock for the device, therefore allowing the ADuC814 to be synchronized to the rest of the application system. The maximum input frequency of this external clock is 16.777216 MHz. If selected, the EXTCLK signal affects the timing of the majority of peripherals on the ADuC814 including the ADC, EEPROM controller, watchdog timer, SPI interface clock, and the MicroConverter core clock.

CFG814	**ADuC814 Configuration Register**
SFR Address	9CH
Power-On Default	04H
Bit Addressable	No

						EXTCLK	SER_EN

Table 17. CFG814 SFR Bit Designations

Bit No.	Name	Description
1	EXTCLK	External Clock Selection Bit. Set to 1 to enable EXTCLK as MCU core clock. Cleared to 0 to enable XTAL and PLL as the MCU core clock.
0	SER_EN	Serial Interface Enable Bit. Set to 1 by the user to enable the SPI interface onto the P3.5, P3.6, and P3.7 pins. Cleared to 0 by the user to enable standard Port 3 functionality on the P3.5, P3.6, and P3.7 pins.

SERIAL PERIPHERAL INTERFACE

The ADuC814 integrates a complete hardware serial peripheral interface (SPI) on-chip. SPI is an industry-standard synchronous serial interface that allows eight bits of data to be synchronously transmitted and received simultaneously, i.e., full duplex. Note that the SPI pins MISO and MOSI are multiplexed with digital outputs P3.6 and P3.7. These pins are controlled via the CFG814.0 bit in the CFG814 SFR (Table 17), which configures the relevant Port 3 pins for normal operation or serial port operation. When the relevant Port 3 pins are configured for serial interface operation via the CFG814 SFR, the SPE bit in the SPICON SFR configures SPI or I^2C operation (see SPE bit description in Table 18). SPI can be configured for master or slave operation, and typically consists of four pins described next.

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

The MISO pin (Pin 23) is configured as an input line in master mode and as an output line in slave mode. The MISO line on the master (data in) should be connected to the MISO line in the slave device (data out). The data is transferred as byte-wide (8-bit) serial data, MSB first.

MOSI (Master Out, Slave In Pin)

The MOSI pin (Pin 24) is configured as an output line in master mode and as 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 SCLOCK pin (Pin 25) 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 18). 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 $\overline{SS}$ input pin (Pin 22) is used only when the ADuC814 is configured in slave mode to enable the SPI peripheral. This line is active low. Data is received or transmitted in slave mode only when the $\overline{SS}$ pin is low, allowing the ADuC814 to be used in single master, multislave SPI configurations. If CPHA = 1, then the $\overline{SS}$ input may be permanently pulled low. With CPHA = 0, the $\overline{SS}$ input must be driven low before the first bit in a byte-wide transmission or reception and return high again after the last bit in that byte-wide transmission or reception. In SPI slave mode, the logic level on the external $\overline{SS}$ pin can be read via the SPR0 bit in the SPICON SFR.

The following SFR registers are used to control the SPI interface.

SPICON	**SPI Control Register**
SFR Address	F8H
Power-On Default	04H
Bit Addressable	Yes

ISPI	WCOL	SPE	SPM	CPOL	CPHA	SPR1	SPR0

Table 18. SPICON SFR Bit Designations

Bit No.	Name	Description
7	ISPI	SPI Interrupt Bit. Set by the MicroConverter at the end of each SPI transfer. Cleared directly by the user code or indirectly by reading the SPIDAT SFR.
6	WCOL	Write Collision Error Bit. Set by the MicroConverter if SPIDAT is written to while an SPI transfer is in progress. Cleared by the user.
5	SPE	SPI Interface Enable Bit. Set by the user to enable the SPI interface. Cleared by the user to enable the I^2C interface.
4	SPIM	SPI Master/Slave Mode Select Bit. Set by the user to enable master mode operation (SCLOCK is an output). Cleared by the user to enable slave mode operation (SCLOCK is an input).
3	CPOL[1]	Clock Polarity Select Bit. Set by the user if SCLOCK idles high. Cleared by the user if SCLOCK idles low.

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 2 OPERATING MODES

This section describes the operating modes for Timer/Counter 2. The operating modes are selected by bits in the T2CON SFR as shown in Table 27.

Table 25. Mode Selection in T2CON

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

16-Bit Autoreload Mode

In autoreload mode, there are two options that are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then when Timer 2 rolls over, it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, then Timer 2 still performs the above, but with the added feature that a 1-to-0 transition at external input T2EX also triggers the 16-bit reload and sets EXF2. The autoreload mode is illustrated in Figure 49.

16-Bit Capture Mode

In the capture mode, there are again two options that are selected by bit EXEN2 in T2CON. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter which, upon overflowing, sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 still performs the above, but a 1-to-0 transition on external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set, and EXF2, like TF2, can generate an interrupt. Capture mode is illustrated in Figure 50.

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1. In either case, if Timer 2 is being used to generate the baud rate, the TF2 interrupt flag does not occur. Therefore Timer 2 interrupts do not occur so they do not have to be disabled. In this mode, the EXF2 flag, however, can still cause interrupts; this can be used as a third external interrupt. Baud rate generation is described as part of the UART Serial Interface section that follows.

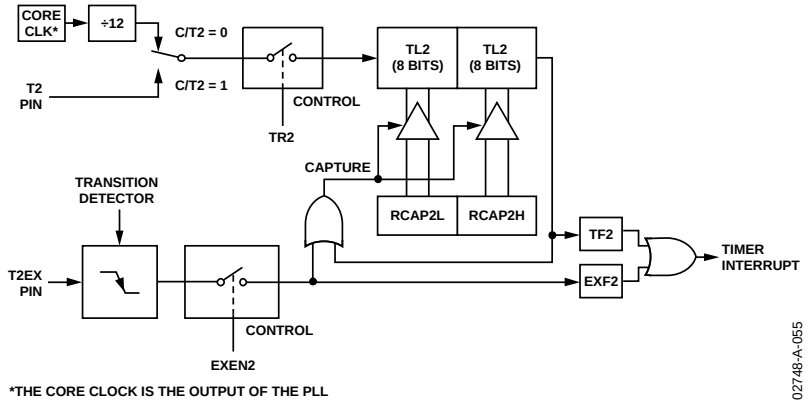

Figure 49. Timer/Counter 2, 16-Bit Autoreload Mode

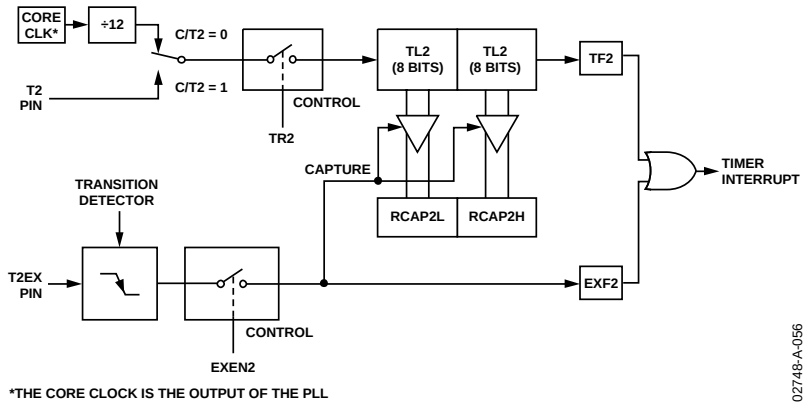

Figure 50. Timer/Counter 2, 16-Bit Capture Mode

Mode 0: 8-Bit Shift Register Mode

Mode 0 is selected by clearing both the SM0 and SM1 bits in the SFR SCON. Serial data enters and exits through RxD. TxD outputs the shift clock. Eight data bits are transmitted or received. Transmission is initiated by any instruction that writes to SBUF. The data is shifted out of the RxD line. The eight bits are transmitted with the least-significant bit (LSB) first, as shown in Figure 52.

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 (Bit 0) and followed by a stop bit (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 51.

Reception is initiated when a 1-to-0 transition is detected on RxD. Assuming a valid start bit was detected, character reception continues. The start bit is skipped and the eight data bits are clocked into the serial port shift register. When all eight bits have been clocked in, the following events occur:

- The eight bits in the receive shift register are latched into SBUF.
- The ninth bit (stop bit) is clocked into RB8 in SCON. The receiver interrupt flag (RI) is set if, and only if, the following conditions are met at the time the final shift pulse is generated:
  - RI = 0
  - Either SM2 = 0 or SM2 = 1 and the received stop bit = 1

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

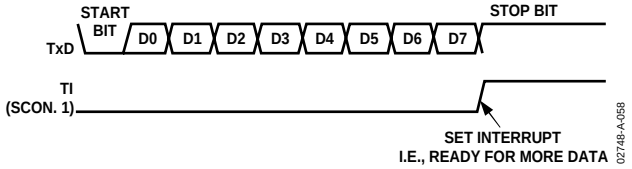

Figure 51. UART Serial Port Transmission, Mode 1

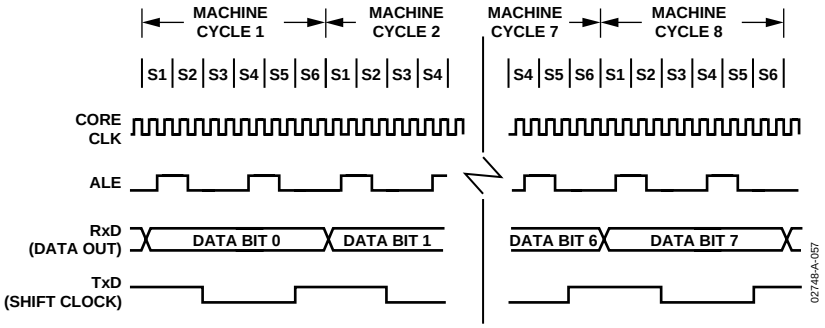

Figure 52. UART Serial Port Transmission, Mode 0

Timer 2 Generated Baud Rates

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

Modes 1 and 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 as before. Therefore, 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 CLK and/or RCLK in T2CON. The baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode as shown in Figure 53.

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

Modes 1 and 3 Baud Rate = (*Core Clk*)/
(32 × [65536 – (*RCAP2H*, *RCAP2L*)])

Table 28 shows some commonly used baud rates and how they could be calculated from a core clock frequency of 2.0971 MHz and 16.7772 MHz.

Table 28. Commonly Used Baud Rates, Timer 2

Ideal Baud	Core CLK	RCAP2H Value	RCAP2L Value	Actual Baud	% Error
19200	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 (D7H)	1191	0.7

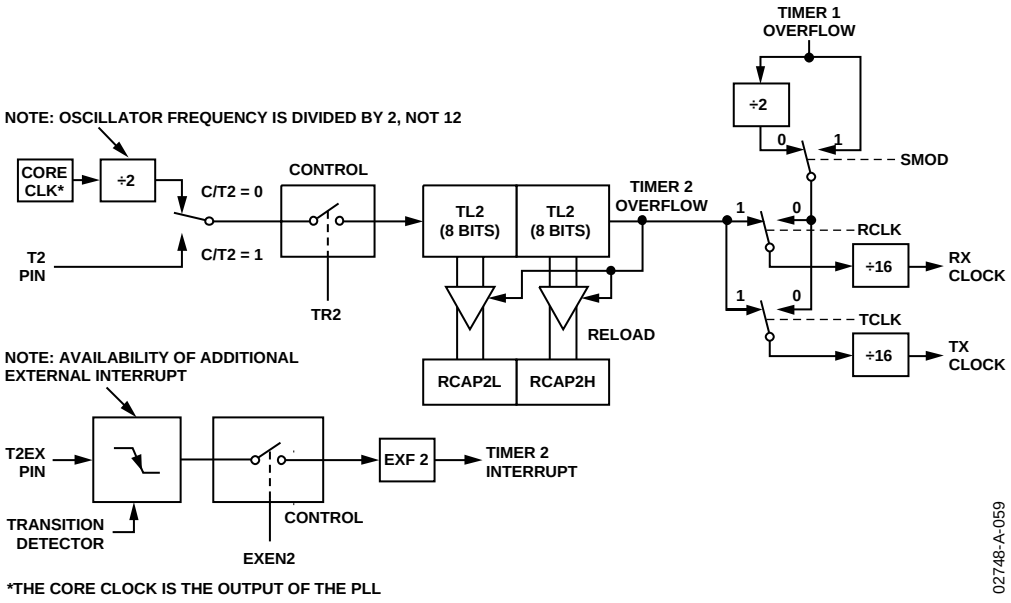

Figure 53. Timer 2, UART Baud Rates

INTERRUPT SYSTEM

The ADuC814 provides a total of twelve 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 and Priority Register

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

EA	EADC	ET2	ES	ET1	EX1	ET0	EX0

Table 29. IE SFR Bit Designations

Bit No.	Name	Description
7	EA	Global Interrupt Enable. Set by the user to enable all interrupt sources. Cleared by the user to disable all interrupt sources.
6	EADC	ADC Interrupt. Set by the user to enable the ADC interrupt. Cleared by the user to disable the ADC interrupt.
5	ET2	Timer 2 Interrupt. Set by the user to enable the Timer 2 interrupt. Cleared by the user to disable the Timer 2 interrupt.
4	ES	UART Serial Port Interrupt. Set by the user to enable the UART serial port interrupt. Cleared by the user to disable the UART serial port interrupt.
3	ET1	Timer 1 Interrupt. Set by the user to enable the Timer 1 interrupt. Cleared by the user to disable the Timer 1 interrupt.
2	EX1	INT1 Interrupt. Set by the user to enable the External Interrupt 1. Cleared by the user to disable the External Interrupt 1.
1	ET0	Timer 0 Interrupt. Set by the user to enable the Timer 0 interrupt. Cleared by the user to disable the Timer 0 interrupt.
0	EX0	INT0 Interrupt. Set by the user to enable the External Interrupt 0. Cleared by the user to disable the External Interrupt 0.

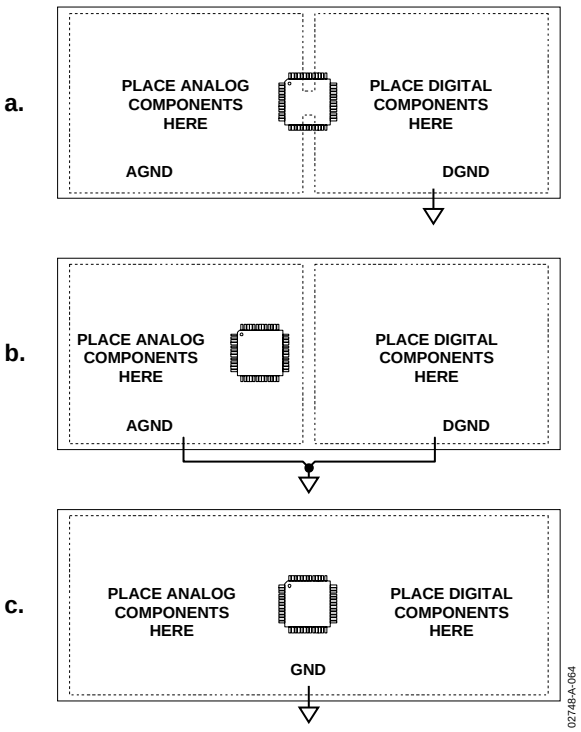

Figure 58. System Grounding Schemes

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.

Table 38. SPI Slave Mode Timing (CPHA = 1)

Parameter		Min	Typ	Max	Unit
t_{SS}	$\overline{SS}$ to SCLOCK Edge	0			ns
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_{SFS}	$\overline{SS}$ High after SCLOCK Edge	0			ns

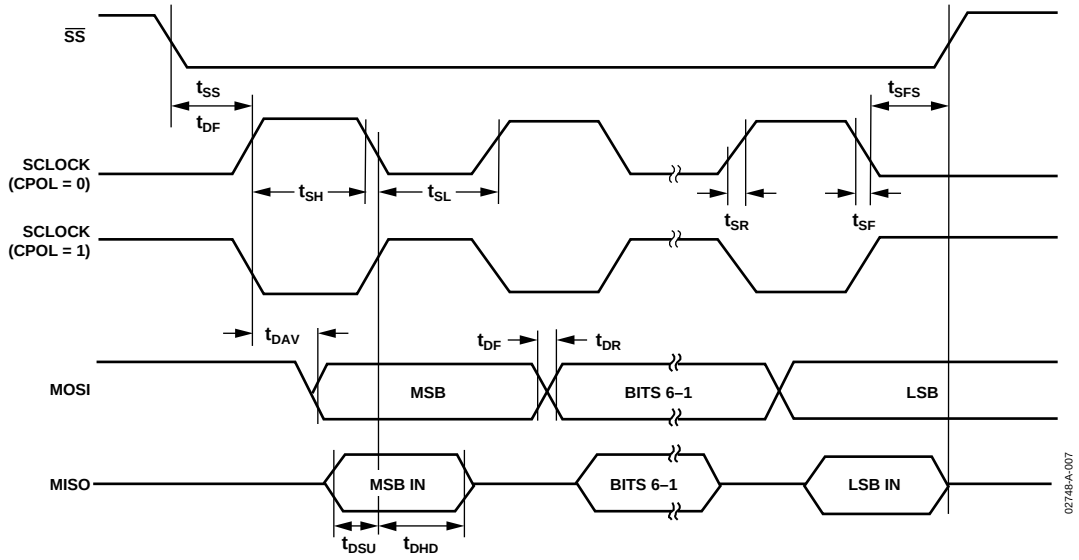

Figure 65. SPI Slave Mode Timing (CPHA = 1)

OUTLINE DIMENSIONS

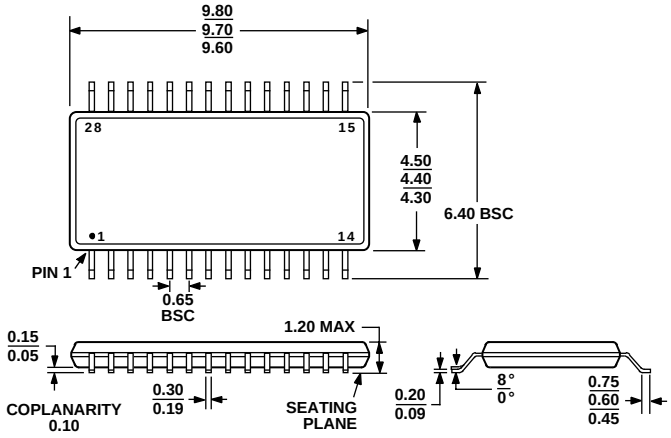

Figure 67. 28-Lead Thin Shrink Small Outline Package (TSSOP) (RU-28)
Dimensions shown in mm