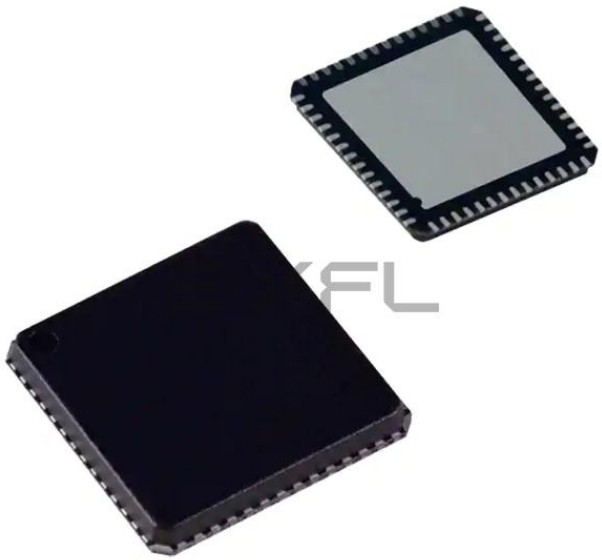

Welcome to E-XFL.COM

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	8.38MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, PSM, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	56-VFQFN Exposed Pad, CSP
Supplier Device Package	56-LFCSP-VQ (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc841bcpz8-3

TABLE OF CONTENTS

REVISION HISTORY

4/16—Rev. 0 to Rev. A

Added Patent Note, Note 1 ... 1
Changes to Figure 3 and Table 3 ... 9
Changes to Figure 4 ... 14
Added Table 4; Renumbered Sequentially ... 14
Changes to Using the DAC Section ... 47
Updated Outline Dimensions ... 94
Changes to Ordering Guide ... 95

11/03—Revision 0: Initial Version

Pin No.	Mnemonic	Type[1]	Description
31	P2.3/A11/A19	I/O	Input/Output Port 2 (P2.3). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A11). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A19). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
32	XTAL1	I	Input to the Inverting Oscillator Amplifier.
33	XTAL2	O	Output of the Inverting Oscillator Amplifier.
36	P2.4/A12/A20	I/O	Input/Output Port 2 (P2.4). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A12). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A20). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
37	P2.5/A13/A21	I/O	Input/Output Port 2 (P2.5). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A13). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A21). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
38	P2.6/PWM0/A14/A22	I/O	Input/Output Port 2 (P2.6). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. PWM 0 Voltage Output (PWM0). PWM outputs can be configured to use Port 2.6 and Port 2.7 or Port 3.4 and Port 3.3. External Memory Addresses (A14). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A22). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
39	P2.7/PWM1/A15/A23	I/O	Input/Output Port 2 (P2.7). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. PWM 1 Voltage Output (PWM1). See the CFG841/CFG842 register for further information. External Memory Addresses (A15). Port 2 emits the middle-order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A23). Port 2 emits the high-order address bytes during accesses to the external 24-bit external data memory space.
40	$\overline{EA}$	I	External Access Enable, Logic Input. When held high, this input enables the device to fetch code from internal program memory locations. The devices do not support external code memory. Do not leave this pin floating.
41	$\overline{PSEN}$	O	Program Store Enable, Logic Output. This pin remains low during internal program execution. $\overline{PSEN}$ enables serial download mode when pulled low through a resistor on power-up or reset. On reset, this pin momentarily becomes an input and the status of the pin is sampled. If there is no pull-down resistor in place, the pin goes momentarily high and then user code executes. If a pull-down resistor is in place, the embedded serial download/debug kernel executes.
42	ALE	O	Address Latch Enable, Logic Output. This output latches the low byte and page byte for 24-bit address space accesses of the address into external data memory.

Pin No.	Mnemonic	Type[1]	Description
43	P0.0/A0	I/O	Input/Output Port 0 (P0.0). Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high impedance inputs. External Memory Address (A0). Port 0 is also the multiplexed low order address and data bus during accesses to external data memory. In this application, it uses strong internal pull-up resistors when emitting 1s.
44	P0.1/A1	I/O	Input/Output Port 0 (P0.1). Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high impedance inputs. External Memory Address (A1). Port 0 is also the multiplexed low order address and data bus during accesses to external data memory. In this application, it uses strong internal pull-up resistors when emitting 1s.
45	P0.2/A2	I/O	Input/Output Port 0 (P0.2). Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high impedance inputs. External Memory Address (A2). Port 0 is also the multiplexed low order address and data bus during accesses to external data memory. In this application, it uses strong internal pull-up resistors when emitting 1s.
46	P0.3/A3	I/O	Input/Output Port 0 (P0.3).Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high impedance inputs. External Memory Address (A3). Port 0 is also the multiplexed low order address and data bus during accesses to external data memory. In this application, it uses strong internal pull-up resistors when emitting 1s.
49	P0.4/A4	I/O	Input/Output Port 0 (P0.4). Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high impedance inputs. External Memory Address (A4). Port 0 is also the multiplexed low order address and data bus during accesses to external data memory. In this application, it uses strong internal pull-up resistors when emitting 1s.
50	P0.5/A5	I/O	Input/Output Port 0 (P0.5). Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high impedance inputs. External Memory Address (A5). Port 0 is also the multiplexed low order address and data bus during accesses to external data memory. In this application, it uses strong internal pull-up resistors when emitting 1s.
51	P0.6/A6	I/O	Input/Output Port 0 (P0.6). Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high impedance inputs. External Memory Address (A6). Port 0 is also the multiplexed low order address and data bus during accesses to external data memory. In this application, it uses strong internal pull-up resistors when emitting 1s.
52	P0.7/A7	I/O	Input/Output Port 0 (P0.7). Port 0 is an 8-bit open-drain bidirectional I/O port. Port 0 pins that have 1s written to them float, and in that state can be used as high impedance inputs. External Memory Address (A7). Port 0 is also the multiplexed low order address and data bus during accesses to external data memory. In this application, it uses strong internal pull-up resistors when emitting 1s.

[1] P = power, G = ground, I= input, O = output., NC = no connect.

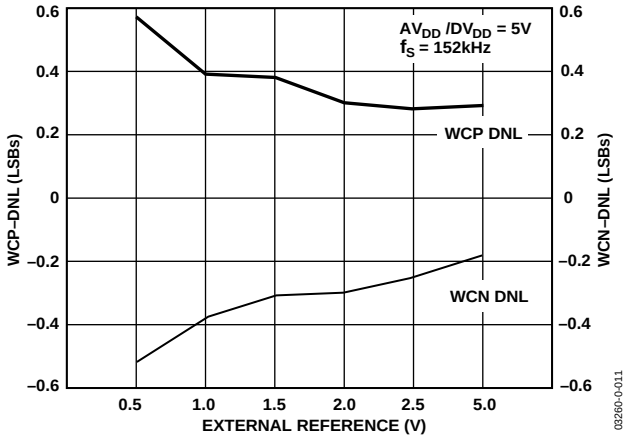

Figure 13. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 5 V

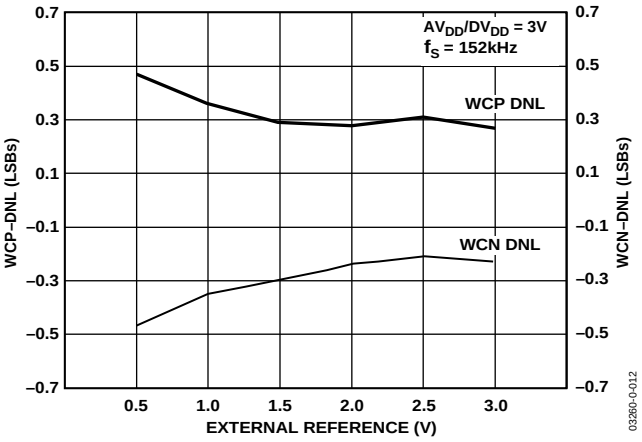

Figure 14. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 3 V

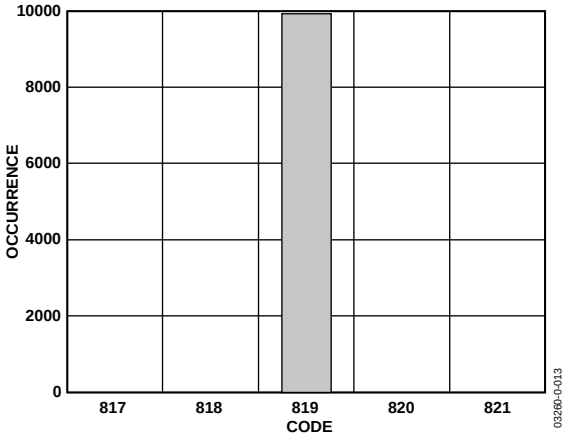

Figure 15. Code Histogram Plot, V_{DD} = 5 V

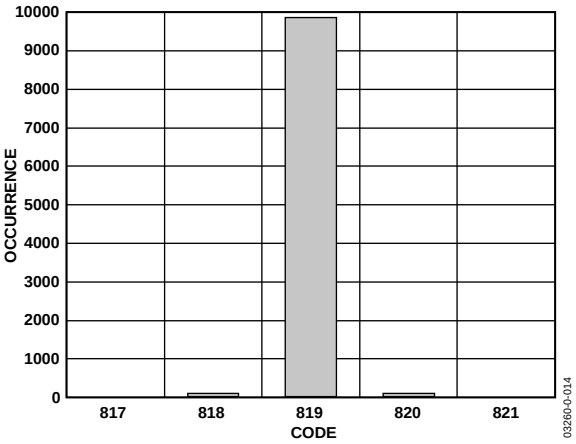

Figure 16. Code Histogram Plot, V_{DD} = 3 V

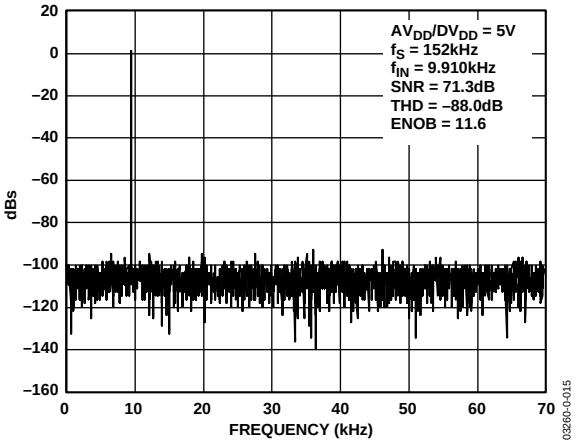

Figure 17. Dynamic Performance at V_{DD} = 5 V

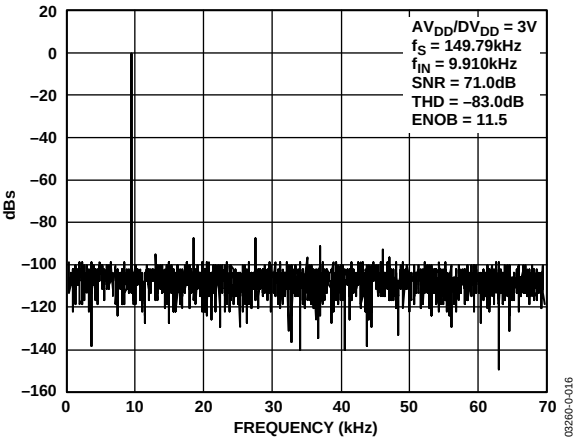

Figure 18. Dynamic Performance at V_{DD} = 3 V

FUNCTIONAL DESCRIPTION

8052 INSTRUCTION SET

Table 5 documents the number of clock cycles required for each instruction. Most instructions are executed in one or two clock cycles, resulting in a 16 MIPS peak performance when operating at PLLCON = 00H on the ADuC842/ADuC843. On the ADuC841, 20 MIPS peak performance is possible with a 20 MHz external crystal.

Table 5. Instructions

Mnemonic	Description	Bytes	Cycles
Arithmetic			
ADD A,Rn	Add register to A	1	1
ADD A,@Ri	Add indirect memory to A	1	2
ADD A,dir	Add direct byte to A	2	2
ADD A,#data	Add immediate to A	2	2
ADDC A,Rn	Add register to A with carry	1	1
ADDC A,@Ri	Add indirect memory to A with carry	1	2
ADDC A,dir	Add direct byte to A with carry	2	2
ADD A,#data	Add immediate to A with carry	2	2
SUBB A,Rn	Subtract register from A with borrow	1	1
SUBB A,@Ri	Subtract indirect memory from A with borrow	1	2
SUBB A,dir	Subtract direct from A with borrow	2	2
SUBB A,#data	Subtract immediate from A with borrow	2	2
INC A	Increment A	1	1
INC Rn	Increment register	1	1
INC @Ri	Increment indirect memory	1	2
INC dir	Increment direct byte	2	2
INC DPTR	Increment data pointer	1	3
DEC A	Decrement A	1	1
DEC Rn	Decrement register	1	1
DEC @Ri	Decrement indirect memory	1	2
DEC dir	Decrement direct byte	2	2
MUL AB	Multiply A by B	1	9
DIV AB	Divide A by B	1	9
DA A	Decimal adjust A	1	2
Logic			
ANL A,Rn	AND register to A	1	1
ANL A,@Ri	AND indirect memory to A	1	2
ANL A,dir	AND direct byte to A	2	2
ANL A,#data	AND immediate to A	2	2
ANL dir,A	AND A to direct byte	2	2
ANL dir,#data	AND immediate data to direct byte	3	3
ORL A,Rn	OR register to A	1	1
ORL A,@Ri	OR indirect memory to A	1	2
ORL A,dir	OR direct byte to A	2	2
ORL A,#data	OR immediate to A	2	2
ORL dir,A	OR A to direct byte	2	2
ORL dir,#data	OR immediate data to direct byte	3	3
XRL A,Rn	Exclusive-OR register to A	1	1
XRL A,@Ri	Exclusive-OR indirect memory to A	2	2
XRL A,#data	Exclusive-OR immediate to A	2	2
XRL dir,A	Exclusive-OR A to direct byte	2	2

Mnemonic	Description	Bytes	Cycles
XRL A,dir	Exclusive-OR indirect memory to A	2	2
XRL dir,#data	Exclusive-OR immediate data to direct	3	3
CLR A	Clear A	1	1
CPL A	Complement A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
Data Transfer			
MOV A,Rn	Move register to A	1	1
MOV A,@Ri	Move indirect memory to A	1	2
MOV Rn,A	Move A to register	1	1
MOV @Ri,A	Move A to indirect memory	1	2
MOV A,dir	Move direct byte to A	2	2
MOV A,#data	Move immediate to A	2	2
MOV Rn,#data	Move register to immediate	2	2
MOV dir,A	Move A to direct byte	2	2
MOV Rn, dir	Move register to direct byte	2	2
MOV dir, Rn	Move direct to register	2	2
MOV @Ri,#data	Move immediate to indirect memory	2	2
MOV dir,@Ri	Move indirect to direct memory	2	2
MOV @Ri,dir	Move direct to indirect memory	2	2
MOV dir,dir	Move direct byte to direct byte	3	3
MOV dir,#data	Move immediate to direct byte	3	3
MOV DPTR,#data	Move immediate to data pointer	3	3
MOVC A,@A+DPTR	Move code byte relative DPTR to A	1	4
MOVC A,@A+PC	Move code byte relative PC to A	1	4
MOVX A,@Ri	Move external (A8) data to A	1	4
MOVX A,@DPTR	Move external (A16) data to A	1	4
MOVX @Ri,A	Move A to external data (A8)	1	4
MOVX @DPTR,A	Move A to external data (A16)	1	4
PUSH dir	Push direct byte onto stack	2	2
POP dir	Pop direct byte from stack	2	2
XCH A,Rn	Exchange A and register	1	1
XCH A,@Ri	Exchange A and indirect memory	1	2
XCHD A,@Ri	Exchange A and indirect memory nibble	1	2
XCH A,dir	Exchange A and direct byte	2	2
Boolean			
CLR C	Clear carry	1	1
CLR bit	Clear direct bit	2	2
SETB C	Set carry	1	1
SETB bit	Set direct bit	2	2
CPL C	Complement carry	1	1
CPL bit	Complement direct bit	2	2
ANL C,bit	AND direct bit and carry	2	2
ANL C,/bit	AND direct bit inverse to carry	2	2
ORL C,bit	OR direct bit and carry	2	2
ORL C,/bit	OR direct bit inverse to carry	2	2
MOV C,bit	Move direct bit to carry	2	2
MOV bit,C	Move carry to direct bit	2	2

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

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

ADC DMA Mode

The on-chip ADC has been designed to run at a maximum conversion speed of 2.38 µs (420 kHz sampling rate). When converting at this rate, the ADuC841/ADuC842/ADuC843 MicroConverter® has 2 µs to read the ADC result and to store the result in memory for further postprocessing; otherwise the next ADC sample could be lost. In an interrupt driven routine, the MicroConverter would also have to jump to the ADC interrupt service routine, which also increases the time required to store the ADC results. In applications where the parts cannot sustain the interrupt rate, an ADC DMA mode is provided.

To enable DMA mode, Bit 6 in ADCCON2 (DMA) must be set, which allows the ADC results to be written directly to a 16 MByte external static memory SRAM (mapped into data memory space) without any interaction from the core of the part. This mode allows the part to capture a contiguous sample stream at full ADC update rates (420 kHz).

Typical DMA Mode Configuration Example

Setting the parts to DMA mode consists of the following steps:

1. The ADC must be powered down. This is done by ensuring that MD1 and MD0 are both set to 0 in ADCCON1.
2. The DMA address pointer must be set to the start address of where the ADC results are to be written. This is done by writing to the DMA mode address pointers DMAL, DMAH, and DMAP. DMAL must be written to first, followed by DMAH, and then by DMAP.
3. The external memory must be preconfigured. This consists of writing the required ADC channel IDs into the top four bits of every second memory location in the external SRAM, starting at the first address specified by the DMA address pointer. Because the ADC DMA mode operates independently from the ADuC841/ADuC842/ADuC843 core, it is necessary to provide it with a stop command. This is done by duplicating the last channel ID to be converted followed by 1111 into the next channel selection field. A typical preconfiguration of external memory is shown in Figure 34.

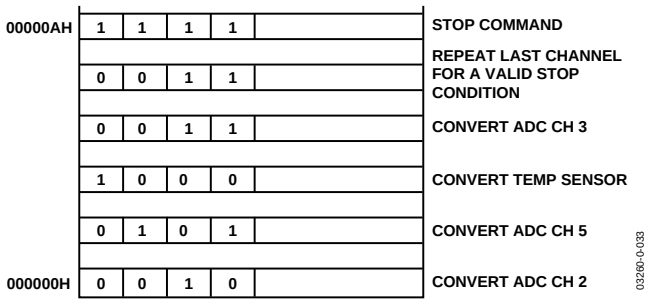

Figure 34. Typical DMA External Memory Preconfiguration

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

When the DMA conversions are complete, the ADC interrupt bit, ADCI, is set by hardware, and the external SRAM contains the new ADC conversion results as shown in Figure 35. Note that no result is written to the last two memory locations.

When the DMA mode logic is active, it takes the responsibility of storing the ADC results away from both the user and the core logic of the part. As the DMA interface writes the results of the ADC conversions to external memory, it takes over the external memory interface from the core. Thus, any core instructions that access the external memory while DMA mode is enabled does not get access to the external memory. The core executes the instructions, and they take the same time to execute, but they cannot access the external memory.

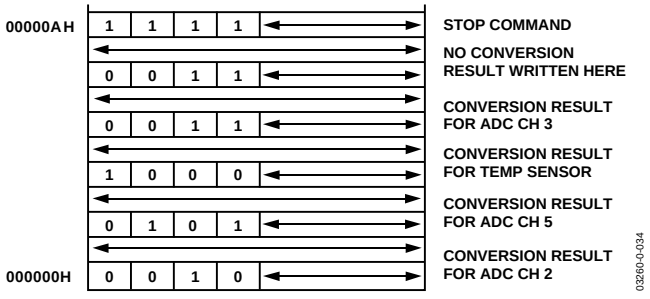

Figure 35. Typical External Memory Configuration Post ADC DMA Operation

A 4 kByte 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.

Flash/EE Memory Reliability

The Flash/EE program and data memory arrays on the parts 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, defined as

1. Initial page erase sequence.
2. Read/verify sequence a single Flash/EE.
3. Byte program sequence memory.
4. Second read/verify sequence endurance cycle.

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 table, the parts' Flash/EE memory endurance qualification has been carried out in accordance with JEDEC Retention Lifetime Specification A117 over the industrial temperature range of –40°C to +25°C and +25°C to +85°C. The results allow the specification of a minimum endurance figure over supply and over 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 parts have been qualified in accordance with the formal JEDEC Retention Lifetime Specification (A117) at a specific junction temperature (T_J = 55°C). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit, described previously, before data retention is characterized. This means that the Flash/EE memory is guaranteed to retain its data for its fully specified retention lifetime every time the Flash/EE memory is reprogrammed. Also note that retention lifetime, based on an activation energy of 0.6 eV, derates with T_J as shown in Figure 38.

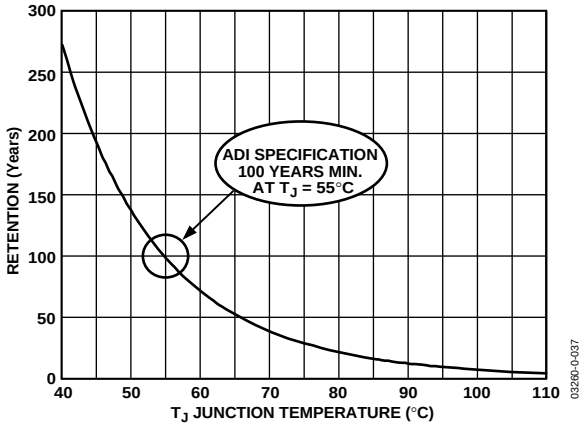

Figure 38. Flash/EE Memory Data Retention

Using the Flash/EE Program Memory

The 62 kByte Flash/EE program memory array is mapped into the lower 62 kBytes of the 64 kByte program space addressable by the parts, and is used to hold user code in typical applications. The program Flash/EE memory array can be programmed in three ways:

Serial Downloading (In-Circuit Programming)

The parts facilitate code download via the standard UART serial port. The parts enter serial download mode after a reset or power cycle if the $\overline{\text{PSEN}}$ pin is pulled low through an external 1 kΩ resistor. Once in serial download mode, the user can download code to the full 62 kBytes of Flash/EE program memory while the device is in-circuit in its target application hardware.

A PC serial download executable is provided as part of the ADuC841/ADuC842 QuickStart development system. The serial download protocol is detailed in MicroConverter Application Note uC004.

Parallel Programming

Parallel programming mode is fully compatible with conventional third party flash or EEPROM device programmers. In this mode, Ports P0, P1, and P2 operate as the external data and address bus interface, ALE operates as the write enable strobe, and Port P3 is used as a general configuration port, which configures the device for various program and erase operations during parallel programming. The high voltage (12 V) supply required for flash programming is generated using on-chip charge pumps to supply the high voltage program lines. The complete parallel programming specification is available on the MicroConverter home page at www.analog.com/microconverter.

User Download Mode (ULOAD)

Figure 39 shows that it is possible to use the 62 kBytes of Flash/EE program memory available to the user as a single block of memory. In this mode, all of the Flash/EE memory is read-only to user code.

However, the Flash/EE program memory can also be written to during runtime simply by entering ULOAD mode. In ULOAD mode, the lower 56 kBytes of program memory can be erased and reprogrammed by user software as shown in Figure 39. ULOAD mode can be used to upgrade your code in the field via any user defined download protocol. By configuring the SPI port on the part as a slave, it is possible to completely reprogram the 56 kBytes of Flash/EE program memory in only 5 seconds (refer to Application Note uC007).

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

The upper 6 kBytes of the 62 kBytes of Flash/EE program memory are programmable only via serial download or parallel programming. This means that this space appears as read-only to user code. Therefore, it cannot be accidentally erased or reprogrammed by erroneous code execution, which makes it very suitable to use the 6 kBytes as a bootloader.

A bootload enable option exists in the serial downloader to "always run from E000H after reset." If using a bootloader, this option is recommended to ensure that the bootloader always executes correct code after reset. Programming the Flash/EE program memory via ULOAD mode is described in more detail in the description of ECON and in Application Note uC007.

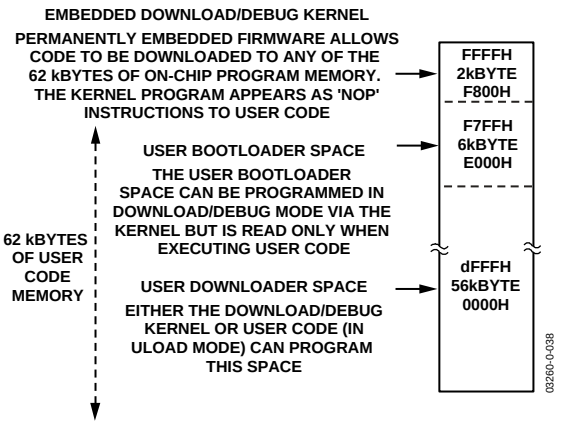

Figure 39. Flash/EE Program Memory Map in ULOAD Mode (62 kByte Part)

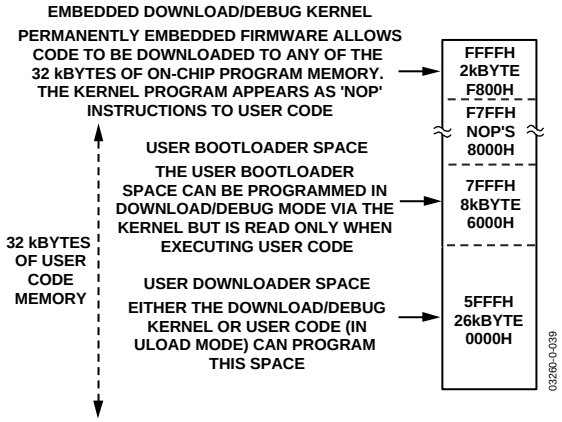

Figure 40. Flash/EE Program Memory Map in ULOAD Mode (32 kByte Part)

Flash/EE Program Memory Security

The ADuC841/ADuC842/ADuC843 facilitate three modes of Flash/EE program memory security. These modes can be independently activated, restricting access to the internal code space. These security modes can be enabled as part of serial download protocol as described in Application Note uC004 or via parallel programming. The security modes available on the parts are as follows:

Lock Mode

This mode locks the code memory, disabling parallel programming of the program memory. However, reading the memory in parallel mode and reading the memory via a MOVC command from external memory is still allowed. This mode is deactivated by initiating a code-erase command in serial download or parallel programming modes.

Secure Mode

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

Serial Safe Mode

This mode disables serial download capability on the device. If serial safe mode is activated and an attempt is made to reset the part into serial download mode, that is, $\overline{\text{RESET}}$ asserted and de-asserted with $\overline{\text{PSEN}}$ low, the part interprets the serial download reset as a normal reset only. It therefore cannot enter serial download mode but can only execute as a normal reset sequence. Serial safe mode can be disabled only by initiating a code-erase command in parallel programming mode.

USING FLASH/EE DATA MEMORY

The 4 kBytes of Flash/EE data memory are configured as 1024 pages, each of 4 bytes. As with the other ADuC841/ADuC842/ADuC843 peripherals, the interface to this memory space is via a group of registers mapped in the SFR space. A group of four data registers (EDATA1–4) is used to hold the four bytes of data at each page. The page is addressed via the two registers, EADRH and EADRL. Finally, ECON is an 8-bit control register that may be written with one of nine Flash/EE memory access commands to trigger various read, write, erase, and verify functions. A block diagram of the SFR interface to the Flash/EE data memory array is shown in Figure 41.

ECON—Flash/EE Memory Control SFR

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

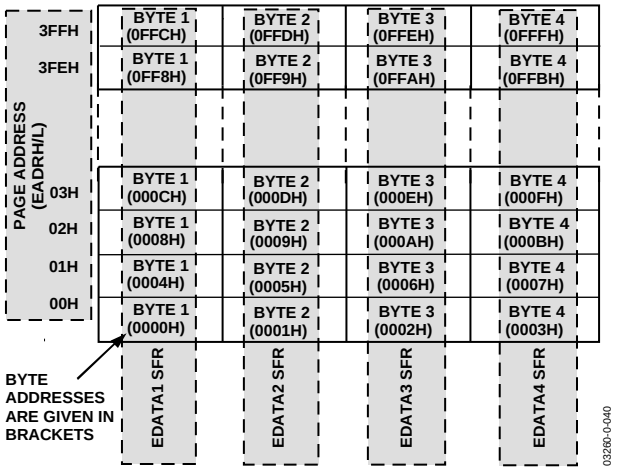

Figure 41. Flash/EE Data Memory Control and Configuration

Table 13. ECON—Flash/EE Memory Commands

ECON VALUE	Command Description (Normal Mode) (Power-On Default)	Command Description (ULOAD Mode)
01H READ	Results in 4 bytes in the Flash/EE data memory, addressed by the page address EADRH/L, being read into EDATA1–4.	Not implemented. Use the MOVC instruction.
02H WRITE	Results in 4 bytes in EDATA1–4 being written to the Flash/EE data memory at the page address given by EADRH/L (0 – EADRH/L < 0400H). Note that the 4 bytes in the page being addressed must be pre-erased.	Results in bytes 0–255 of internal XRAM being written to the 256 bytes of Flash/EE program memory at the page address given by EADRH (0 – EADRH < E0H). Note that the 256 bytes in the page being addressed must be pre-erased.
03H	Reserved.	Reserved.
04H VERIFY	Verifies that the data in EDATA1–4 is contained in the page address given by EADRH/L. A subsequent read of the ECON SFR results in 0 being read if the verification is valid, or a nonzero value being read to indicate an invalid verification.	Not implemented. Use the MOVC and MOVX instructions to verify the write in software.
05H ERASE PAGE	Results in erasing the 4-byte page of Flash/EE data memory addressed by the page address EADRH/L.	Results in the 64 byte page of Flash/EE program memory, addressed by the byte address EADRH/L, being erased. EADRL can equal any of 64 locations within the page. A new page starts whenever EADRL is equal to 00H, 40H, 80H, or C0H.
06H ERASE ALL	Results in erasing the entire 4 kBytes of Flash/EE data memory.	Results in erasing the entire 56 kBytes of ULOAD Flash/EE program memory.
81H READBYTE	Results in the byte in the Flash/EE data memory, addressed by the byte address EADRH/L, being read into EDATA1 (0 – EADRH / L – 0FFFH).	Not implemented. Use the MOVC command.
82H WRITEBYTE	Results in the byte in EDATA1 being written into Flash/EE data memory at the byte address EADRH/L	Results in the byte in EDATA1 being written into Flash/EE program memory at the byte address EADRH/L (0 – EADRH/L – DFFFH).
0FH EXULOAD	Leaves the ECON instructions to operate on the Flash/EE data memory.	Enters normal mode directing subsequent ECON instructions to operate on the Flash/EE data memory.
F0H ULOAD	Enters ULOAD mode, directing subsequent ECON instructions to operate on the Flash/EE program memory.	Leaves the ECON instructions to operate on the Flash/EE program memory.

ADuC842/ADuC843 Configuration SFR (CFG842)

The CFG842 SFR contains the necessary bits to configure the internal XRAM, external clock select, PWM output selection, DAC buffer, and the extended SP for both the ADuC842 and the ADuC843. By default, it configures the user into 8051 mode, that is, extended SP is disabled and internal XRAM is disabled. On the ADuC841, this register is the CFG841 register and is described on the next page.

CFG842	ADuC842/ADuC843 Config SFR
SFR Address	AFH
Power-On Default	00H
Bit Addressable	No

Table 14. CFG842 SFR Bit Designations

Bit No.	Name	Description
7	EXSP	Extended SP Enable. When set to 1 by the user, the stack rolls over from SPH/SP = 00FFH to 0100H. When set to 0 by the user, the stack rolls over from SP = FFH to SP = 00H.
6	PWPO	PWM Pin Out Selection. Set to 1 by the user to select P3.4 and P3.3 as the PWM output pins. Set to 0 by the user to select P2.6 and P2.7 as the PWM output pins.
5	DBUF	DAC Output Buffer. Set to 1 by the user to bypass the DAC output buffer. Set to 0 by the user to enable the DAC output buffer.
4	EXTCLK	Set by the user to 1 to select an external clock input on P3.4. Set by the user to 0 to use the internal PLL clock.
3	RSVD	Reserved. This bit should always contain 0.
2	RSVD	Reserved. This bit should always contain 0.
1	MSPI	Set to 1 by the user to move the SPI functionality of MISO, MOSI, and SCLOCK to P3.3, P3.4, and P3.5, respectively. Set to 0 by the user to leave the SPI functionality as usual on MISO, MOSI, and SCLOCK pins.
0	XRAMEN	XRAM Enable Bit. When set to 1 by the user, the internal XRAM is mapped into the lower 2 kBytes of the external address space. When set to 0 by the user, the internal XRAM is not accessible, and the external data memory is mapped into the lower 2 kBytes of external data memory.

Mode 4: Dual NRZ 16-Bit Σ-Δ DAC

Mode 4 provides a high speed PWM output similar to that of a Σ-Δ DAC. Typically, this mode is used with the PWM clock equal to 16.777216 MHz. In this mode, P2.6 and P2.7 are updated every PWM clock (60 ns in the case of 16 MHz). Over any 65536 cycles (16-bit PWM) PWM0 (P2.6) is high for PWM0H/L cycles and low for (65536 – PWM0H/L) cycles. Similarly, PWM1 (P2.7) is high for PWM1H/L cycles and low for (65536 – PWM1H/L) cycles.

For example, if PWM1H is set to 4010H (slightly above one quarter of FS), then P2.7 is typically low for three clocks and high for one clock (each clock is approximately 60 ns). Over every 65536 clocks, the PWM compensates for the fact that the output should be slightly above one quarter of full scale by having a high cycle followed by only two low cycles.

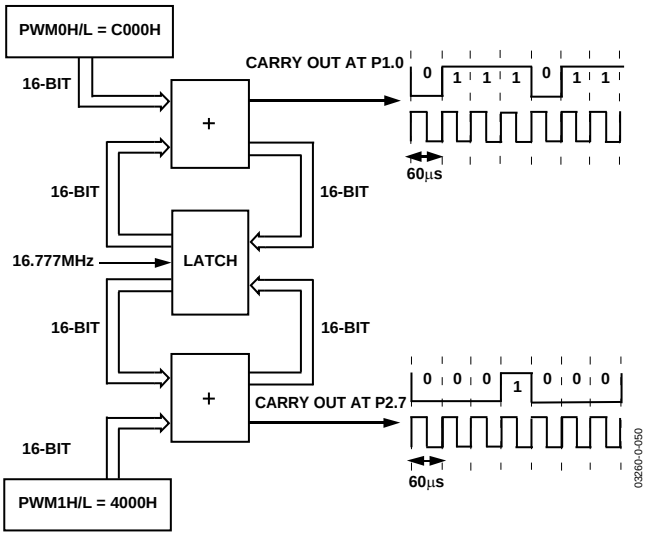

Figure 51. PWM Mode 4

For faster DAC outputs (at lower resolution), write 0s to the LSBs that are not required. If, for example, only 12-bit performance is required, write 0s to the four LSBs. This means that a 12-bit accurate Σ-Δ DAC output can occur at 4.096 kHz. Similarly writing 0s to the 8 LSBs gives an 8-bit accurate Σ-Δ DAC output at 65 kHz.

Mode 5: Dual 8-Bit PWM

In Mode 5, the duty cycle of the PWM outputs and the resolution of the PWM outputs are individually programmable. The maximum resolution of the PWM output is 8 bits. The output resolution is set by the PWM1L and PWM1H SFRs for the P2.6 and P2.7 outputs, respectively. PWM0L and PWM0H sets the duty cycles of the PWM outputs at P2.6 and P2.7, respectively. Both PWMs have the same clock source and clock divider.

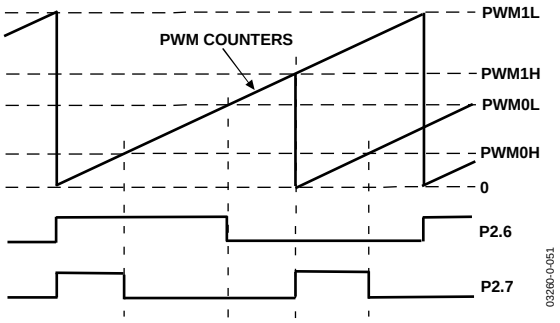

Figure 52. PWM Mode 5

Mode 6: Dual RZ 16-Bit Σ-Δ DAC

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

If PWM1H is set to 4010H (slightly above one quarter of FS), P2.7 is typically low for three full clocks (3 × 60 ns), high for half a clock (30 ns), and then low again for half a clock (30 ns) before repeating itself. Over every 65536 clocks, the PWM compensates for the fact that the output should be slightly above one quarter of full scale by leaving the output high for two half clocks in four. The rate at which this happens depends on the value and degree of compensation required.

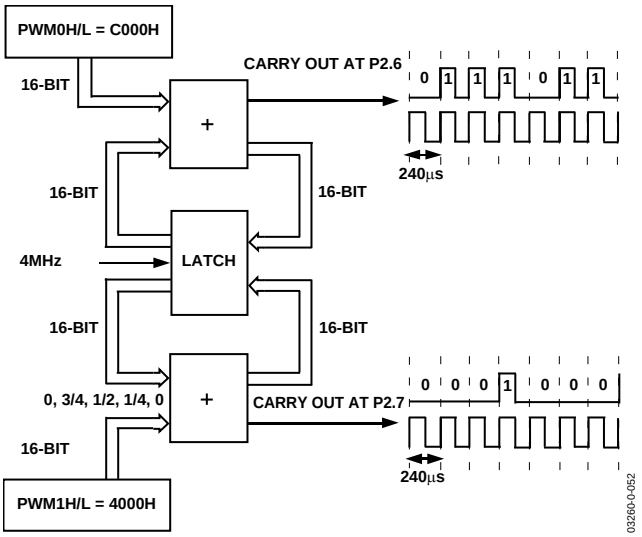

Figure 53. PWM Mode 6

SERIAL PERIPHERAL INTERFACE (SPI)

The ADuC841/ADuC842/ADuC843 integrate a complete hardware serial peripheral interface on-chip. SPI is an industry-standard synchronous serial interface that allows 8 bits of data to be synchronously transmitted and received simultaneously, that is, full duplex. Note that the SPI pins are shared with the I^2C pins. Therefore, the user can enable only one interface or the other on these pins at any given time (see SPE in Table 19). SPI can be operated at the same time as the I^2C interface if the MSPI bit in CFG841/CFG8842 is set. This moves the SPI outputs (MISO, MOSI, and SCLOCK) to P3.3, P3.4, and P3.5, respectively). The SPI port can be configured for master or slave operation and typically consists of four pins, described in the following sections.

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

The MISO pin 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 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 master serial clock (SCLOCK) is used to synchronize the data being transmitted and received through the MOSI and MISO data lines. A single data bit is transmitted and received in each SCLOCK period. Therefore, a byte is transmitted/received after eight SCLOCK periods. The SCLOCK pin is configured as an output in master mode and as an input in slave mode. In master mode, the bit rate, polarity, and phase of the clock are controlled by the CPOL, CPHA, SPR0, and SPR1 bits in the SPICON SFR (see Table 19). 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 CPHA and CPOL are configured the same for the master and slave devices.

$\overline{SS}$ (Slave Select Input Pin)

The $\overline{SS}$ pin is shared with the ADC5 input. To configure this pin as a digital input, the bit must be cleared, for example, CLR P1.5. This line is active low. Data is received or transmitted in slave mode only when the $\overline{SS}$ pin is low, allowing the parts to be used in single-master, multislave SPI configurations. If CPHA = 1, the $\overline{SS}$ input may be permanently pulled low. If 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 SFR registers, described in the following tables, are used to control the SPI interface.

SPICON SPI Control Register

SFR Address	F8H
Power-On Default	04H
Bit Addressable	Yes

Table 19. 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 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 user code.
5	SPE	SPI Interface Enable Bit. Set by the user to enable the SPI interface. Cleared by the user to enable the I^2C pins, this is not requiredto enable the I^2C interface if the MSPI bit is set in CFG841/CFG842. In this case, the I^2C interface is automatically enabled.
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.
2	CPHA[1]	Clock Phase Select Bit. Set by the user if leading SCLOCK edge is to transmit data. Cleared by the user if trailing SCLOCK edge is to transmit data.
1	SPR1	SPI Bit Rate Select Bits.
0	SPR0	These bits select the SCLOCK rate (bit rate) in master mode as follows: SPR1 SPR0 Selected Bit Rate 0 0 $f_{osc}/2$ 0 1 $f_{osc}/4$ 1 0 $f_{osc}/8$ 1 1 $f_{osc}/16$ In SPI slave mode, that is, SPIM = 0, the logic level on the external $\overline{SS}$ pin can be read via the SPR0 bit.

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

SPIDAT	**SPI Data Register**
Function	SPIDAT SFR is written by the user to transmit data over the SPI interface or read by user code to read data just received by the SPI interface.
SFR Address	F7H
Power-On Default	00H
Bit Addressable	No

I²C COMPATIBLE INTERFACE

The ADuC841/ADuC842/ADuC843 support a fully licensed I²C serial interface. The I²C interface is implemented as a full hardware slave and software master. SDATA is the data I/O pin, and SCLOCK is the serial clock. These two pins are shared with the MOSI and SCLOCK pins of the on-chip SPI interface. To enable the I²C interface, the SPI interface must be turned off (see SPE in Table 19) or the SPI interface must be moved to P3.3, P3.4, and P3.5 via the CFG841.1/CFG842.1 bit. Application Note uC001 describes the operation of this interface as implemented and is available from the MicroConverter website at www.analog.com/microconverter.

Three SFRs are used to control the I²C interface and are described in the following tables.

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

Table 20. I2CCON SFR Bit Designations, Master Mode

Bit No.	Name	Description
7	MDO	I²C Software Master Data Output Bit (Master Mode Only). This data bit is used to implement a master I²C transmitter interface in software. Data written to this bit is output on the SDATA pin if the data output enable (MDE) bit is set.
6	MDE	I²C Software Master Data Output Enable Bit (Master Mode Only). Set by the user to enable the SDATA pin as an output (Tx). Cleared by the user to enable the SDATA pin as an input (Rx).
5	MCO	I²C Software Master Clock Output Bit (Master Mode Only). This data bit is used to implement a master I²C transmitter interface in software. Data written to this bit is output on the SCLOCK pin.
4	MDI	I²C Software Master Data Input Bit (Master Mode Only). This data bit is used to implement a master I²C receiver interface in software. Data on the SDATA pin is latched into this bit on SCLOCK if the data output enable (MDE) bit is 0.
3	I2CM	I²C Master/Slave Mode Bit. Set by the user to enable I²C software master mode. Cleared by the user to enable I²C hardware slave mode.
2	----	Reserved.
1	----	Reserved.
0	----	Reserved.

Table 21. I2CCON SFR Bit Designations, Slave Mode

Bit No.	Name	Description
7	I2CSI	I²C Stop Interrupt Enable Bit. Set by the user to enable I²C stop interrupts. If set, a stop bit that follows a valid start condition generates an interrupt. Cleared by the user to disable I²C stop interrupts.
6	I2CGC	I²C General Call Status Bit. Set by hardware after receiving a general call address. Cleared by the user.
5	I2CID1	I²C Interrupt Decode Bits.
4	I2CID0	Set by hardware to indicate the source of an I²C interrupt. 00 Start and Matching Address. 01 Repeated Start and Matching Address. 10 User Data. 11 Stop after a Start and Matching Address.
3	I2CM	I²C Master/Slave Mode Bit. Set by the user to enable I²C software master mode. Cleared by the user to enable I²C hardware slave mode.

POWER SUPPLY MONITOR

As its name suggests, the power supply monitor, once enabled, monitors the DV_{DD} supply on the ADuC841/ADuC842/ADuC843. It indicates when any of the supply pins drops below one of two user selectable voltage trip points, 2.93 V and 3.08 V. For correct operation of the power supply monitor function, AV_{DD} must be equal to or greater than 2.7 V. Monitor function is controlled via the PSMCON SFR. If enabled via the IEIP2 SFR, the monitor interrupts the core using the PSMI bit in the PSMCON SFR. This bit is not cleared until the failing power supply has returned above the trip point for at least 250 ms. This monitor function allows the user to save working registers to avoid possible data loss due to the low supply condition, and also ensures that normal code execution does not resume until a safe supply level has been well established. The supply monitor is also protected against spurious glitches triggering the interrupt circuit.

Note that the 5 V part has an internal POR trip level of 4.5 V, which means that there are no usable PSM levels on the 5 V part. The 3 V part has a POR trip level of 2.45 V, allowing all PSM trip points to be used.

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

Table 23. PSMCON SFR Bit Designations

Bit No.	Name	Description
7	----	Reserved.
6	CMPD	DV_{DD} Comparator Bit. This is a read-only bit that directly reflects the state of the DV_{DD} comparator. Read 1 indicates that the DV_{DD} supply is above its selected trip point. Read 0 indicates that the DV_{DD} supply is below its selected trip point.
5	PSMI	Power Supply Monitor Interrupt Bit. This bit is set high by the MicroConverter if either CMPA or CMPD is low, indicating low analog or digital supply. The PSMI bit can be used to interrupt the processor. Once CMPD and/or CMPA return (and remain) high, a 250 ms counter is started. When this counter times out, the PSMI interrupt is cleared. PSMI can also be written by the user. However, if either comparator output is low, it is not possible for the user to clear PSMI.
4	TPD1	DV_{DD} Trip Point Selection Bits.
3	TPD0	These bits select the DV_{DD} trip point voltage as follows: TPD1 TPD0 Selected DV_{DD} Trip Point (V) 0 0 Reserved 0 1 3.08 1 0 2.93 1 1 Reserved
2	----	Reserved.
1	----	Reserved.
0	PSMEN	Power Supply Monitor Enable Bit. Set to 1 by the user to enable the power supply monitor circuit. Cleared to 0 by the user to disable the power supply monitor circuit.

Timer 3 Generated Baud Rates

The high integer dividers in a UART block mean that high speed baud rates are not always possible using some particular crystals. For example, using a 12 MHz crystal, a baud rate of 115200 is not possible. To address this problem, the part has added a dedicated baud rate timer (Timer 3) specifically for generating highly accurate baud rates. Timer 3 can be used instead of Timer 1 or Timer 2 for generating very accurate high speed UART baud rates including 115200 and 230400. Timer 3 also allows a much wider range of baud rates to be obtained. In fact, every desired bit rate from 12 bit/s to 393216 bit/s can be generated to within an error of ±0.8%. Timer 3 also frees up the other three timers, allowing them to be used for different applications. A block diagram of Timer 3 is shown in Figure 74.

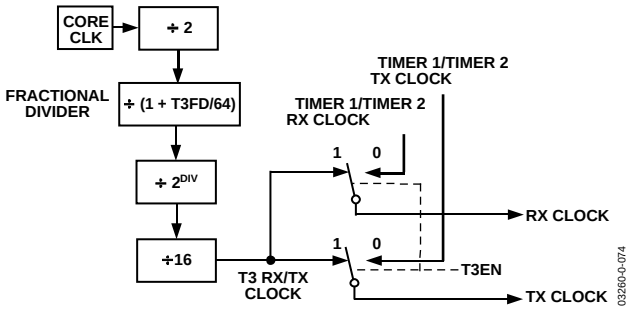

Figure 74. Timer 3, UART Baud Rates

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

The appropriate value to write to the DIV2-1-0 bits can be calculated using the following formula where f_{CORE} is defined in PLLCON SFR. Note that the DIV value must be rounded down.

$$DIV = \frac{\log\left(\frac{f_{CORE}}{16 \times Baud\ Rate}\right)}{\log(2)}$$

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

$$T3FD = \frac{2 \times f_{CORE}}{2^{DIV-1} \times Baud\ Rate} - 64$$

Note that *T3FD* should be rounded to the nearest integer. Once the values for *DIV* and *T3FD* are calculated, the actual baud rate can be calculated with the following formula:

$$Actual\ Baud\ Rate = \frac{2 \times f_{CORE}}{2^{DIV-1} \times (T3FD + 64)}$$

For example, to get a baud rate of 115200 while operating at 16.7 MHz, that is, CD = 0

$$DIV = \log(16777216/(16 \times 115200))/\log 2 = 3.18 = 3$$

$$T3FD = (2 \times 16777216)/(2^2 \times 115200) - 64 = 9 = 09H$$

Therefore, the actual baud rate is 114912 bit/s.

Table 34. T3CON SFR Bit Designations

Bit No.	Name	Description			
7	T3BAUDEN	T3UARTBAUD Enable.			
		Set to enable Timer 3 to generate the baud rate. When set, PCON.7, T2CON.4, and T2CON.5 are ignored.			
		Cleared to let the baud rate be generated as per a standard 8052.			
6		Reserved.			
5		Reserved.			
4		Reserved.			
3		Reserved.			
2	DIV2	Binary Divider Factor.			
1	DIV1	DIV2	DIV1	DIV0	Bin Divider
0	DIV0	0	0	0	1
		0	0	1	1
		0	1	0	1
		0	1	1	1
		1	0	0	1
		1	0	1	1
		1	1	0	1
		1	1	1	1

Parameter					
SPI MASTER MODE TIMING (CPHA = 0)		**Min**	**Typ**	**Max**	**Unit**
t_{SL}	SCLOCK Low Pulse Width[1]		476		ns
t_{SH}	SCLOCK High Pulse Width[1]		476		ns
t_{DAV}	Data Output Valid after SCLOCK Edge			50	ns
t_{DOSU}	Data Output Setup before SCLOCK Edge			150	ns
t_{DSU}	Data Input Setup Time before SCLOCK Edge	100			ns
t_{DHD}	Data Input Hold Time after SCLOCK Edge	100			ns
t_{DF}	Data Output Fall Time		10	25	ns
t_{DR}	Data Output Rise Time		10	25	ns
t_{SR}	SCLOCK Rise Time		10	25	ns
t_{SF}	SCLOCK Fall Time		10	25	ns

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

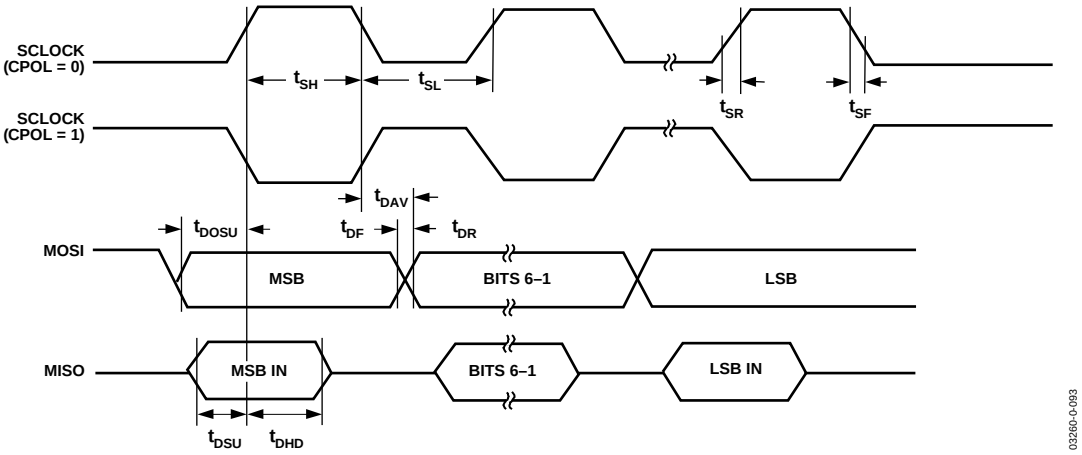

Figure 92. SPI Master Mode Timing (CPHA = 0)

Parameter		Min	Typ	Max	Unit
SPI SLAVE MODE TIMING (CPHA = 1)					
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

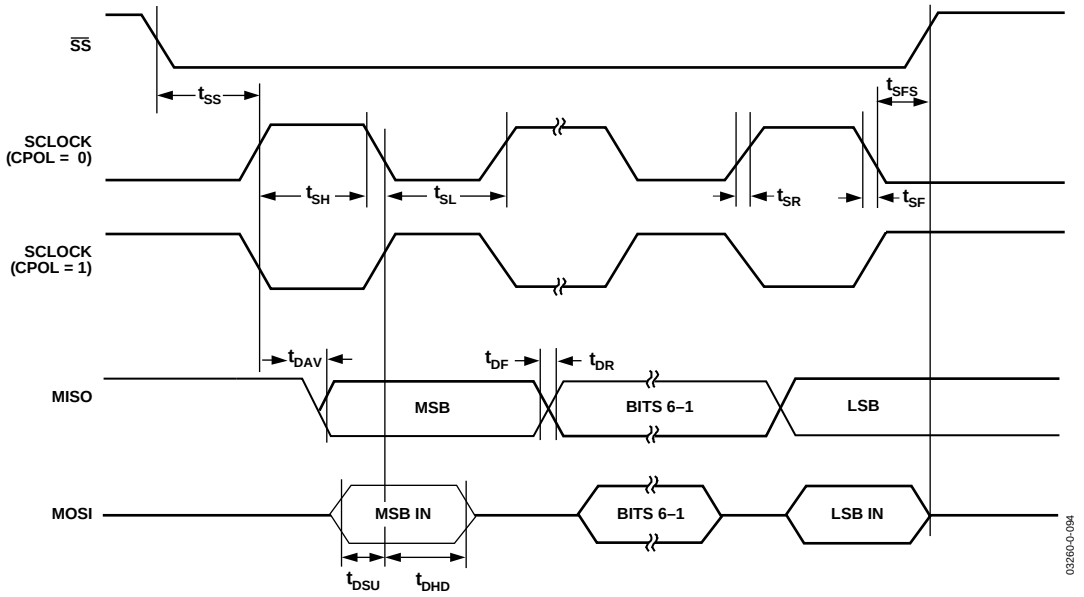

Figure 93. SPI Slave Mode Timing (CPHA = 1)

Parameter					
SPI SLAVE MODE TIMING (CPHA = 0)		**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_{DOSS}	Data Output Valid after $\overline{SS}$ Edge			20	ns
t_{SFS}	$\overline{SS}$ High after SCLOCK Edge				ns

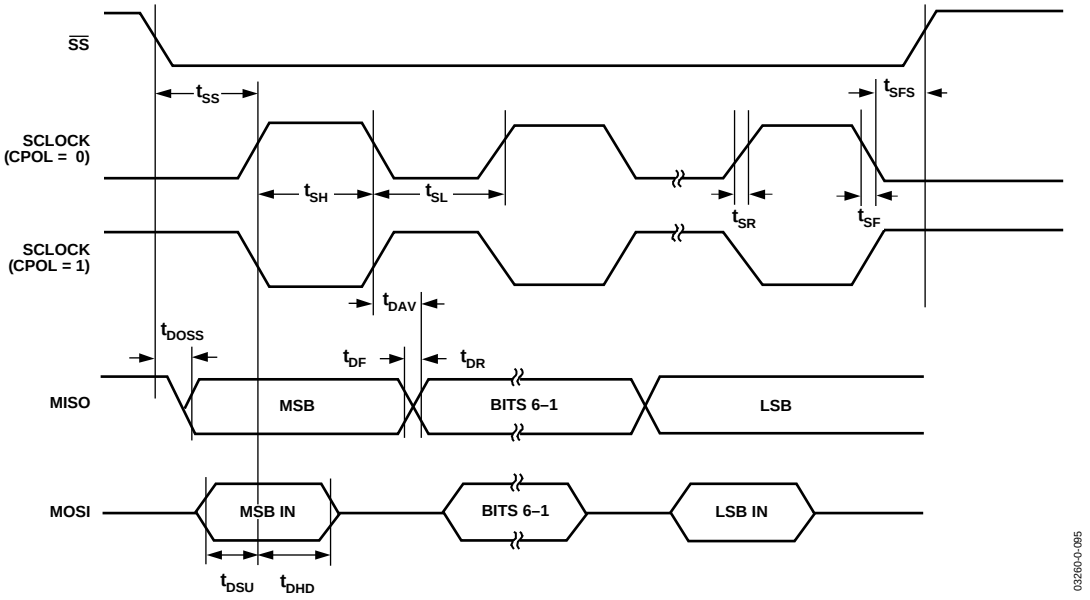

Figure 94. SPI Slave Mode Timing (CPHA = 0)