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	Obsolete
Core Processor	ARM7®
Core Size	16/32-Bit
Speed	20.48MHz
Connectivity	LINbus, SPI, UART/USART
Peripherals	POR, PSM, Temp Sensor, WDT
Number of I/O	9
Program Memory Size	96KB (48K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1.5K x 32
Voltage - Supply (Vcc/Vdd)	3.5V ~ 18V
Data Converters	A/D 2x16b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7032bstz-88-rl

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
PACKAGE THERMAL SPECIFICATIONS					
Thermal Shutdown[31]		140	150	160	°C
Thermal Impedance (θ_{JA})[32]	48-lead LQFP, stacked die				
	Top die		50		°C/W
	Bottom die		25		°C/W
POWER REQUIREMENTS					
Power Supply Voltages					
VDD (Battery Supply)		3.5		18	V
REG_DVDD, REG_AVDD[33]		2.5	2.6	2.7	V
Power Consumption					
I_{DD} (MCU Normal Mode)[34]	MCU clock rate = 10.24 MHz, ADC off		10	20	mA
	MCU clock rate = 20.48 MHz, ADC off		20	30	mA
I_{DD} (MCU Powered Down)[1]	ADC low power mode, measured over an ambient temperature range of −10°C to +40°C (continuous ADC conversion)		300	400	µA
	ADC low power mode, measured over an ambient temperature range of −40°C to +85°C (continuous ADC conversion)		300	500	µA
	ADC low power plus mode, measured over an ambient temperature range of −10°C to +40°C (continuous ADC conversion)		520	700	µA
	Average current, measured with wake-up and watchdog timer clocked from low power oscillator (−40°C to +85°C)		120	300	µA
	Average current, measured with wake-up and watchdog timer clocked from low power oscillator over an ambient temperature range of −10°C to +40°C		120	175	µA
I_{DD} (Current ADC)			1.7		mA
I_{DD} (Voltage/Temperature ADC)	Per ADC		0.5		mA
I_{DD} (Precision Oscillator)			400		µA

[1] Not guaranteed by production test, but by design and/or characterization data at production release.
[2] Valid for current ADC gain setting of PGA = 4 to 64.
[3] These numbers include temperature drift.
[4] Tested at gain range = 4; self-offset calibration removes this error.
[5] Measured with an internal short after an initial offset calibration.
[6] Measured with an internal short.
[7] Includes internal reference temperature drift.
[8] Factory calibrated at gain = 1.
[9] System calibration at specific gain range removes the error at this gain range at that temperature.
[10] Valid when used in conjunction with the ADCREF (the low power mode reference error) MMR.
[11] Typical noise in low power modes is measured with chop enabled.
[12] Voltage channel specifications include resistive attenuator input stage.
[13] Includes an initial system calibration.
[14] System calibration removes this error at that temperature.
[15] RMS noise is referred to voltage attenuator input. For example, at f_{ADC} = 1 kHz, typical rms noise at the ADC input is 7.5 µV, which, when scaled by the attenuator (24), yields these input referred noise figures.
[16] ADC self-offset calibration removes this error.
[17] Valid after an initial self-calibration.
[18] Factory calibrated for the internal temperature sensor during final production test.
[19] In ADC low power mode, the input range is fixed at ±9.375 mV. In ADC low power plus mode, the input range is fixed at ±2.34375 mV.
[20] It is possible to extend the ADC input range by up to 10% by modifying the factory set value of the gain calibration register or using system calibration. This approach can also be used to reduce the ADC input range (LSB size).
[21] Limited by minimum/maximum absolute input voltage range.
[22] Valid for a differential input less than 10 mV.
[23] Measured using box method.
[24] The long-term stability specification is noncumulative. The drift in subsequent 1000 hour periods is significantly lower than in the first 1000 hour period.
[25] References of up to REG_AVDD can be accommodated by enabling an internal divide-by-2.
[26] Die temperature.
[27] Endurance is qualified to 10,000 cycles, as per JEDEC Std. 22 Method A117, and measured at −40°C, +25°C, and +125°C. Typical endurance at 25°C is 170,000 cycles.
[28] Retention lifetime equivalent at junction temperature (T_J) = 85°C, as per JEDEC Std. 22 Method A117. Retention lifetime derates with junction temperature.
[29] Low power oscillator can be calibrated against either the precision oscillator or the external 32.768 kHz crystal in user code.
[30] These numbers are not production tested but are supported by LIN compliance testing.
[31] The MCU core is not shut down, but an interrupt is generated, if enabled.
[32] Thermal impedance can be used to calculate the thermal gradient from ambient to die temperature.
[33] Internal regulated supply available at REG_DVDD (I_{SOURCE} = 5 mA) and REG_AVDD (I_{SOURCE} = 1 mA).
[34] Typical additional supply current consumed during Flash/EE memory program and erase cycles is 7 mA and 5 mA, respectively.

Table 3. SPI Master Mode Timing (PHASE Mode = 0)

Parameter	Description	Min	Typ	Max	Unit
t_{SL}	SCLK low pulse width[1]		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLK high pulse width[1]		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLK edge[2]			(2 × t_{UCLK}) + (2 × t_{HCLK})	ns
t_{DOSU}	Data output setup time before SCLK edge		½ t_{SL}		ns
t_{DSU}	Data input setup time before SCLK edge	0			ns
t_{DHD}	Data input hold time after SCLK edge[2]	3 × t_{UCLK}			ns
t_{DF}	Data output fall time		3.5		ns
t_{DR}	Data output rise time		3.5		ns
t_{SR}	SCLK rise time		3.5		ns
t_{SF}	SCLK fall time		3.5		ns

[1] t_{HCLK} depends on the clock divider or CD bits in PLLCON MMR. $t_{HCLK} = t_{UCLK}/2^{CD}$.
[2] t_{UCLK} = 48.8 ns. It corresponds to the 20.48 MHz internal clock from the PLL before the clock divider.

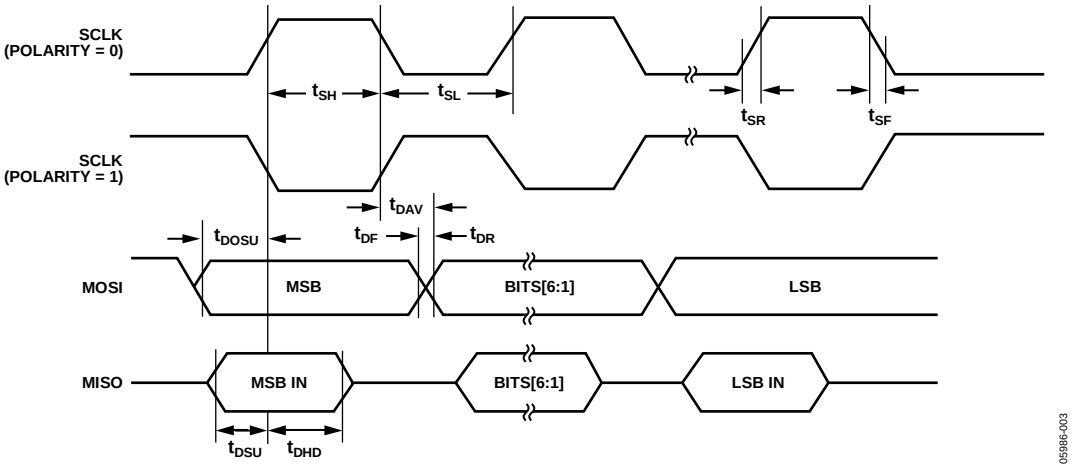

Figure 3. SPI Master Mode Timing (PHASE Mode = 0)

Remap

The ARM exception vectors are all situated at the bottom of the memory array, from Address 0x00000000 to Address 0x00000020.

By default, after a reset, the Flash/EE memory is logically mapped to Address 0x00000000.

It is possible to logically remap the SRAM to Address 0x00000000 by setting Bit 0 of the SYSMAP0 MMR, which is located at 0xFFFF0220. To revert Flash/EE to Address 0x00000000, Bit 0 of SYSMAP0 is cleared.

It may be desirable to remap RAM to Address 0x00000000 to optimize the interrupt latency of the ADuC7032-8L, as code can be run in full 32-bit ARM mode and at the maximum core speed. It should be noted that when an exception occurs, the core defaults to ARM mode.

Remap Operation

When a reset occurs on the ADuC7032-8L, execution starts automatically in the factory-programmed internal configuration code. This so-called kernel is hidden and cannot be accessed by user code.

If the ADuC7032-8L is in normal mode, it executes the power-on configuration routine of the kernel and then jumps to the reset vector, Address 0x00000000, to execute the user reset exception routine.

Because the Flash/EE is mirrored at the bottom of the memory array at reset, the reset routine must always be written in Flash/EE.

Precautions must be taken to execute the remap command from the absolute Flash/EE address, and not from the mirrored, remapped segment of memory, because this segment may be replaced by the SRAM. If a remap operation is executed while operating code from the mirrored location, prefetch/data aborts may occur; or the user may observe abnormal program operation.

This operation is reversible. The Flash/EE can be remapped to Address 0x00000000 by clearing Bit 0 of the SYSMAP0 MMR. Precautions must again be taken to execute the remap function from outside the mirrored area.

Any kind of reset logically remaps the Flash/EE memory to the bottom of the memory array.

SYSMAP0 Register

Name: SYSMAP0
Address: 0xFFFF0220
Default Value: Updated by the kernel
Access: Read/write
Function: This 8-bit register allows user code to remap either RAM or Flash/EE space into the bottom of the ARM memory space, starting at Address 0x00000000.

Table 11. SYSMAP0 MMR Bit Designations

Bit	Description
7 to 1	Reserved. These bits are reserved and should be written as 0 by user code.
0	Remap Bit. Set by the user to remap the SRAM to 0x00000000. Cleared automatically after reset to remap the Flash/EE memory to 0x00000000.

FEE0MOD and FEE1MOD Registers

Name: FEE0MOD and FEE1MOD
Address: 0xFFFF0E04 and 0xFFFF0E84
Default Value (Both Registers): 0x00
Access: Read/write
Function: These registers are written by user code to configure the mode of operation of the Flash/EE memory controllers.

Table 16. FEE0MOD and FEE1MOD MMR Bit Designations

Bit	Description[1]
15 to 7	Not Used. These bits are reserved for future functionality and should be written as 0 by user code.
6 to 5	Flash/EE Security Lock Bits. These bits must be written as [6:5] = 10 to complete the Flash security protect sequence.
4	Flash/EE Controller Command Complete Interrupt Enable. Set to 1 by user code to enable the Flash/EE controller to generate an interrupt upon completion of a Flash/EE command. Cleared to disable the generation of a Flash/EE interrupt upon completion of a Flash/EE command.
3	Flash/EE Erase/Write Enable. Set by user code to enable the Flash/EE erase and write access via FEExCON. Cleared by user code to disable the Flash/EE erase and write access via FEExCON.
2	Reserved. Should be written as 0.
1	Flash/EE Controller Abort Enable. Set to 1 by user code to enable the Flash/EE controller abort functionality.
0	Reserved. Should be written as 0.

[1] x is 0 or 1 to designate Flash/EE Block 0 or Flash/EE Block 1.

FLASH/EE MEMORY SECURITY

The 94 kB of Flash/EE memory available to the user can be read-protected and write-protected using the FFE0HID and FEE1HID registers.

In Block 0, the FEE0HID MMR protects the 30 kB of Flash/EE memory. Bit 0 to Bit 28 of this register protect Page 0 to Page 57 from writing. Each bit protects two pages, that is, 1 kB. Bit 29 to Bit 30 protect Page 58 and Page 59, respectively; that is, each bit write-protects a single page of 512 bytes. The MSB of this register (Bit 31) protects Block 0 from being read through JTAG.

The FEE0PRO register mirrors the bit definitions of the FEE0HID MMR. The FEE0PRO MMR allows user code to lock the protection or security configuration of the Flash/EE memory so that the protection configuration is automatically loaded on subsequent power-on or reset events.

This flexibility allows the user to set and test protection settings temporarily using the FEE0HID MMR and subsequently lock the required protection configuration (using FEE0PRO) when shipping protection systems into the field.

In Block 1 (64 kB), the FEE1HID MMR protects the 64 kB of Flash/EE memory. Bit 0 to Bit 29 of this register protect Page 0 to Page 119 from writing. Each bit protects four pages, that is, 2 kB. Bit 30 protects Page 120 to Page 127; that is, Bit 30 write-protects eight pages of 512 bytes. The MSB of this register (Bit 31) protects Flash/EE Block 1 from being read through JTAG.

As with Block 0, the FEE1PRO register mirrors the bit definitions of the FEE1HID MMR. The FEE1PRO MMR allows user code to lock the protection or security configuration of the Flash/EE memory so that the protection configuration is automatically loaded on subsequent power-on or reset events.

MEMORY MAPPED REGISTERS

The memory mapped register (MMR) space is mapped into the top 4 kB of the MCU memory space and accessed by indirect addressing, load, and store commands through the ARM7 banked registers. An outline of the memory mapped register bank of the ADuC7032-8L is shown in Figure 14.

The MMR space provides an interface between the CPU and all on-chip peripherals. All registers except the ARM7 core registers (described in the ARM Registers section) reside in the MMR area.

As seen in the Complete MMR Listing section (Table 20 to Table 30), the MMR data widths vary from one byte (eight bits) to four bytes (32 bits). The ARM7 core can access any of the MMRs (single-byte or multiple-byte width registers) with a 32-bit read or write access.

The resultant read, for example, is aligned per the little endian format described in the Memory Format section. However, errors result if the ARM7 core tries to access 4-byte (32-bit) MMRs with a 16-bit access. In the case of a (16-bit) write access to a 32-bit MMR, the (upper) 16 most significant bits are written as 0s. More obviously, in the case of a 16-bit read access to a 32-bit MMR, only 16 of the MMR bits can be read.

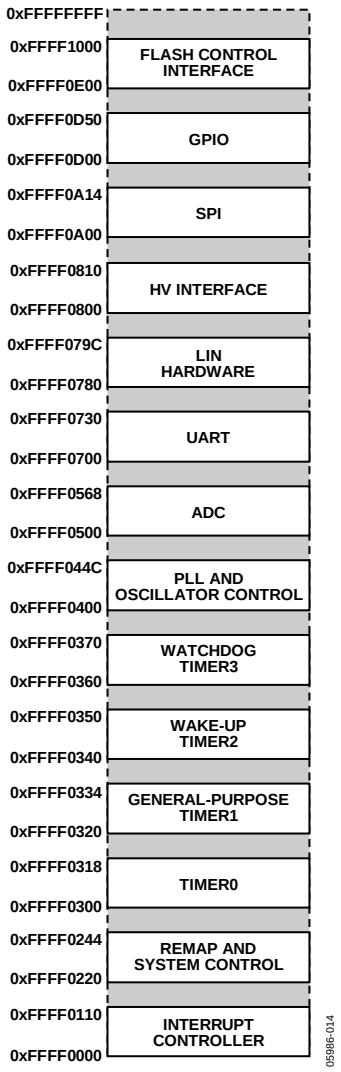

Figure 14. Top Level MMR Map

16-BIT, SIGMA-DELTA ANALOG-TO-DIGITAL CONVERTERS

The ADuC7032-8L incorporates three independent Σ-Δ analog-to-digital converters (ADCs), namely, the current channel ADC (I-ADC), the voltage channel ADC (V-ADC), and the temperature channel ADC (T-ADC). These precision measurement channels integrate on-chip buffering; programmable gain amplifiers; 16-bit, Σ-Δ modulators; and digital filtering and are intended for the precision measurement of current, voltage, and temperature variables in 12 V automotive battery systems.

CURRENT CHANNEL ADC (I-ADC)

This ADC is intended to convert battery current sensed through an external 100 μΩ shunt resistor. On-chip programmable gain means that the I-ADC can be configured to accommodate battery current levels from ±1 A to ±1500 A.

As shown in Figure 15, the I-ADC employs a Σ-Δ conversion technique to realize 16 bits of no missing codes performance. The modulator converts the sampled input signal into a digital pulse train, whose duty cycle contains the digital information.

A modified Sinc3 programmable low-pass filter is then employed to decimate the modulator output data stream to give a valid 16-bit data conversion result at programmable output rates from 4 Hz to 8 kHz in normal mode and 1 Hz to 2 kHz in low power mode.

The I-ADC also incorporates counter, comparator, and accumulator logic. This allows the I-ADC result to generate an interrupt after a predefined number of conversions has elapsed or if the I-ADC result exceeds a programmable threshold value. A fast ADC overrange feature is also supported. Once enabled, a 32-bit accumulator automatically sums the 16-bit I-ADC results.

The time to a first valid (fully settled) result on the current channel is three ADC conversion cycles with chop mode turned off and two ADC conversion cycles with chop mode turned on.

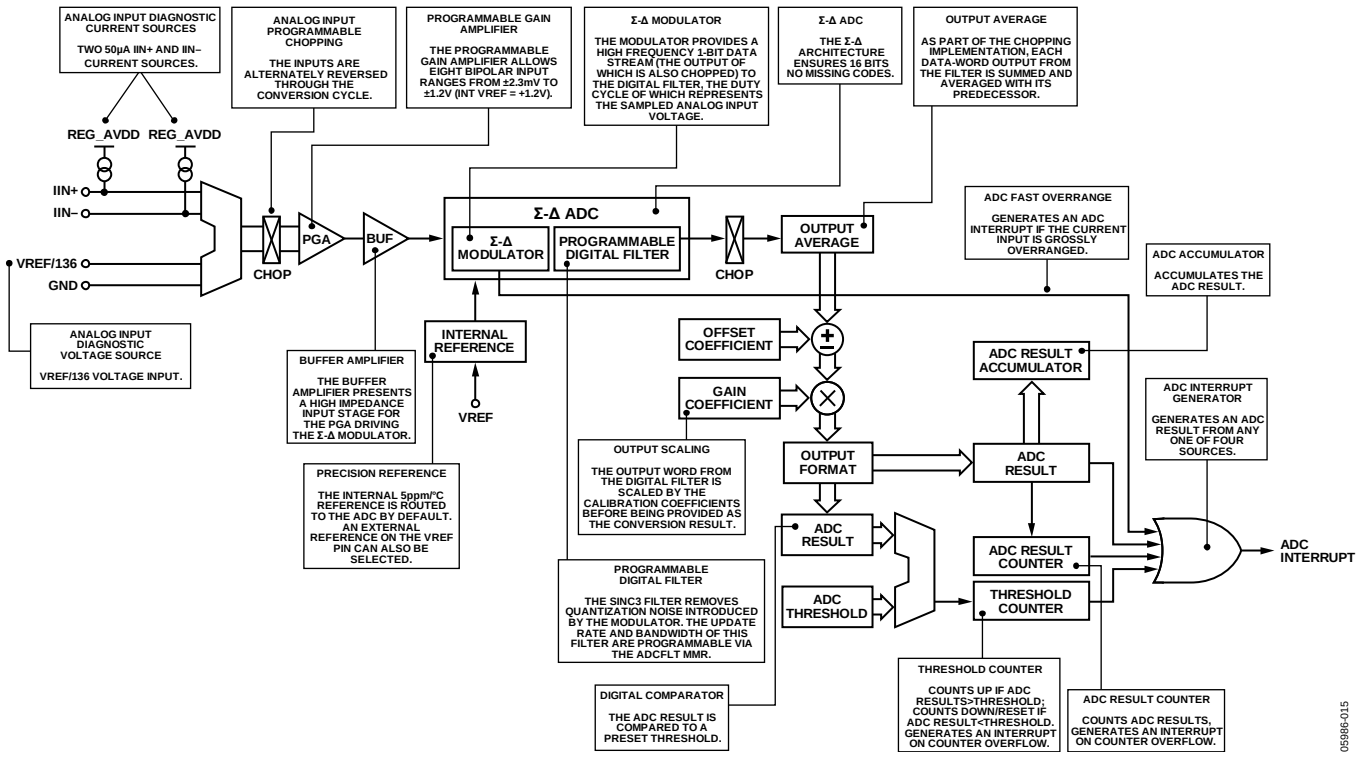

Figure 15. Current ADC, Top Level Overview

ADC Mode Register

Name: ADCMDE
Address: 0xFFFF0508
Default Value: 0x00
Access: Read/write
Function: The ADC Mode MMR is an 8-bit register that configures the mode of operation of the ADC subsystem.

Table 36. ADCMDE MMR Bit Designations

Bit	Description
7	Not Used. This bit is reserved for future functionality and must be written as 0 by user code.
6	20 kΩ Resistor Select. Set to 1 to select the 20 kΩ resistor, as shown in Figure 18. Set to 0 to select the direct path to ground, as shown in Figure 18 (default).
5	Low Power Mode Reference Select. Set to 1 to enable the precision voltage reference in ADC low power mode. This increases current consumption. Set to 0 to enable the low power voltage reference in ADC low power mode (default).
4 to 3	ADC Power Mode Configuration. 00 = ADC normal mode. If enabled, the ADC operates with normal current consumption yielding optimum electrical performance. 01 = ADC low power mode. If enabled, the I-ADC operates with reduced current consumption. This limitation in current consumption is achieved (at the expense of ADC noise performance) by fixing the gain to 128 and using the on-chip low power (131 kHz) oscillator to drive the ADC circuits directly. 10 = ADC low power plus mode. If enabled, the I-ADC again operates with reduced current consumption. In this mode, the gain is fixed to 512, and the current consumed is 200 µA (approximately) more than ADC low power mode, shown previously. The additional current consumed also ensures ADC noise performance is better than that achieved in ADC low power mode. 11 = not defined.
2 to 0	ADC Operation Mode Configuration. 000 = ADC power-down mode. All ADC circuits (including internal reference) are powered-down. 001 = ADC continuous conversion mode. In this mode, any enabled ADC continuously converts. 010 = ADC single conversion mode. In this mode, any enabled ADC performs a single conversion. The ADC enters idle mode once the single-shot conversion is complete. A single conversion takes two to three ADC clock cycles, depending on the chop mode. 011 = ADC idle mode. In this mode, the ADC is fully powered on but is held in r. 100 = ADC self-offset calibration. In this mode, an offset calibration is performed on any enabled ADC using an internally generated 0 V. The calibration is carried out at the user-programmed ADC settings; therefore, as with a normal single ADC conversion, it takes two to three ADC conversion cycles before a fully settled calibration result is ready. The calibration result is automatically written to the ADCxOF MMR of the respective ADC. The ADC returns to idle mode and the calibration- and conversion-ready status bits are set at the end of an offset calibration cycle. 101 = ADC self-gain calibration. In this mode, a gain calibration against an internal reference voltage is performed on all enabled ADCs. A gain calibration is a two-stage process that takes twice the time of an offset calibration. The calibration result is automatically written to the ADCxGN MMR of the respective ADC. The ADC returns to idle mode, and the calibration- and conversion-ready status bits are set at the end of a gain-calibration cycle. An ADC self-gain calibration should only be carried out on the current channel ADC, while preprogrammed, factory calibration coefficients (downloaded automatically from internal Flash) should be used for voltage temperature measurements. If an external NTC is used, an ADC self-calibration should be done on the temperature channel. 110 = ADC system zero-scale calibration. In this mode, a zero-scale calibration is performed on enabled ADC channels against an external zero-scale voltage driven at the ADC input pins. The calibration is carried out at the user programmed ADC settings; therefore, as with a normal single ADC conversion, it takes three ADC conversion cycles before a fully settled calibration result is ready. 111 = ADC system full-scale calibration.

ADC Filter Register

Name: ADCFLT
Address: 0xFFFF0518
Default Value: 0x0007
Access: Read/write
Function: The ADC filter MMR is a 16-bit register that controls the speed and resolution of the on-chip ADCs.
Note: If ADCFLT is modified, the current, voltage, and temperature ADCs are reset. An additional time of 60 μs per enabled ADC is required before the first ADC result is available.

Table 40. ADCFLT MMR Bit Designations

Bit	Description
15	Chop Enable. Set by user to enable system chopping of all active ADCs. When this bit is set, the ADC has very low offset errors and drift, but the ADC output rate is reduced by a factor of 3 if AF = 0 (see Sinc3 decimation factor bits, Bit 6 to Bit 0). If AF ≠ 0, then the ADC output update rate is the same with chop on or off. When chop is enabled, the settling time is two output periods.
14	Running Average. Set by user to enable a running average-by-2 function, reducing ADC noise. This function is automatically enabled when chopping is active. It is an optional feature when chopping is inactive and, if enabled (when chopping is inactive), it does not reduce ADC output rate but increases the settling time by one conversion period. Cleared by user to disable the running average function.
13 to 8	Averaging Factor (AF). The value written to these bits is used to implement a programmable first-order Sinc3 post filter. The averaging factor can further reduce ADC noise at the expense of output rate, as described in the Sinc3 decimation factor bits (Bit 6 to Bit 0).
7	Sinc3 Modify. Set by user to modify the standard Sinc3 frequency response to increase the filter stop-band rejection by 5 dB approximate. This is achieved by inserting a second notch (NOTCH2) at $f_{NOTCH2} = 1.333 \times f_{NOTCH}$ where f_{NOTCH} is the location of the first notch in the response.
6 to 0	Sinc3 Decimation Factor (SF). The value (SF) written in these bits controls the oversampling (decimation factor) of the Sinc3 filter. The output rate from the Sinc3 filter is given by $f_{ADC} = (512{,}000/([SF+1] \times 64))$ Hz when the chop bit (Bit 15) = 0 and AF = 0 (AF = averaging factor).[1, 2]

[1] This is valid for all SF values ≤ 125. For SF = 126, f_{ADC} is forced to 60 Hz. For SF = 127, f_{ADC} is forced to 50 Hz.
[2] For information on calculating the f_{ADC} for SF (other than 126 and 127) and AF values, see Table 41.

Note that due to limitations on the digital filter internal datapath, there are some limitations on the combinations of SF (Sinc3 decimation factor) and AF (averaging factor) that can be used to generate a required ADC output rate. This restriction limits the minimum ADC update to 4 Hz in normal power mode or to 1 Hz in low power mode. If all three ADCs are enabled, then the minimum value of SF written by user code must be 1.

In low power mode and low power plus mode, the ADC is driven directly by the low power oscillator, or 131 kHz, and not 512 kHz. All f_{ADC} calculations should be divided by 4 (approximate).

For optimal ADC performance, SF should be increased before AF is used.

Current Channel ADC Data Register

Name: ADC0DAT
Address: 0xFFFF0520
Default Value: 0x0000
Access: Read only
Function: This current channel ADC data MMR holds the 16-bit conversion result from the I-ADC. The ADC does not update this MMR if the ADC conversion result-ready bit (ADCSTA[0]) is set. A read of this MMR by the MCU clears all asserted ready flags (ADCSTA[2:0]).

Voltage Channel Data Register

Name: ADC1DAT
Address: 0xFFFF0524
Default Value: 0x0000
Access: Read only
Function: This V-ADC data MMR holds the 16-bit conversion result from the V-ADC. The ADC does not update this MMR if the voltage conversion result-ready bit (ADCSTA[1]) is set. If I-ADC is not active, a read of this MMR by the MCU clears all asserted ready flags (ADCSTA[2:1]).

Temperature Channel ADC Data Register

Name: ADC2DAT
Address: 0xFFFF0528
Default Value: 0x0000
Access: Read only
Function: This T-ADC data MMR holds the 16-bit conversion result from the T-ADC. The ADC does not update this MMR if the temperature conversion result-ready bit (ADCSTA[2]) is set. A read of this MMR clears ADCSTA[2].

ADC FIFO Register

Name: ADCFIFO
Address: 0xFFFF052C
Default Value: 0x0000
Access: Read only
Function: This 32-bit, read-only register returns the value of the I-ADC and V-ADC conversion result held in the FIFO location currently pointed to by the FIFO read pointer. The low 16 bits [15:0] of this 32-bit word are the I-ADC result, and the high 16 bits [31:16] are the V-ADC result. The FIFO function is enabled via the ADCCFG[1] bit, and three flags available in the ADCSTA register allow user code to monitor and read the FIFO contents.

Current Channel ADC Offset Calibration Register

Name: ADC0OF
Address: 0xFFFF0530
Default Value: Part-specific, factory programmed
Access: Read/write
Function: This ADC offset MMR holds a 16-bit offset calibration coefficient for the I-ADC. The register is configured at power-on with a factory default value. However, this register is automatically overwritten if an offset calibration of the I-ADC is initiated by the user via bits in the ADCMDE MMR. User code can write to this calibration register only if the ADC is in idle mode. An ADC must be enabled and in idle mode before being written to any offset or gain register. The ADC must be in idle mode for at least 23 μs.

Voltage Channel ADC Offset Calibration Register

Name: ADC1OF
Address: 0xFFFF0534
Default Value: Part-specific, factory programmed
Access: Read/write
Function: This V-ADC offset MMR holds a 16-bit offset calibration coefficient for the voltage channel. The register is configured at power-on with a factory default value. However, this register is automatically overwritten if an offset calibration of the voltage channel is initiated by the user via bits in the ADCMDE MMR. User code can write to this calibration register only if the ADC is in idle mode. An ADC must be enabled and in idle mode before being written to any offset or gain register. The ADC must be in idle mode for at least 23 μs.

Temperature Channel ADC Offset Calibration Register

Name: ADC2OF
Address: 0xFFFF0538
Default Value: Part-specific, factory programmed
Access: Read/write
Function: This T-ADC offset MMR holds a 16-bit offset calibration coefficient for the temperature channel. The register is configured at power-on with a factory default value. However, this register is automatically overwritten if an offset calibration of the temperature channel is initiated by the user via bits in the ADCMDE MMR. User code can write to this calibration register only if the ADC is in idle mode. An ADC must be enabled and in idle mode before writing to any offset or gain register. The ADC must be in idle mode for at least 23 μs.

ADuC7032-8L SYSTEM CLOCKS

The ADuC7032-8L integrates a highly flexible clocking system that can be clocked from one of three sources: an integrated on-chip precision oscillator, an integrated on-chip low power oscillator, or an external watch crystal. These three options are shown in Figure 27.

Each of the internal oscillators is divided by four to generate a clock frequency of 32.768 kHz. The PLL locks onto a multiple (625) of 32.768 kHz, supplied by either of the internal oscillators or the external crystal, providing a stable 20.48 MHz clock for the system. The core can operate at this frequency or at binary submultiples of it, allowing power saving if peak performance is not required.

By default, the PLL is driven by the low power oscillator, which generates a 20.48 MHz clock source. The ARM7TDMI core is driven by a CD divided clock derived from the output of the PLL. By default, the CD divider is configured to divide the PLL output by two, which generates a core clock of 10.24 MHz. The divide factor can be modified to generate a binary weighted divider factor from 1 to 128, which can be altered dynamically by user code.

The ADC is driven by the output of the PLL, divided to give an ADC clock source of 512 kHz. In low-power mode, the ADC clock source is switched from the standard 512 kHz to the low power 131 kHz oscillator.

It should also be noted that the low power oscillator drives both the watchdog and core wake-up timers through a divide-by-4 circuit. A detailed block diagram of the ADuC7032-8L clocking system is shown in Figure 27.

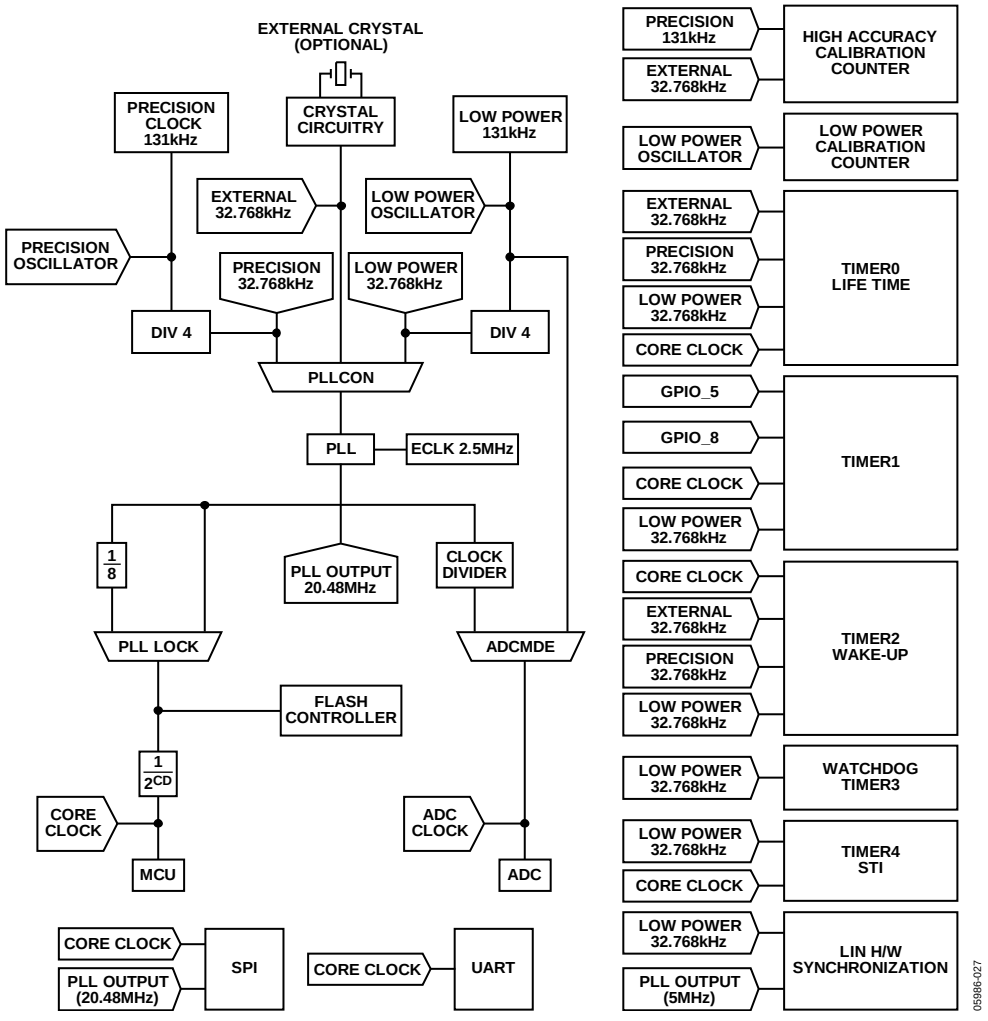

Figure 27. ADuC7032-8L System Clock Generation

IRQ

The IRQ is the exception signal to enter the IRQ mode of the processor. It is used to service general-purpose interrupt handling of internal and external events.

There are four 32-bit registers dedicated to IRQ.

IRQSIG

Reflects the status of the different IRQ sources. If a peripheral generates an IRQ signal, the corresponding bit in the IRQSIG is set; otherwise, it is cleared. The IRQSIG bits are cleared when the interrupt in the particular peripheral is cleared. All IRQ sources can be masked in the IRQEN MMR. IRQSIG is read only and can be used to poll interrupt sources.

IRQEN

Provides the value of the current enable mask. When the bit is set to 1, the source request is enabled to create an IRQ exception. When the bit is set to 0, the source request is disabled or masked and does not create an IRQ exception.

IRQCLR

A write-only register that allows clearing the IRQEN register to mask an interrupt source. Each bit set to 1 clears the corresponding bit in the IRQEN register without affecting the remaining bits. The pair of registers, IRQEN and IRQCLR, allow independent manipulation of the enable mask without requiring an automatic read-modify-write.

IRQSTA

A read-only register that provides the current enabled IRQ source status (effectively a Logic AND of the IRQSIG and IRQEN bits). When the bit is set to 1, that source generates an active IRQ request to the ARM7TDMI core. There is no priority encoder or interrupt vector generation. This function is implemented in software in a common interrupt handler routine. All 32 bits are logically OR'ed to create a single IRQ signal to the ARM7TDMI core.

FIQ

The fast interrupt request (FIQ) is the exception signal to enter the FIQ mode of the processor. It is provided to service data transfer or communication channel tasks with low latency. The FIQ interface is identical to the IRQ interface, providing the second level interrupt (highest priority). Four 32-bit registers are dedicated to FIQ, FIQSIG, FIQEN, FIQCLR, and FIQSTA.

Bit 31 to Bit 1 of FIQSTA are logically OR'ed to create the FIQ signal to the core and to Bit 0 of both the FIQ and IRQ registers (FIQ source).

The logic for FIQEN and FIQCLR does not allow an interrupt source to be enabled in both IRQ and FIQ masks. A bit set to 1 in FIQEN, as a side effect, clears the same bit in IRQEN.

A bit set to 1 in IRQEN, as a side effect, clears the same bit in FIQEN. An interrupt source can be disabled in both IRQEN and FIQEN masks.

Programmed Interrupts

Because the programmed interrupts are nonmaskable, they are controlled by another register, SWICFG, which writes into both IRQSTA and IRQSIG registers and/or FIQSTA and FIQSIG registers at the same time.

The 32-bit register dedicated to software interrupt is SWICFG, described in Table 52. This MMR allows the control of programmed source interrupt.

Table 52. SWICFG MMR Bit Designations

Bit	Description
31 to 3	Reserved.
2	Programmed Interrupt FIQ. Setting/clearing this bit corresponds to setting/clearing Bit 1 of FIQSTA and FIQSIG.
1	Programmed Interrupt IRQ. Setting/clearing this bit corresponds to setting/clearing Bit 1 of IRQSTA and IRQSIG.
0	Reserved.

Note that to be detected by the interrupt controller and to be detected by the user in the IRQSTA and FIQSTA registers, any interrupt signal must be active for at least the minimum interrupt latency time.

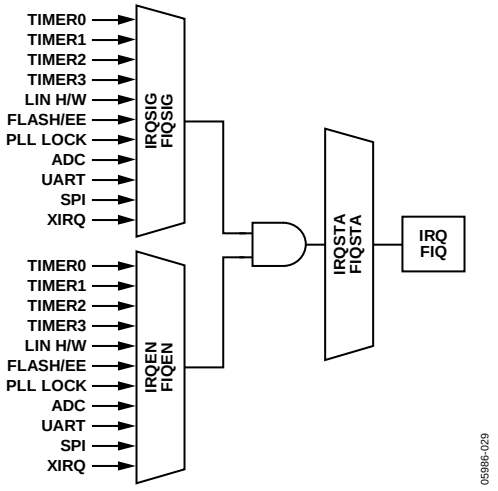

Figure 29. Interrupt Structure

Timer1 Capture Register

Name: T1CAP
Address: 0xFFFF0330
Default Value: 0x00000000
Access: Read/write
Function: This 32-bit register holds the 32-bit value captured by an enabled IRQ event.

Timer1 Control Register

Name: T1CON
Address: 0xFFFF0328
Default Value: 0x01000000
Access: Read/write
Function: This 32-bit MMR configures the mode of operation of Timer1.

Table 55. T1CON MMR Bit Designations

Bit	Description
31 to 24	Timer1 8-Bit Postscalar. By writing to these 8 bits, a value is loaded into the postscalar. Writing 0 to these bits is interpreted as a 1. By reading these 8 bits, the current value of the counter is loaded.
23	Timer1 Enable Postscalar. Set to enable Timer1 postscalar. If enabled, an interrupt is generated after T1CON[31:24] periods, as defined by T1LD. Cleared to disable Timer1 postscalar.
22 to 20	Reserved. This bit is reserved and should be written as 0 by user code.
19	Postscalar Compare Flag. Set if the number of Timer1 overflows is equal to the number written to the postscalar.
18	Timer1 Interrupt Source. Set to select interrupt generation from postscalar counter. Cleared to select interrupt generation direct from Timer1.
17	Event Select Bit. Set by user to enable time capture of an event. Cleared by user to disable time capture of an event.
16 to 12	Event Select Range, 0 to 31. The events are described in Table 53.
11 to 9	Clock Select. 000 = core clock (default). 001 = low power 32.768 kHz oscillator. 010 = GPIO_8. 011 = GPIO_5.
8	Count Up. Set by user for Timer1 to count up. Cleared by user for Timer1 to count down (default).
7	Timer1 Enable Bit. Set by user to enable Timer1. Cleared by user to disable Timer1 (default).
6	Timer1 Mode. Set by user to operate in periodic mode. Cleared by user to operate in free-running mode (default).
5 to 4	Format. 00 = binary (default). 01 = reserved. 10 = hours:minutes:seconds:hundredths—23 hours to 0 hour. 11 = hours:minutes:seconds:hundredths—255 hours to 0 hour.
3 to 0	Prescalar. 0000 = source clock/1 (default). 0100 = source clock/16. 1000 = source clock/256. 1111 = source clock/32,768.

TIMER2—WAKE-UP TIMER

Timer2 is a 32-bit wake-up timer, count-down or count-up, with a programmable prescalar. The prescalar is clocked directly from one of four clock sources, namely, the core clock (default selection), the low power 32.768 kHz oscillator, the external 32.768 kHz watch crystal, or the precision 32.768 kHz oscillator. The selected clock source can be scaled by a factor of 1, 16, 256, or 32,768. The wake-up timer continues to run when the core clock is disabled. This gives a minimum resolution of 48.83 ns when operating at CD zero; the core is operating at 20.48 MHz, and with a prescalar of 1. Capture of the current timer value is enabled if the Timer2 interrupt is enabled via IRQEN[4].

The counter can be formatted as plain 32-bit value or as hours:minutes:seconds:hundreths.

Timer2 reloads the value from T2LD either when TIMER2 overflows, or immediately when T2CLRI is written.

The Timer2 interface consists of four MMRs.

- T2LD and T2VAL: 32-bit registers that hold 32-bit unsigned integers. T2VAL is read only.
- T2CLRI: 8-bit register. Writing any value to this register clears the Timer2 interrupt.
- T2CON: configuration MMR described in Table 56.

Timer2 Load Register

Name: T2LD
Address: 0xFFFF0340
Default Value: 0x00000000
Access: Read/write
Function: This 32-bit register holds the 32-bit value loaded into the counter.

Timer2 Clear Register

Name: T2CLRI
Address: 0xFFFF034C
Access: Write only
Function: This 8-bit, write-only MMR is written (with any value) by user code to clear the interrupt.

Timer2 Value Register

Name: T2VAL
Address: 0xFFFF0344
Default Value: 0xFFFFFFFF
Access: Read only
Function: This 32-bit register holds the current value of Timer2.

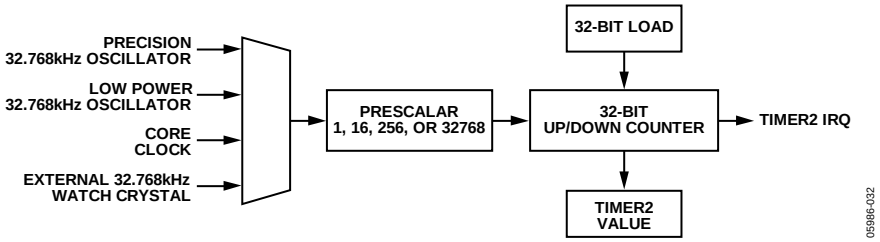

Figure 34. Timer2 Block Diagram

Timer2 Control Register

Name: T2CON
Address: 0xFFFF0348
Default Value: 0x0000
Access: Read/write
Function: This 32-bit MMR configures the mode of operation of Timer2.

Table 56. T2CON MMR Bit Designations

Bit	Description
31 to 11	Reserved.
10 to 9	Clock Source Select. 00 = core clock (default). 01 = low power 32.768 kHz oscillator. 10 = external 32.768 kHz watch crystal. 11 = precision 32.768 kHz oscillator.
8	Count Up. Set by user for Timer2 to count up. Cleared by user for Timer2 to count down (default).
7	Timer2 Enable Bit. Set by user to enable Timer2. Cleared by user to disable Timer2 (default).
6	Timer2 Mode. Set by user to operate in periodic mode. Cleared by user to operate in free running mode (default).
5 to 4	Format. 00 = binary (default). 01 = reserved. 10 = hours:minutes:seconds:hundredths—23 hours to 0 hour. 11 = hours:minutes:seconds:hundredths—255 hours to 0 hour.
3 to 0	Prescalar. 0000 = source clock/1 (default). 0100 = source clock/16. 1000 = source clock/256 (should be used in conjunction with Timer2 Format 10 and Timer2 Format 11). 1111 = source clock/32,768.

GENERAL-PURPOSE I/O

The ADuC7032-8L features nine general-purpose bidirectional I/O pins (GPIO). In general, many of the GPIO pins have multiple functions that can be configured by user code. By default, the GPIO pins are configured in GPIO mode. All GPIO pins have an internal pull-up resistor, their sink capability is 0.8 mA, and they can source 0.1 mA.

The nine GPIOs are grouped into three ports: Port0, Port1, and Port2. Port0 is five bits wide. Port1 and Port2 are both two bits wide. The GPIO assignment within each port is shown in Table 58.

A typical GPIO structure is shown in Figure 36.

External interrupts are present on GPIO_0, GPIO_5, GPIO_7, and GPIO_8. These interrupts are level triggered and are active high. These interrupts are not latched; therefore, the interrupts source must be present until either IRQSTA or FIQSTA is interrogated.

The interrupt source must be active for at least one CD-divided core clock to guarantee recognition.

All port pins are configured and controlled by four sets (one set for each port) of four port-specific MMRs.

- GPxCON: Portx control register
- GPxDAT: Portx configuration and data register
- GPxSET: Portx data set
- GPxCLR: Portx data clear

where x corresponds to the port number (0, 1, or 2).

During normal operation, user code can control the function and state of the external GPIO pins via these general-purpose registers. All GPIO pins retain their external high or low during power-down (POWCON) mode.

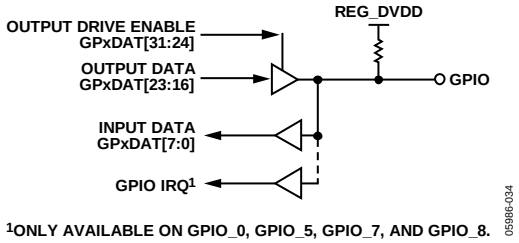

Figure 36. ADuC7032-8L GPIO

UART Interrupt Enable Register 0

Name: COMIEN0
Address: 0xFFFF0704
Default Value: 0x00
Access: Read/write
Function: This 8-bit register enables and disables the individual UART interrupt sources.

Table 84. COMIEN0 MMR Bit Designations

Bit	Name	Description
7 to 3		Not Used. 0 by default.
2	ELSI	COMRX Status Interrupt Enable Bit. Set by user to enable generation of an interrupt if any of COMSTA0[3:1] are set. Cleared by user.
1	ETBEI	Enable Transmit Buffer Empty Interrupt. Set by user to enable interrupt when buffer is empty during a transmission, that is, when COMSTA[5] is set. Cleared by user.
0	ERBFI	Enable Receive Buffer Full Interrupt. Set by user to enable interrupt when buffer is full during a reception. Cleared by user.

UART Interrupt Identification Register 0

Name: COMIID0
Address: 0xFFFF0708
Default Value: 0x01
Access: Read only
Function: This 8-bit register reflects the source of the UART interrupt.

Table 85. COMIID0 MMR Bit Designations

Bits[2:1] Status Bits	Bit 0 NINT	Priority	Definition	Clearing Operation
00	1	N/A	No Interrupt.	N/A
11	0	1	Receive Line Status Interrupt.	Read COMSTA0.
10	0	2	Receive Buffer Full Interrupt.	Read COMRX.
01	0	3	Transmit Buffer Empty Interrupt.	Write data to COMTX or read COMIID0.
00	0	4	Modem Status Interrupt.	Read COMSTA1.

UART Fractional Divider Register

Name: COMDIV2
Address: 0xFFFF072C
Default Value: 0x0000
Access: Read/write
Function: This 16-bit register controls the operation of the ADuC7032-8L fractional divider.

Table 86. COMDIV2 MMR Bit Designations

Bit	Name	Description
15	FBEN	Fractional Baud Rate Generator Enable Bit. Set by user to enable the fractional baud rate generator. Cleared by user to generate baud rate using the standard 450 UART baud rate generator.
14 to 13		Reserved.
12 to 11	FBM[1:0]	M. If FBM = 0, M = 4.
10 to 0	FBN[10:0]	N.

SERIAL PERIPHERAL INTERFACE

The ADuC7032-8L features a complete hardware serial peripheral interface (SPI) on-chip. SPI is an industry-standard synchronous serial interface that allows eight bits of data to be synchronously transmitted and received simultaneously, that is, full duplex.

The SPI is operational only with core clock divider bits (POWCON[2:0] = 0 or 1).

The SPI port can be configured for master or slave operation and consists of four pins that are multiplexed with four GPIOs. The four SPI pins are MISO, MOSI, SCLK, and $\overline{SS}$. The pins to which these signals are connected are shown in Table 87.

Table 87. SPI Output Pins

Pin	Signal	Description
GPIO_0 (GPIO MODE 1)	$\overline{SS}$	Chip Select
GPIO_1 (GPIO MODE 1)	SCLK	Serial Clock
GPIO_2 (GPIO MODE 1)	MISO	Master In, Slave Out
GPIO_3 (GPIO MODE 1)	MOSI	Master Out, Slave In

MISO (MASTER IN, SLAVE OUT DATA I/O PIN)

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

MOSI (MASTER OUT, SLAVE IN PIN)

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

SCLK (SERIAL CLOCK I/O PIN)

The master serial clock (SCLK) is used to synchronize the data being transmitted and received through the MOSI SCLK period. Therefore, a byte is transmitted/received after eight SCLK periods. The SCLK pin is configured as an output in master mode and as an input in slave mode.

In master mode, polarity and phase of the clock are controlled by the SPICON register, and the bit rate is defined in the SPIDIV register as follows:

$$f_{SERIALCLOCK} = \frac{20.48\ \text{MHz}}{2 \times (1 + SPIDIV)}$$

The maximum speed of the SPI clock is dependent on the clock divider bits and is summarized in Table 88.

Table 88. SPI Speed vs. Clock Divider Bits in Master Mode

CD Bits	0	1
SPIDIV	0x05	0x0B
Maximum SCLK	1.667 MHz	0.833 MHz

In slave mode, the SPICON register must be configured with the phase and polarity of the expected input clock. The slave accepts data from an external master up to 5.12 Mb at CD = 0. The formula to determine the maximum speed follows:

$$f_{SERIALCLOCK} = \frac{f_{HCLK}}{4}$$

In both master and slave modes, data is transmitted on one edge of the SCL signal and sampled on the other. Therefore, it is important that the polarity and phase be configured in the same way for the master and slave devices.

CHIP SELECT ($\overline{SS}$) INPUT PIN

In SPI slave mode, a transfer is initiated by the assertion of $\overline{SS}$, which is an active low input signal. The SPI port then transmits and receives 8-bit data until the transfer is concluded by deassertion of $\overline{SS}$. In slave mode, $\overline{SS}$ is always an input.

SPI REGISTERS DEFINITIONS

The following MMR registers are used to control the SPI interface:

- SPICON: 16-bit control register
- SPISTA: 8-bit read-only status register
- SPIDIV: 8-bit serial clock divider register
- SPIRX: 8-bit read-only receive register
- SPITX: 8-bit write-only transmit register

LIN Hardware Synchronization Control Register 0

Name: LHSCON0
Address: 0xFFFF0784
Default Value: 0x0000
Access: Read/write
Function: This LHS control register is a 16-bit register that, in conjunction with the LHSCON1 register, is used to configure the LIN mode of operation.

Table 92. LHSCON0 MMR Bit Designations

Bit	Description
15 to 12	Reserved. These bits are reserved for future use and should be written as 0 by user software.
11	Break Timer Compare Interrupt Disable. Set to 1 to disable the break timer compare interrupt. Cleared to 0 to enable the break timer compare interrupt.
10	Break Timer Error Interrupt Disable. Set to 1 to disable the break timer error interrupt. Cleared to 0 to enable the break timer error interrupt.
9	LIN Transceiver, Standalone Test Mode. Cleared to 0 by user code to operate the LIN in normal mode, driven directly from the on-chip UART. Set to 1 by user code to enable external GPIO_7 and GPIO_8 pins to drive the LIN transceiver TxD and RxD, respectively, independent of the UART. The functions of GPIO_7 and GPIO_8 should first be configured by user code via GPIO Function Select Bit 0 and GPIO Function Select Bit 4 in the GP2CON register.
8	Gate UART Bit. Set to 1 by user code to disable the internal UART RxD (receive data) by gating it high until both the break field and subsequent LIN sync byte have been detected. This ensures the UART does not receive any spurious serial data during break or sync field periods that need to be flushed out of the UART before valid data fields can be received. Set to 0 by user code to enable the internal UART RxD (receive data) after the break field and subsequent LIN sync byte have been detected so that the UART can receive the subsequent LIN data fields.
7	Sync Timer Stop Edge Type Bit. Cleared to 0 by user code to stop the sync timer on the falling edge count configured via the LHSCON1[7:4] register. Set to 1 by user code to stop the sync timer on the rising edge count configured via the LHSCON1[7:4] register.
6 to 5	Reserved. These bits are reserved for future use and should be written as 0 by user software.
4	Enable Stop Interrupt. Cleared to 0 by user code to disable interrupts when a stop condition occurs. Set to 1 by user code to generate an interrupt when a stop condition occurs.
3	Enable Start Interrupt. Cleared to 0 by user code to disable interrupts when a start condition occurs. Set to 1 by user code to generate an interrupt when a start condition occurs.
2	LIN Sync Enable Bit. Cleared to 0 by user code to disable LHS functionality. Set to 1 by user code to enable LHS functionality.
1	Edge Counter Clear Bit. Cleared to 0 by user code to enable the rising or falling edge counters to function normally. Set to 1 by user code to clear the internal edge counters in the LHS peripheral. This bit does not reset to 0 automatically and requires user code to write 0 to re-enable the edge counters.
0	LHS Reset Bit. Cleared to 0 automatically after 15 µs delay. Set to 1 by user code to reset all LHS logic to default conditions.

Example LIN Hardware Synchronization Routine

Consider the following C-source code LIN initialization routine:

```
void LIN_INIT(void )
{
     char HVstatus;
     GP2CON = 0x110000;             // Enable LHS on GPIO pins

     LHSCON0 = 0x1;                 // Reset LHS interface

     do{
         HVDAT = 0x02;              // Enable normal LIN TX mode
         HVCON = 0x08;              // Write to Config0
              do{
                 HVstatus = HVCON;
                }
              while(HVstatus & 0x1); // Wait until command is finished
         }
     while (!(HVstatus & 0x4));     // Transmit command is correct

     while((LHSSTA & 0x20) == 0 )
     {                              // Wait until the LHS hardware is reset
     }

     LHSCON1 = 0x062;               // Sets stop edge as the fifth falling edge
                                    // and the start edge as the first falling
                                    // edge in the sync byte
     LHSCON0 = 0x0114;              // Gates UART RX line, ensure no interference
                                    // from the LIN into the UART
                                    // Selects the stop condition as a falling edge
                                    // Enables generation of an interrupt on the
                                    // stop condition
                                    // Enables the interface
     LHSVAL1 = 0x03F;               // Set number of 131 kHz periods to generate
                                    // a break interrupt 0x3F / 131 kHz ~ 480 µs
                                    // which is just over 9.5 tbits
}
```

Using this configuration, LHSVAL1 begins to count on the first falling edge received on the LIN bus. If LHSVAL1 exceeds the value written to LHSVAL1, in this case 0x3F, a break compare interrupt is generated.

On the next falling edge, LHSVAL0 begins counting. LHSVAL0 monitors the number of falling edges and compares this number to the value written to LHSCON1. In this example, the edge to monitor is the sixth falling edge of the LIN frame, or the fifth falling edge of the sync byte.

After the number of falling edges is received, a stop condition interrupt is generated. It is at this point that the UART is configured to receive the protected identifier.

The UART must not be ungated, through LHSCON0[8], before the LIN bus returns high. If this occurs, UART communication errors may occur. This process is shown in detail in Figure 46. Example code follows.

ADuC7032-8L ON-CHIP DIAGNOSTICS

The ADuC7032-8L integrates multiple diagnostic support circuits on-chip. These circuits allow the device to test core digital functionality, and analog front-end and high voltage I/O ports in-circuit.

ADC DIAGNOSTICS

Internal Test Voltage

The current channel can be configured to convert on an internal 8.3 mV test voltage. On any gain range, the result should be within ±2% of the expected result.

Internal Short Mode

The current and voltage input channels can also be shorted internally. Converting on the internal short allows an assessment of the internal ADC noise.

Internal Current Sources

Internal current sources can also be enabled on both current and temperature channels. These current sources can be used to determine external short-circuit or open-circuit conditions in both external shunt or temperature sensor configurations.

HIGH VOLTAGE I/O DIAGNOSTICS

High Voltage I/O Readback

All high voltage I/O pins are supported with readback capability, which allows the detection of external short conditions.

High Voltage Current Detection

All high voltage I/O pins also have a high current detection capability, allowing high-side connections to VBAT to be detected and controlled.