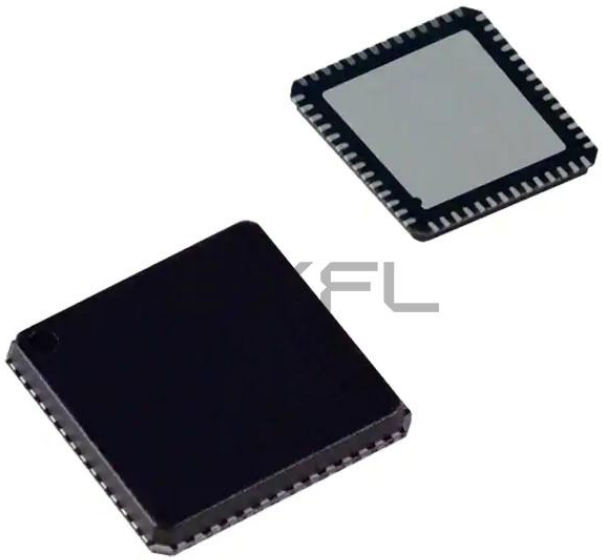

Welcome to E-XFL.COM

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	EBI/EMI, 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)	2.7V ~ 5.25V
Data Converters	A/D 7x16b; D/A 1x12b
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/aduc836bcpz

SPECIFICATIONS (continued)

Parameter	ADuC836	Test Conditions/Comments	Unit
LOGIC OUTPUTS (Not Including XTAL2)[2]			
V_{OH}, Output High Voltage	2.4	V_{DD} = 5 V, I_{SOURCE} = 80 μA	V min
	2.4	V_{DD} = 3 V, I_{SOURCE} = 20 μA	V min
V_{OL}, Output Low Voltage[14]	0.4	I_{SINK} = 8 mA, SCLOCK, MOSI/SDATA	V max
	0.4	I_{SINK} = 10 mA, P1.0 and P1.1	V max
	0.4	I_{SINK} = 1.6 mA, All Other Outputs	V max
Floating State Leakage Current[2]	±10		μA max
Floating State Output Capacitance	5		pF typ
POWER SUPPLY MONITOR (PSM)			
AV_{DD} Trip Point Selection Range	2.63	Four Trip Points Selectable in This Range	V min
	4.63	Programmed via TPA1–0 in PSMCON	V max
AV_{DD} Power Supply Trip Point Accuracy	±3.0	T_{MAX} = 85°C	% max
	±4.0	T_{MAX} = 125°C	% max
DV_{DD} Trip Point Selection Range	2.63	Four Trip Points Selectable in This Range	V min
	4.63	Programmed via TPD1–0 in PSMCON	V max
DV_{DD} Power Supply Trip Point Accuracy	±3.0	T_{MAX} = 85°C	% max
	±4.0	T_{MAX} = 125°C	% max
WATCHDOG TIMER (WDT)			
Timeout Period	0	Nine Timeout Periods in This Range	ms min
	2000	Programmed via PRE3–0 in WDCON	ms max
MCU CORE CLOCK RATE		Clock Rate Generated via On-Chip PLL	
MCU Clock Rate[2]	98.3	Programmable via CD2–0 Bits in PLLCON SFR	kHz min
	12.58		MHz max
START-UP TIME			
At Power-On	300		ms typ
After External RESET in Normal Mode	3		ms typ
After WDT Reset in Normal Mode	3	Controlled via WDCON SFR	ms typ
From Idle Mode	10		μs typ
From Power-Down Mode			
Oscillator Running		OSC_PD Bit = 0 in PLLCON SFR	
Wake-Up with $\overline{INT0}$ Interrupt	20		μs typ
Wake-Up with SPI Interrupt	20		μs typ
Wake-Up with TIC Interrupt	20		μs typ
Wake-Up with External RESET	3		ms typ
Oscillator Powered Down		OSC_PD Bit = 1 in PLLCON SFR	
Wake-Up with $\overline{INT0}$ Interrupt	20		μs typ
Wake-Up with SPI Interrupt	20		μs typ
Wake-Up with External RESET	5		ms typ
FLASH/EE MEMORY RELIABILITY CHARACTERISTICS[15]			
Endurance[16]	100,000		Cycles min
Data Retention[17]	100		Years min

NOTES

[1] Temperature range for ADuC836BS (MQFP package) is –40°C to +125°C. Temperature range for ADuC836BCP (CSP package) is –40°C to +85°C.

[2] These numbers are not production tested but are guaranteed by design and/or characterization data on production release.

[3] System Zero-Scale Calibration can remove this error.

[4] The primary ADC is factory calibrated at 25°C with AV_{DD} = DV_{DD} = 5 V yielding this full-scale error of 10 μV. If user power supply or temperature conditions are significantly different from these, an Internal Full-Scale Calibration will restore this error to 10 μV. A system zero-scale and full-scale calibration will remove this error altogether.

[5] Gain Error Drift is a span drift. To calculate Full-Scale Error Drift, add the Offset Error Drift to the Gain Error Drift times the full-scale input.

[6] The auxiliary ADC is factory calibrated at 25°C with AV_{DD} = DV_{DD} = 5 V yielding this full-scale error of –2.5 LSB. A system zero-scale and full-scale calibration will remove this error altogether.

[7] DAC linearity and ac specifications are calculated using: reduced code range of 48 to 4095, 0 to V_{REF}; reduced code range of 100 to 3950, 0 to V_{DD}.

[8] Gain Error is a measurement of the span error of the DAC.

[9] In general terms, the bipolar input voltage range to the primary ADC is given by $Range_{ADC} = \pm(V_{REF}\ 2^{RN})/125$, where:
V_{REF} = REFIN(+) to REFIN(–) voltage and V_{REF} = 1.25 V when internal ADC V_{REF} is selected. RN = decimal equivalent of RN2, RN1, RN0, e.g.,
V_{REF} = 2.5 V and RN2, RN1, RN0 = 1, 1, 0 the $Range_{ADC}$ = ±1.28 V. In Unipolar mode, the effective range is 0 V to 1.28 V in our example.

[10] 1.25 V is used as the reference voltage to the auxiliary ADC when internal V_{REF} is selected via XREF0 and XREF1 bits in ADC0CON and ADC1CON, respectively.

[11] In Bipolar mode, the auxiliary ADC can only be driven to a minimum of AGND – 30 mV as indicated by the auxiliary ADC absolute AIN voltage limits. The bipolar range is still $–V_{REF}$ to $+V_{REF}$; however, the negative voltage is limited to –30 mV.

[12] The ADuC836BCP (CSP package) has been qualified and tested with the base of the CSP package floating.

[13] Pins configured in SPI mode, pins configured as digital inputs during this test.

[14] Pins configured in I^2C mode only.

[15] Flash/EE Memory Reliability Characteristics apply to both the Flash/EE program memory and Flash/EE data memory.

[16] Endurance is qualified to 100 Kcycles as per JEDEC Std. 22 method A117 and measured at –40°C, +25°C, +85°C, and +125°C. Typical endurance at 25°C is 700 Kcycles.

[17] Retention lifetime equivalent at junction temperature (T_J) = 55°C as per JEDEC Std. 22, Method A117. Retention lifetime based on an activation energy of 0.6 eV will derate with junction temperature as shown in Figure 16 in the Flash/EE Memory section.

[18] Power Supply current consumption is measured in Normal, Idle, and Power-Down modes under the following conditions:
Normal mode: Reset = 0.4 V, Digital I/O pins = open circuit, Core Clk changed via CD bits in PLLCON, Core Executing internal software loop.
Idle mode: Reset = 0.4 V, Digital I/O pins = open circuit, Core Clk changed via CD bits in PLLCON, PCON.0 = 1, Core Execution suspended in idle mode.
Power-Down mode: Reset = 0.4 V, All P0 pins and P1.2–P1.7 pins = 0.4 V, all other digital I/O pins are open circuit, Core Clk changed via CD bits in PLLCON, PCON.1 = 1, Core Execution suspended in Power-Down mode, OSC turned on or off via OSC_PD bit (PLLCON.7) in PLLCON SFR.

[19] DV_{DD} power supply current will increase typically by 3 mA (3 V operation) and 10 mA (5 V operation) during a Flash/EE memory program or erase cycle.

Specifications subject to change without notice.

PIN FUNCTION DESCRIPTIONS (continued)

Pin No. 52-Lead MQFP	Pin No. 56-Lead CSP	Mnemonic	Type*	Description
		P1.4/AIN1	I	Primary ADC, Positive Analog Input
		P1.5/AIN2	I	Primary ADC, Negative Analog Input
		P1.6/AIN3	I	Auxiliary ADC Input or Muxed Primary ADC, Positive Analog Input
		P1.7/AIN4/DAC	I/O	Auxiliary ADC Input or Muxed Primary ADC, Negative Analog Input. The voltage output from the DAC can also be configured to appear at this pin.
5	4, 5	AV_{DD}	S	Analog Supply Voltage, 3 V or 5 V
6	6, 7, 8	AGND	S	Analog Ground. Ground reference pin for the analog circuitry.
7	9	REFIN(–)	I	Reference Input, Negative Terminal
8	10	REFIN(+)	I	Reference Input, Positive Terminal
13	15	$\overline{SS}$	I	Slave Select Input for the SPI Interface. A weak pull-up is present on this pin.
14	16	MISO	I/O	Master Input/Slave Output for the SPI Interface. A weak pull-up is present on this input pin.
15	17	RESET	I	Reset Input. A high level on this pin for 16 core clock cycles while the oscillator is running resets the device. There is an internal weak pull-down and a Schmitt trigger input stage on this pin.
16–19, 22–25	18–21, 24–27	P3.0–P3.7	I/O	P3.0–P3.7 are bidirectional port pins 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 will source current because of the internal pull-up resistors. When driving a 0-to-1 output transition, a strong pull-up is active for two core clock periods of the instruction cycle. Port 3 pins also have various secondary functions including:
		P3.0/RXD	I/O	Receiver Data for UART Serial Port
		P3.1/TXD	I/O	Transmitter Data for UART Serial Port
		P3.2/$\overline{INT0}$	I/O	External Interrupt 0. This pin can also be used as a gate control input to Timer 0.
		P3.3/$\overline{INT1}$	I/O	External Interrupt 1. This pin can also be used as a gate control input to Timer 1.
		P3.4/T0/PWMCLK	I/O	Timer/Counter 0 External Input. If the PWM is enabled, an external clock may be input at this pin.
		P3.5/T1	I/O	Timer/Counter 1 External Input
		P3.6/$\overline{WR}$	I/O	External Data Memory Write Strobe. Latches the data byte from Port 0 into an external data memory.
		P3.7/$\overline{RD}$	I/O	External Data Memory Read Strobe. Enables the data from an external data memory to Port 0.
20, 34, 48	22, 36, 51,	DV_{DD}	S	Digital Supply, 3 V or 5 V
21, 35, 47	23, 37, 38, 50	DGND	S	Digital Ground. Ground reference point for the digital circuitry.
26		SCLOCK	I/O	Serial Interface Clock for either the I^2C or SPI Interface. As an input, this pin is a Schmitt-triggered input, and a weak internal pull-up is present on this pin unless it is outputting logic low. This pin can also be directly controlled in software as a digital output pin.
27		MOSI/SDATA	I/O	Serial Data I/O for the I^2C Interface or Master Output/Slave Input for the SPI Interface. A weak internal pull-up is present on this pin unless it is outputting logic low. This pin can also be directly controlled in software as a digital output pin.
28–31 36–39	30–33 39–42	P2.0–P2.7 (A8–A15) (A16–A23)	I/O	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 will source current because of the internal pull-up resistors. Port 2 emits the high order address bytes during fetches from external program memory and middle and high order address bytes during accesses to the 24-bit external data memory space.
32	34	XTAL1	I	Input to the Crystal Oscillator Inverter
33	35	XTAL2	O	Output from the Crystal Oscillator Inverter. (See the Hardware Design Considerations section for description.)

PIN FUNCTION DESCRIPTIONS (continued)

Pin No. 52-Lead MQFP	Pin No. 56-Lead CSP	Mnemonic	Type*	Description
40	43	$\overline{EA}$	I/O	External Access Enable, Logic Input. When held high, this input enables the device to fetch code from internal program memory locations 0000h to F7FFh. When held low, this input enables the device to fetch all instructions from external program memory. To determine the mode of code execution, i.e., internal or external, the $\overline{EA}$ pin is sampled at the end of an external RESET assertion or as part of a device power cycle. $\overline{EA}$ may also be used as an external emulation I/O pin, and therefore the voltage level at this pin must not be changed during normal mode operation as it may cause an emulation interrupt that will halt code execution.
41	44	$\overline{PSEN}$	O	Program Store Enable, Logic Output. This output is a control signal that enables the external program memory to the bus during external fetch operations. It is active every six oscillator periods except during external data memory accesses. This pin remains high during internal program execution. $\overline{PSEN}$ can also be used to enable Serial Download mode when pulled low through a resistor at the end of an external RESET assertion or as part of a device power cycle.
42	45	ALE	O	Address Latch Enable, Logic Output. This output is used to latch the low byte (and page byte for 24-bit data address space accesses) of the address to external memory during external code or data memory access cycles. It is activated every six oscillator periods except during an external data memory access. It can be disabled by setting the PCON.4 bit in the PCON SFR.
43–46 49–52	46–49 52–55	P0.0–P0.7 (AD0–AD3)	I/O	These pins are part of Port 0, which is an 8-bit, open-drain, bidirectional I/O port. Port 0 pins that have 1s written to them float and in that state can be used (AD4–AD7)as high impedance inputs. An external pull-up resistor will be required on P0 outputs to force a valid logic high level externally. Port 0 is also the multiplexed low order address and data bus during accesses to external program or data memory. In this application, it uses strong internal pull-ups when emitting 1s.
		EPAD		Exposed Pad. The LFCSP has an exposed pad that must be soldered to the metal plate on the printed circuit board (PCB) for mechanical reasons and to DGND.

* I = Input, O = Output, S = Supply.

MEMORY ORGANIZATION

The ADuC836 contains four different memory blocks:

- 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

(1) Flash/EE Program Memory

The ADuC836 provides 62 Kbytes of Flash/EE program memory to run user code. The user can choose to run code from this internal memory or run code from an external program memory.

If the user applies power or resets the device while the $\overline{EA}$ pin is pulled low externally, the part will execute code from the external program space; otherwise, if $\overline{EA}$ is pulled high externally, the part defaults to code execution from its internal 62 Kbytes of Flash/EE program memory.

Unlike the ADuC816, where code execution can overflow from the internal code space to external code space once the PC becomes greater than 1FFFH, the ADuC836 does not support the rollover from F7FFH in internal code space to F800H in external code space. Instead, the 2048 bytes between F800H and FFFFH will appear as NOP instructions to user code.

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.

56 Kbytes of the program memory can be reprogrammed during runtime; thus the code space can be upgraded in the field 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.

(2) 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 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.

(3) General-Purpose RAM

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

The lower 128 bytes of internal data memory are mapped as shown in Figure 2. The lowest 32 bytes are grouped into four banks of eight registers addressed as R0 through R7. The next 16 bytes (128 bits), locations 20H through 2FH above the register banks, form a block of directly addressable bit locations at bit addresses 00H through 7FH. The stack can be located anywhere in the internal memory address space, and the stack depth can be expanded up to 2048 bytes.

GENERAL NOTES PERTAINING TO THIS DATA SHEET

1. SET implies a Logic 1 state and CLEARED implies a Logic 0 state, unless otherwise stated.
2. SET and CLEARED also imply that the bit is set or automatically cleared by the ADuC836 hardware, unless otherwise stated.
3. User software should not write 1s to reserved or unimplemented bits as they may be used in future products.
4. Any pin numbers used throughout this data sheet refer to the 52-lead MQFP package, unless otherwise stated.

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.

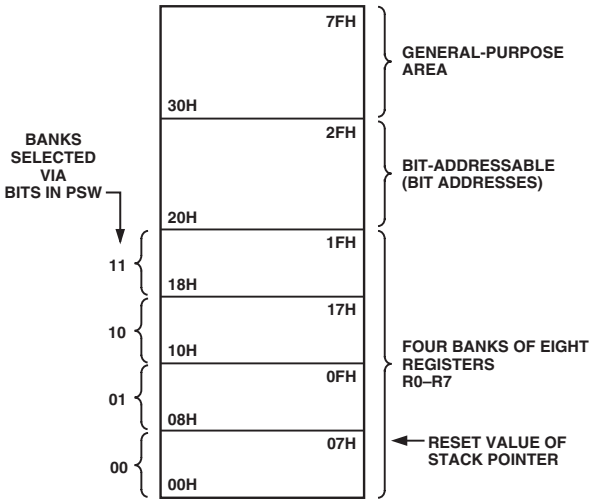

Figure 2. Lower 128 Bytes of Internal Data Memory

(4) Internal XRAM

The ADuC836 contains 2 Kbytes of on-chip extended data memory. This memory, although on-chip, is accessed via the MOVX instruction. The 2 Kbytes of internal XRAM are mapped into the bottom 2 Kbytes of the external address space if the CFG836.0 bit is set. Otherwise, access to the external data memory will occur just like a standard 8051.

Even with the CFG836.0 bit set, access to the external XRAM will occur once the 24-bit DPTR is greater than 0007FFH.

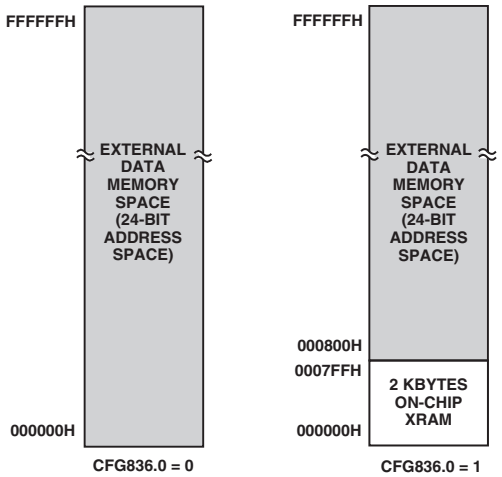

Figure 3. Internal and External XRAM

When accessing the internal XRAM, the P0 and P2 port pins, as well as the $\overline{RD}$ and $\overline{WR}$ strobes, will not be output as per a standard 8051 MOVX instruction. This allows the user to use these port pins as standard I/O.

The upper 1792 bytes of the internal XRAM can be configured to be used as an extended 11-bit stack pointer. By default, the stack will operate exactly like an 8052 in that it will roll over from FFH to 00H in the general-purpose RAM. On the ADuC836 however, it is possible (by setting CFG836.7) to enable the 11-bit extended stack pointer. In this case, the stack will roll 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 three extra bits necessary to extend the 8-bit stack pointer into an 11-bit stack pointer.

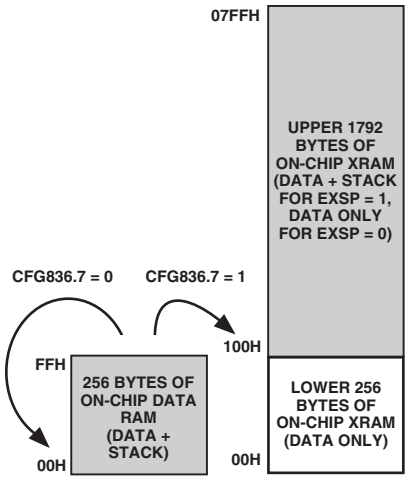

Figure 4. Extended Stack Pointer Operation

External Data Memory (External XRAM)
Just like a standard 8051 compatible core, the ADuC836 can access external data memory using a MOVX instruction. The MOVX instruction automatically outputs the various control strobes required to access the data memory.

The ADuC836, however, can access up to 16 Mbytes of external data memory. This is an enhancement of the 64 Kbytes external data memory space available on a standard 8051 compatible core.

The external data memory is discussed in more detail in the ADuC836 Hardware Design Considerations section.

SPECIAL FUNCTION REGISTERS (SFRS)
The SFR space is mapped into the upper 128 bytes of internal data memory space and accessed by direct addressing only. It provides an interface between the CPU and all on-chip peripherals. A block diagram showing the programming model of the ADuC836 via the SFR area is shown in Figure 5.

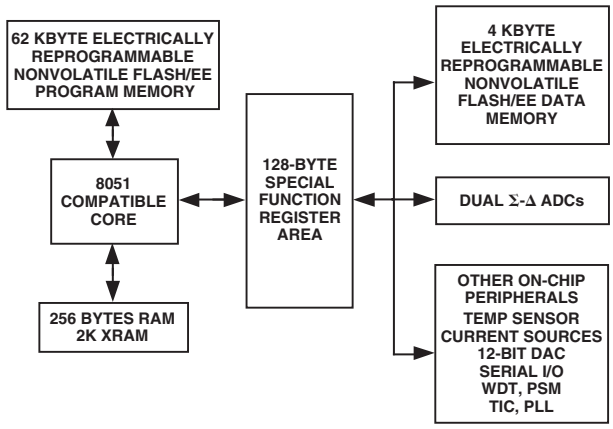

Figure 5. Programming Model

All registers, except the Program Counter (PC) and the four general-purpose register banks, reside in the SFR area. The SFR registers include control, configuration, and data registers that provide an interface between the CPU and all on-chip peripherals.

Accumulator SFR (ACC)
ACC is the Accumulator Register, which is used for math operations including addition, subtraction, integer multiplication, and division, and Boolean bit manipulations. The mnemonics for accumulator-specific instructions, refer to the Accumulator as A.

B SFR (B)
The B Register is used with the ACC for multiplication and division operations. For other instructions, it can be treated as a general-purpose scratch pad register.

Data Pointer (DPTR)
The Data Pointer is made up of three 8-bit registers, named DPP (page byte), DPH (high byte), and DPL (low byte). These are used to provide memory addresses for internal and external code access and external data access. It may be manipulated as a 16-bit register (DPTR = DPH, DPL), although INC DPTR instructions will automatically carry over to DPP, or as three independent 8-bit registers (DPP, DPH, DPL).

The ADuC836 supports dual data pointers. For more information, refer to the Dual Data Pointer section.

Reference Input
The ADuC836's reference inputs, REFIN(+) and REFIN(–), provide a differential reference input capability. The common-mode range for these differential inputs is from AGND to AV_{DD}. The nominal reference voltage, V_{REF} (REFIN(+) – REFIN(–)), for specified operation is 2.5 V with the primary and auxiliary reference enable bits set in the respective ADC0CON and/or ADC1CON SFRs.

The part is also functional (although not specified for performance) when the XREF0 or XREF1 bits are 0, which enables the on-chip internal band gap reference. In this mode, the ADCs will see the internal reference of 1.25 V, therefore halving all input ranges. As a result of using the internal reference voltage, a noticeable degradation in peak-to-peak resolution will result. Therefore, for best performance, operation with an external reference is strongly recommended.

In applications where the excitation (voltage or current) for the transducer on the analog input also drives the reference voltage for the part, the effect of the low frequency noise in the excitation source will be removed as the application is ratiometric. If the ADuC836 is not used in a ratiometric application, a low noise reference should be used. Recommended reference voltage sources for the ADuC836 include the AD780, REF43, and REF192.

It should also be noted that the reference inputs provide a high impedance, dynamic load. Because the input impedance of each reference input is dynamic, resistor/capacitor combinations on these inputs can cause dc gain errors depending on the output impedance of the source that is driving the reference inputs. Reference voltage sources, like those recommended above (e.g., AD780), will typically have low output impedances and therefore decoupling capacitors on the REFIN(+) input would be recommended. Deriving the reference input voltage across an external resistor, as shown in Figure 66, will mean that the reference input sees a significant external source impedance. External decoupling on the REFIN(+) and REFIN(–) pins would not be recommended in this type of circuit configuration.

Burnout Currents
The primary ADC on the ADuC836 contains two 100 nA constant current generators: one sourcing current from AV_{DD} to AIN(+) and one sinking from AIN(–) to AGND. The currents are switched to the selected analog input pair. Both currents are either on or off, depending on the Burnout Current Enable (BO) bit in the ICON SFR (see Table IX). These currents can be used to verify that an external transducer is still operational before attempting to take measurements on that channel. Once the burnout currents are turned on, they will flow in the external transducer circuit, and a measurement of the input voltage on the analog input channel can be taken. If the resultant voltage measured is full-scale, it indicates that the transducer has gone open-circuit. If the voltage measured is 0 V, it indicates that the transducer has short circuited. For normal operation, these burnout currents are turned off by writing a 0 to the BO bit in the ICON SFR. The current sources work over the normal absolute input voltage range specifications.

Excitation Currents
The ADuC836 also contains two identical, 200 μA constant current sources. Both source current from AV_{DD} to Pin 3 (IEXC1) or Pin 4 (IEXC2). These current sources are controlled via bits in the ICON SFR shown in Table IX. They can be configured to source 200 μA individually to both pins or a combination of both currents, i.e., 400 μA, to either of the selected pins. These current sources can be used to excite external resistive bridge or RTD sensors.

Reference Detect
The ADuC836 includes on-chip circuitry to detect if the part has a valid reference for conversions or calibrations. If the voltage between the external REFIN(+) and REFIN(–) pins goes below 0.3 V or either the REFIN(+) or REFIN(–) inputs is open circuit, the ADuC836 detects that it no longer has a valid reference. In this case, the NOXREF bit of the ADCSTAT SFR is set to a 1. If the ADuC836 is performing normal conversions and the NOXREF bit becomes active, the conversion results revert to all 1s. It is not necessary to continuously monitor the status of the NOXREF bit when performing conversions. It is only necessary to verify its status if the conversion result read from the ADC Data Register is all 1s.

If the ADuC836 is performing either an offset or gain calibration and the NOXREF bit becomes active, the updating of the respective calibration registers is inhibited to avoid loading incorrect coefficients to these registers, and the appropriate ERR0 or ERR1 bits in the ADCSTAT SFR are set. If the user is concerned about verifying that a valid reference is in place every time a calibration is performed, the status of the ERR0 or ERR1 bit should be checked at the end of the calibration cycle.

Σ-Δ Modulator
A Σ-Δ ADC generally consists of two main blocks, an analog modulator and a digital filter. In the case of the ADuC836 ADCs, the analog modulators consist of a difference amplifier, an integrator block, a comparator, and a feedback DAC, as illustrated in Figure 10.

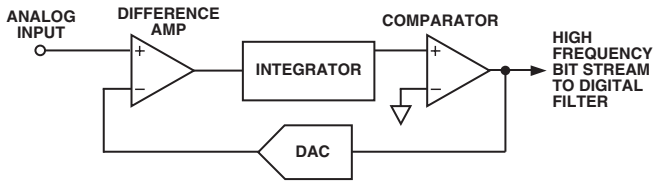

Figure 10. Σ-Δ Modulator Simplified Block Diagram

In operation, the analog signal sample is fed to the difference amplifier along with the output of the feedback DAC. The difference between these two signals is integrated and fed to the comparator. The output of the comparator provides the input to the feedback DAC so the system functions as a negative feedback loop that tries to minimize the difference signal. The digital data that represents the analog input voltage is contained in the duty cycle of the pulse train appearing at the output of the comparator. This duty cycle data can be recovered as a data-word using a subsequent digital filter stage. The sampling frequency of the modulator loop is many times higher than the bandwidth of the input signal. The integrator in the modulator shapes the quantization noise (which results from the analog-to-digital conversion) so that the noise is pushed toward one-half of the modulator frequency.

NONVOLATILE FLASH/EE MEMORY

Flash/EE Memory Overview

The ADuC836 incorporates Flash/EE memory technology on-chip to provide the user with nonvolatile, in-circuit, reprogrammable code and data memory space. Flash/EE memory is a relatively recent type of nonvolatile memory technology and is based on a single transistor cell architecture. This technology is basically an outgrowth of EPROM technology and was developed through the late 1980s. Flash/EE memory takes the flexible in-circuit reprogrammable features of EEPROM and combines them with the space efficient/density features of EPROM (see Figure 15).

Because Flash/EE technology is based on a single transistor cell architecture, a Flash memory array, like EPROM, can be implemented to achieve the space efficiencies or memory densities required by a given design.

Like EEPROM, flash memory can be programmed in-system at a byte level, although it must first be erased; the erase being performed in page blocks. Thus, flash memory is often and more correctly referred to as Flash/EE memory.

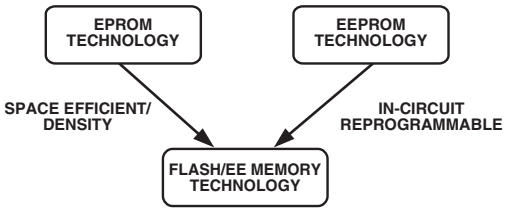

Figure 15. Flash/EE Memory Development

Overall, Flash/EE memory represents a step closer to the ideal memory device that includes nonvolatility, in-circuit programmability, high density, and low cost. Incorporated into the ADuC836, Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one-time programmable (OTP) devices at remote operating nodes.

Flash/EE Memory and the ADuC836

The ADuC836 provides two arrays of Flash/EE memory for user applications. 62 Kbytes of Flash/EE program space are provided on-chip to facilitate code execution without any external discrete ROM device requirements. The program memory can be programmed in-circuit, using the serial download mode provided, using conventional third party memory programmers, or via any user defined protocol in User Download (ULOAD) mode.

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

ADuC836 Flash/EE Memory Reliability

The Flash/EE program and data memory arrays on the ADuC836 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, which are defined as:

a. Initial page erase sequence
b. Read/verify sequence
c. Byte program sequence
d. Second read/verify sequence

A Single Flash/EE Memory 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 Specification tables, the ADuC836 Flash/EE Memory Endurance qualification has been carried out in accordance with JEDEC Specification A117 over the industrial temperature range of –40°C, +25°C, +85°C, and +125°C. The results allow the specification of a minimum endurance figure over supply and 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 ADuC836 has been qualified in accordance with the formal JEDEC Retention Lifetime Specification (A117) at a specific junction temperature (T_J = 55°C). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit described above, before data retention is characterized. This means that the Flash/EE memory is guaranteed to retain its data for its full specified retention lifetime every time the Flash/EE memory is reprogrammed. It should also be noted that retention lifetime, based on an activation energy of 0.6 eV, will derate with T_J, as shown in Figure 16.

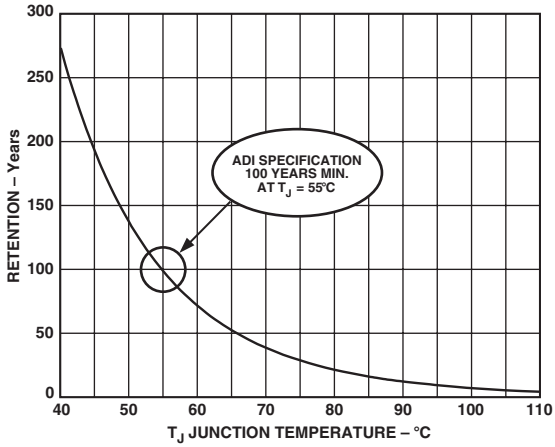

Figure 16. Flash/EE Memory Data Retention

Flash/EE Program Memory

The ADuC836 contains a 64 Kbyte array of Flash/EE program memory. The lower 62 Kbytes of this program memory are available to the user, and can be used for program storage or indeed 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 (ADC, temperature sensor, current sources, band gap references, and so on).

This 2 Kbyte embedded firmware is hidden from user code. Attempts to read this space will read 0s, i.e., the embedded firmware appears as NOP instructions to user code.

In normal operating mode (power-up 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 17.

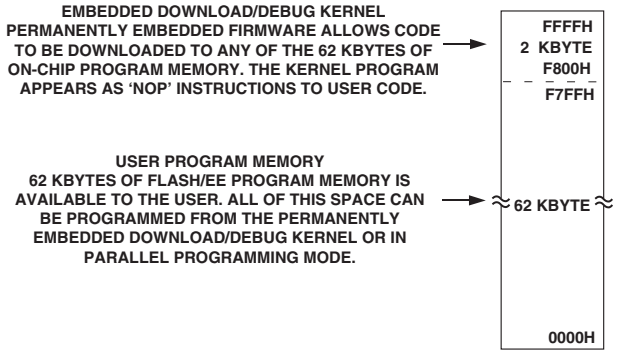

Figure 17. Flash/EE Program Memory Map in Normal Mode

In Normal mode, the 62 Kbytes of Flash/EE program memory can be programmed by serial downloading or parallel processing:

(1) Serial Downloading (In-Circuit Programming)

The ADuC836 facilitates code download via the standard UART serial port. The ADuC836 will 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 hidden embedded download kernel will execute. 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 is provided as part of the ADuC836 QuickStart development system. Application Note uC004 fully describes the serial download protocol that is used by the embedded download kernel. This Application Note is available at www.analog.com/microconverter.

(2) 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 18. 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. Port 1.1, P1.2, P1.3, and P1.4 are used as a general configuration port that configures the device for various program and erase operations during parallel programming.

Table XIII. 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 Modes
1	0	0	1	Read Device Signature/ID
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

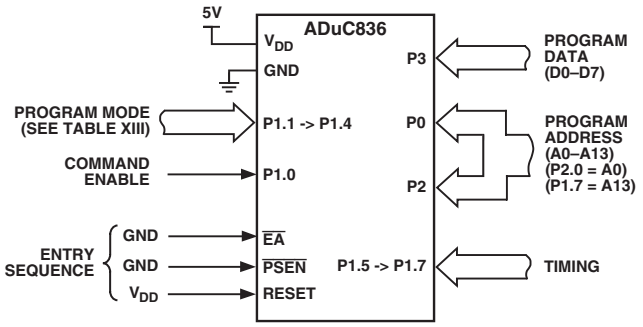

Figure 18. Flash/EE Memory Parallel Programming

User Download Mode (ULOAD)

In Figure 17 we can see that it was 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 of the Flash/EE memory is read only to user code.

However, the Flash/EE program memory can also be written to during runtime simply by entering ULOAD mode. In ULOAD mode, the lower 56 Kbytes of program memory can be erased and reprogrammed by user software, as shown in Figure 19. ULOAD mode can be used to upgrade your code in the field via any user defined download protocol. Configuring the SPI port on the ADuC836 as a slave, it is possible to completely reprogram the 56 Kbytes of Flash/EE program memory in only 5 seconds (see Application Note uC007).

Alternatively, ULOAD mode can be used to save data to the 56 Kbytes of Flash/EE memory. This can be extremely useful in data logging applications where the ADuC836 can provide up to 60 Kbytes of NV data memory on-chip (4 Kbytes of dedicated Flash/EE data memory also exist).

The upper 6 Kbytes of the 62 Kbytes of Flash/EE program memory are only programmable via serial download or parallel programming. This means that this space appears as read-only to user code. Therefore, it cannot be accidently erased or reprogrammed by erroneous code execution. This makes it very suitable to use the 6 Kbytes as a bootloader. A Bootload Enable option exists in the serial downloader to "Always RUN from E000h after Reset." If using a bootloader, this option is recommended to ensure that the bootloader always executes correct code after reset.

Programming the Flash/EE program memory via ULOAD mode is described in more detail in the description of ECON and also in Application Note uC007.

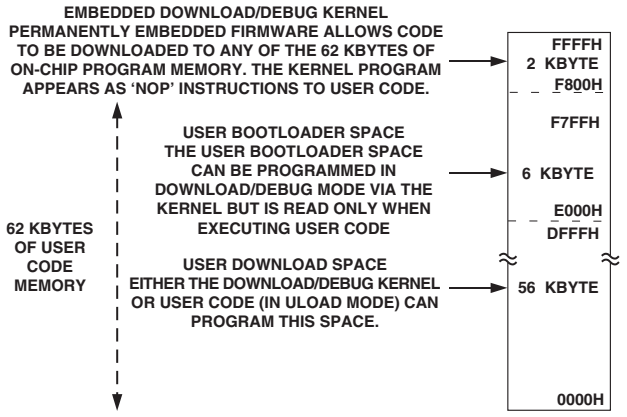

Figure 19. Flash/EE Program Memory Map in ULOAD Mode

Flash/EE Program Memory Security

The ADuC836 facilitates three modes of Flash/EE program memory security. These modes can be independently activated, restricting access to the internal code space. These security modes can be enabled as part of serial download protocol, as described in Application Note uC004, or via parallel programming. The ADuC836 offers the following security modes:

Lock Mode

This mode locks the code memory, disabling parallel programming of the program memory. However, reading the memory in Parallel mode and reading the memory via a MOVC command from external memory is still allowed. This mode is deactivated by initiating 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 is 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 part into Serial Download mode, i.e., RESET asserted and deasserted with $\overline{\text{PSEN}}$ low, the part will interpret the serial download reset as a normal reset only. It will therefore not enter Serial Download mode, but only execute a normal reset sequence. Serial Safe mode can only be disabled by initiating an "erase code and data" command in parallel programming mode.

INTVAL	**User Time Interval Select Register**
Function	User code writes the required time interval to this register. When the 8-bit interval counter is equal to the time interval value loaded in the INTVAL SFR, the TII bit (TIMECON.2) is set and generates an interrupt if enabled. (See IEIP2 SFR description under the Interrupt System section.)
SFR Address	A6H
Power-On Default Value	00H
Reset Default Value	00H
Bit Addressable	No
Valid Value	0 to 255 decimal

HTHSEC	**Hundredths Seconds Time Register**
Function	This register is incremented in 1/128 second intervals once TCEN in TIMECON is active. The HTHSEC SFR counts from 0 to 127 before rolling over to increment the SEC time register.
SFR Address	A2H
Power-On Default Value	00H
Reset Default Value	00H if TCEN = 0, previous value before reset if TCEN = 1
Bit Addressable	No
Valid Value	0 to 127 decimal

SEC	**Seconds Time Register**
Function	This register is incremented in 1-second intervals once TCEN in TIMECON is active. The SEC SFR counts from 0 to 59 before rolling over to increment the MIN time register.
SFR Address	A3H
Power-On Default Value	00H
Reset Default Value	00H if TCEN = 0, previous value before reset if TCEN = 1
Bit Addressable	No
Valid Value	0 to 59 decimal

MIN	**Minutes Time Register**
Function	This register is incremented in 1-minute intervals once TCEN in TIMECON is active. The MIN counts from 0 to 59 before rolling over to increment the HOUR time register.
SFR Address	A4H
Power-On Default Value	00H
Reset Default Value	00H if TCEN = 0, previous value before reset if TCEN = 1
Bit Addressable	No
Valid Value	0 to 59 decimal

HOUR	**Hours Time Register**
Function	This register is incremented in 1-hour intervals once TCEN in TIMECON is active. The HOUR SFR counts from 0 to 23 before rolling over to 0.
SFR Address	A5H
Power-On Default Value	00H
Reset Default Value	00H if TCEN = 0, previous value before reset if TCEN = 1
Bit Addressable	No
Valid Value	0 to 23 decimal

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

Depending on the configuration of the bits in the SPICON SFR shown in Table XXI, the ADuC836 SPI interface will transmit or receive data in a number of possible modes. Figure 34 shows all possible ADuC836 SPI configurations and the timing relationships and synchronization between the signals involved. Also shown in this figure is the SPI Interrupt bit (ISPI) and how it is triggered at the end of each byte-wide communication.

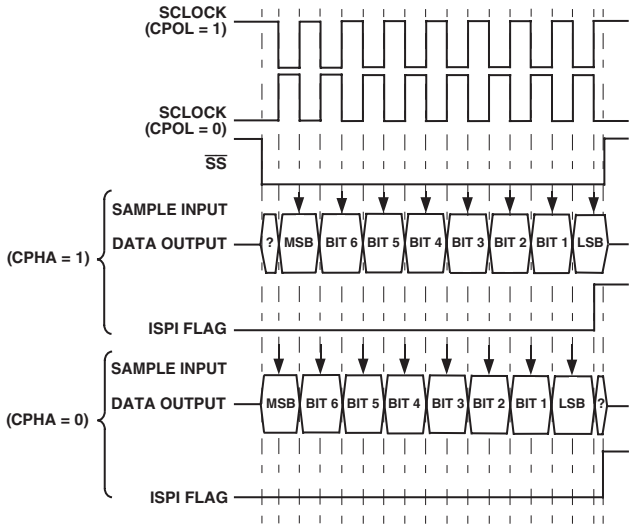

Figure 34. SPI Timing, All Modes

SPI Interface—Master Mode

In Master mode, the SCLOCK pin is always an output and generates a burst of eight clocks whenever user code writes to the SPIDAT Register. The SCLOCK bit rate is determined by SPR0 and SPR1 in SPICON. It should also be noted that the $\overline{SS}$ pin is not used in Master mode. If the ADuC836 needs to assert the $\overline{SS}$ pin on an external slave device, a port digital output pin should be used.

In Master mode, a byte transmission or reception is initiated by a write to SPIDAT. Eight clock periods are generated via the SCLOCK pin and the SPIDAT byte being transmitted via MOSI. With each SCLOCK period, a data bit is also sampled via MISO. After eight clocks, the transmitted byte will have been completely transmitted and the input byte will be waiting in the input shift register. The ISPI flag will be set automatically and an interrupt will occur if enabled. The value in the shift register will be latched into SPIDAT.

SPI Interface—Slave Mode

In Slave mode, the SCLOCK is an input. The $\overline{SS}$ pin must also be driven low externally during the byte communication. Transmission is also initiated by a write to SPIDAT. In Slave mode, a data bit is transmitted via MISO and a data bit is received via MOSI through each input SCLOCK period. After eight clocks, the transmitted byte will have been completely transmitted and the input byte will be waiting in the input shift register. The ISPI flag will be set automatically and an interrupt will occur if enabled. The value in the shift register will be latched into SPIDAT only when the transmission/reception of a byte has been completed. The end of transmission occurs after the eighth clock has been received, if CPHA = 1 or when $\overline{SS}$ returns high if CPHA = 0.

P1.2 to P1.7

The remaining Port 1 pins (P1.2 to P1.7) can only be configured as analog input (ADC) or digital input pins. By (power-on) default, these pins are configured as analog inputs, i.e., 1 written in the corresponding Port 1 register bit. To configure any of these pins as digital inputs, the user should write a 0 to these port bits to configure the corresponding pin as a high impedance digital input. Figure 39 illustrates this function. Note that there are no output drivers for Port 1 pins, and they therefore cannot be used as outputs.

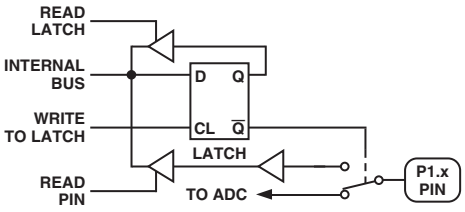

Figure 39. P1.2 to P1.7 Bit Latch and I/O Buffer

Port 2

Port 2 is a bidirectional port with internal pull-up resistors directly controlled via the P2 SFR. Port 2 also emits the high order address bytes during fetches from external program memory and middle and high order address bytes during accesses to the 24-bit external data memory space.

As shown in Figure 40, the output drivers of Port 2 are switchable to an internal ADDR bus by an internal CONTROL signal for use in external memory accesses (as for Port 0). In external memory addressing mode (CONTROL = 1), the port pins feature push/pull operation controlled by the internal address bus (ADDR line). However, unlike the P0 SFR during external memory accesses, the P2 SFR remains unchanged.

In general-purpose I/O port mode, Port 2 pins that have 1s written to them are pulled high by the internal pull-ups (Figure 38), and in that state can be used as inputs. As inputs, Port 2 pins being pulled externally low will source current because of the internal pull-up resistors. Port 2 pins with 0s written to them will drive a logic low output voltage (V_{OL}) and will be capable of sinking 1.6 mA.

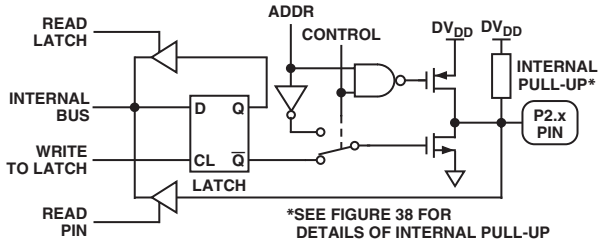

Figure 40. Port 2 Bit Latch and I/O Buffer

Port 3

Port 3 is a bidirectional port with internal pull-ups directly controlled via the P3 SFR.

Port 3 pins that have 1s written to them are pulled high by the internal pull-ups, and in that state can be used as inputs. As inputs, Port 3 pins being pulled externally low will source current because of the internal pull-ups. Port 3 pins with 0s written to them will drive a logic low output voltage (V_{OL}) and will be capable of sinking 1.6 mA.

Port 3 pins also have various secondary functions described in Table XXV. The alternate functions of Port 3 pins can be activated only if the corresponding bit latch in the P3 SFR contains a 1. Otherwise, the port pin is stuck at 0.

Table XXV. Port 3, Alternate Pin Functions

Pin	Alternate Function
P3.0	RxD (UART Input Pin) (or Serial Data I/O in Mode 0)
P3.1	TxD (UART Output Pin) (or Serial Clock Output in Mode 0)
P3.2	$\overline{INT0}$ (External Interrupt 0)
P3.3	$\overline{INT1}$ (External Interrupt 1)
P3.4	T0 (Timer/Counter 0 External Input) PWMCLK (PWM External Clock)
P3.5	T1 (Timer/Counter 1 External Input)
P3.6	$\overline{WR}$ (External Data Memory Write Strobe)
P3.7	$\overline{RD}$ (External Data Memory Read Strobe)

Port 3 pins have the same bit latch and I/O buffer configurations as the P1.0 and P1.1, as shown in Figure 41. The internal pull-up configuration is also defined by the one in Figure 38.

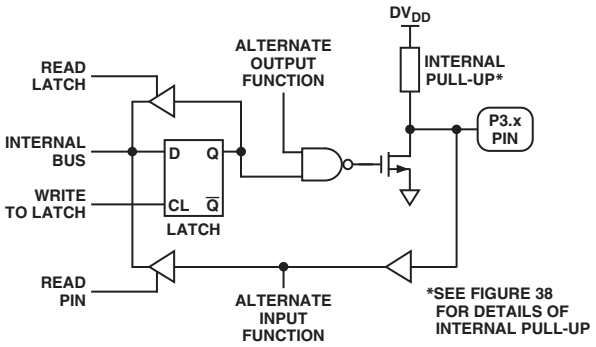

Figure 41. Port 3 Bit Latch and I/O Buffer

Additional Digital I/O

In addition to the port pins, the dedicated SPI/I2C pins (SCLOCK and SDATA/MOSI) also feature both input and output functions. Their equivalent I/O architectures are illustrated in Figure 42 and Figure 44, respectively, for SPI operation, and in Figure 43 and Figure 45 for I^2C operation.

Notice that in I^2C mode (SPE = 0), the strong pull-up FET (Q1) is disabled leaving only a weak pull-up (Q2) present. By contrast, in SPI mode (SPE = 1), the strong pull-up FET (Q1) is controlled directly by SPI hardware, giving the pin push/pull capability.

In I^2C mode (SPE = 0), two pull-down FETs (Q3 and Q4) operate in parallel in order to provide an extra 60% or 70% of current sinking capability. In SPI mode, however, (SPE = 1), only one of the pull-down FETs (Q3) operates on each pin resulting in sink capabilities identical to that of Port 0 and Port 2 pins.

On the input path of SCLOCK, notice that a Schmitt trigger conditions the signal going to the SPI hardware to prevent false triggers (double triggers) on slow incoming edges. For incoming signals from the SCLOCK and SDATA pins going to I^2C hardware, a filter conditions the signals to reject glitches of up to 50 ns in duration.

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

Table XXVII. TCON SFR Bit Designations

Bit	Name	Description
7	TF1	Timer 1 Overflow Flag. Set by hardware on a Timer/Counter 1 overflow. Cleared by hardware when the Program Counter (PC) vectors to the interrupt service routine.
6	TR1	Timer 1 Run Control Bit. Set by user to turn on Timer/Counter 1. Cleared by user to turn off Timer/Counter 1.
5	TF0	Timer 0 Overflow Flag. Set by hardware on a Timer/Counter 0 overflow. Cleared by hardware when the PC vectors to the interrupt service routine.
4	TR0	Timer 0 Run Control Bit. Set by user to turn on Timer/Counter 0. Cleared by user to turn off Timer/Counter 0.
3	IE1*	External Interrupt 1 (INT1) Flag. Set by hardware by a falling edge or zero level being applied to external interrupt pin INT1, depending on bit IT1 state. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source rather than the on-chip hardware, controls the request flag.
2	IT1*	External Interrupt 1 (IE1) Trigger Type. Set by software to specify edge-sensitive detection (i.e., 1-to-0 transition). Cleared by software to specify level-sensitive detection (i.e., zero level).
1	IE0*	External Interrupt 0 (INT0) Flag. Set by hardware by a falling edge or zero level being applied to external interrupt pin INT0, depending on bit IT0 state. Cleared by hardware when the PC vectors to the interrupt service routine only if the interrupt was transition-activated. If level-activated, the external requesting source controls the request flag, rather than the on-chip hardware.
0	IT0*	External Interrupt 0 (IE0) Trigger Type. Set by software to specify edge-sensitive detection (i.e., 1-to-0 transition). Cleared by software to specify level-sensitive detection (i.e., zero level).

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

Timer/Counter 0 and 1 Data Registers
Both Timer 0 and Timer 1 consist of two 8-bit registers. These can be used as independent registers or combined to be a single 16-bit register, depending on the timer mode configuration.

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

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

TIMER/COUNTER 2 OPERATING MODES

The following paragraphs describe the operating modes for Timer/Counter 2. The operating modes are selected by bits in the T2CON SFR, as shown in Table XXIX.

Table XXVIII. Timer 2 Operating Modes

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

16-Bit Autoreload Mode

Autoreload mode has two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, when Timer 2 rolls over, it not only sets TF2 but also causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2L and RCAP2H, which are preset by software. If EXEN2 = 1, Timer 2 still performs the above, but with the added feature that a 1-to-0 transition at external input T2EX will also trigger the 16-bit reload and set EXF2. The Autoreload mode is illustrated in Figure 52.

16-Bit Capture Mode

Capture Mode has two options, which are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter that, upon overflowing, sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, Timer 2 still performs the above, but a l-to-0 transition on external input T2EX causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into registers RCAP2L and RCAP2H, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set; EXF2, like TF2, can generate an interrupt. Capture mode is illustrated in Figure 53.

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1.

In either case, if Timer 2 is being used to generate the baud rate, the TF2 interrupt flag will not occur. Therefore Timer 2 interrupts will not occur so they do not have to be disabled. However, in this mode, the EXF2 flag can still cause interrupts and this can be used as a third external interrupt.

Baud rate generation will be described as part of the UART serial port operation.

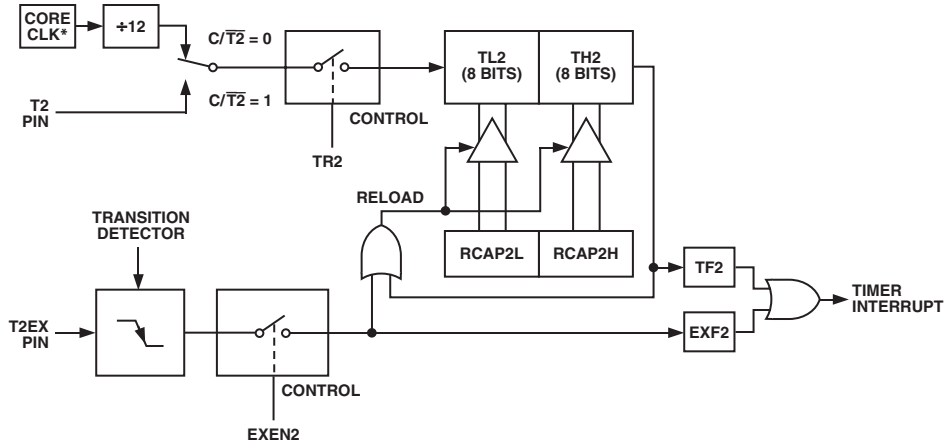

Figure 52. Timer/Counter 2, 16-Bit Autoreload Mode

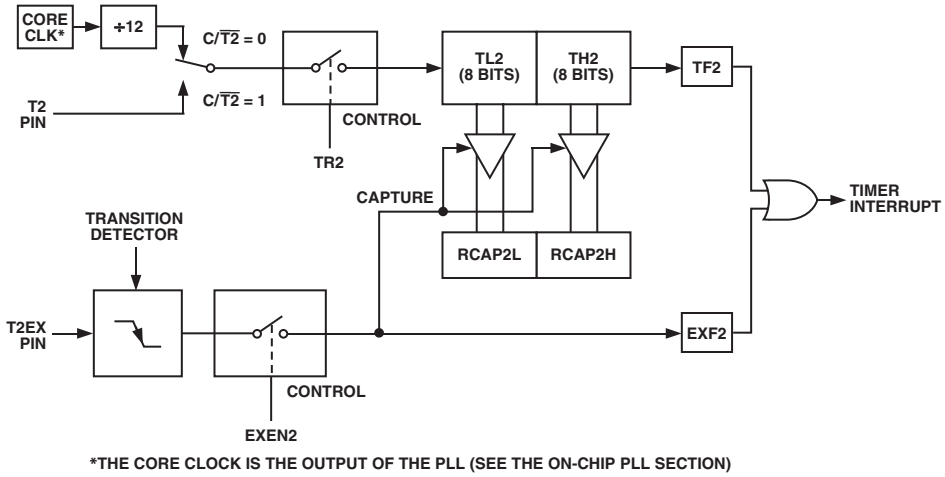

Figure 53. Timer/Counter 2, 16-Bit Capture Mode

UART SERIAL INTERFACE

The serial port is full-duplex, meaning it can transmit and receive simultaneously. It is also receive-buffered, meaning it can commence reception of a second byte before a previously received byte has been read from the receive register. However, if the first byte still has not been read by the time reception of the second byte is complete, the first byte will be 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 the following registers:

SBUF

The serial port receive and transmit registers are both 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 Registers**
SFR Address	98H
Power-On Default Value	00H
Bit Addressable	Yes

Table XXX. SCON SFR Bit Designations

Bit	Name	Description
7	SM0	UART Serial Mode Select Bits.
6	SM1	These bits select the Serial Port operating mode as follows: SM0 SM1 Selected Operating Mode 0 0 Mode 0: Shift Register, fixed baud rate (f_{CORE}/12) 0 1 Mode 1: 8-bit UART, variable baud rate 1 0 Mode 2: 9-bit UART, fixed baud rate (f_{CORE}/64) or (f_{CORE}/32) 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 will not be activated if a valid stop bit was not received. If SM2 is cleared, RI will be set as soon as the byte of data has been received. In Modes 2 or 3, if SM2 is set, RI will not be activated if the received ninth data bit in RB8 is 0. If SM2 is cleared, RI will be set as soon as the byte of data has been received.
4	REN	Serial Port Receive Enable Bit. Set by user software to enable serial port reception. Cleared by user software to disable serial port reception.
3	TB8	Serial Port Transmit (Bit 9). The data loaded into TB8 will be the ninth data bit that will be transmitted in Modes 2 and 3.
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.

UART OPERATING MODES

Mode 0: 8-Bit Shift Register Mode

Mode 0 is selected by clearing both the SM0 and SM1 bits in the SFR SCON. Serial data enters and exits through RxD. TxD outputs the shift clock. Eight data bits are transmitted or received. Transmission is initiated by any instruction that writes to SBUF. The data is shifted out of the RxD line. The 8 bits are transmitted with the least significant bit (LSB) first, as shown in Figure 54.

Reception is initiated when the Receive Enable bit (REN) is 1 and the Receive Interrupt bit (RI) is 0. When RI is cleared, the data is clocked into the RxD line and the clock pulses are output from the TxD line.

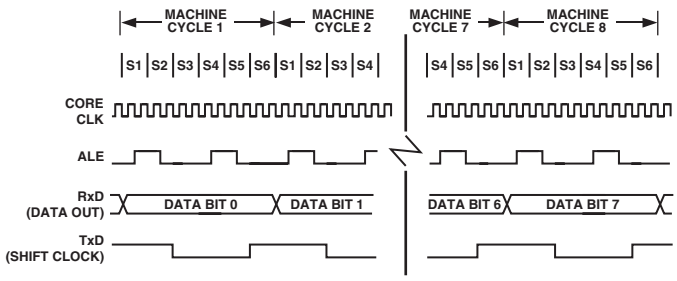

Figure 54. UART Serial Port Transmission, Mode 0

BAUD RATE GENERATION USING TIMER 3

The high integer dividers in a UART block means that high speed baud rates are not always possible using some particular crystals, e.g., using a 12 MHz crystal, a baud rate of 115200 is not possible. To address this problem, the ADuC836 has added a dedicated baud rate timer (Timer 3) specifically for generating highly accurate baud rates.

Timer 3 can be used instead of Timer 1 or Timer 2 for generating very accurate high speed UART baud rates including 115200 and 230400. Timer 3 also allows a much wider range of baud rates to be obtained. In fact, every desired bit rate from 12 bits to 393216 bits 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 57.

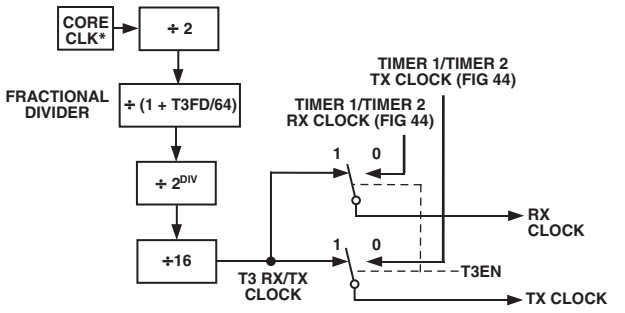

Figure 57. Timer 3, UART Baud Rates

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

Table XXXIII. T3CON SFR Bit Designations

Bit	Name	Description			
7	T3EN	Set to enable Timer 3 to generate the baud rate. When set PCON.7, T2CON.4 and T2CON.5 are ignored. Cleared to let the baud rate be generated as per a standard 8052.			
6	—	Reserved for Future Use			
5	—	Reserved for Future Use			
4	—	Reserved for Future Use			
3	—	Reserved for Future Use			
2	DIV2	Binary Divider Factor			
1	DIV1	DIV2	DIV1	DIV0	Bin Divider
0	DIV0	0	0	0	1
		0	0	1	2
		0	1	0	4
		0	1	1	8
		1	0	0	16
		1	0	1	32
		1	1	0	64
		1	1	1	128

The appropriate value to write to the DIV2-1-0 bits can be calculated using the following formula where f_{CORE} is the output of the PLL, as described in the On-Chip PLL section. Note that the *DIV* value must be rounded down.

$$DIV = \frac{\log\left(\frac{f_{CORE}}{32 \times Baud\ Rate}\right)}{\log(2)}$$

T3FD is the fractional divider ratio required to achieve the required baud rate. We can calculate the appropriate value for T3FD using the following formula. Note that the *T3FD* should be rounded to the nearest integer.

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

Once the values for *DIV* and *T3FD* are calculated, the actual baud rate can be calculated using the following formula:

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

For a baud rate of 115200 while operating from the maximum core frequency (CD = 0), we have:

$$DIV = \log(12582912/32 \times 115200) / \log 2 = 1.77 = 1$$

$$T3FD = (2 \times 12.582912) // (2^1 \times 115200) - 64 = 45.22 = 2Dh$$

Therefore, the actual baud rate is 115439 bits.

Table XXXIV. Commonly Used Baud Rates Using Timer 3

Ideal Baud	CD	DIV	T3CON	T3FD	% Error
230400	0	0	80H	2DH	0.2
115200	0	1	81H	2DH	0.2
115200	1	0	80H	2DH	0.2
57600	0	2	82H	2DH	0.2
57600	1	1	81H	2DH	0.2
57600	2	0	80H	2DH	0.2
38400	0	3	83H	12H	0.1
38400	1	2	82H	12H	0.1
38400	2	1	81H	12H	0.1
38400	3	0	80H	12H	0.1
19200	0	4	84H	12H	0.1
19200	1	3	83H	12H	0.1
19200	2	2	82H	12H	0.1
19200	3	1	81H	12H	0.1
19200	4	0	80H	12H	0.1
9600	0	5	85H	12H	0.1
9600	1	4	84H	12H	0.1
9600	2	3	83H	12H	0.1
9600	3	2	82H	12H	0.1
9600	4	1	81H	12H	0.1
9600	5	0	80H	12H	0.1
38400	0	3	83H	12H	0.1

Power Saving Modes
Setting the Idle and Power-Down Mode Bits, PCON.0 and PCON.1, respectively, in the PCON SFR described in Table II allows the chip to be switched from Normal mode into Idle mode, and also into full Power-Down mode.

In Idle mode, the oscillator continues to run, but the core clock generated from the PLL is halted. The on-chip peripherals continue to receive the clock and remain functional. The CPU status is preserved with the stack pointer, program counter, and all other internal registers maintain their data during Idle mode. Port pins and DAC output pins also retain their states, and ALE and $\overline{\text{PSEN}}$ outputs go high in this mode. The chip will recover from Idle mode upon receiving any enabled interrupt, or upon receiving a hardware reset.

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, being driven directly from the oscillator, can also be enabled during power-down. All other on-chip peripherals, however, 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 $\overline{\text{PSEN}}$ outputs are held low. During full Power-Down mode with the oscillator and wake-up timer running, the ADuC836 typically consumes a total of 15 μA. There are five ways of terminating Power-Down mode:

Asserting the RESET Pin (Pin 15)
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 deasserted.

Cycling Power
All registers are set to their default state and program execution starts at the reset vector approximately 128 ms later.

Time Interval Counter (TIC) Interrupt
If the OSC_PD bit in the PLLCON SFR is clear, the 32 kHz oscillator will remain powered up even in Power-Down mode. If the Time Interval Counter (Wakeup/RTC timer) is enabled, a TIC interrupt will wake the ADuC836 up from Power-Down mode. The CPU services the TIC interrupt. The RETI at the end of the TIC ISR will return the core to the instruction after the one that enabled power-down.

SPI Interrupt
If the SERIPD bit in the PCON SFR is set, then an SPI interrupt, if enabled, will wake up the ADuC836 from Power-Down mode. The CPU services the SPI interrupt. The RETI at the end of the ISR will return the core to the instruction after the one that enabled power-down.

$\overline{\text{INT0}}$ Interrupt
If the INT0PD bit in the PCON SFR is set, an external interrupt 0, if enabled, will wake up the ADuC836 from power-down. The CPU services the SPI interrupt. The RETI at the end of the ISR will return the core to the instruction after the one that enabled power-down.

Wake-Up from Power-Down Latency
Even with the 32 kHz crystal enabled during power-down, the PLL will take some time to lock after a wake-up from power-down. Typically, the PLL will take about 1 ms to lock. During this time, code will execute, but not at the specified frequency. Some operations require an accurate clock, for example, UART communications, to achieve specified 50 Hz/60 Hz rejection from the ADCs. The following code may be used to wait for the PLL to lock:

```
WAITFORLOCK:
    MOV A, PLLCON
    JNB ACC.6, WAITFORLOCK
```

If the crystal has been powered down during power-down, there is an additional delay associated with the startup of the crystal oscillator before the PLL can lock. 32 kHz crystals are inherently slow to oscillate, typically taking about 150 ms. Once again, during this time before lock, code will execute, but the exact frequency of the clock cannot be guaranteed. Again for any timing sensitive operations, it is recommended to wait for lock using the lock bit in PLLCON, as shown in the code above.

Grounding and Board Layout Recommendations
As with all high resolution data converters, special attention must be paid to grounding and PC board layout of ADuC836 based designs in order to achieve optimum performance from the ADCs and DAC.

Although the ADuC836 has separate pins for analog and digital ground (AGND and DGND), the user must not tie these to two separate ground planes unless the two ground planes are connected together very close to the ADuC836, as illustrated in the simplified example of Figure 64a. In systems where digital and analog ground planes are connected together somewhere else (at the system's power supply, for example), they cannot be connected again near the ADuC836 since a ground loop would result. In these cases, tie the ADuC836's AGND and DGND pins all to the analog ground plane, as illustrated in Figure 64b. In systems with only one ground plane, ensure that the digital and analog components are physically separated onto separate halves of the board such that digital return currents do not flow near analog circuitry and vice versa. The ADuC836 can then be placed between the digital and analog sections, as illustrated in Figure 64c.

In all of these scenarios, and in more complicated real-life applications, keep in mind the flow of current from the supplies and back to ground. Make sure the return paths for all currents are as close as possible to the paths the currents took to reach their destinations. For example, do not power components on the analog side of Figure 64b with DV_{DD} since that would force return currents from DV_{DD} to flow through AGND. Also, try to avoid digital currents flowing under analog circuitry, which could happen if the user placed a noisy digital chip on the left half of the board in Figure 64c. Whenever possible, avoid large discontinuities in the ground plane(s) (such as those formed by a long trace on the same layer), since they force return signals to travel a longer path. And of course, make all connections directly to the ground plane with little or no trace separating the pin from its via to ground.

Parameter		12.58 MHz Core_Clk Min	12.58 MHz Core_Clk Max	Variable Core_Clk Min	Variable Core_Clk Max	Unit
EXTERNAL DATA MEMORY READ CYCLE						
t_{RLRH}	$\overline{RD}$ Pulsewidth	377		$6t_{CORE} - 100$		ns
t_{AVLL}	Address Valid after ALE Low	39		$t_{CORE} - 40$		ns
t_{LLAX}	Address Hold after ALE Low	44		$t_{CORE} - 35$		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		232		$5t_{CORE} - 165$	ns
t_{RHDX}	Data and Address Hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float after $\overline{RD}$		89		$2t_{CORE} - 70$	ns
t_{LLDV}	ALE Low to Valid Data In		486		$8t_{CORE} - 150$	ns
t_{AVDV}	Address to Valid Data In		550		$9t_{CORE} - 165$	ns
t_{LLWL}	ALE Low to $\overline{RD}$ Low	188	288	$3t_{CORE} - 50$	$3t_{CORE} + 50$	ns
t_{AVWL}	Address Valid to $\overline{RD}$ Low	188		$4t_{CORE} - 130$		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ High to ALE High	39	119	$t_{CORE} - 40$	$t_{CORE} + 40$	ns

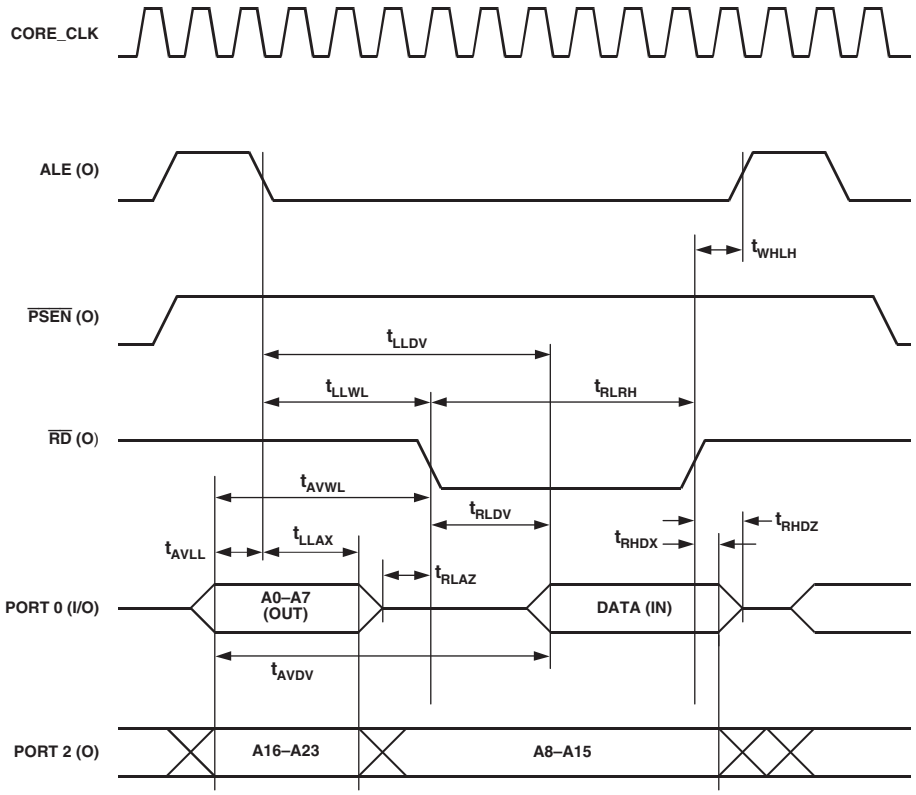

Figure 72. External Data Memory Read Cycle

Parameter		12.58 MHz Core_Clk Min	12.58 MHz Core_Clk Max	Variable Core_Clk Min	Variable Core_Clk Max	Unit
EXTERNAL DATA MEMORY WRITE CYCLE						
t_{WLWH}	$\overline{WR}$ Pulsewidth	377		$6t_{CORE} - 100$		ns
t_{AVLL}	Address Valid after ALE Low	39		$t_{CORE} - 40$		ns
t_{LLAX}	Address Hold after ALE Low	44		$t_{CORE} - 35$		ns
t_{LLWL}	ALE Low to $\overline{WR}$ Low	188	288	$3t_{CORE} - 50$	$3t_{CORE} + 50$	ns
t_{AVWL}	Address Valid to $\overline{WR}$ Low	188		$4t_{CORE} - 130$		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	29		$t_{CORE} - 50$		ns
t_{QVWH}	Data Setup before $\overline{WR}$	406		$7t_{CORE} - 150$		ns
t_{WHQX}	Data and Address Hold after $\overline{WR}$	29		$t_{CORE} - 50$		ns
t_{WHLH}	$\overline{WR}$ High to ALE High	39	119	$t_{CORE} - 40$	$t_{CORE} + 40$	ns

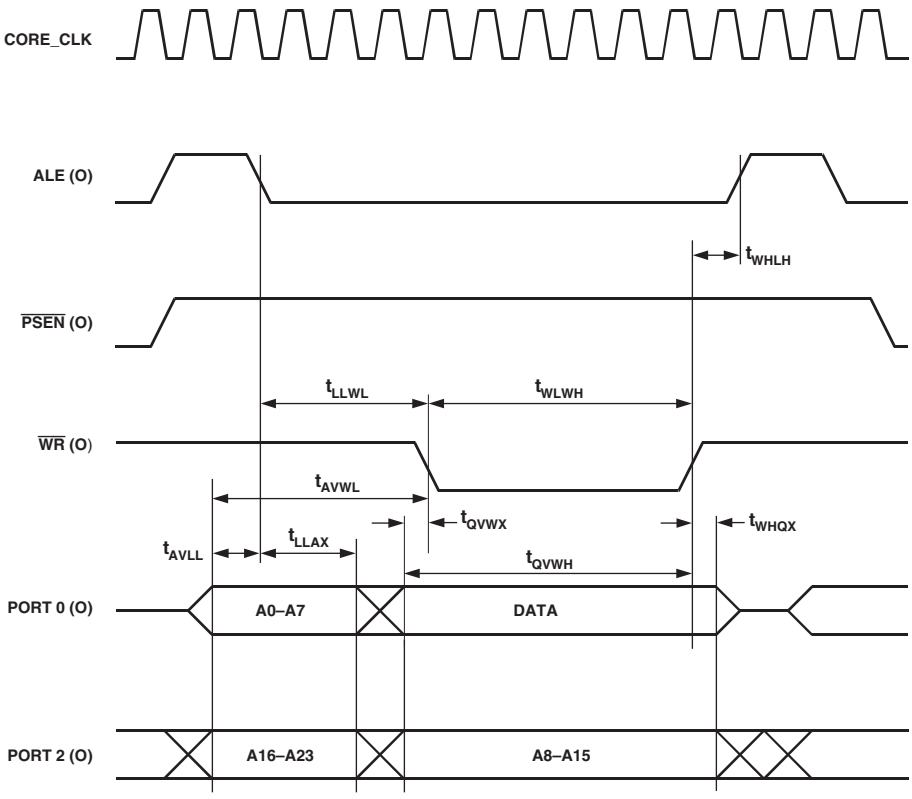

Figure 73. External Data Memory Write Cycle