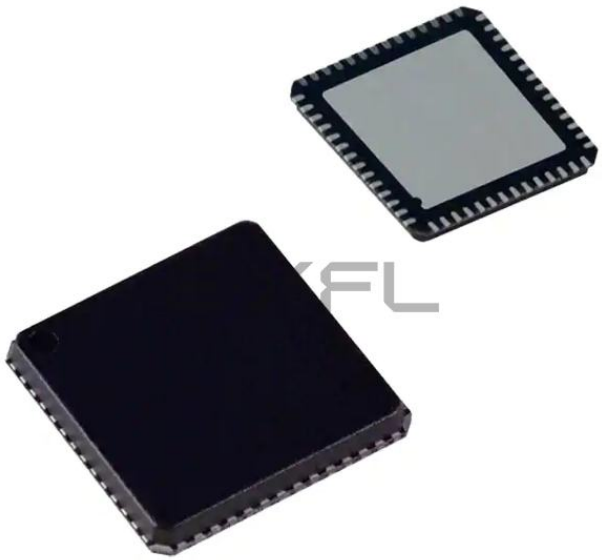


Welcome to E-XFL.COM

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	DMA, PSM, PWM, Temp Sensor, WDT
Number of I/O	32
Program Memory Size	62KB (62K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
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/aduc842bcpz62-5

SPECIFICATIONS[1]

Table 1. AV_{DD} = DV_{DD} = 2.7 V to 3.6 V or 4.75 V to 5.25 V; V_{REF} = 2.5 V internal reference, f_{CORE} = 16.78 MHz @ 5 V 8.38 MHz @ 3 V; all specifications T_A = T_{MIN} to T_{MAX}, unless otherwise noted

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
ADC CHANNEL SPECIFICATIONS				
DC ACCURACY[2, 3]				f_{SAMPLE} = 120 kHz, see the Typical Performance Characteristics for typical performance at other values of f_{SAMPLE}
Resolution	12	12	Bits	
Integral Nonlinearity	±1	±1	LSB max	2.5 V internal reference
	±0.3	±0.3	LSB typ	
Differential Nonlinearity	+1/−0.9	+1/−0.9	LSB max	2.5 V internal reference
	±0.3	±0.3	LSB typ	
Integral Nonlinearity[4]	±2	±1.5	LSB max	1 V external reference
Differential Nonlinearity[4]	+1.5/−0.9	+1.5/−0.9	LSB max	1 V external reference
Code Distribution	1	1	LSB typ	ADC input is a dc voltage
CALIBRATED ENDPOINT ERRORS[5, 6]				
Offset Error	±3	±2	LSB max	
Offset Error Match	±1	±1	LSB typ	
Gain Error	±3	±2	LSB max	
Gain Error Match	±1	±1	LSB typ	
DYNAMIC PERFORMANCE				f_{IN} = 10 kHz sine wave f_{SAMPLE} = 120 kHz
Signal-to-Noise Ratio (SNR)[7]	71	71	dB typ	
Total Harmonic Distortion (THD)	−85	−85	dB typ	
Peak Harmonic or Spurious Noise	−85	−85	dB typ	
Channel-to-Channel Crosstalk[8]	−80	−80	dB typ	
ANALOG INPUT				
Input Voltage Range	0 to V_{REF}	0 to V_{REF}	V	
Leakage Current	±1	±1	µA max	
Input Capacitance	32	32	pF typ	
TEMPERATURE SENSOR[9]				
Voltage Output at 25°C	700	700	mV typ	
Voltage TC	−1.4	−1.4	mV/°C typ	
Accuracy	±1.5	±1.5	°C typ	Internal/External 2.5 V V_{REF}
DAC CHANNEL SPECIFICATIONS **Internal Buffer Enabled** ADuC841/ADuC842 **Only**				DAC load to AGND R_L = 10 kΩ, C_L = 100 pF
DC ACCURACY[10]				
Resolution	12	12	Bits	
Relative Accuracy	±3	±3	LSB typ	
Differential Nonlinearity[11]	−1	−1	LSB max	Guaranteed 12-bit monotonic
	±1/2	±1/2	LSB typ	
Offset Error	±50	±50	mV max	V_{REF} range
Gain Error	±1	±1	% max	AV_{DD} range
	±1	±1	% typ	V_{REF} range
Gain Error Mismatch	0.5	0.5	% typ	% of full-scale on DAC1
ANALOG OUTPUTS				
Voltage Range_0	0 to V_{REF}	0 to V_{REF}	V typ	DAC V_{REF} = 2.5 V
Voltage Range_1	0 to V_{DD}	0 to V_{DD}	V typ	DAC V_{REF} = V_{DD}
Output Impedance	0.5	0.5	Ω typ	

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
LOGIC INPUTS[4]				
INPUT VOLTAGES				
All Inputs Except SCLOCK, SDATA, RESET, and XTAL1				
VINL, Input Low Voltage	0.8	0.4	V max	
VINH, Input High Voltage	2.0	2.0	V min	
SDATA				
VINL, Input Low Voltage	0.8	0.8	V max	
VINH, Input High Voltage	2.0	2.0	V min	
SCLOCK and RESET ONLY[4]				
(Schmitt-Triggered Inputs)				
V_{T+}	1.3	0.95	V min	
	3.0	0.25	V max	
V_{T-}	0.8	0.4	V min	
	1.4	1.1	V max	
$V_{T+} - V_{T-}$	0.3	0.3	V min	
	0.85	0.85	V max	
CRYSTAL OSCILLATOR				
Logic Inputs, XTAL1 Only				
V_{INL}, Input Low Voltage	0.8	0.4	V typ	
V_{INH}, Input High Voltage	3.5	2.5	V typ	
XTAL1 Input Capacitance	18	18	pF typ	
XTAL2 Output Capacitance	18	18	pF typ	
MCU CLOCK RATE	16.78	8.38	MHz max	ADuC842/ADuC843 Only
	20	8.38	MHz max	ADuC841 Only
DIGITAL OUTPUTS				
Output High Voltage (V_{OH})	2.4		V min	V_{DD} = 4.5 V to 5.5 V
	4		V typ	I_{SOURCE} = 80 µA
		2.4	V min	V_{DD} = 2.7 V to 3.3 V
		2.6	V typ	I_{SOURCE} = 20 µA
Output Low Voltage (V_{OL})				
ALE, Ports 0 and 2	0.4	0.4	V max	I_{SINK} = 1.6 mA
	0.2	0.2	V typ	I_{SINK} = 1.6 mA
Port 3	0.4	0.4	V max	I_{SINK} = 4 mA
SCLOCK/SDATA	0.4	0.4	V max	I_{SINK} = 8 mA, I^2C Enabled
Floating State Leakage Current[4]	±10	±10	µA max	
	±1	±1	µA typ	
STARTUP TIME				At any core CLK
At Power-On	500	500	ms typ	
From Idle Mode	100	100	µs typ	
From Power-Down Mode				
Wake-up with $\overline{INT0}$ Interrupt	150	400	µs typ	
Wake-up with SPI/I^2C Interrupt	150	400	µs typ	
Wake-up with External RESET	150	400	µs typ	
After External RESET in Normal Mode	30	30	ms typ	
After WDT Reset in Normal Mode	3	3	ms typ	Controlled via WDCON SFR

Parameter	V_{DD} = 5 V	V_{DD} = 3 V	Unit	Test Conditions/Comments
POWER REQUIREMENTS[19, 20]				
Power Supply Voltages				
AV_{DD}/DV_{DD} – AGND		2.7	V min	AV_{DD}/DV_{DD} = 3 V nom
		3.6	V max	
	4.75		V min	AV_{DD}/DV_{DD} = 5 V nom
	5.25		V max	
Power Supply Currents **Normal** Mode[21]				
DV_{DD} Current[4]	10	4.5	mA typ	Core CLK = 2.097 MHz
AV_{DD} Current	1.7	1.7	mA max	Core CLK = 2.097 MHz
DV_{DD} Current	38	12	mA max	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
	33	10	mA typ	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
AV_{DD} Current	1.7	1.7	mA max	Core CLK = 16.78MHz/8.38 MHz 5 V/3 V
DV_{DD} Current[4]	45	N/A	mA max	Core CLK = 20MHz **ADuC841 Only**
Power Supply Currents **Idle** Mode[21]				
DV_{DD} Current	4.5	2.2	mA typ	Core CLK = 2.097 MHz
AV_{DD} Current	3	2	µA typ	Core CLK = 2.097 MHz
DV_{DD} Current[4]	12	5	mA max	Core CLK = 16.78 MHz/8.38 MHz 5 V/3 V
	10	3.5	mA typ	Core CLK = 16.78 MHz/8.38 MHz 5 V/3 V
AV_{DD} Current	3	2	µA typ	Core CLK = 16.78 MHz/8.38 MHz 5 V/3 V
Power Supply Currents **Power-Down** Mode[21]				Core CLK = any frequency
DV_{DD} Current	28	18	µA max	Oscillator Off / TIMECON.1 = 0
	20	10	µA typ	
AV_{DD} Current	2	1	µA typ	Core CLK = any frequency, **ADuC841 Only**
DV_{DD} Current[4]	3	1	mA max	TIMECON.1 = 1
DV_{DD} Current[4]	50	22	µA max	Core CLK = any frequency
	40	15	µA typ	**ADuC842/ADuC843 Only,** oscillator on
Typical Additional Power Supply Currents				
PSM Peripheral	15	10	µA typ	AV_{DD} = DV_{DD}
ADC[4]	1.0	1.0	mA min	MCLK Divider = 32
	2.8	1.8	mA max	MCLK Divider = 2
DAC	150	130	µA typ	

See footnotes on the next page.

Pin No.	Mnemonic	Type[1]	Description
22	P3.4/T0/PWMC/PWM0/EXTCLK	I/O	Input/Output Port 3 (P3.4). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors. Timer/Counter 0 Input (T0). PWM Clock Input (PWMC). 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. Input for External Clock Signal (EXTCLK). This pin function must be enabled via the CFG842 register.
23	P3.5/T1/$\overline{\text{CONVST}}$	I/O	Input/Output Port 3 (P3.5). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors. Timer/Counter 1 Input (T1). Active Low Convert Start Logic Input for the ADC Block When the External Convert Start Function is Enabled ($\overline{\text{CONVST}}$). A low to high transition on this input puts the track-and-hold into hold mode and starts the conversion.
24	P3.6/$\overline{\text{WR}}$	I/O	Input/Output Port 3 (P3.6). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors. Write Control Signal, Logic Output ($\overline{\text{WR}}$). Latches the data byte from Port 0 into the external data memory.
25	P3.7/$\overline{\text{RD}}$	I/O	Input/Output Port 3 (P3.7). Port 3 is a bidirectional port with internal pull-up resistors. Port 3 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low source current because of the internal pull-up resistors. Read Control Signal, Logic Output ($\overline{\text{RD}}$). Enables the external data memory to Port 0.
26	SCLOCK	I/O	Serial Clock Pin for I^2C-Compatible Clock or for SPI Serial Interface Clock.
27	SDATA/MOSI	I/O	User Selectable, I^2C Compatible, or SPI Data Input/Output Pin (SDATA). SPI Master Output/Slave Input Data I/O Pin for SPI Interface (MOSI).
28	P2.0/A8/A16	I/O	Input/Output Port 2 (P2.0). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A8). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A16). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
29	P2.1/A9/A17	I/O	Input/Output Port 2 (P2.1). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A9). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A17). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.
30	P2.2/A10/A18	I/O	Input/Output Port 2 (P2.2). Port 2 is a bidirectional port with internal pull-up resistors. Port 2 pins that have 1s written to them are pulled high by the internal pull-up resistors, and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low source current because of the internal pull-up resistors. External Memory Addresses (A10). Port 2 emits the middle order address byte during accesses to the external 24-bit external data memory space. External Memory Addresses (A18). Port 2 emits the high order address byte during accesses to the external 24-bit external data memory space.

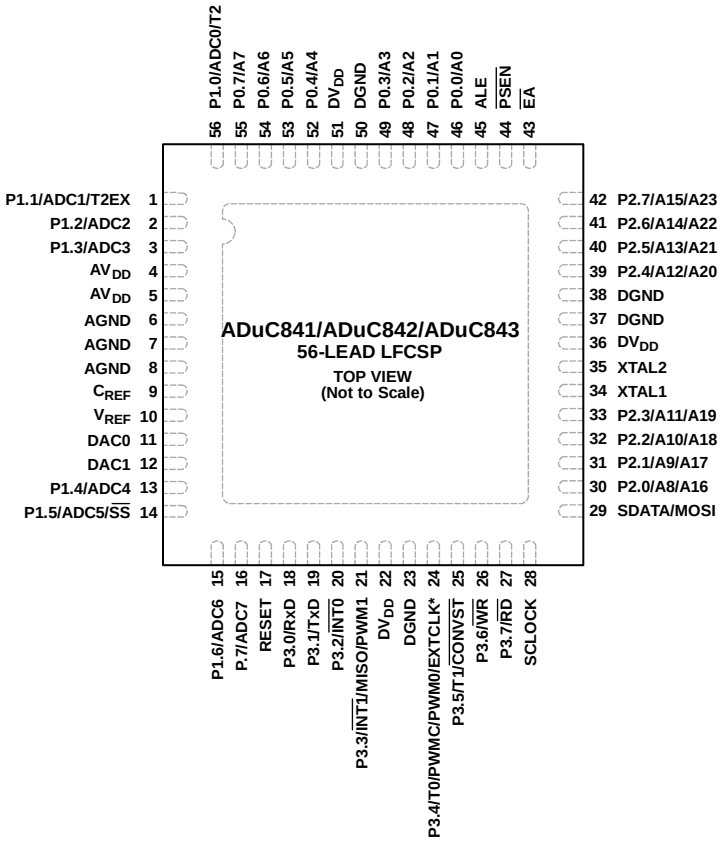


Figure 4. 56-Lead LFCSP Pin Configuration

Table 4. 56-Lead LFCSP Pin Function Descriptions

Pin No.	Mnemonic	Type[1]	Description
1	P1.1/ADC1/T2EX	I	Input Port 1 (P1.1). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input 1 (ADC1). Channel selection is via ADCCON2 SFR.
			Capture/Reload Trigger for Counter 2 (T2EX). Digital Input. This pin also functions as an up/down control input for Counter 2.
2	P1.2/ADC2	I	Input Port 1 (P1.2). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input (ADC2). Channel selection is via ADCCON2 SFR.
3	P1.3/ADC3	I	Input Port 1 (P1.3). Port 1 is an 8-bit input port only. Unlike the other ports, Port 1 defaults to analog input mode. To configure this port pin as a digital input, write a 0 to the port bit.
			Single-Ended Analog Input (ADC3). Channel selection is via ADCCON2 SFR.
4, 5	AV_{DD}	P	Analog Positive Supply Voltage. 3 V or 5 V nominal.
6, 7, 8	AGND	G	Analog Ground. AGND is the ground reference point for the analog circuitry.
9	C_{REF}	I/O	Decoupling Input for On-Chip Reference. Connect a 0.47 µF capacitor between this pin and AGND.
10	V_{REF}	NC	Not Connected. This was a reference output on the ADuC812; use the C_{REF} pin instead.
11	DAC0	O	Voltage Output from DAC0. This pin is a no connect on the ADuC843.

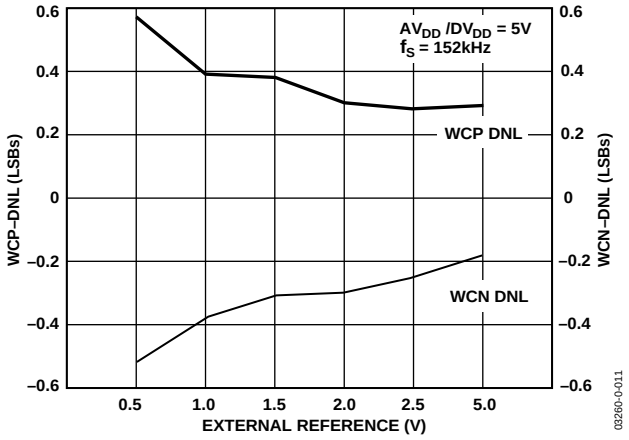


Figure 13. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 5 V

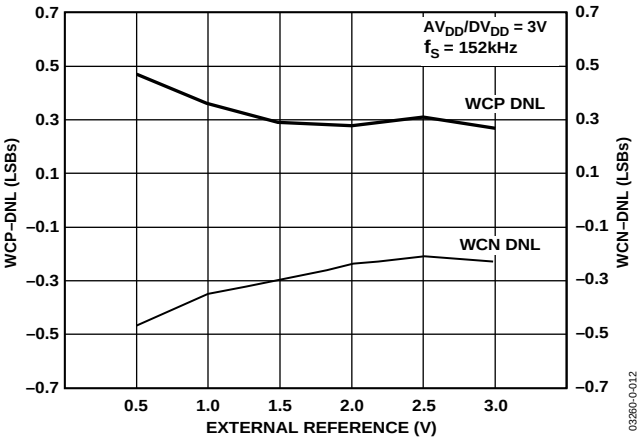


Figure 14. Typical Worst-Case DNL Error vs. V_{REF}, V_{DD} = 3 V

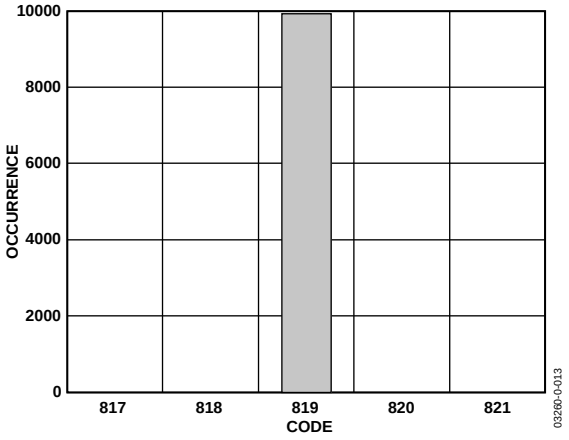


Figure 15. Code Histogram Plot, V_{DD} = 5 V

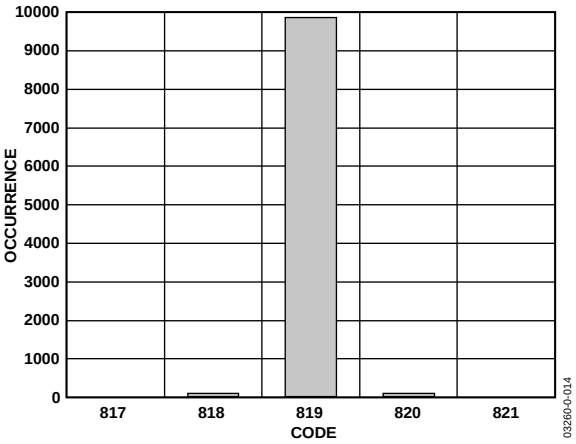


Figure 16. Code Histogram Plot, V_{DD} = 3 V

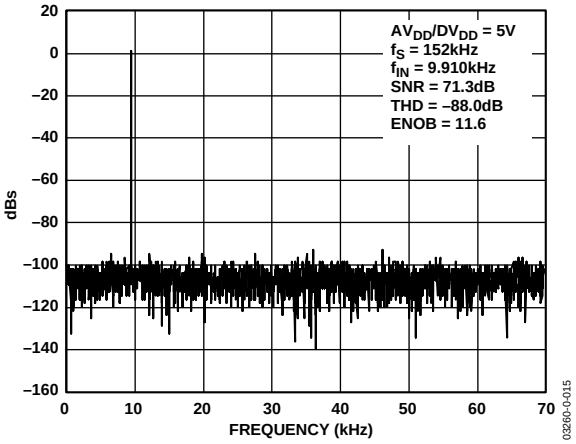


Figure 17. Dynamic Performance at V_{DD} = 5 V

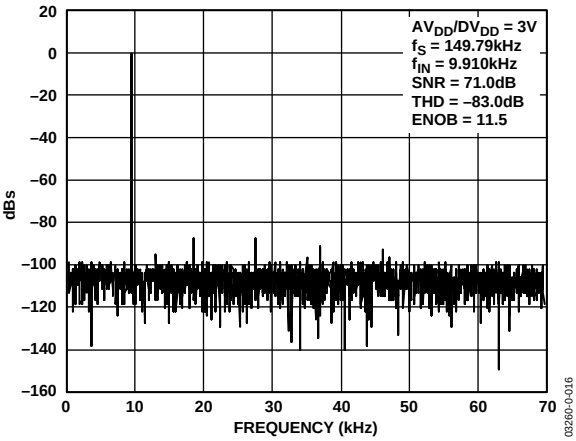


Figure 18. Dynamic Performance at V_{DD} = 3 V

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

MEMORY ORGANIZATION

The ADuC841/ADuC842/ADuC843 each contain four different memory blocks:

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

Flash/EE Program Memory

The parts provide up to 62 kBytes of Flash/EE program memory to run user code. The user can run code from this internal memory only. Unlike the ADuC812, where code execution can overflow from the internal code space to external code space once the PC becomes greater than 1FFFH, the parts do not support the roll-over from F7FFH in internal code space to F800H in external code space. Instead, the 2048 bytes between F800H and FFFFH appear as NOP instructions to user code.

This internal code space can be downloaded via the UART serial port while the device is in-circuit. 56 kBytes of the program memory can be reprogrammed during run time; thus the code space can be upgraded in the field by using a user defined protocol, or it can be used as a data memory. This is discussed in more detail in the Flash/EE Memory section.

For the 32 kBytes memory model, the top 8 kBytes function as the ULOAD space; this is explained in the Flash/EE Memory section.

Flash/EE Data Memory

4 kBytes of Flash/EE data memory are available to the user and can be accessed indirectly via a group of control registers mapped into the special function register (SFR) area. Access to the Flash/EE data memory is discussed in detail in the Flash/EE Memory section.

General-Purpose RAM

The general-purpose RAM is divided into two separate memories: the upper and the lower 128 bytes of RAM. The lower 128 bytes of RAM can be accessed through direct or indirect addressing. The upper 128 bytes of RAM can be accessed only through indirect addressing because it shares the same address space as the SFR space, which can be accessed only through direct addressing.

The lower 128 bytes of internal data memory are mapped as shown in Figure 23. 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 to 7FH. The stack can be located anywhere in the internal memory address space, and the stack depth can be expanded up to 2048 bytes.

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

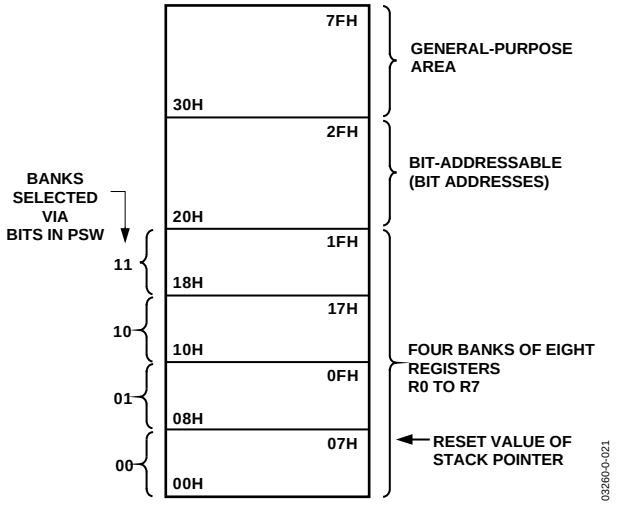


Figure 23. Lower 128 Bytes of Internal Data Memory

The parts contain 2048 bytes of internal XRAM, 1792 bytes of which can be configured to an extended 11-bit stack pointer.

By default, the stack operates exactly like an 8052 in that it rolls over from FFH to 00H in the general-purpose RAM. On the parts, however, it is possible (by setting CFG841.7 or CFG842.7) to enable the 11-bit extended stack pointer. In this case, the stack rolls over from FFH in RAM to 0100H in XRAM.

The 11-bit stack pointer is visible in the SP and SPH SFRs. The SP SFR is located at 81H as with a standard 8052. The SPH SFR is located at B7H. The 3 LSBs of this SFR contain the 3 extra bits necessary to extend the 8-bit stack pointer into an 11-bit stack pointer.

SPECIAL FUNCTION REGISTER BANKS

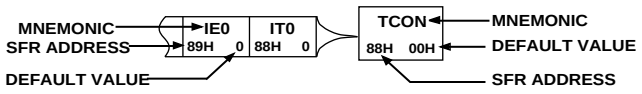
All registers except the program counter and the four general-purpose register banks reside in the special function register (SFR) area. The SFR registers include control, configuration, and data registers, which provide an interface between the CPU and other on-chip peripherals. Figure 27 shows a full SFR memory map and SFR contents on reset. Unoccupied SFR locations are shown dark-shaded in the figure (NOT USED). Unoccupied locations in the SFR address space are not implemented, that is, no register exists at this location. If an unoccupied location is read, an unspecified value is returned. SFR locations reserved for on-chip testing are shown lighter shaded (RESERVED) and should not be accessed by user software. Sixteen of the SFR locations are also bit addressable and denoted by 1 in Figure 27, that is, the bit addressable SFRs are those whose address ends in 0H or 8H.

ISPI FFH 0	WCOL FEH 0	SPE FDH 0	SPIM FCH 0	CPOL FBH 0	CPHA FAH 1	SPR1 F9H 0	SPR0 F8H 0	BITS	SPICON[1] F8H 04H
F7H 0	F6H 0	F5H 0	F4H 0	F3H 0	F2H 0	F1H 0	F0H 0	BITS	B[1] F0H 00H
I2CSI/MDO EFH 0	I2CGC/MDE EEH 0	I2C1O1MCO EDH 0	I2C1O0/MDI ECH 0	I2CM EBH 0	I2CRS EAH 0	I2CTX E9H 0	I2CI E8H 0	BITS	I2CCON[1] E8H 00H
E7H 0	E6H 0	E5H 0	E4H 0	E3H 0	E2H 0	E1H 0	E0H 0	BITS	ACC[1] E0H 00H
ADCI DFH 0	DMA DEH 0	CCONV DDH 0	SCONV DCH 0	CS3 DBH 0	CS2 DAH 0	CS1 D9H 0	CS0 D8H 0	BITS	ADCCON2[1] D8H 00H
CY D7H 0	AC D6H 0	F0 D5H 0	RS1 D4H 0	RS0 D3H 0	OV D2H 0	FI D1H 0	P D0H 0	BITS	PSW[1] D0H 00H
TF2 CFH 0	EXF2 CEH 0	RCLK CDH 0	TCLK CCH 0	EXEN2 CBH 0	TR2 CAH 0	CNT2 C9H 0	CAP2 C8H 0	BITS	T2CON[1] C8H 00H
PRE3 C7H 0	PRE2 C6H 0	PRE1 C5H 0	PRE0 C4H 1	WDIR C3H 0	WDS C2H 0	WDE C1H 0	WDWR C0H 0	BITS	WDCON[1] C0H 10H
PSI BFH 0	PADC BEH 0	PT2 BDH 0	PS BCH 0	PT1 BBH 0	PX1 BAH 0	PT0 B9H 0	PX0 B8H 0	BITS	IP[1] B8H 00H
RD B7H 1	WR B6H 1	T1 B5H 1	T0 B4H 1	INT1 B3H 1	INT0 B2H 1	TxD B1H 1	RxD B0H 1	BITS	P3[1] B0H FFH
EA AFH 0	EADC AEH 0	ET2 ADH 0	ES ACH 0	ET1 ABH 0	EX1 AAH 0	ET0 A9H 0	EX0 A8H 0	BITS	IE[1] A8H 00H
A7H 1	A6H 1	A5H 1	A4H 1	A3H 1	A2H 1	A1H 1	A0H 1	BITS	P2[1] A0H FFH
SM0 9FH 0	SM1 9EH 0	SM2 9DH 0	REN 9CH 0	TB8 9BH 0	RB8 9AH 0	TI 99H 0	RI 98H 0	BITS	SCON[1] 98H 00H
97H 1	96H 1	95H 1	94H 1	93H 1	92H 1	T2EX 91H 1	T2 90H 1	BITS	P1[1, 2] 90H FFH
TF1 8FH 0	TR1 8EH 0	TF0 8DH 0	TR0 8CH 0	IE1 8BH 0	IT1 8AH 0	IE0 89H 0	IT0 88H 0	BITS	TCON[1] 88H 00H
87H 1	86H 1	85H 1	84H 1	83H 1	82H 1	81H 1	80H 1	BITS	P0[1] 80H FFH

DAC0L F9H 00H	DAC0H FAH 00H	DAC1L FBH 00H	DAC1H FCH 00H	DACCON FDH 04H	RESERVED	RESERVED	
ADCOFSL[3] F1H 00H	ADCOFSH[3] F2H 20H	ADCGAINL[3] F3H 00H	ADCGAINH[3] F4H 00H	ADCCON3 F5H 00H	RESERVED	SPIDAT F7H 00H	
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	ADCCON1 EFH 40H	
RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	
ADCDATAL D9H 00H	ADCDATAH DAH 00H	RESERVED	RESERVED	RESERVED	RESERVED	PSMCON DFH DEH	
RESERVED	DMAL D2H 00H	DMAH D3H 00H	DMAP D4H 00H	RESERVED	RESERVED	PLLCON D7H 53H	
RESERVED	RCAP2L CAH 00H	RCAP2H CBH 00H	TL2 CCH 00H	TH2 CDH 00H	RESERVED	RESERVED	
RESERVED	CHIPID C2H XXH	RESERVED	RESERVED	RESERVED	EDARL C6H 00H	EDARH C7H 00H	
ECON B9H 00H	RESERVED	RESERVED	EDATA1 BCH 00H	EDATA2 BDH 00H	EDATA3 BEH 00H	EDATA4 BFH 00H	
PWM0L B1H 00H	PWM0H B2H 00H	PWM1L B3H 00H	PWM1H B4H 00H	NOT USED	NOT USED	SPH B7H 00H	
IEIP2 A9H A0H	RESERVED	RESERVED	RESERVED	RESERVED	PWMCON AEH 00H	CFG841/ CFG842 AFH 00H	
TIMECON A1H 00H	HTHSEC A2H 00H	SEC A3H 00H	MIN A4H 00H	HOUR A5H 00H	INTVAL A6H 00H	DPCON A7H 00H	
SBUF 99H 00H	I2CDAT 9AH 00H	I2CADD 9BH 55H	NOT USED	T3FD 9DH 00H	T3CON 9EH 00H	NOT USED	
I2CADD1 91H 7FH	I2CADD2 92H 7FH	I2CADD3 93H 7FH	NOT USED	NOT USED	NOT USED	NOT USED	
TMOD 89H 00H	TL0 8AH 00H	TL1 8BH 00H	TH0 8CH 00H	TH1 8DH 00H	RESERVED	RESERVED	
SP 81H 07H	DPL 82H 00H	DPH 83H 00H	DPP 84H 00H	RESERVED	RESERVED	PCON 87H 00H	

SFR MAP KEY:

THESE BITS ARE CONTAINED IN THIS BYTE.



NOTES

[1]SFRs WHOSE ADDRESS ENDS IN 0H OR 8H ARE BIT ADDRESSABLE.

[2]THE PRIMARY FUNCTION OF PORT1 IS AS AN ANALOG INPUT PORT; THEREFORE, TO ENABLE THE DIGITAL SECONDARY FUNCTIONS ON THESE PORT PINS, WRITE A 0 TO THE CORRESPONDING PORT 1 SFR BIT.

[3]CALIBRATION COEFFICIENTS ARE PRECONFIGURED ON POWER-UP TO FACTORY CALIBRATED VALUES.

03260-0-025

Figure 27. Special Function Register Locations and Reset Values

ADCCON2—(ADC Control SFR 2)

The ADCCON2 register controls ADC channel selection and conversion modes as detailed below.

SFR Address	D8H
SFR Power-On Default	00H
Bit Addressable	Yes

Table 9. ADCCON2 SFR Bit Designations

Bit No.	Name	Description
7	ADCI	ADC Interrupt Bit. Set by hardware at the end of a single ADC conversion cycle or at the end of a DMA block conversion. Cleared by hardware when the PC vectors to the ADC interrupt service routine. Otherwise, the ADCI bit is cleared by user code.
6	DMA	DMA Mode Enable Bit. Set by the user to enable a preconfigured ADC DMA mode operation. A more detailed description of this mode is given in the ADC DMA Mode section. The DMA bit is automatically set to 0 at the end of a DMA cycle. Setting this bit causes the ALE output to cease; it starts again when DMA is started and operates correctly after DMA is complete.
5	CCONV	Continuous Conversion Bit. Set by the user to initiate the ADC into a continuous mode of conversion. In this mode, the ADC starts converting based on the timing and channel configuration already set up in the ADCCON SFRs; the ADC automatically starts another conversion once a previous conversion has completed.
4	SCONV	Single Conversion Bit. Set to initiate a single conversion cycle. The SCONV bit is automatically reset to 0 on completion of the single conversion cycle.
3 2 1 0	CS3 CS2 CS1 CS0	Channel Selection Bits. Allow the user to program the ADC channel selection under software control. When a conversion is initiated, the converted channel is the one pointed to by these channel selection bits. In DMA mode, the channel selection is derived from the channel ID written to the external memory.

CS3	CS2	CS1	CS0	CH#	
0	0	0	0	0	
0	0	0	1	1	
0	0	1	0	2	
0	0	1	1	3	
0	1	0	0	4	
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0	Temp Monitor	Requires minimum of 1 µs to acquire.
1	0	0	1	DAC0	Only use with internal DAC output buffer on.
1	0	1	0	DAC1	Only use with internal DAC output buffer on.
1	0	1	1	AGND	
1	1	0	0	V_{REF}	
1	1	1	1	DMA STOP	Place in XRAM location to finish DMA sequence; refer to the ADC DMA Mode section.

All other combinations reserved.

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.

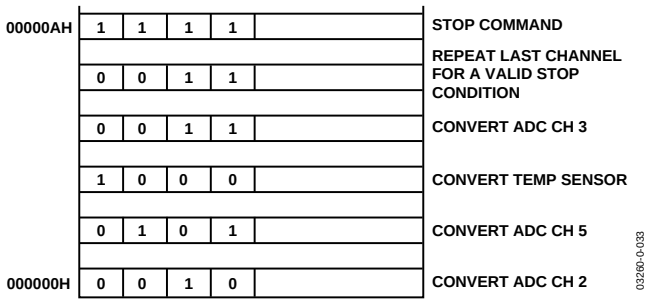


Figure 34. Typical DMA External Memory Preconfiguration

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.

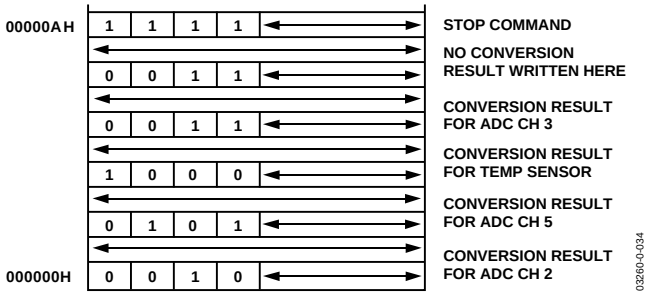


Figure 35. Typical External Memory Configuration Post ADC DMA Operation

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.

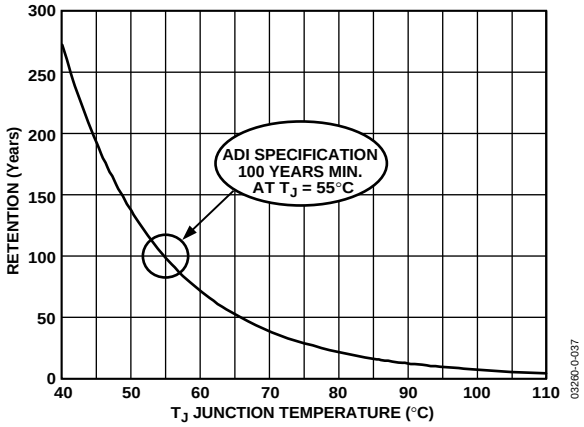


Figure 38. Flash/EE Memory Data Retention

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.

Example: Programming the Flash/EE Data Memory

A user wants to program F3H into the second byte on Page 03H of the Flash/EE data memory space while preserving the other 3 bytes already in this page. A typical program of the Flash/EE data array involves

1. Setting EADRH/L with the page address.
2. Writing the data to be programmed to the EDATA1–4.
3. Writing the ECON SFR with the appropriate command.

Step 1: Set Up the Page Address

Address registers EADRH and EADRL hold the high byte address and the low byte address of the page to be addressed. The assembly language to set up the address may appear as

```
MOV EADRH,#0            ; Set Page Address Pointer
MOV EADRL,#03H
```

Step 2: Set Up the EDATA Registers

Write the four values to be written into the page into the four SFRs, EDATA1–4. Unfortunately, the user does not know three of them. Thus, the user must read the current page and overwrite the second byte.

```
MOV ECON,#1             ; Read Page into EDATA1-4
MOV EDATA2,#0F3H        ; Overwrite byte 2
```

Step 3: Program Page

A byte in the Flash/EE array can be programmed only if it has previously been erased. To be more specific, a byte can be programmed only if it already holds the value FFH. Because of the Flash/EE architecture, this erase must happen at a page level; therefore, a minimum of 4 bytes (1 page) are erased when an erase command is initiated. Once the page is erase, the user can program the 4 bytes in-page and then perform a verification of the data.

```
MOV ECON,#5             ; ERASE Page
MOV ECON,#2             ; WRITE Page
MOV ECON,#4             ; VERIFY Page
MOV A,ECON              ; Check if ECON=0 (OK!)
JNZ ERROR
```

Although the 4 kBytes of Flash/EE data memory are shipped from the factory pre-erased, that is, byte locations set to FFH, it is nonetheless good programming practice to include an ERASEALL routine as part of any configuration/setup code running on the parts. An ERASEALL command consists of writing 06H to the ECON SFR, which initiates an erase of the 4-kByte Flash/EE array. This command coded in 8051 assembly would appear as

```
MOV ECON,#06H           ; Erase all Command
                        ; 2 ms Duration
```

Flash/EE Memory Timing

Typical program and erase times for the parts are as follows:

Normal Mode (operating on Flash/EE data memory)

READPAGE (4 bytes)	22 machine cycles
WRITEPAGE (4 bytes)	380 µs
VERIFYPAGE (4 bytes)	22 machine cycles
ERASEPAGE (4 bytes)	2 ms
ERASEALL (4 kBytes)	2 ms
READBYTE (1 byte)	9 machine cycles
WRITEBYTE (1 byte)	200 µs

ULOAD Mode (operating on Flash/EE program memory)

WRITEPAGE (256 bytes)	16.5 ms
ERASEPAGE (64 bytes)	2 ms
ERASEALL (56 kBytes)	2 ms
WRITEBYTE (1 byte)	200 µs

Note that a given mode of operation is initiated as soon as the command word is written to the ECON SFR. The core microcontroller operation on the parts is idled until the requested program/read or erase mode is completed. In practice, this means that even though the Flash/EE memory mode of operation is typically initiated with a two machine cycle MOV instruction (to write to the ECON SFR), the next instruction is not executed until the Flash/EE operation is complete. This means that the core cannot respond to interrupt requests until the Flash/EE operation is complete, although the core peripheral functions like counter/timers continue to count and time as configured throughout this period.

ON-CHIP PLL

The ADuC842 and ADuC843 are intended for use with a 32.768 kHz watch crystal. A PLL locks onto a multiple (512) of this to provide a stable 16.78 MHz clock for the system. The ADuC841 operates directly from an external crystal. The core can operate at this frequency or at binary submultiples of it to allow power saving in cases where maximum core performance is not required. The default core clock is the PLL clock divided by 8 or 2.097152 MHz. The ADC clocks are also derived from the PLL clock, with the modulator rate being the same as the crystal oscillator frequency. The preceding choice of frequencies ensures that the modulators and the core are synchronous, regardless of the core clock rate. The PLL control register is PLLCON.

At 5 V the core clock can be set to a maximum of 16.78 MHz, while at 3 V the maximum core clock setting is 8.38 MHz. The CD bits should not be set to 0 on a 3 V part.

Note that on the ADuC841, changing the CD bits in PLLCON causes the core speed to change. The core speed is crystal freq/2^{CD}. The other bits in PLLCON are reserved in the case of the ADuC841 and should be written with 0.

PLLCON PLL	**Control Register**
SFR Address	D7H
Power-On Default	53H
Bit Addressable	No

Table 17. PLLCON SFR Bit Designations

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

PWM Modes of Operation

Mode 0: PWM Disabled

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

Mode 1: Single Variable Resolution PWM

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

PWM1H/L sets the period of the output waveform. Reducing PWM1H/L reduces the resolution of the PWM output but increases the maximum output rate of the PWM. For example, setting PWM1H/L to 65536 gives a 16-bit PWM with a maximum output rate of 266 Hz (16.777 MHz/65536). Setting PWM1H/L to 4096 gives a 12-bit PWM with a maximum output rate of 4096 Hz (16.777 MHz/4096).

PWM0H/L sets the duty cycle of the PWM output waveform, as shown in Figure 48.

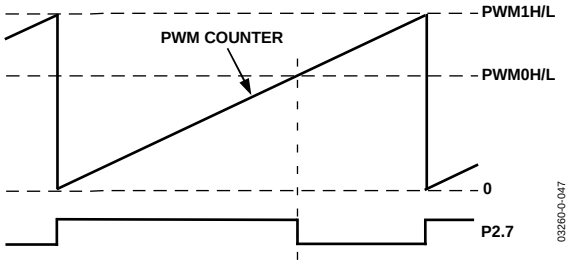


Figure 48. PWM in Mode 1

Mode 2: Twin 8-Bit PWM

In Mode 2, the duty cycle of the PWM outputs and the resolution of the PWM outputs are both programmable. The maximum resolution of the PWM output is 8 bits.

PWM1L sets the period for both PWM outputs. Typically, this is set to 255 (FFH) to give an 8-bit PWM, although it is possible to reduce this as necessary. A value of 100 could be loaded here to give a percentage PWM, that is, the PWM is accurate to 1%.

The outputs of the PWM at P2.6 and P2.7 are shown in Figure 49. As can be seen, the output of PWM0 (P2.6) goes low when the PWM counter equals PWM0L. The output of PWM1 (P2.7) goes high when the PWM counter equals PWM1H and goes low again when the PWM counter equals PWM0H. Setting PWM1H to 0 ensures that both PWM outputs start simultaneously.

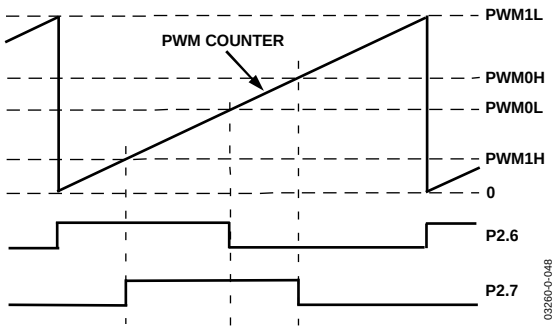


Figure 49. PWM Mode 2

Mode 3: Twin 16-Bit PWM

In Mode 3, the PWM counter is fixed to count from 0 to 65536, giving a fixed 16-bit PWM. Operating from the 16.777 MHz core clock results in a PWM output rate of 256 Hz. The duty cycle of the PWM outputs at P2.6 and P2.7 is independently programmable.

As shown in Figure 50, while the PWM counter is less than PWM0H/L, the output of PWM0 (P2.6) is high. Once the PWM counter equals PWM0H/L, PWM0 (P2.6) goes low and remains low until the PWM counter rolls over.

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

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

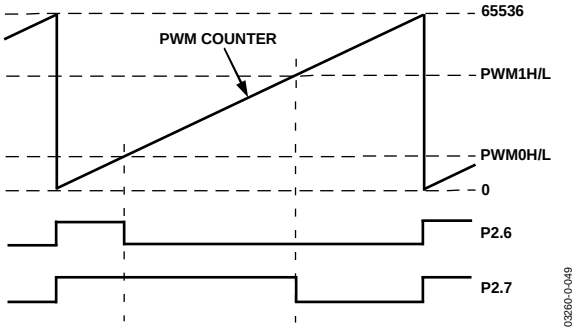


Figure 50. PWM Mode 3

- An I²C slave can respond to repeated start conditions without a stop bit in between. This allows a master to change direction of transfer without giving up the bus. Note that the repeated start is detected only when a slave has previously been configured as a receiver.
- On-chip filtering rejects <50 ns spikes on the SDATA and the SCLOCK lines to preserve data integrity.

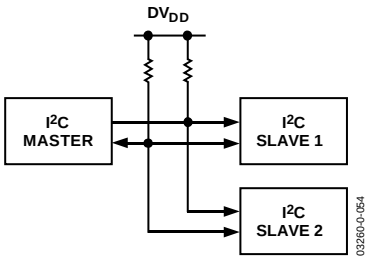


Figure 55. Typical I²C System

Software Master Mode

The ADuC841/ADuC842/ADuC843 can be used as I²C master devices by configuring the I²C peripheral in master mode and writing software to output the data bit by bit. This is referred to as a software master. Master mode is enabled by setting the I2CM bit in the I2CCON register.

To transmit data on the SDATA line, MDE must be set to enable the output driver on the SDATA pin. If MDE is set, the SDATA pin is pulled high or low depending on whether the MDO bit is set or cleared. MCO controls the SCLOCK pin and is always configured as an output in master mode. In master mode, the SCLOCK pin is pulled high or low depending on the whether MCO is set or cleared.

To receive data, MDE must be cleared to disable the output driver on SDATA. Software must provide the clocks by toggling the MCO bit and reading the SDATA pin via the MDI bit. If MDE is cleared, MDI can be used to read the SDATA pin. The value of the SDATA pin is latched into MDI on a rising edge of SCLOCK. MDI is set if the SDATA pin was high on the last rising edge of SCLOCK. MDI is clear if the SDATA pin was low on the last rising edge of SCLOCK.

Software must control MDO, MCO, and MDE appropriately to generate the start condition, slave address, acknowledge bits, data bytes, and stop conditions. These functions are described in Application Note uC001.

Hardware Slave Mode

After reset, the ADuC841/ADuC842/ADuC843 default to hardware slave mode. The I²C interface is enabled by clearing the SPE bit in SPICON (this is not necessary if the MSPI bit is set). Slave mode is enabled by clearing the I2CM bit in I2CCON. The parts have a full hardware slave. In slave mode, the I²C address is stored in the I2CADD register. Data received or to be transmitted is stored in the I2CDAT register.

Once enabled in I²C slave mode, the slave controller waits for a start condition. If the part detects a valid start condition, followed by a valid address, followed by the R/$\overline{W}$ bit, the I2CI interrupt bit is automatically set by hardware. The I²C peripheral generates a core interrupt only if the user has pre-configured the I²C interrupt enable bit in the IEIP2 SFR as well as the global interrupt bit, $\overline{EA}$, in the IE SFR. That is,

```
;Enabling I2C Interrupts for the ADuC842
MOV IEIP2,#01h            ; enable I2C interrupt
SETB EA
```

An autoclear of the I2CI bit is implemented on the parts so that this bit is cleared automatically on a read or write access to the I2CDAT SFR.

```
MOV I2CDAT, A             ; I2CI auto-cleared
MOV A, I2CDAT             ; I2CI auto-cleared
```

If for any reason the user tries to clear the interrupt more than once, that is, access the data SFR more than once per interrupt, then the I²C controller halts. The interface then must be reset using the I2CRS bit.

The user can choose to poll the I2CI bit or to enable the interrupt. In the case of the interrupt, the PC counter vectors to 003BH at the end of each complete byte. For the first byte, when the user gets to the I2CI ISR, the 7-bit address and the R/$\overline{W}$ bit appear in the I2CDAT SFR.

The I2CTX bit contains the R/$\overline{W}$ bit sent from the master. If I2CTX is set, the master is ready to receive a byte. Therefore the slave transmits data by writing to the I2CDAT register. If I2CTX is cleared, the master is ready to transmit a byte. Therefore the slave receives a serial byte. Software can interrogate the state of I2CTX to determine whether it must write to or read from I2CDAT.

Once the part has received a valid address, hardware holds SCLOCK low until the I2CI bit is cleared by software. This allows the master to wait for the slave to be ready before transmitting the clocks for the next byte.

The I2CI interrupt bit is set every time a complete data byte is received or transmitted, provided it is followed by a valid ACK. If the byte is followed by a NACK, an interrupt is not generated.

The part continues to issue interrupts for each complete data byte transferred until a stop condition is received or the interface is reset.

When a stop condition is received, the interface resets to a state in which it is waiting to be addressed (idle). Similarly, if the interface receives a NACK at the end of a sequence, it also returns to the default idle state. The I2CRS bit can be used to reset the I²C interface. This bit can be used to force the interface back to the default idle state.

DUAL DATA POINTER

The ADuC841/ADuC842/ADuC843 incorporate two data pointers. The second data pointer is a shadow data pointer and is selected via the data pointer control SFR (DPCON). DPCON also includes some useful features such as automatic hardware post-increment and post-decrement as well as automatic data pointer toggle. DPCON is described in Table 22.

DPCON	Data Pointer Control SFR
SFR Address	A7H
Power-On Default	00H
Bit Addressable	No

Table 22. DPCON SFR Bit Designations

Bit No.	Name	Description
7	----	Reserved.
6	DPT	Data Pointer Automatic Toggle Enable. Cleared by the user to disable autoswapping of the DPTR. Set in user software to enable automatic toggling of the DPTR after each each MOVX or MOVC instruction.
5	DP1m1	Shadow Data Pointer Mode.
4	DP1m0	These two bits enable extra modes of the shadow data pointer's operation, allowing for more compact and more efficient code size and execution. m1 m0 Behavior of the shadow data pointer. 0 0 8052 behavior. 0 1 DPTR is post-incremented after a MOVX or a MOVC instruction. 1 0 DPTR is post-decremented after a MOVX or MOVC instruction. 1 1 DPTR LSB is toggled after a MOVX or MOVC instruction. (This instruction can be useful for moving 8-bit blocks to/from 16-bit devices.)
3	DP0m1	Main Data Pointer Mode.
2	DP0m0	These two bits enable extra modes of the main data pointer operation, allowing for more compact and more efficient code size and execution. m1 m0 Behavior of the main data pointer. 0 0 8052 behavior. 0 1 DPTR is post-incremented after a MOVX or a MOVC instruction. 1 0 DPTR is post-decremented after a MOVX or MOVC instruction. 1 1 DPTR LSB is toggled after a MOVX or MOVC instruction. (This instruction can be useful for moving 8-bit blocks to/from 16-bit devices.)
1	----	This bit is not implemented to allow the INC DPCON instruction toggle the data pointer without incrementing the rest of the SFR.
0	DPSEL	Data Pointer Select. Cleared by the user to select the main data pointer. This means that the contents of this 24-bit register are placed into the three SFRs: DPL, DPH, and DPP. Set by the user to select the shadow data pointer. This means that the contents of a separate 24-bit register appears in the three SFRs: DPL, DPH, and DPP.

Note 1: This is the only place where the main and shadow data pointers are distinguished. Everywhere else in this data sheet wherever the DPTR is mentioned, operation on the active DPTR is implied.

Note 2: Only MOVC/MOVX @DPTR instructions are relevant above. MOVC/MOVX PC/@Ri instructions do not cause the DPTR to automatically post increment/decrement, and so on.

To illustrate the operation of DPCON, the following code copies 256 bytes of code memory at address D000H into XRAM starting from Address 0000H.

```
    MOV DPTR,#0          ; Main DPTR = 0
    MOV DPCON,#55H       ; Select shadow DPTR
                         ; DPTR1 increment mode,
                         ; DPTR0 increment mode
                         ; DPTR auto toggling ON
    MOV DPTR,#0D000H     ; Shadow DPTR = D000H
MOVELOOP:
    CLR A
    MOVC A,@A+DPTR       ; Get data
                         ; Post Inc DPTR
                         ; Swap to Main DPTR (Data)
    MOVX @DPTR,A         ; Put ACC in XRAM
                         ; Increment main DPTR
                         ; Swap Shadow DPTR (Code)
    MOV A, DPL
    JNZ MOVELOOP
```

WATCHDOG TIMER

The purpose of the watchdog timer is to generate a device reset or interrupt within a reasonable amount of time if the ADuC841/ADuC842/ADuC843 enter an erroneous state, possibly due to a programming error or electrical noise. The watchdog function can be disabled by clearing the WDE (watchdog enable) bit in the watchdog control (WDCON) SFR. When enabled, the watchdog circuit generates a system reset or interrupt (WDS) if the user program fails to set the watchdog (WDE) bit within a predetermined amount of time (see PRE3-0 bits in Table 24. The watchdog timer is clocked directly from the 32 kHz external crystal on the ADuC842/ADuC843. On the ADuC841, the watchdog timer is clocked by an internal R/C oscillator at 32 kHz ±10%. The WDCON SFR can be written only by user software if the double write sequence described in WDWR below is initiated on every write access to the WDCON SFR.

WDCON Watchdog Timer	Control Register
SFR Address	C0H
Power-On Default	10H
Bit Addressable	Yes

Table 24. WDCON SFR Bit Designations

Bit No.	Name	Description
7	PRE3	Watchdog Timer Prescale Bits.
6	PRE2	The watchdog timeout period is given by the equation $t_{WD} = (2^{PRE} \times (2^9/f_{XTAL}))$
5	PRE1	(0 – PRE – 7; f_{XTAL} = 32.768 kHz (ADuC842/ADuC843), or 32kHz ± 10% (ADuC841))
4	PRE0	PRE3 PRE2 PRE1 PRE0 Timeout Period (ms) Action 0 0 0 0 15.6 Reset or Interrupt 0 0 0 1 31.2 Reset or Interrupt 0 0 1 0 62.5 Reset or Interrupt 0 0 1 1 125 Reset or Interrupt 0 1 0 0 250 Reset or Interrupt 0 1 0 1 500 Reset or Interrupt 0 1 1 0 1000 Reset or Interrupt 0 1 1 1 2000 Reset or Interrupt 1 0 0 0 0.0 Immediate Reset PRE3–0 > 1000 Reserved
3	WDIR	Watchdog Interrupt Response Enable Bit. If this bit is set by the user, the watchdog generates an interrupt response instead of a system reset when the watchdog timeout period has expired. This interrupt is not disabled by the CLR $\overline{EA}$ instruction, and it is also a fixed, high priority interrupt. If the watchdog is not being used to monitor the system, it can be used alternatively as a timer. The prescaler is used to set the timeout period in which an interrupt is generated.
2	WDS	Watchdog Status Bit. Set by the watchdog controller to indicate that a watchdog timeout has occurred. Cleared by writing a 0 or by an external hardware reset. It is not cleared by a watchdog reset.
1	WDE	Watchdog Enable Bit. Set by the user to enable the watchdog and clear its counters. If this bit is not set by the user within the watchdog timeout period, the watchdog generates a reset or interrupt, depending on WDIR. Cleared under the following conditions: user writes 0, watchdog reset (WDIR = 0); hardware reset; PSM interrupt.

0	WDWR	Watchdog Write Enable Bit.
		To write data to the WDCON SFR involves a double instruction sequence. The WDWR bit must be set and the very next instruction must be a write instruction to the WDCON SFR.
		For example:
		`CLR EA ;disable interrupts while writing`
		`;to WDT`
		`SETB WDWR ;allow write to WDCON`
		`MOV WDCON,#72H ;enable WDT for 2.0s timeout`
		`SETB EA ;enable interrupts again (if rqd)`

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.

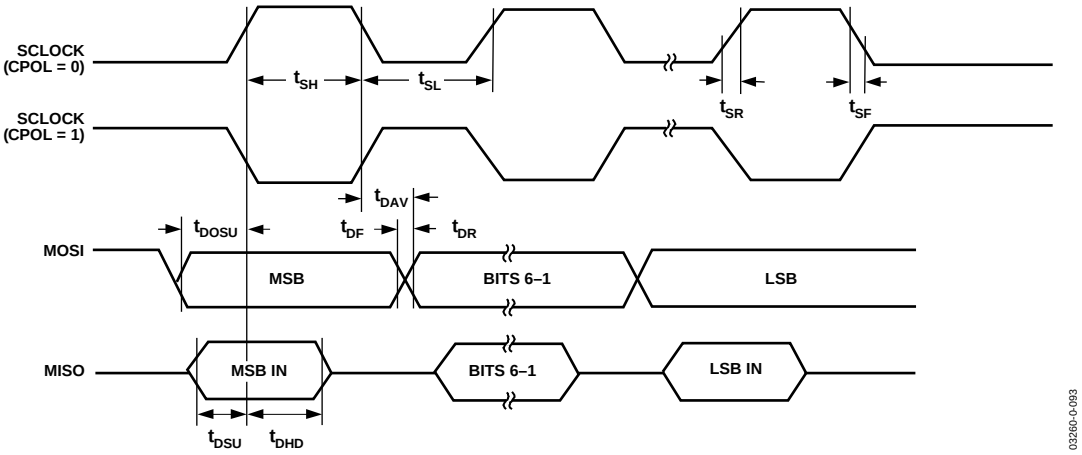


Figure 92. SPI Master Mode Timing (CPHA = 0)

Parameter					
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

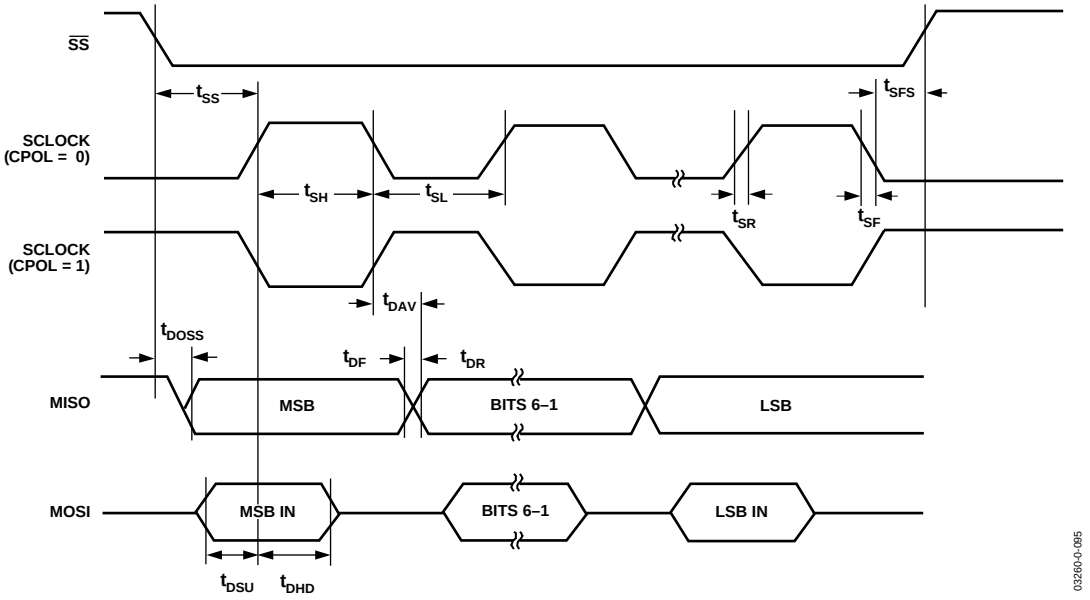


Figure 94. SPI Slave Mode Timing (CPHA = 0)