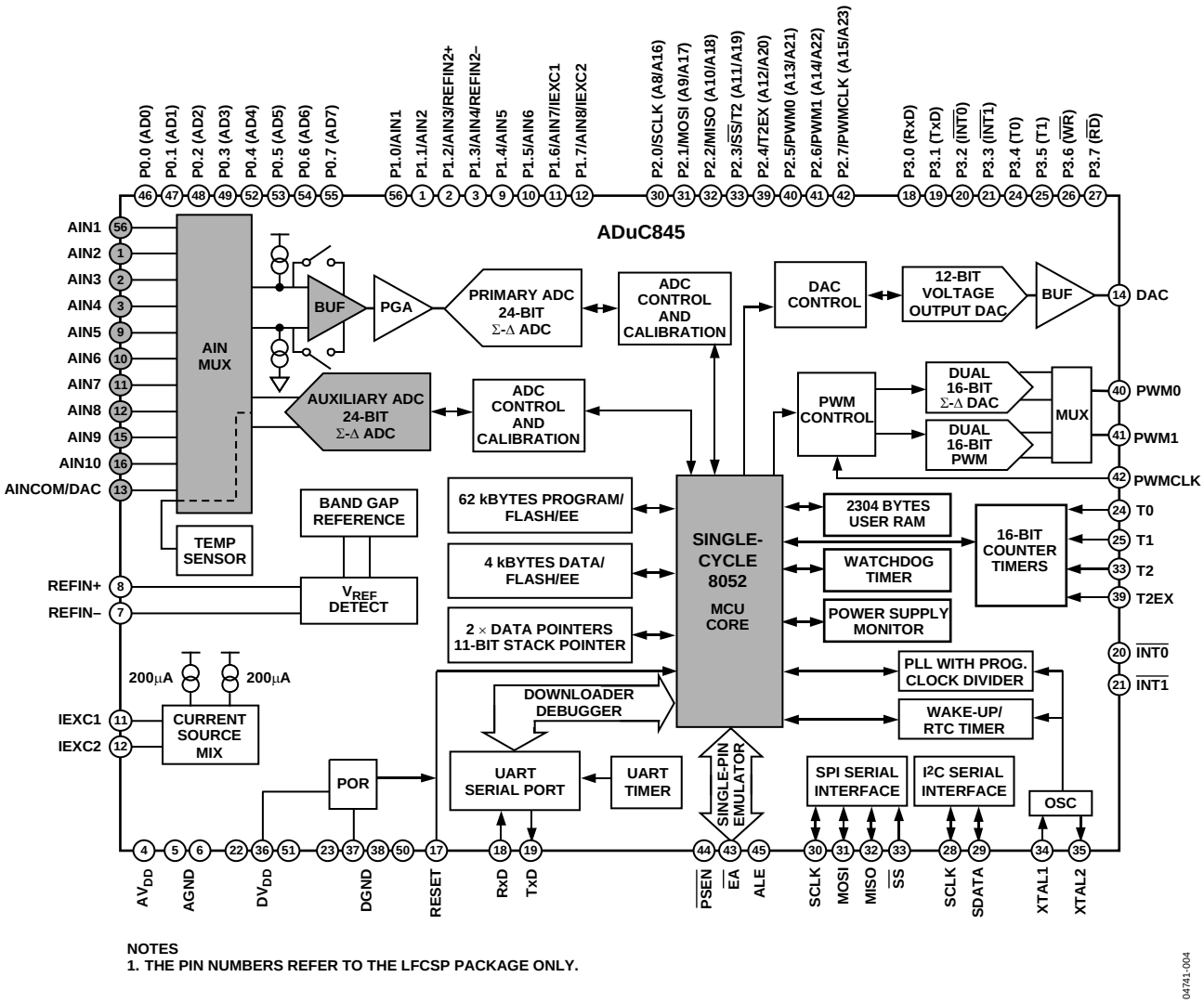
Welcome to E-XFL.COM

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	12.58MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	POR, PSM, PWM, Temp Sensor, WDT
Number of I/O	34
Program Memory Size	62KB (62K x 8)
Program Memory Type	FLASH
EEPROM Size	4K x 8
RAM Size	2.25K x 8
Voltage - Supply (Vcc/Vdd)	4.75V ~ 5.25V
Data Converters	A/D 10x24b; D/A 1x12b, 2x16b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	52-QFP
Supplier Device Package	52-MQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc845bsz62-5



NOTES
1. THE PIN NUMBERS REFER TO THE LFCSP PACKAGE ONLY.

Figure 4. Detailed Block Diagram of the ADuC845

Mnemonic	Description	Bytes	Cycles[1]
SJMP rel	Short jump (relative address)	2	3
JC rel	Jump on carry = 1	2	3
JNC rel	Jump on carry = 0	2	3
JZ rel	Jump on accumulator = 0	2	3
JNZ rel	Jump on accumulator ! = 0	2	3
DJNZ Rn,rel	Decrement register, JNZ relative	2	3
LJMP	Long jump unconditional	3	4
LCALL[3] addr16	Long jump to subroutine	3	4
JB bit,rel	Jump on direct bit = 1	3	4
JNB bit,rel	Jump on direct bit = 0	3	4
JBC bit,rel	Jump on direct bit = 1 and clear	3	4
CJNE A,dir,rel	Compare A, direct JNE relative	3	4
CJNE A,#data,rel	Compare A, immediate JNE relative	3	4
CJNE Rn,#data,rel	Compare register, immediate JNE relative	3	4
CJNE @Ri,#data,rel	Compare indirect, immediate JNE relative	3	4
DJNZ dir,rel	Decrement direct byte, JNZ relative	3	4
Miscellaneous			
NOP	No operation	1	1

[1] One cycle is one clock.
[2] MOVX instructions are four cycles when they have 0 wait state. Cycles of MOVX instructions are 4 + *n* cycles when they have *n* wait states as programmed via EWAIT.
[3] LCALL instructions are three cycles when the LCALL instruction comes from an interrupt.

MEMORY ORGANIZATION

The ADuC845, ADuC847, and ADuC848 contain four memory blocks:

- 62 kbytes/32 kbytes/8 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 devices provide up to 62 kbytes of Flash/EE program memory to run user code. All further references to Flash/EE program memory assume the 62-kbyte option.

When $\overline{EA}$ is pulled high externally during a power cycle or a hardware reset, the devices default to code execution from their internal 62 kbytes of Flash/EE program memory. The devices do not support the rollover from internal code space to external code space. No external code space is available on the devices. Permanently embedded firmware allows code to be serially downloaded to the 62 kbytes of internal code space via the UART serial port while the device is in-circuit. No external hardware is required.

During run time, 56 kbytes of the 62-kbyte program memory can be reprogrammed. This means that the code space can be upgraded in the field by using a user-defined protocol running on the devices, or it can be used as a data memory. For details, see the Nonvolatile Flash/EE Memory Overview section.

Flash/EE Data Memory

The user has 4 kbytes of Flash/EE data memory available that can be accessed indirectly by using a group of registers mapped into the special function register (SFR) space. For details, see the Nonvolatile Flash/EE Memory Overview 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 must be accessed through direct addressing.

The lower 128 bytes of internal data memory are mapped as shown in Figure 8. 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. Any call or push pre-increments the SP before loading the stack. Therefore, loading the stack starts from location 08H, which is also the first register (R0) of Register Bank 1. Thus, if one is going to use more than one register bank, the stack pointer should be initialized to an area of RAM not used for data storage.

Signal Chain Overview with Chop Disabled ($\overline{CHOP}$ = 1)

With $\overline{CHOP}$ = 1, chop is disabled and the available output rates vary from 16.06 Hz to 1.365 kHz. The range of applicable SF words is from 3 to 255. When switching between channels with chop disabled, the channel throughput rate is higher than when chop is enabled. The drawback with chop disabled is that the drift performance is degraded and offset calibration is required following a gain range change or significant temperature change. A block diagram of the ADC input channel with chop disabled is shown in Figure 15.

The signal chain includes a multiplex or buffer, PGA, Σ-Δ modulator, and digital filter. The modulator bit stream is applied to a Sinc^3 filter. Programming the Sinc^3 decimation factor is restricted to an 8-bit register SF; the actual decimation factor is the register value times 8. The decimated output rate from the Sinc^3 filter (and the ADC conversion rate) is therefore

$$f_{ADC} = \frac{1}{8 \times SF} \times f_{MOD}$$

where:
f_{ADC} is the ADC conversion rate.
SF is the decimal equivalent of the word loaded to the filter register, valid range is from 3 to 255.
f_{MOD} is the modulator sampling rate of 32.768 kHz.

The settling time to a step input is governed by the digital filter. A synchronized step change requires a settling time of three times the programmed update rate; a channel change can be treated as a synchronized step change. This is one conversion longer than the case for chop enabled. However, because the ADC throughput is three times faster with chop disabled than it is with chop enabled, the actual time to a settled ADC output is significantly less also. This means that following a synchronized step change, the ADC requires three conversions (note: data is not output following a synchronized ADC change until data has settled) before the result accurately reflects the new input voltage.

$$t_{SETTLE} = \frac{3}{f_{ADC}} = 3 \times t_{ADC}$$

An unsynchronized step change requires four conversions to accurately reflect the new analog input at its output. Note that with an unsynchronized change the ADC continues to output data and so the user must take unsettled outputs into account. Again, this is one conversion longer than with chop enabled, but because the ADC throughput with chop disabled is faster than with chop enabled, the actual time taken to obtain a settled ADC output is less.

The allowable range for SF is 3 to 255 with a default of 69 (45H). The corresponding conversion rates, rms, and peak-to-peak noise performances are shown in Table 14, Table 15, Table 16, and Table 17. Note that the conversion time increases by 0.244 ms for each increment in SF.

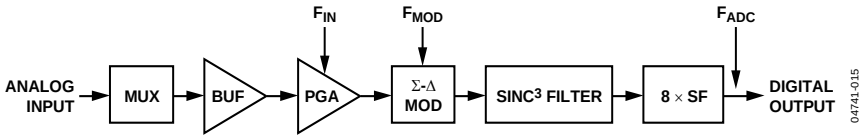


Figure 15. Block Diagram of ADC Input Channel with Chop Disabled

(REJ60 bit, ADCMODE.6). This fixed filter can be enabled or disabled by setting or clearing the REJ60 bit in the ADCMODE register (ADCMODE.6). This 60 Hz drop-in notch filter can be enabled for any SF word that yields an ADC throughput that is less than 20 Hz with chop enabled (SF ≥ 68 decimal).

ADC CHOPPING

The ADCs on the ADuC845/ADuC847/ADuC848 implement a chopping scheme whereby the ADC repeatedly reverses its inputs. The decimated digital output words from the Sinc^3 filter, therefore, have a positive and negative offset term included. As a result, a final summing stage is included in each ADC so that each output word from the filter is summed and averaged with the previous filter output to produce a new valid output result to be written to the ADC data SFRs. The ADC throughput or update rate is listed in Table 29. The chopping scheme incorporated into the devices results in excellent dc offset and offset drift specifications, and is extremely beneficial in applications where drift, noise rejection, and optimum EMI performance are important. ADC chop can be disabled via the chop bit in the ADCMODE SFR (ADCMODE.3). Setting this bit to 1 (logic high) disables chop mode.

CALIBRATION

The ADuC845/ADuC847/ADuC848 incorporate four calibration modes that can be programmed via the mode bits in the ADCMODE SFR detailed in Table 24. Every device is calibrated before it leaves the factory. The resulting offset and gain calibration coefficients for both the primary and auxiliary (ADuC845 only) ADCs are stored on-chip in manufacturing-specific Flash/EE memory locations. At power-on or after a reset, these factory calibration registers are automatically downloaded to the ADC calibration registers in the SFR space of the device. To facilitate user calibration, each of the primary and auxiliary (ADuC845 only) ADCs have dedicated calibration control SFRs, which are described in the ADC SFR Interface section. Once a user initiates a calibration procedure, the factory calibration values that were initially downloaded during the power-on sequence to the ADC calibration SFRs are overwritten. The ADC to be calibrated must be enabled via the ADC enable bits in the ADCMODE register.

Even though an internal offset calibration mode is described in this section, note that the ADCs can be chopped. This chopping scheme inherently minimizes offset errors and means that an offset calibration should never be required. Also, because factory 5 V/25°C gain calibration coefficients are automatically present at power-on, an internal full-scale calibration is required only if the device is operated at 3 V or at temperatures significantly different from 25°C.

If the device is operated in chop disabled mode, a calibration may need to be done with every gain range change that occurs via the PGA.

The ADuC845/ADuC847/ADuC848 each offer internal or system calibration facilities. For full calibration to occur on the selected ADC, the calibration logic must record the modulator output for two input conditions: zero-scale and full-scale points. These points are derived by performing a conversion on the different input voltages (zero-scale and full-scale) provided to the input of the modulator during calibration. The result of the zero-scale calibration conversion is stored in the offset calibration registers for the appropriate ADC. The result of the full-scale calibration conversion is stored in the gain calibration registers for the appropriate ADC. With these readings, the calibration logic can calculate the offset and the gain slope for the input-to-output transfer function of the converter.

During an internal zero-scale or full-scale calibration, the respective zero-scale input or full-scale input is automatically connected to the ADC inputs internally. A system calibration, however, expects the system zero-scale and system full-scale voltages to be applied externally to the ADC pins by the user before the calibration mode is initiated. In this way, external errors are taken into account and minimized. Note that all ADuC845/ADuC847/ADuC848 ADC calibrations are carried out at the user-selected SF word update rate. To optimize calibration accuracy, it is recommended that the slowest possible update rate be used.

Internally in the devices, the coefficients are normalized before being used to scale the words coming out of the digital filter. The offset calibration coefficient is subtracted from the result prior to the multiplication by the gain coefficient.

From an operational point of view, a calibration should be treated just like an ordinary ADC conversion. A zero-scale calibration (if required) should always be carried out before a full-scale calibration. System software should monitor the relevant ADC RDY0/1 bit in the ADCSTAT SFR to determine the end of calibration by using a polling sequence or an interrupt driven routine. If required, the NOEXREF0/1 bits can be monitored to detect unconnected or low voltage errors in the reference during conversion. In the event of the reference becoming disconnected, causing a NOXREF flag during a calibration, the calibration is immediately halted and no write to the calibration SFRs takes place.

Internal Calibration Example

With chop enabled, a zero-scale or offset calibration should never be required, although a full-scale or gain calibration may be required. However, if a full internal calibration is required, the procedure should be to select a PGA gain of 1 (±2.56 V) and perform a zero-scale calibration (MD2...0 = 100B in the ADCMODE register). Next, select and perform full-scale calibration by setting MD2...0 = 101B in the ADCMODE SFR. Now select the desired PGA range and perform a zero-scale calibration again (MD2..0 = 100B in ADCMODE) at the new PGA range. The reason for the double zero-scale calibration is

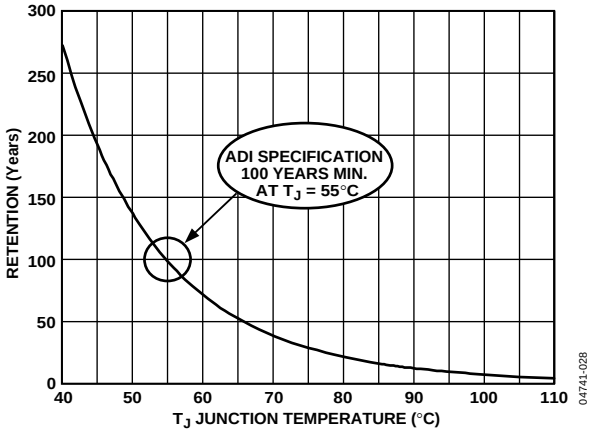


Figure 27. Flash/EE Memory Data Retention

FLASH/EE PROGRAM MEMORY

The ADuC845/ADuC847/ADuC848 contain a 64-kbyte array of Flash/EE program memory. The lower 62 kbytes of this program memory are available to the user for program storage or as additional NV data memory.

The upper 2 kbytes of this Flash/EE program memory array contain permanently embedded firmware, allowing in-circuit serial download, serial debug, and nonintrusive single-pin emulation. These 2 kbytes of embedded firmware also contain a power-on configuration routine that downloads factory calibrated coefficients to the various calibrated peripherals such as ADC, temperature sensor, current sources, band gap, and references.

These 2 kbytes of embedded firmware are hidden from the user code. Attempts to read this space read 0s; therefore, the embedded firmware appears as NOP instructions to user code.

In normal operating mode (power-on default), the 62 kbytes of user Flash/EE program memory appear as a single block. This block is used to store the user code as shown in Figure 28.

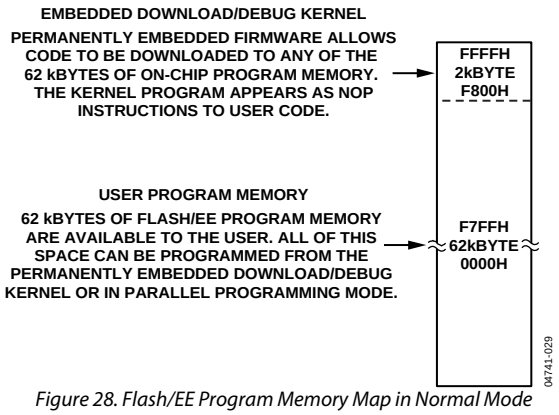


Figure 28. Flash/EE Program Memory Map in Normal Mode

In normal mode, the 62 kbytes of Flash/EE program memory can be programmed by serial downloading and by parallel programming.

Serial Downloading (In-Circuit Programming)

The ADuC845/ADuC847/ADuC848 facilitate code download via the standard UART serial port. The devices enter serial download mode after a reset or a power cycle if the $\overline{\text{PSEN}}$ pin is pulled low through an external 1 kΩ resistor. Once in serial download mode, the hidden embedded download kernel executes. This allows the user to 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 (WSD.EXE) is provided as part of the ADuC845/ADuC847/ADuC848 Quick Start development system. The AN-1074 Application Note fully describes the serial download protocol that is used by the embedded download kernel.

Parallel Programming

The parallel programming mode is fully compatible with conventional third-party flash or EEPROM device programmers. A block diagram of the external pin configuration required to support parallel programming is shown in Figure 29. In this mode, Ports 0 and 2 operate as the external address bus interface, P3 operates as the external data bus interface, and P1.0 operates as the write enable strobe. P1.1, P1.2, P1.3, and P1.4 are used as general configuration ports that configure the device for various program and erase operations during parallel programming.

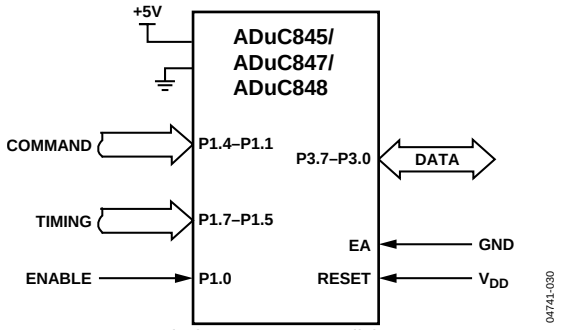


Figure 29. Flash/EE Memory Parallel Programming

The command words that are assigned to P1.1, P1.2, P1.3, and P1.4 are described in Table 31.

Table 31. Flash/EE Memory Parallel Programming Modes

Port 1 Pins				
P1.4	**P1.3**	**P1.2**	**P1.1**	**Programming Mode**
0	0	0	0	Erase Flash/EE Program, Data, and Security Mode
1	0	1	0	Program Code Byte
0	0	1	0	Program Data Byte
1	0	1	1	Read Code Byte
0	0	1	1	Read Data Byte
1	1	0	0	Program Security Modes
1	1	0	1	Read/Verify Security Modes
All other codes				Redundant

USER DOWNLOAD MODE (ULOAD)

Figure 28 shows that it is possible to use the 62 kbytes of Flash/EE program memory available to the user as one single block of memory. In this mode, all the Flash/EE memory is read-only to user code.

However, most of the Flash/EE program memory can also be written to during run time simply by entering ULOAD mode. In ULOAD mode, the lower 56 kbytes of program memory can be erased and reprogrammed by the user software as shown in Figure 30. ULOAD mode can be used to upgrade the code in the field via any user-defined download protocol. By configuring the SPI port on the ADuC845/ADuC847/ADuC848 as a slave, it is possible to completely reprogram the 56 kbytes of Flash/EE program memory in under 5 s (see the AN-1074 Application Note).

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 devices can provide up to 60 kbytes of 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 (8 kbytes on the 32-kbyte parts) 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, making it very suitable to use the 6 kbytes as a bootloader. A bootload enable option exists in the Windows® serial downloader (WSD) 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 the Flash/EE Memory Control SFR section of ECON and also in the AN-1074 Application Note.

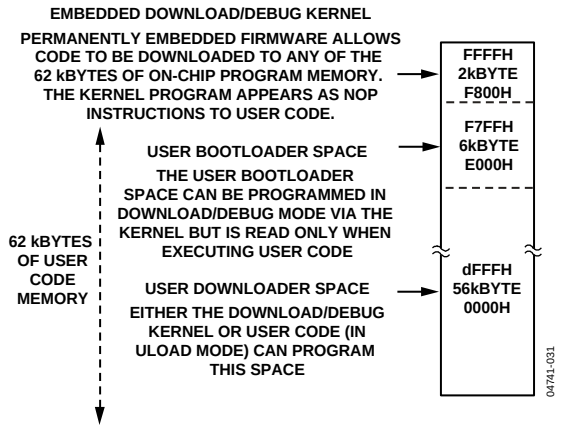


Figure 30. Flash/EE Program Memory Map in ULOAD Mode (62-kbyte Part)

The 32-kbyte memory parts have the user bootload space starting at 6000H. The memory mapping is shown in Figure 31.

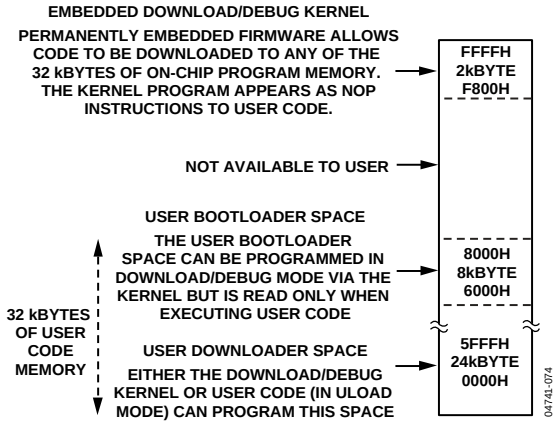


Figure 31. Flash/EE Program Memory Map in ULOAD Mode (32-kbyte Part)

ULOAD mode is not available on the 8-kbyte Flash/EE program memory parts.

Flash/EE Program Memory Security

The ADuC845/ADuC847/ADuC848 facilitate three modes of Flash/EE program memory security: the lock, secure, and serial safe modes. These modes can be independently activated, restricting access to the internal code space. They can be enabled as part of serial download protocol, as described in the AN-1074 Application Note, or via parallel programming.

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 are still allowed. This mode is deactivated by initiating an ERASE CODE AND DATA command in serial download or parallel programming modes.

Secure Mode

This mode locks the code memory, disabling parallel programming of the program memory. Reading/verifying the memory in parallel mode and reading the internal memory via a MOVC command from external memory are also disabled. This mode is deactivated by initiating an ERASE CODE AND DATA 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 device into serial download mode, that is, RESET asserted (pulled high) and de-asserted (pulled low) with $\overline{\text{PSEN}}$ low, the device interprets the serial download reset as a normal reset only. It therefore does not enter serial download mode, but executes only a normal reset sequence. Serial safe mode can be disabled only by initiating an ERASE CODE AND DATA 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 ADuC845/ADuC847/ADuC848 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) holds the 4 bytes of data at each page. The page is addressed via the EADRH and EADRL registers. Finally, ECON is an 8-bit control register that can be written to 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 32.

ECON—Flash/EE Memory Control SFR

Programming 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.

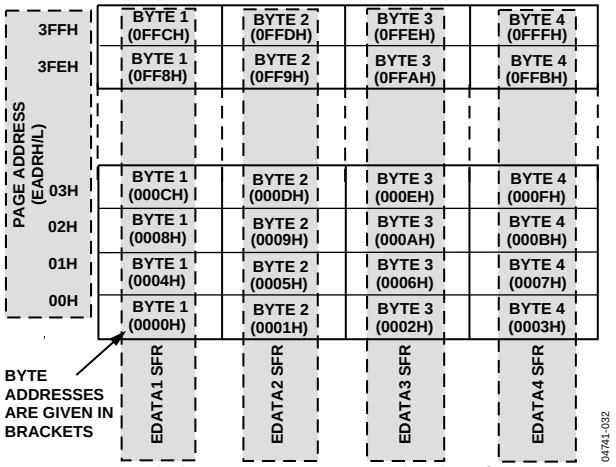


Figure 32. Flash/EE Data Memory Control and Configuration

Table 32. ECON—Flash/EE Memory Commands

ECON Value	Command Description (Normal Mode, Power-On Default)	Command Description (ULOAD Mode)
01H Read	4 bytes in the Flash/EE data memory, addressed by the page address EADRH/L, are 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 (0 ≤ EADRH < 0400H). Note that the 4 bytes in the page being addressed must be pre-erased.	Bytes 0 to 255 of internal XRAM are written to the 256 bytes of Flash/EE program memory at the page address given by EADRH/L (0 ≤ EADRH/L < 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 a 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	4-byte page of Flash/EE data memory address is erased by the page address EADRH/L.	64-byte page of FLASH/EE program memory addressed by the byte address EADRH/L is erased. A new page starts when EADRL is equal to 00H, 80H, or C0H.
06H Erase All	4 kbytes of Flash/EE data memory are erased.	The entire 56 kbytes of ULOAD are erased.
81H ReadByte	The byte in the Flash/EE data memory, addressed by the byte address EADRH/L, is read into EDATA1 (0 ≤ EADRH/L ≤ 0FFFH).	Not implemented. Use the MOVC command.
82H WriteByte	The byte in EDATA1 is written into Flash/EE data memory at the byte address EADRH/L.	The byte in EDATA1 is written into Flash/EE program memory at the byte address EADRH/L (0 ≤ EADRH/L ≤ DFFFH).
0FH EXULOAD	Configures the ECON instructions (above) to operate on Flash/EE data memory.	Enters normal mode, directing subsequent ECON instructions to operate on the Flash/EE data memory.
F0H ULOAD	Enters ULOAD mode; subsequent ECON instructions operate on Flash/EE program memory.	Enables the ECON instructions to operate on the Flash/EE program memory. ULOAD entry mode.

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. Specifically, a byte can be programmed only if it already holds the value FFH. Because of the Flash/EE architecture, this erasure 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 erased, 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!)
```

Although the 4 kbytes of Flash/EE data memory are factory pre-erased, that is, byte locations set to FFH, it is good programming practice to include an ERASEALL routine as part of any configuration/set-up code running on the devices. 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 language would appear as

```
MOV ECON, #06H     ;ERASE all Command
                   ;2ms duration
```

FLASH/EE MEMORY TIMING

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

Normal Mode (Operating on Flash/EE Data Memory)

Command	Bytes Affected	
READPAGE	4 bytes	25 machine cycles
WRITEPAGE	4 bytes	380 µs
VERIFYPAGE	4 bytes	25 machine cycles
ERASEPAGE	4 bytes	2 ms
ERASEALL	4 kbytes	2 ms
READBYTE	1 byte	10 machine cycles
WRITEBYTE	1 byte	200 µs

ULOAD Mode (Operating on Flash/EE Program Memory)

WRITEPAGE	256 bytes	15 ms
ERASEPAGE	64 bytes	2 ms
ERASEALL	56 kbytes	2 ms
WRITEBYTE	1 byte	200 µs

A given mode of operation is initiated as soon as the command word is written to the ECON SFR. The core microcontroller operation 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 such as counter/timers continue to count as configured throughout this period.

PWMCON PWM Control SFR

SFR Address: AEH
Power-On Default: 00H
Bit Addressable: No

Table 34. PWMCON PWM Control SFR

Bit No.	Name	Description
7	–––	Not Implemented. Write Don't Care.
6, 5, 4	PWM2, PWM1, PWM0	PMW Mode Selection. PWM2 PWM1 PWM0 0 0 0 Mode 0: PWM disabled. 0 0 1 Mode 1: Single 16-bit output with programmable pulse and cycle time. 0 1 0 Mode 2: Twin 8-bit outputs. 0 1 1 Mode 3: Twin 16-bit outputs. 1 0 0 Mode 4: Dual 16-bit pulse density outputs. 1 0 1 Mode 5: Dual 8-bit outputs. 1 1 0 Mode 6: Dual 16-bit pulse density RZ outputs. 1 1 1 Mode 7: PWM counter reset with outputs not used.
3, 2	PWS1, PWS0	PWM Clock Source Divider. PWS1 PWS0 0 0 Selected clock. 0 1 Selected clock divided by 4. 1 0 Selected clock divided by 16. 1 1 Selected clock divided by 64.
1, 0	PWC1, PWC0	PWM Clock Source Selection. PWC1 PWC0 0 0 F_{XTAL}/15 (2.184 kHz). 0 1 F_{XTAL} (32.768 kHz). 1 0 External input on P2.7. 1 1 F_{VCO} (12.58 MHz).

PWM Pulse Width High Byte (PWM0H)

SFR Address: B2H
Power-On Default: 00H
Bit Addressable: No

Table 35. PWM0H: PWM Pulse Width High Byte

PWM0H.7	PWM0H.6	PWM0H.5	PWM0H.4	PWM0H.3	PWM0H.2	PWM0H.1	PWM0H.0
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWM Pulse Width Low Byte (PWM0L)

SFR Address: B1H
Power-On Default: 00H
Bit Addressable: No

I²C SERIAL INTERFACE

The ADuC845/ADuC847/ADuC848 support a fully licensed I²C serial interface. The I²C interface is implemented as a full hardware slave and software master. SDATA (Pin 27 on the MQFP package and Pin 29 on the LFCSP package) is the data I/O pin. SCLK (Pin 26 on the MQFP package and Pin 28 on the LFCSP package) is the serial interface clock for the SPI interface. The I²C interface on the devices is fully independent of all other pin/function multiplexing. The I²C interface incorporated on the ADuC845/ADuC847/ADuC848 also includes a second address register (I2CADD1) at SFR Address F2H with a default power-on value of 7FH. The I²C interface is always available to the user and is not multiplexed with any other I/O functionality on the chip. This means that the I²C and SPI interfaces can be used at the same time.

Note that when using the I²C and SPI interfaces simultaneously, they both use the same interrupt routine (Vector Address 3BH). When an interrupt occurs from one of these, it is necessary to interrogate each interface to see which one has triggered the ISR request.

The four SFRs that are used to control the I²C interface are described next.

I2CCON—I²C Control Register

SFR Address: E8H
Power-On Default: 00H
Bit Addressable: Yes

Table 40. I2CCON SFR Bit Designations

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 bit (MDE) is set.
6	MDE	I²C Software 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 bit is used to implement the SCLK for a master I²C transmitter in software. Data written to this bit is output on the SCLK 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 an SCLK transition 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	I2CRS	I²C Reset Bit (slave mode only). Set by the user to reset the I²C interface. Cleared by the user code for normal I²C operation.
1	I2CTX	I²C Direction Transfer Bit (slave mode only). Set by the MicroConverter if the I²C interface is transmitting. Cleared by the MicroConverter if the I²C interface is receiving.
0	I2CI	I²C Interrupt Bit (slave mode only). Set by the MicroConverter after a byte has been transmitted or received. Cleared by the MicroConverter when the user code reads the I2CDAT SFR. I2CI should not be cleared by user code.

I2CADD—I²C Address Register 1

Function:	Holds one of the I²C peripheral addresses for the device. It may be overwritten by user code. The uC001 Application Note describes the format of the I²C standard 7-bit address.
SFR Address:	9BH
Power-On Default:	55H
Bit Addressable:	No

I2CADD1—I²C Address Register 2

Function:	Same as the I2CADD.
SFR Address:	F2H
Power-On Default:	7FH
Bit Addressable:	No

I2CDAT—I²C Data Register

Function:	The I2CDAT SFR is written to by user code to transmit data, or read by user code to read data just received by the I²C interface. Accessing I2CDAT automatically clears any pending I²C interrupt and the I2CI bit in the I2CCON SFR. User code should access I2CDAT only once per interrupt cycle.
SFR Address:	9AH
Power-On Default:	00H
Bit Addressable:	No

The main features of the MicroConverter I²C interface are

- Only two bus lines are required: a serial data line (SDATA) and a serial clock line (SCLOCK).
- An I²C master can communicate with multiple slave devices. Because each slave device has a unique 7-bit address, single master/slave relationships can exist at all times even in a multislave environment.
- The ability to respond to two separate addresses when operating in slave mode.
- On-chip filtering rejects <50 ns spikes on the SDATA and the SCLOCK lines to preserve data integrity.

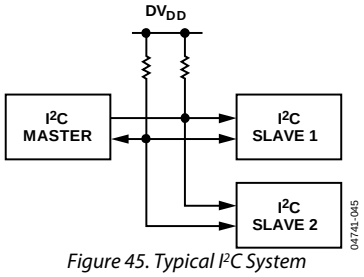


Figure 45. Typical I²C System

Software Master Mode

The ADuC845/ADuC847/ADuC848 can be used as an I²C master device 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 is high on the last rising edge of SCLOCK. MDI is cleared if the SDATA pin is 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 the uC001 Application Note.

TIMECON—TIC Control Register

SFR Address: A1H
Power-On Default: 00H
Bit Addressable: No

Table 45. TIMECON SFR Bit Designations

Bit No.	Name	Description
7	----	Not Implemented. Write Don't Care.
6	TFH	Twenty-Four Hour Select Bit. Set by the user to enable the hour counter to count from 0 to 23. Cleared by the user to enable the hour counter to count from 0 to 255.
5, 4	ITS1, ITS0	Interval Timebase Selection Bits. ITS1 ITS0 Interval Timebase 0 0 1/128 Second 0 1 Seconds 1 0 Minutes 1 1 Hours
3	ST1	Single Time Interval Bit. Set by the user to generate a single interval timeout. If set, a timeout clears the TIEN bit. Cleared by the user to allow the interval counter to be automatically reloaded and start counting again at each interval timeout.
2	TII	TIC Interrupt Bit. Set when the 8-bit interval counter matches the value in the INTVAL SFR. Cleared by user software.
1	TIEN	Time Interval Enable Bit. Set by the user to enable the 8-bit time interval counter. Cleared by the user to disable the interval counter.
0	TCEN	Time Clock Enable Bit. Set by the user to enable the time clock to the time interval counters. Cleared by the user to disable the clock to the time interval counters and reset the time interval SFRs to the last value written to them by the user. The time registers (HTHSEC, SEC, MIN, and HOUR) can be written while TCEN is low.

UART SERIAL INTERFACE

The serial port is full duplex, meaning that it can transmit and receive simultaneously. It is also receive buffered, meaning that it can begin receiving a second byte before a previously received byte is read from the receive register. However, if the first byte is still not read by the time reception of the second byte is complete, the first byte is lost. The physical interface to the serial data network is via Pins RxD(P3.0) and TxD(P3.1), while the SFR interface to the UART comprises SBUF and SCON, as described in this section.

SBUF SFR

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

SCON UART—Serial Port Control Register

SFR Address: 98H
Power-On Default: 00H
Bit Addressable: Yes

Table 54. SCON SFR Bit Designations

Bit No.	Name	Description
7, 6	SM0, SM1	UART Serial Mode Select Bits. These bits select the serial port operating mode as follows:
		SM0 SM1 Selected Operating Mode.
		0 0 Mode 0: Shift register, fixed baud rate (Core_Clk/2).
		0 1 Mode 1: 8-bit UART, variable baud rate.
		1 0 Mode 2: 9-bit UART, fixed baud rate (Core_Clk/32) or (Core_Clk/16).
		1 1 Mode 3: 9-bit UART, variable baud rate.
5	SM2	Multiprocessor Communication Enable Bit.
		Enables multiprocessor communication in Modes 2 and 3.
		In Mode 0, SM2 should be cleared.
		In Mode 1, if SM2 is set, RI is not activated if a valid stop bit was not received. If SM2 is cleared, RI is set as soon as the byte of data is received.
		In Modes 2 or 3, if SM2 is set, RI is not activated if the received ninth data bit in RB8 is 0.
		If SM2 is cleared, RI is set as soon as the byte of data is received.
4	REN	Serial Port Receive Enable Bit.
		Set by user software to enable serial port reception.
3	TB8	Serial Port Transmit (Bit 9).
		The data loaded into TB8 is the ninth data bit transmitted in Modes 2 and 3. Cleared by user software to disable serial port reception.
2	RB8	Serial Port Receiver Bit 9.
		The ninth data bit received in Modes 2 and 3 is latched into RB8. For Mode 1, the stop bit is latched into RB8.
1	TI	Serial Port Transmit Interrupt Flag.
		Set by hardware at the end of the eighth bit in Mode 0, or at the beginning of the stop bit in Modes 1, 2, and 3. TI must be cleared by user software.
0	RI	Serial Port Receive Interrupt Flag.
		Set by hardware at the end of the eighth bit in Mode 0, or halfway through the stop bit in Modes 1, 2, and 3. RI must be cleared by software.

SBUF—UART Serial Port Data Register

SFR Address: 99H
Power-On Default: 00H
Bit Addressable: No

Timer 3 Generated Baud Rates

The high integer dividers in a UART block mean that high speed baud rates are not always possible. Also, generating baud rates requires the exclusive use of a timer, rendering it unusable for other applications when the UART is required. To address this problem, the ADuC845/ADuC847/ADuC848 have 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 bps to 393216 bps 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 61.

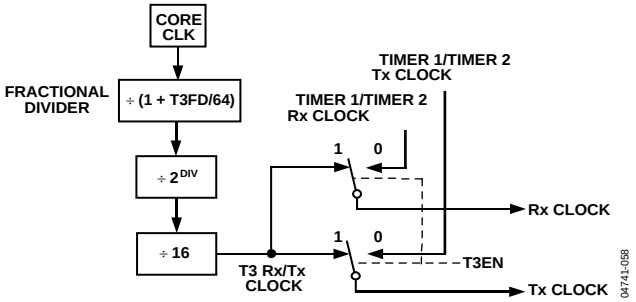


Figure 61. Timer 3, UART Baud Rate

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 to set 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.

$$DIV = \frac{\log\left(\frac{Core\ Clock\ Frequency}{16 \times Baud\ Rate}\right)}{\log(2)}$$

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:

$$T3FD = \frac{2 \times Core\ Clock\ Frequency}{2^{DIV-1} \times Baud\ Rate} - 64$$

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:

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

For example, to get a baud rate of 9600 while operating at a core clock frequency of 1.5725 MHz, that is, CD = 3,

$$DIV = \log(1572500/(16 \times 9600))/\log 2 = 3.35 = 3$$

Note that the DIV result is rounded down.

$$T3FD = (2 \times 1572500)/(2^{3-1} \times 9600) - 64 = 18 = 12H$$

Therefore, the actual baud rate is 9588 bps, which gives an error of 0.12%.

The T3CON and T3FD registers are used to control Timer 3.

T3CON – Timer 3 Control Register

SFR Address: 9EH
Power-On Default: 00H
Bit Addressable: No

Table 57. Common Baud Rates Using Timer 3 with a 12.58 MHz PLL Clock

Ideal Baud	CD	DIV	T3CON	T3FD	% Error
230400	0	1	81H	2DH	0.18
115200	0	2	82H	2DH	0.18
115200	1	1	81H	2DH	0.18
57600	0	3	83H	2DH	0.18
57600	1	2	82H	2DH	0.18
57600	2	1	81H	2DH	0.18
38400	0	4	84H	12H	0.12
38400	1	3	83H	12H	0.12
38400	2	2	82H	12H	0.12
38400	3	1	81H	12H	0.12
19200	0	5	85H	12H	0.12
19200	1	4	84H	12H	0.12
19200	2	3	83H	12H	0.12
19200	3	2	82H	12H	0.12
19200	4	1	81H	12H	0.12
9600	0	6	86H	12H	0.12
9600	1	5	85H	12H	0.12
9600	2	4	84H	12H	0.12
9600	3	3	83H	12H	0.12
9600	4	2	82H	12H	0.12
9600	5	1	81H	12H	0.12

HARDWARE DESIGN CONSIDERATIONS

This section outlines some of the key hardware design considerations that must be addressed when integrating the ADuC845/ADuC847/ADuC848 into any hardware system.

EXTERNAL MEMORY INTERFACE

In addition to their internal program and data memories, the devices can access up to 16 Mbytes of external data memory (SRAM). No external program memory access is available.

To begin executing code, tie the $\overline{EA}$ (external access) pin high. When $\overline{EA}$ is high (pulled up to V_{DD}—see Figure 70), user program execution starts at Address 0 in the internal 62-kbyte Flash/EE code space. When executing from internal code space, accesses to the program space above F7FFH (62 kbytes) are read as NOP instructions.

Note that a second very important function of the $\overline{EA}$ pin is described in the Single-Pin Emulation Mode section under the Other Hardware Considerations section.

Figure 62 shows a hardware configuration for accessing up to 64 kbytes of external data memory. This interface is standard to any 8051-compatible MCU.

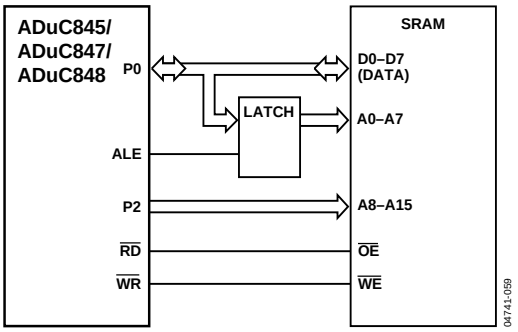


Figure 62. External Data Memory Interface (64-kbyte Address Space)

If access to more than 64 kbytes of RAM is desired, a feature unique to the MicroConverter allows addressing up to 16 Mbytes of external RAM simply by adding another latch as shown in Figure 63.

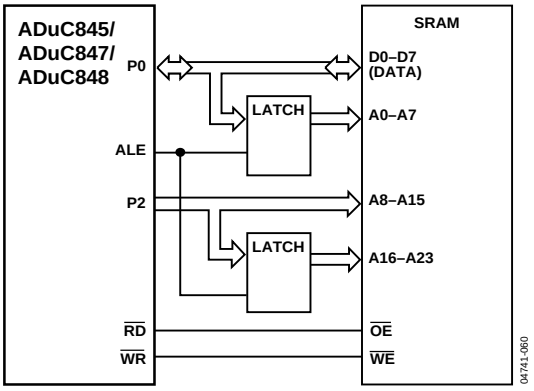


Figure 63. External Data Memory Interface (16-Mbtye Address Space)

In either implementation, Port 0 (P0) serves as a multiplexed address/data bus. It emits the low byte of the data pointer (DPL) as an address, which is latched by ALE prior to data being placed on the bus by the devices (write operation) or the external data memory (read operation). Port 2 (P2) provides the data pointer page byte (DPP) to be latched by ALE, followed by the data pointer high byte (DPH). If no latch is connected to P2, DPP is ignored by the SRAM, and the 8051 standard of 64-kbyte external data memory access is maintained.

The following example shows the code used to write data to external data memory.

```
MOV DPP, #10h ;Set addr to 100000h
MOV DPH, #00h
MOV DPL, #00h
MOV A,   #'B' ;Write Char 'B' (42h)
MOVX @DPTR,A  ;Move to DPP:DPH:DPL addr
```

POWER SUPPLIES

The operational power supply voltage range of the device is 2.7 V to 5.25 V. Although the guaranteed data sheet specifications are given only for power supplies within 2.7 V to 3.6 V and 4.75 V to 5.25 V (±5% of the nominal 5 V level), the chip functions equally well at any power supply level between 2.7 V and 5.25 V.

Separate analog and digital power supply pins (AV_{DD} and DV_{DD}, respectively) allow AV_{DD} to be kept relatively free of the noisy digital signals often present on a system DV_{DD} line. In this mode, the device can also operate with split supplies, that is, using different voltage supply levels for each supply. For example, the system can be designed to operate with a DV_{DD} voltage level of 3 V and the AV_{DD} level can be at 5 V, or vice versa, if required. A typical split-supply configuration is shown in Figure 64.

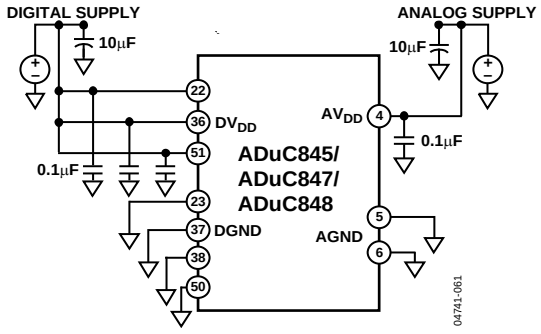


Figure 64. External Dual-Supply Connections (56-Lead LFCSP Pin Numbering)

As an alternative to providing two separate power supplies, AV_{DD} can be kept quiet by placing a small series resistor and/or ferrite bead between it and DV_{DD}, and then decoupling AV_{DD} separately to ground. An example of this configuration is shown in Figure 65. In this configuration, other analog circuitry (such

as op amps and voltage reference) can be powered from the AV_{DD} supply line as well.

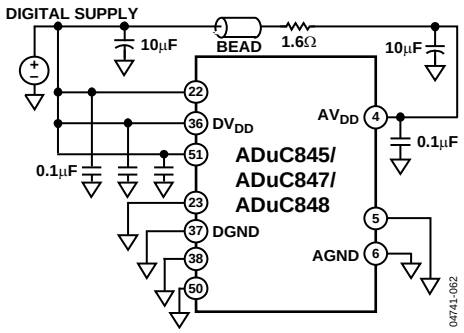


Figure 65. External Single-Supply Connections (56-Lead LFCSP Pin Numbering)

Notice that in both Figure 64 and Figure 65 a large value (10 µF) reservoir capacitor sits on DV_{DD} and a separate 10 µF capacitor sits on AV_{DD}. Also, local decoupling capacitors (0.1 µF) are located at each V_{DD} pin of the chip. As per standard design practice, be sure to include all of these capacitors and ensure that the smaller capacitors are closer than the 10 µF capacitors to each V_{DD} pin with lead lengths as short as possible. Connect the ground terminal of each of these capacitors directly to the underlying ground plane. Finally, note that, at all times, the analog and digital ground pins on the device must be referenced to the same system ground reference point. It is recommended that the LFCSP paddle be soldered to ensure mechanical stability but be floated with respect to system V_{DD}s or grounds.

POWER-ON RESET OPERATION

An internal power-on reset (POR) is implemented on the ADuC845/ADuC847/ADuC848.

3 V Part

For DV_{DD} below 2.63 V, the internal POR holds the device in reset. As DV_{DD} rises above 2.63 V, an internal timer times out for typically 128 ms before the device is released from reset. The user must ensure that the power supply has at least reached a stable 2.7 V minimum level by this time. Likewise on power-down, the internal POR holds the device in reset until the power supply drops below 1 V. Figure 66 illustrates the operation of the internal POR.

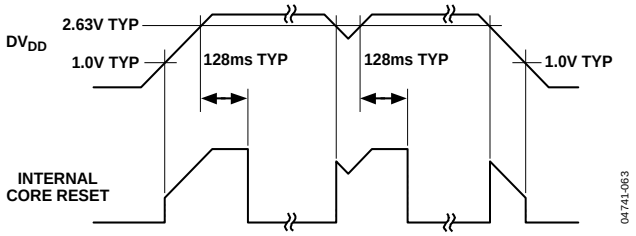


Figure 66. 3 V Part POR operation

5 V Part

For DV_{DD} below 4.5 V, the internal POR holds the device in reset. As DV_{DD} rises above 4.5 V, an internal timer times out for approximately 128 ms before the device is released from reset. The user must ensure that the power supply has reached a stable 4.75 V minimum level by this time. Likewise on power-down, the internal POR holds the device in reset until the power supply drops below 1 V. Figure 67 illustrates this operation.

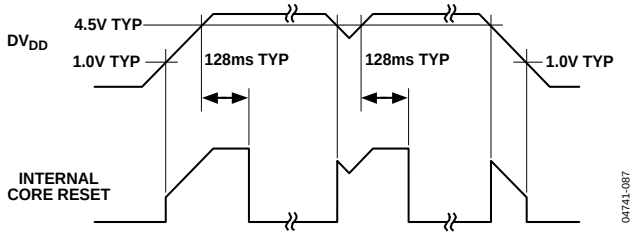


Figure 67. 5 V Part POR Operation

POWER CONSUMPTION

The DV_{DD} power supply current consumption is specified in normal and power-down modes. The AV_{DD} power supply current is specified with the analog peripherals disabled. The normal mode power consumption represents the current drawn from DV_{DD} by the digital core. The other on-chip peripherals (such as the watchdog timer and power supply monitor) consume negligible current and are therefore included with the normal operating current. The user must add any currents sourced by the parallel and serial I/O pins, and those sourced by the DAC to determine the total current needed at the ADuC845/ADuC847/ADuC848 DV_{DD} and AV_{DD} supply pins. Also, current drawn from the DV_{DD} supply increases by approximately 5 mA during Flash/EE erase and program cycles.

POWER-SAVING MODES

Setting the power-down mode bit, PCON.1, in the PCON SFR described in Table 6, allows the chip to be switched from normal mode into full power-down mode.

In power-down mode, both the PLL and the clock to the core are stopped. The on-chip oscillator can be halted or can continue to oscillate, depending on the state of the oscillator power-down bit (OSC_PD) in the PLLCON SFR. The TIC, driven directly from the oscillator, can also be enabled during power-down. However, all other on-chip peripherals are shut down. Port pins retain their logic levels in this mode, but the DAC output goes to a high impedance state (three-state) while ALE and $\overline{PSEN}$ outputs are held low. There are five ways to terminate power-down mode:

- **Asserting the RESET Pin**
  Returns to normal mode. All registers are set to their reset default value and program execution starts at the reset vector once the RESET pin is de-asserted.

Table 66. EXTERNAL DATA MEMORY WRITE CYCLE Parameter

		12.58 MHz Core Clock		6.29 MHz Core Clock		
		Min	Max	Min	Max	Unit
t_{WLWH}	$\overline{WR}$ Pulse Width	65		130		ns
t_{AVLL}	Address Valid After ALE Low	60		120		ns
t_{LLAX}	Address Hold After ALE Low	65		135		ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low		130		260	ns
t_{AVWL}	Address Valid to $\overline{RD}$ or $\overline{WR}$ Low	190		375		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	60		120		ns
t_{QVWH}	Data Setup Before $\overline{WR}$	120		250		ns
t_{WHQX}	Data and Address Hold After $\overline{WR}$	380		755		ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	60		125		ns

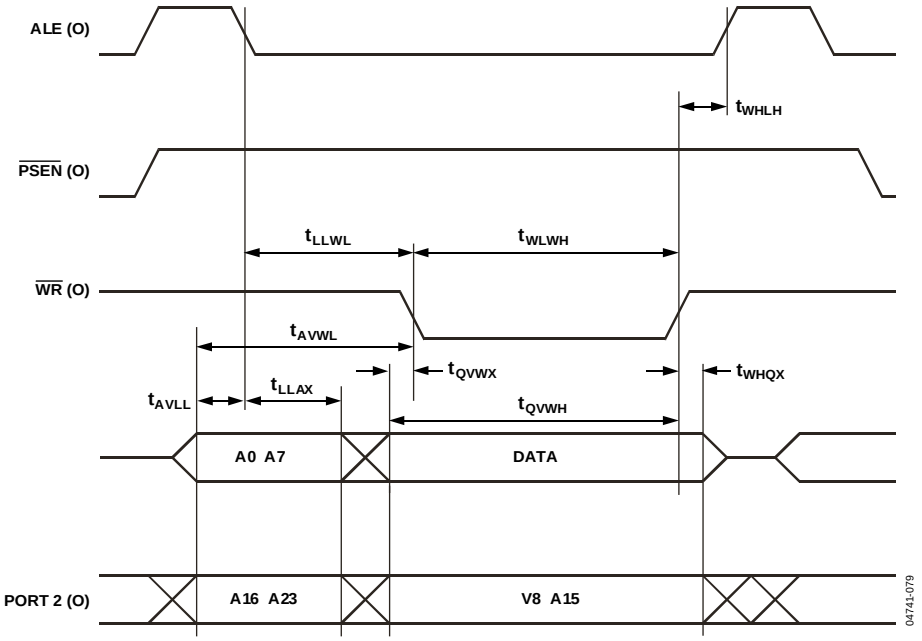


Figure 74. External Data Memory Write Cycle

Table 67. I²C-COMPATIBLE INTERFACE TIMING Parameter

Parameter		Min	Max	Unit
t_L	SCLCK Low Pulse Width	1.3		µs
t_H	SCLCK High Pulse Width	0.6		µs
t_{SHD}	Start Condition Hold Time	0.6		µs
t_{DSU}	Data Setup Time	100		µs
t_{DHD}	Data Hold Time		0.9	µs
t_{RSU}	Setup Time for Repeated Start	0.6		µs
t_{PSU}	Stop Condition Setup Time	0.6		µs
t_{BUF}	Bus Free Time Between a Stop Condition and a Start Condition	1.3		µs
t_R	Rise Time of Both SCLCK and SDATA		300	ns
t_F	Fall Time of Both SCLCK and SDATA		300	ns
t_{SUP}[1]	Pulse Width of Spike Suppressed		50	ns

[1] Input filtering on both the SCLOCK and SDATA inputs suppresses noise spikes less than 50 ns.

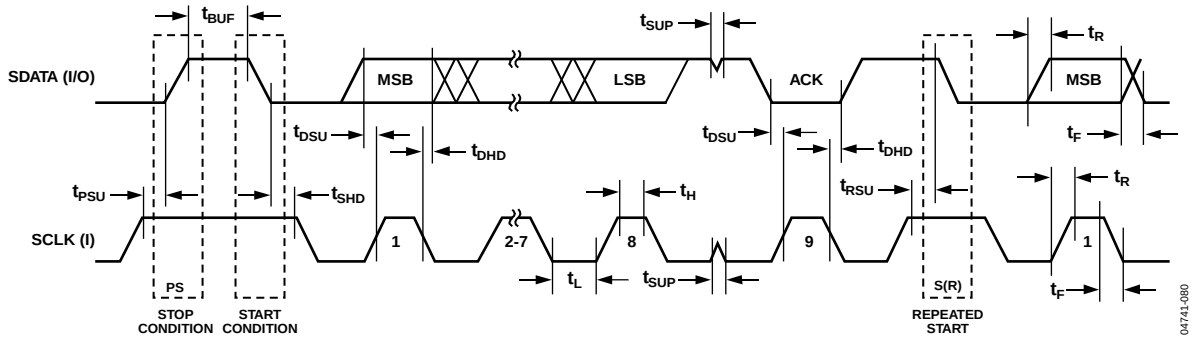


Figure 75. I²C-Compatible Interface Timing

Table 69. SPI MASTER MODE TIMING (CPHA = 0) Parameter

		Min	Typ	Max	Unit
t_{SL}	SCLOCK Low Pulse Width[1]		635		ns
t_{SH}	SCLOCK High Pulse Width[1]		635		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 in PLLCON SFR set to 0, 1, and 1, respectively, that is, core clock frequency = 1.57 MHz.
b. SPI bit-rate selection bits SPR1 and SPR0 in SPICON SFR set to 0 and 0, respectively.

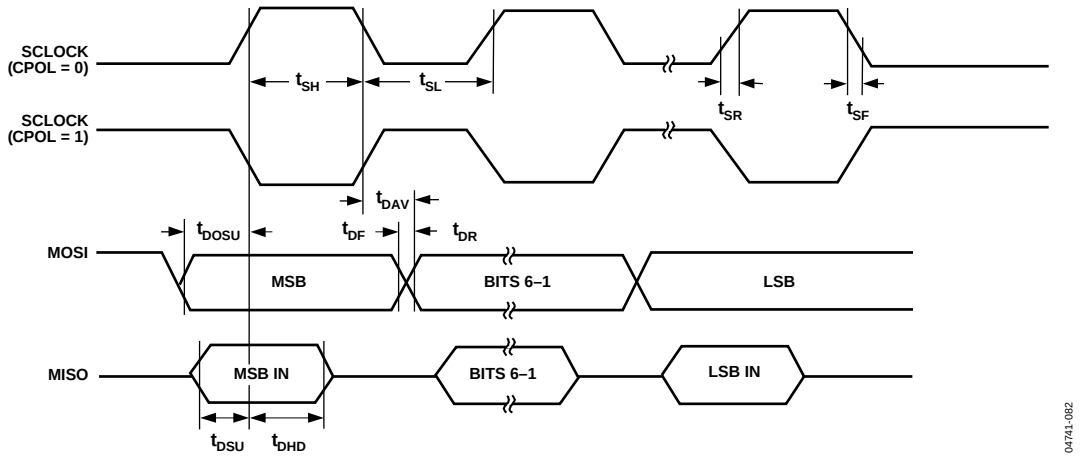


Figure 77. SPI Master Mode Timing (CHPA = 0)