Integrated Precision Battery Sensor For Automotive

Preliminary Technical Data

ADuC7030/ADuC7033

FEATURES

High precision ADCs
- Dual channel, simultaneous sampling, 16-bit ΣΔ ADCs
- Programmable ADC throughput from 1 Hz to 8 kHz
- On-chip 5ppm/°C voltage reference
- Current channel
  - Fully differential, buffered Input
  - Programmable gain from 1 to 512
  - ADC input range: –200 mV to +300 mV
  - Digital comparators, with current accumulator feature
- Voltage channel
  - Buffered, on-chip attenuator for 12V battery inputs
- Temperature channel
  - External and on-chip temperature sensor options

Microcontroller
- ARM7TDMI core, 16-/32-bit RISC architecture
- 20.48 MHz PLL with programmable divider
- PLL Input Source
  - On-chip precision oscillator
  - On-chip low-power oscillator
  - External (32.768 kHz) watch crystal
- JTAG port supports code download and debug

Memory
- 32 kbytes Flash/EE memory, 4 kbytes SRAM (ADuC7030)
- 96 kbytes Flash/EE memory, 6 kbytes SRAM (ADuC7033)
- 10 kcycles Flash/EE endurance, 20 years Flash/EE retention
- In-circuit download via JTAG and LIN

On-chip peripherals
- LIN 2.0 (slave) compatible support via UART with hardware synchronization
- Flexible wake-up I/O pin, master/slave SPI serial I/O
- 9-pin GPIO port, 3 X general purpose timers
- Wake-up and watchdog timers
- Power supply monitor, on-chip power-on-reset

Power
- Operates directly from 12 V battery supply
- Current consumption
  - Normal mode 10 mA at 10 MHz
  - Low power monitor mode

Packages and temperature range
- 48 Pin LFCSP 7X7 mm or 48 Pin LQFP 7X7 mm body package
- Fully specified for –40°C to +115°C operation

APPLICATIONS

Battery sensing/management for automotive systems

FUNCTIONAL BLOCK DIAGRAM

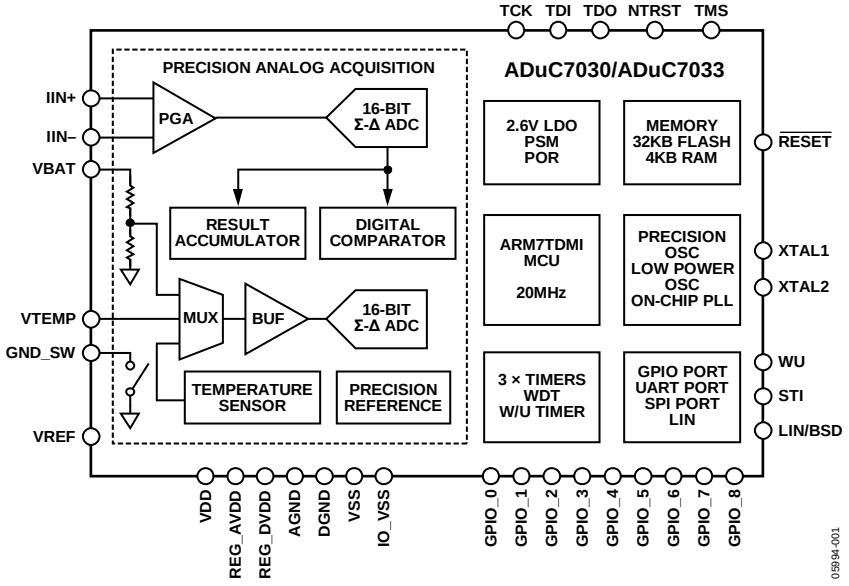


Figure 1.

Rev. PrE

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113

TABLE OF CONTENTS

SPECIFICATIONS

ELECTRICAL SPECIFICATIONS

V_{DD} = 3.5 V to 18 V, V_{REF} = 1.2 V internal reference, f_{CORE} = 10.24 MHz driven from external 32.768 kHz watch crystal or on-chip precision oscillator, all specifications T_A = −40°C to 115°C, unless otherwise noted.

Table 1. ADuC7030/ADuC7033 SPECIFICATIONS

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
ADC SPECIFICATIONS					
Conversion Rate[1]	Chop Off, ADC Normal Operating Mode	4		8000	Hz
	Chop On, ADC Normal Operating Mode	4		2600	Hz
	Chop On, ADC Low Power Mode	1		650	Hz
Current Channel					
No Missing Codes[1]	Valid for all ADC update rates and ADC modes	16			Bits
Integral Nonlinearity[1, 2]			±10	±60	PPM of FSR
Offset Error[2, 3, 4, 5]	Chop off, 1 LSB = (36.6/gain) μV	−10	±3	+10	LSB
Offset Error[1, 3, 6]	Chop on	−2	±0.5	+2	μV
Offset Error Drift[6]	Chop off, valid for ADC gains of 4 to 64, normal mode		0.03		LSB/°C
Offset Error Drift[6]	Chop off, valid for ADC gains of 128 to 512, normal mode		30		nV/°C
Offset Error Drift[6]	Chop on		10		nV/°C
Total Gain Error[1, 3, 7, 8, 9, 10]	Normal mode	−0.5	±0.1	+0.5	%
Total Gain Error[1, 3, 7, 9, 11]	Low power mode	−4	±0.2	+4	%
Total Gain Error[1, 3, 7, 9, 12]	Low power-plus mode, using precision V_{REF}	−1	±0.2	+1	%
Gain Drift			3		ppm/°C
PGA Gain Mismatch Error			±0.1		%
Output Noise[1, 13]					
	4 Hz update rate, gain = 512, chop enabled		60	90	nV rms
	10 Hz update rate, gain = 512, chop enabled		100	150	nV rms
	1 kHz update rate, gain = 512		0.6	0.9	μV rms
	1 kHz update rate, gain = 32		0.8	1.2	μV rms
	1 kHz update rate, gain = 4		2.0	2.8	μV rms
	8 kHz update rate, gain = 32		2.5	3.5	μV rms
	8 kHz update rate, gain = 4		14	21	μV rms
	ADC low power mode, FADC = 10 Hz, gain = 128		1.25	1.9	μV rms
	ADC low power mode, FADC = 1 Hz, gain = 128		0.35	0.5	μV rms
	ADC low power-plus mode, FADC = 1 Hz, gain = 512		0.1	0.15	μV rms
Voltage Channel [14]					
No Missing Codes [1]	Valid at all ADC update rates	16			Bits
Integral Nonlinearity [1]			±10	±60	ppm of FSR
Offset Error[3, 5]	Chop off , 1 LSB 16 = 439.5 μV	−10	±1	+10	LSB
Offset Error[1, 3]	Chop on		0.3	1	LSB
Offset Error Drift	Chop off		0.03		LSB/°C
Total Gain Error[1, 3, 7, 15, 10]	Includes resistor mismatch	−0.25	±0.06	+0.25	%
Total Gain Error[1, 3, 7, 15, 10]	Temperature range = −25°C to +65°C	−0.15	±0.03	+0.15	%
Gain Drift	Includes resistor mismatch drift		3		ppm/°C
Output Noise[1, 16]					
	4 Hz update rate		60	90	μV rms
	10 Hz update rate		60	90	μV rms
	1 kHz update rate		180	270	μV rms
	8 kHz Update Rate		1600	2400	μV rms

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
Temperature Channel					
No Missing Codes[1]	Valid at all ADC Update Rates	16			Bits
Integral Nonlinearity[1]			±10	±60	ppm of FSR
Offset Error[3, 5, 17, 18]	Chop Off , 1 LSB16=19.84uV	−10	±3	+10	LSB
Offset Error [1, 3]	Chop On	−5	1	5	LSB
Offset Error Drift	Chop Off		0.03		LSB/°C
Total Gain Error[1, 3, 19]		−0.2	±0.06	+0.2	%
Gain Drift			3		ppm/°C
Output Noise[1]	1 kHz update rate		7.5	11.25	µV rms
ADC SPECIFICATIONS ANALOG INPUT	Internal V_{REF} = 1.2 V				
Current Channel					
Absolute Input Voltage Range	Applies to both IIN+ and IIN−	−200		+300	mV
Input Voltage Range[20,21]	Gain = 1[22]		±1.2		V
	Gain = 2[22]		±600		mV
	Gain = 4[22]		±300		mV
	Gain = 8		±150		mV
	Gain = 16		±75		mV
	Gain = 32		±37.5		mV
	Gain = 64		±18.75		mV
	Gain = 128		±9.375		mV
	Gain = 256		±4.68		mV
	Gain = 512		±2.3		mV
Input Leakage Current[1]		−3		+3	nA
Input Offset Current[1, 23]			0.5	1.5	nA
Voltage Channel					
Absolute Input Voltage Range		4		18	V
Input Voltage Range			0 to 28.8		V
VBAT Input Current	VBAT = 18V	3	5.5	8	µA
Temperature Channel	V_{REF} = (REG_AVDD, GND_SW) / 2				
Absolute Input Voltage Range		100		1300	mV
Input Voltage Range			0 to V_{REF}		V
VTEMP Input Current[1]			2.5	100	nA
VOLTAGE REFERENCE					
ADC Precision Reference					
Internal V_{REF}			1.2		V
Power Up Time[1]			0.5		Ms
Initial Accuracy[1]	Measured at T_A = 25°C	−0.15		0.15	%
Temperature Coefficient[1, 24]		−20	±5	+20	ppm/°C
Reference long-term stability[25]			100		ppm/1000 hr
External Reference Input Range [26]		0.1		1.3	V
V_{REF} Divide by 2 Initial Error[1]			0.1	0.3	%

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
ADC Low Power Reference					
Internal V_{REF}			1.2		V
Initial Accuracy	Measured at T_A = 25°C	−5		5	%
Initial Accuracy[11]	Using ADCREF. Measured at T_A = 25°C		0.1		%
Temperature Coefficient[1, 24]		−300	±150	+300	ppm/°C
RESISTIVE ATTENUATOR					
Divider Ratio			24		
Resistor Mismatch Drift			3		ppm/°C
ADC GROUND SWITCH					
Resistance	Direct path to ground		10		Ω
	20 kΩ resistor selected	10	20	30	k Ω
Input Current				6	mA
TEMPERATURE SENSOR[27]	After user calibration				
Accuracy	MCU in power down or standby mode		±3		°C
	MCU in power down or standby mode, Temperature range = −25°C to +65°C		±2		°C
POWER-ON RESET (POR)					
POR Trip Level	Refers to voltage at VDD pin	2.85	3.0	3.15	V
POR Hysteresis			300		mV
RESET Time-Out from POR			20		ms
LOW VOLTAGE FLAG (LVF)					
LVF Level	Refers to voltage at VDD pin	1.9	2.1	2.3	V
POWER SUPPLY MONITOR (PSM)					
PSM Trip Level	Refers to voltage at VDD pin		6.0		V
WATCHDOG TIMER (WDT)					
Timeout Period[1]	32.768 kHz clock, 256 pre-scale	0.008		512	sec
Timeout Step Size			7.8		ms
FLASH/EE MEMORY[1]					
Endurance[28]		10,000			Cycles
Data Retention[29]	T_J = 85°C	20			Years
DIGITAL INPUTS	All digital inputs except NTRST				
Input Leakage Current	Input (high) = REG_DVDD		±1	±10	µA
Input Pull-up Current	Input (low) = 0V	10	20	80	µA
Input Capacitance			10		pF
Input Leakage Current	NTRST only: Input (low) = 0 V		±1	±10	µA
Input Pull-down Current	NTRST only: input (high) = REG_DVDD	30	55	100	µA
LOGIC INPUTS[1]	All logic inputs				
VINL, Input Low Voltage				0.4	V
VINH, Input High Voltage		2.0			V
CRYSTAL OSCILLATOR[1]					
Logic Inputs, XTAL1 Only					
VINL, Input Low Voltage				0.8	V
VINH, Input High Voltage		1.7			V
XTAL1 Capacitance			12		pF
XTAL2 Capacitance			12		pF

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
ON-CHIP OSCILLATORS					
Low Power Oscillator			131.072		kHz
Accuracy[30]	Includes drift data from 1000 hr life-test	−6		3	%
Precision Oscillator			131.072		kHz
Accuracy	Includes drift data from 1000 hr life-test	-1.2		1.2	%
MCU CLOCK RATE	8 programmable core clock selections within this range (binary divisions 1, 2, 4, 8…..64, 128)	0.160	10.24	20.48	MHz
MCU START-UP TIME					
At Power-On	Includes kernel power-on execution time		25		ms
After Reset Event	Includes kernel power-on execution time		5		ms
From MCU Power-Down					
Oscillator Running					
Wakeup from Interrupt			2		ms
Wakeup from LIN			2		ms
Crystal Powered Down					
Wakeup from Interrupt			500		ms
Internal PLL Lock Time			1		ms
LIN I/O GENERAL					
Baud Rate		1000		20,000	Bits/sec
VDD	Supply voltage range for which the LIN interface is functional	7		18	V
Input capacitance			5.5		pF
LIN comparator response time[1]	Using 22Ohm resistor		38	90	µs
$I_{LIN\ DOM\ MAX}$	Current limit for driver when LIN bus is in dominant state. $V_{BAT} = V_{BAT\ (MAX)}$	40		200	mA
$I_{LIN\_PAS\_REC}$	Driver Off ; 7.0 V < V_{BUS} < 18 V; $V_{DD} = V_{LIN}$ −0.7 V			20	µA
$I_{LIN\_PAS\_DOM}$[1]	Input Leakage VLIN = 0 V	−1			mA
$I_{LIN\_NO\_GND}$[31]	Control Unit disconnected from ground GND = V_{DD}; 0 V V_{LIN} < 18 V ; V_{BAT} = 12 V	−1		+1	mA
V_{LIN_DOM}[1]	LIN receiver dominant state, V_{DD} > 7.0 V			0.4 V_{DD}	V
V_{LIN_REC}[1]	LIN receiver recessive state, V_{DD} > 7.0 V	0.6 V_{DD}			V
V_{LIN_CNT}[1]	LIN receiver centre voltage, V_{DD} > 7.0 V	0.475 V_{DD}	0.5 V_{DD}	0.525 V_{DD}	V
V_{HYS}[1]	LIN receiver hysteresis voltage			0.175 V_{DD}	V
$V_{LIN\_DOM\_DRV\_LOSUP}$[1]	LIN dominant output voltage, V_{DD} 7 V, R_L 500 Ω			1.2	V
$V_{LIN\_DOM\_DRV\_HISUP}$	LIN dominant output voltage, V_{DD} 18 V, R_L 500 Ω			2	V
$V_{LIN\_DOM\_DRV\_LOSUP}$[1]	LIN dominant output voltage, VDD 7V, R_L 1000 Ω	0.6			V
$V_{LIN\_DOM\_DRV\_HISUP}$[1]	LIN dominant output voltage, VDD 18 V, R_L 1000 Ω	0.8			V
$V_{LIN\_RECESSIVE}$	LIN recessive output voltage	0.8 V_{DD}			V
V_{BAT}-Shift[31]		0		0.1 V_{DD}	V
GND-Shift[31]		0		0.1 V_{DD}	V
R_{slave}	Slave termination resistance	20	30	47	k Ω
$V_{Serial\ Diode}$[31]	Voltage drop at the serial diode D_{Ser_Int}	0.4	0.7	1	V
LIN I/O GENERAL Contd	Bus load conditions (CBUS||RBU): 1nF||1k Ω; 6.8nF|| 660 Ω ; 10nF || 500 Ω				
Symmetry of Transmit Propagation Delay[1]	VDD_{MIN} = 7 V	−2		+2	µs
Receive Propagation Delay[1]	VDD_{MIN} = 7 V			6	µs
Symmetry of Receive Propagation Delay[1]	VDD_{MIN} = 7 V	−2		+2	µs

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
LIN V2.0 SPECIFICATION	Bus load conditions (CBUS||RBUS):				
	1 nF||1 kΩ; 6.8 nF|| 660 Ω; 10 nF||500 Ω				
D1	Duty Cycle 1	0.396			
	$TH_{REC(MAX)} = 0.744 \times V_{BAT}$				
	$TH_{DOM(MAX)} = 0.581 \times V_{BAT}$				
	V_{SUP} = 7.0 V…18 V; t_{BIT} = 50 µs				
	$D1 = t_{BUS\_REC(MIN)}/(2 \times t_{BIT})$				
D2	Duty Cycle 2			0.581	
	$TH_{REC(MIN)} = 0..284 \times V_{BAT}$				
	$TH_{DOM(MIN)} = 0.422 \times V_{BAT}$				
	V_{SUP} = 7.0 V…18 V; t_{BIT} = 50 µs				
	$D2 = t_{BUS\_REC(MAX)}/(2 \times t_{BIT})$				
BSD I/O[32]					
Baud Rate		1164	1200	1236	Bits/sec
VOL, Output Lo-Voltage				1.2	V
VOH, Output Hi-Voltage		0.8 V_{DD}		V_{DD}	V
ILH, High Leakage Current	$V_{BSD} = V_{DD}$ or 0 V	−5	5	20	uA
$I_{o(sc)}$ Short-Circuit Output Current	$V_{BSD} = V_{DD}$ = 12 V	50	80	120	mA
VINL, Input Low Voltage				1.8	V
VINH, Input High Voltage		0.7 V_{DD}			V
WAKE					
	R_L = 1 kΩ, C_{BUS} = 91 nF, R_{LIMIT} = 39 Ω				
VDD[1]	Supply voltage range for which the Wake pin is functional	7		18	V
V_{OH}[33]	Output high level	5			V
V_{OL}[33]	Output low level			2	V
V_{IH}	Input high level	4.6			V
V_{IL}	Input low level			1.2	V
Monoflop Timeout	Timeout period	0.5	1.3	2	sec
$I_{o(sc)}$ Short-Circuit Output Current			100		mA
SERIAL TEST INTERFACE	R_L = 500 Ω, C_{BUS} = 2.4 nF, R_{LIMIT} = 39 Ω				
Baud Rate				40	kbps
VDD	Supply voltage range for which STI is functional	7		18	V
V_{OH}	Output high level	0.6			V_{DD}
V_{OL}	Output low level			0.4	V_{DD}
V_{IH}	Input high level	0.6			V_{DD}
V_{IL}	Input low level			0.4	V_{DD}
PACKAGE THERMAL SPECIFICATIONS					
Thermal Shutdown[134]		140	150	160	°C
Thermal Impedance (θ_{ja})[35]	48 LFCSP, Stacked Die		45		°C/W
Thermal Impedance (θ_{ja})[35]	48 LQFP, Stacked Die		75		°C/W
POWER REQUIREMENTS					
Power Supply Voltages					
VDD (Battery Supply)		3.5		18	V
REG_DVDD, REG_AVDD [36]		2.5	2.6	2.7	V
Power Consumption					
IDD – MCU Normal Mode[37]	MCU Clock Rate = 10.24MHz, ADC Off		10	20	mA
IDD – MCU Normal Mode[37]	MCU Clock Rate = 20.48MHz, ADC Off		20		mA
IDD – MCU Powered Down[1]	ADC Low Power Mode, measured over an ambient temperature range of −10°C to +40°C		300	400	µA
	(Continuous ADC Conversion)				

Parameter	Test Conditions/Comments	Min	Typ	Max	Unit
IDD–MCU Powered Down[1]	ADC Low Power Mode, measured over an ambient temperature range of −40°C to +85°C (Continuous ADC Conversion)		300	500	µA
IDD – MCU Powered Down[1]	ADC Low Power-Plus Mode, measured over an ambient temperature range of −10°C to +40°C (Continuous ADC Conversion)		520	700	µA
IDD – MCU Powered Down	Average Current, Measured with Wake and Watchdog Timer clocked from Low Power Oscillator (−40°C to +85°C)		120	300	µA
IDD – MCU Powered Down[1]	Average Current, Measured with Wake and Watchdog Timer clocked from Low Power Oscillator over an ambient temperature range of −10°C to +40°C		120	175	µA
IDD –Current ADC			1.7		mA
IDD –Voltage/Temperature ADC			0.5		mA
IDD – Precision Oscillator			400		µA

[1] These numbers are not production tested, but are guaranteed 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] These numbers include internal reference temperature drift.
[8] Factory Calibrated at Gain = 1.
[9] System calibration at specific gain range will remove the error at this gain range
[10] Includes an initial system calibration
[11] When used in conjunction with ADCREF, the Low Power Mode Reference error MMR.
[12] Using ADC Normal Mode Voltage Reference
[13] Typical Noise in Low Power modes is measured with Chop enabled.
[14] Voltage Channel Specifications include resistive attenuator input stage
[15] System Calibration will remove this error
[16] rms noise is referred to Voltage attenuator input, for example at F_{ADC}=1KHz, typical rms noise at the ADC input is 7.5uV, scaled by the attenuator (24) yields these input referred noise figures
[17] ADC Self Offset calibration will remove this error.
[18] Valid after an initial Self Calibration
[19] System Calibration will remove this error
[20] In ADC Low Power Mode the input range is fixed at ±9.375mV. In ADC Low Power Plus Mode the input range is fixed at ±2.34375mV.
[21] 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).
[22] Limited by minimum absolute input voltage range.
[23] Valid for a differential input less than 10mV
[24] Measured using Box Method
[25] The long-term stability specification is non cumulative. The drift in subsequent 1,000 hour periods is significantly lower than in the first 1,000 hour period.
[26] References of up to REG_AVDD can be accommodated by enabling an internal Divide-by-2
[27] Die Temperature.
[28] 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.
[29] Retention lifetime equivalent at junction temperature (Tj) = 85°C as per JEDEC Std. 22 method A117. Retention lifetime will de-rate with junction temperature.
[30] Low Power oscillator can be calibrated against either the precision oscillator or the external 32.768kHz crystal in user code
[31] These numbers are not production tested, but are supported by LIN Compliance testing.
[32] BSD Electrical Specifications, except High and Low voltage levels, are per LIN2.0 with pull-up resistor disabled and C_{Load}= 10nF max.
[33] Specified after Rlimit of 39Ohms
[34] The MCU core is not shutdown but interrupted and high voltage I/O pins are disabled in response to a thermal shutdown event.
[35] Thermal Impedance can be used to calculate the thermal gradient from ambient to die temperature.
[36] Internal Regulated Supply available at REG_DVDD (I_{SOURCE} =5mA), and REG_AVDD (I_{SOURCE} =1mA)
[37] Typical, additional supply current consumed during Flash memory program and erase cycles is 7mA and 5mA respectively.

TIMING SPECIFICATIONS

SPI Timing Specifications

Table 2. SPI Master Mode Timing (PHASE Mode = 1)

Parameter	Description	Min	Typ	Max	Unit
t_{SL}	SCLOCK low pulse width		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLOCK high pulse width		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge				ns
t_{DSU}	Data input setup time before SCLOCK edge				ns
t_{DHD}	Data input hold time0 after SCLOCK edge				ns
t_{DF}	Data output fall time				ns
t_{DR}	Data output rise time				ns
t_{SR}	SCLOCK rise time				ns
t_{SF}	SCLOCK fall time				ns

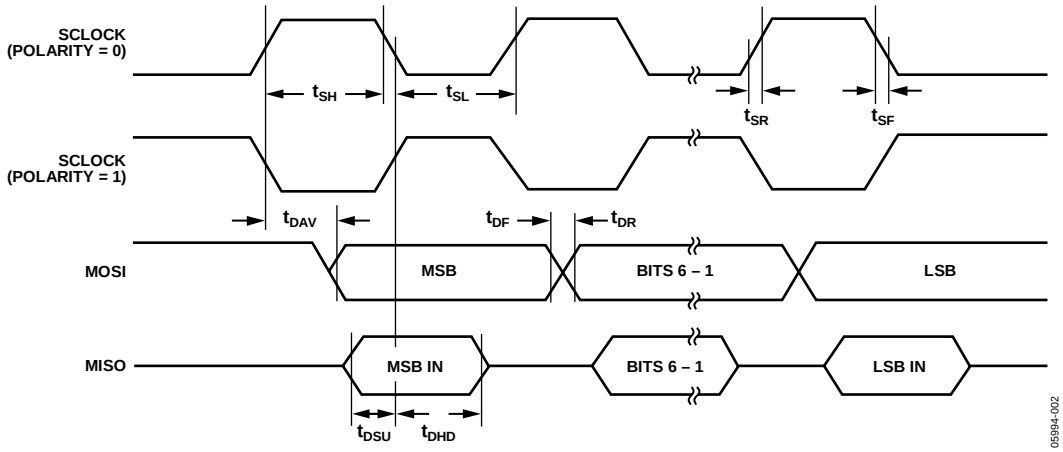


Figure 2. SPI Master Mode Timing (PHASE Mode = 1)

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

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

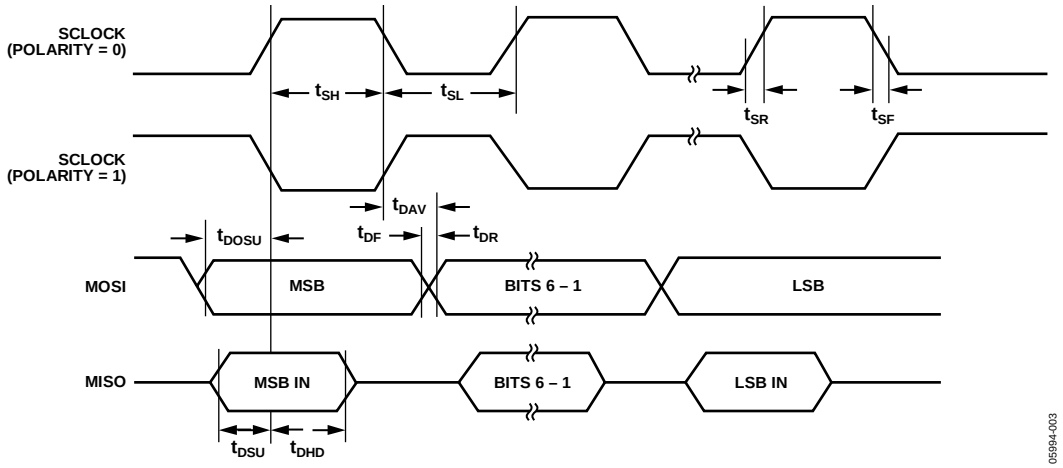


Figure 3. SPI Master Mode Timing (PHASE Mode = 0)

Table 4. SPI Slave Mode Timing (PHASE Mode = 1)

Parameter	Description	Min	Typ	Max	Unit
t_{CS}	CS to SCLOCK edge				ns
t_{SL}	SCLOCK low pulse width		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLOCK high pulse width		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge				ns
t_{DSU}	Data input setup time before SCLOCK edge				ns
t_{DHD}	Data input hold time after SCLOCK edge				ns
t_{DF}	Data output fall time				ns
t_{DR}	Data output rise time				ns
t_{SR}	SCLOCK rise time				ns
t_{SF}	SCLOCK fall time				ns
t_{SFS}	CS high after SCLOCK edge				ns

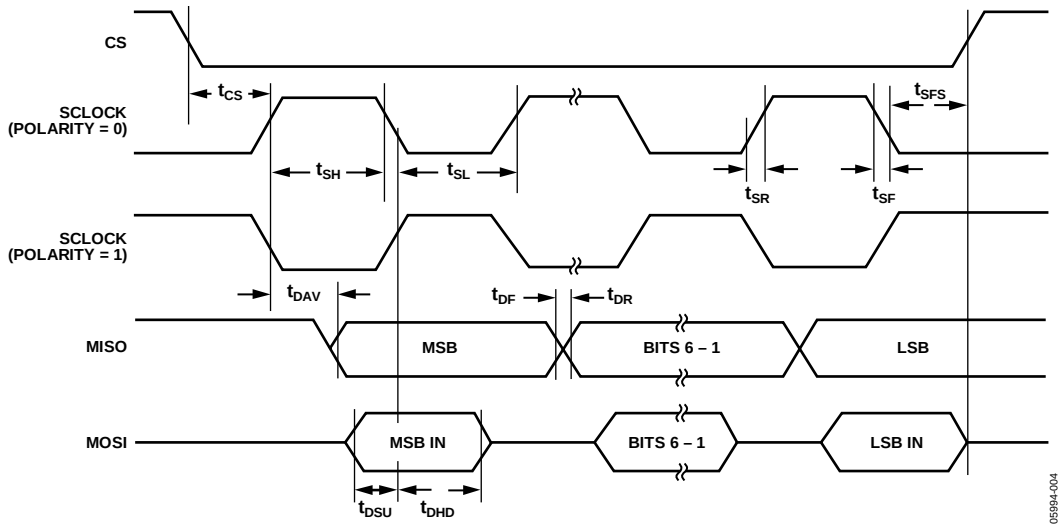


Figure 4. SPI Slave Mode Timing (PHASE Mode = 1)

Table 5. SPI Slave Mode Timing (PHASE Mode = 0)

Parameter	Description	Min	Typ	Max	Unit
t_{CS}	CS to SCLOCK edge				ns
t_{SL}	SCLOCK low pulse width		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLOCK high pulse width		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge				ns
t_{DSU}	Data input setup time before SCLOCK edge				ns
t_{DHD}	Data input hold time after SCLOCK edge				ns
t_{DF}	Data output fall time				ns
t_{DR}	Data output rise time				ns
t_{SR}	SCLOCK rise time				ns
t_{SF}	SCLOCK fall time				ns
t_{DOCS}	Data output valid after CS edge				ns
t_{SFS}	CS high after SCLOCK edge				ns

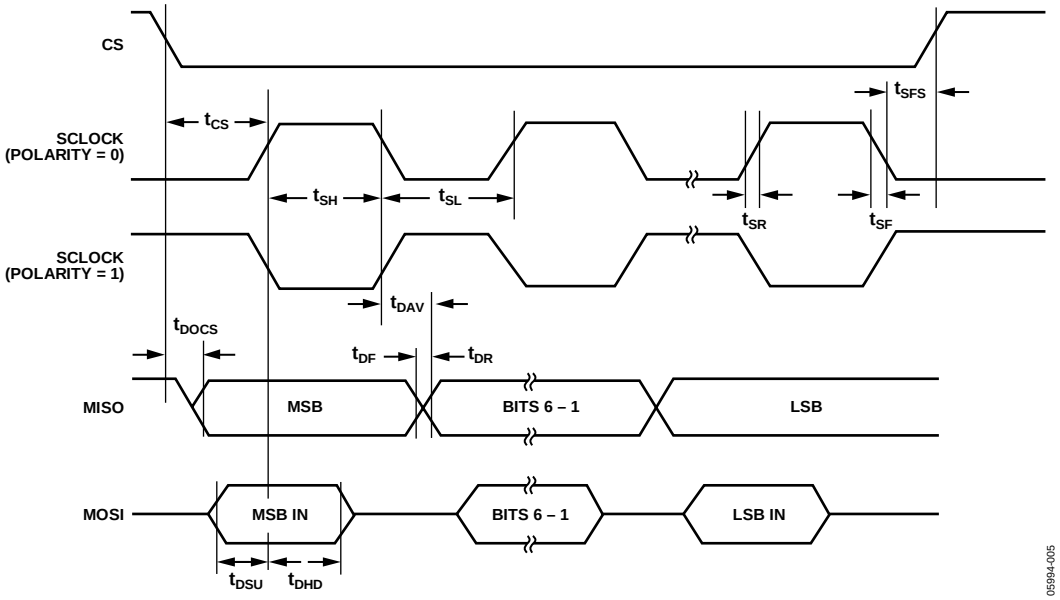


Figure 5. SPI Slave Mode Timing (PHASE Mode = 0)

LIN Timing Specifications

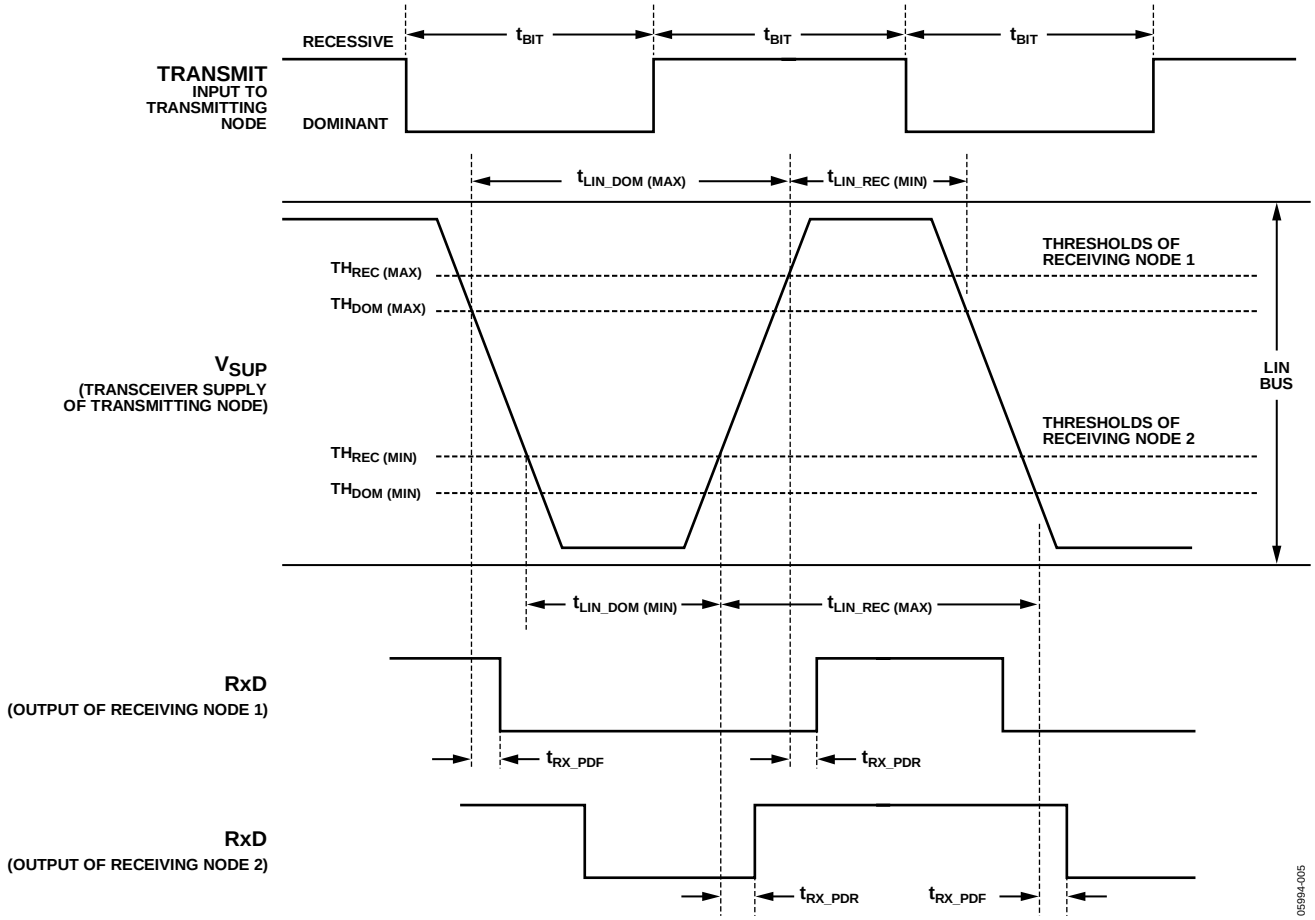


Figure 6 : LIN V2.0 Timing Specification

TERMINOLOGY

Conversion Rate
The conversion rate specifies the rate at which an output result is available from the ADC, once the ADC has settled.

The sigma-delta conversion techniques used on this part mean that while the ADC front-end signal is over-sampled at a relatively high sample rate, a subsequent digital filter is employed to decimate the output to give a valid 16-Bit data conversion result at output rates from 1Hz to 8 KHz.

It should also be noted that when software switches from one input to another (on the same ADC), the digital filter must first be cleared and then allowed to average a new result. Depending on the configuration of the ADC and the type of filter this can take multiple conversion cycles.

Integral Non Linearity (INL)
This is the maximum deviation of any code from a straight line passing through the endpoints of the transfer function. The endpoints of the transfer function are zero scale, a point 0.5 LSB below the first code transition and full scale, a point 0.5 LSB above the last code transition (111 . . . 110 to 111 . . . 111). The error is expressed as a percentage of full scale.

No Missing Codes
This is a measure of the Differential Non-Linearity of the ADC. The error is expressed in bits and specifies the number of codes (ADC results) as 2^N Bits, where is N = No Missing Codes, guaranteed to occur through the full ADC input range.

Offset Error
This is the deviation of the first code transition ADC input voltage from the ideal first code transition.

Offset Error Drift
Offset Error Drift is the variation in absolute offset error with respect to temperature. This error is expressed as LSBs per °C.

Gain Error
This is a measure of the span error of the ADC. It is a measure

of the difference between the measured and the ideal span between any two points in the transfer function.

Output Noise
The output noise is specified as the standard deviation (or 1 X Sigma) of ADC output codes distribution collected when the ADC input voltage is at a dc voltage. It is expressed as µ rms. The output or RMS noise can be used to calculate the Effective Resolution of the ADC as defined by the following equation

Effective Resolution= Log_2 (full-scale range/rms noise) Bits

The peak-to-peak noise is defined as the deviation of codes that fall within 6.6 X Sigma of the distribution of ADC output codes collected when the ADC input voltage is at dc. The peak-to-peak noise is therefore calculated as 6.6 times the rms noise.

The peak-to-peak noise can be used to calculate the ADC (Noise Free, Code) Resolution for which there will be no code flicker within a 6.6-Sigma limit as defined by the following equation

Noise Free Code Resolution = Log_2 (full-scale range/peak-to-peak noise) Bits

Data Sheet Acronyms

ADC	Analog to Digital Converter
ARM	Advanced RISC Machine
JTAG	Joint Test Action Group
LIN	Local Interconnect Network
LSB	Least Significant Byte/Bit
LVF	Low Voltage Flag
MCU	MicroController
MMR	Memory Mapped Register
MSB	Most Significant Byte/Bit
PID	Protected Identifier
POR	Power On Reset
PSM	Power Supply Monitor
RMS	Root Mean Square
STI	Serial Test Interface

ABSOLUTE MAXIMUM RATINGS

T_A = −40°C to 115°C unless otherwise noted

Table 6.

Parameter	Rating
AGND to DGND to VSS to IO_VSS	−0.3 V to +0.3 V
VBAT to AGND	−22 V to +40 V
V_{DD} to VSS	−0.3 V to +33 V
V_{DD} to VSS for 1 second	−0.3 V to +40 V
LIN to IO_VSS	−16 V to +40 V
STI/WU to IO_VSS	−3 V to +33 V
Wake Continuous Current	50 mA
HV IO Pins Short-Circuit Current	100 mA
Digital I/O Voltage to DGND	−0.3 V to REG_DV_{DD} + 0.3 V
V_{REF} to AGND	−0.3 V to REG_AV_{DD} + 0.3 V
ADC Inputs to AGND	−0.3 V to REG_AV_{DD}+0.3 V
ESD (HBM) Rating	
LIN, STI, WU and VBAT	±4 kV
All Other Pins	±2 kV
Storage Temperature	125°C
Junction Temperature	
Transient	150°C
Continuous	130°C
Lead Temperature, Soldering Reflow (15 sec)	260°C

Stresses above those listed under Absolute Maximum Ratings may cause permanent damage to the device. This is a stress rating only; functional operation of the device at these or any other conditions above those indicated in the operational section of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although this product features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

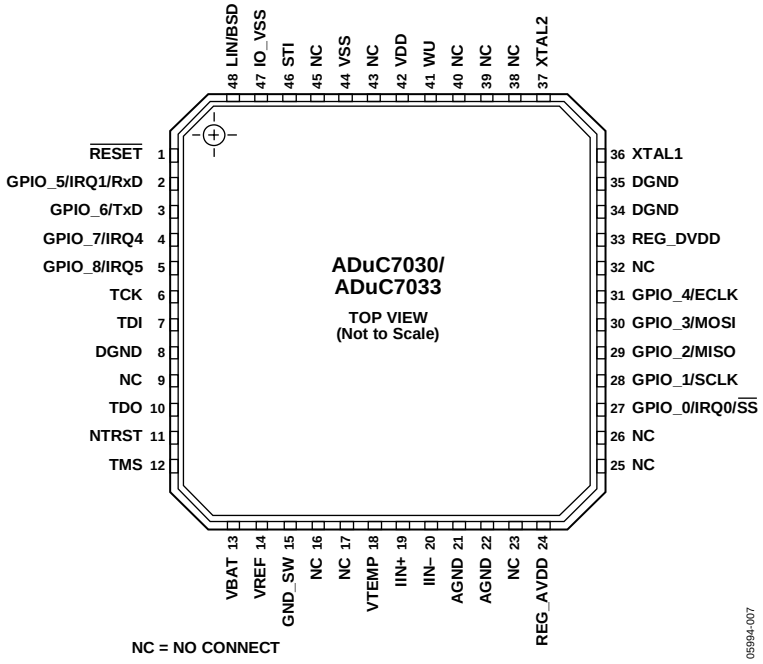


Figure 7. Pin Configuration

Table 7. Pin Function Descriptions

Pin No.	Mnemonic	Type[1]	Function
1	$\overline{\text{RESET}}$	I	Reset Input Pin, Active Low. This pin has an internal, weak pull-up resistor to REG_DVDD. If this pin is not being used, it can be left not connected. For added security and robustness, it is recommended that this pin be strapped via a resistor to REG_DVDD.
2	GPIO_5/IRQ1/RxD	I/O	General Purpose Digital Input/Output 5, External Interrupt Request 1, or Receive Data. This is a multifunction pin. By default and after power-on-reset, this pin configures as an input. The pin has an internal weak pull-up resistor and when not in use, it can be left unconnected. This multifunction pin can be configured in one of three states, namely: General purpose digital I/O 5 External Interrupt Request 1, active high Receive data for UART serial port
3	GPIO_6/TxD	I/O	General Purpose Digital Input/Output 6, Transmit Data. This is a multifunction pin. By default and after power-on-reset, this pin configures as an input. The pin has an internal weak pull-up resistor and when not in use, it can be left unconnected. This multifunction pin can be configured in one of two states, namely: General purpose digital I/O 6 Transmit Data for UART Serial Port
4	GPIO_7/IRQ4	I/O	General Purpose Digital Input/Output 7, External Interrupt Request. This is a multifunction pin. By default and after power-on-reset, this pin configures as an input. The pin has an internal weak pull-up resistor and when not in use, it can be left unconnected. This multifunction pin can be configured in one of two states, namely: General purpose digital I/O 7 External Interrupt Request 4, active high
5	GPIO_8/ IRQ5	I/O	General Purpose Digital Input/Output 8, External Interrupt Request. This is a multifunction pin. By default and after power-on-reset, this pin configures as an input. The pin has an internal weak pull-up resistor and when not in use, it can be left unconnected. This multifunction pin can be configured in one of two states, namely: General purpose digital I/O 8 External Interrupt Request 5, active high
6	TCK	I	JTAG Test Clock. This clock input pin is one of the standard 5-pin JTAG debug ports on the part. TCK is an input pin only and has an internal weak pull-up resistor. This pin can be left unconnected when not in use.

Pin No.	Mnemonic	Type[1]	Function
7	TDI	I	JTAG Test Data Input. This data input pin is one of the standard 5-pin JTAG debug ports on the part. TDI is an input pin only and has an internal weak pull-up resistor. This pin can be left unconnected when not in use.
8, 34, 35	DGND	S	Ground Reference for On-Chip Digital Circuits.
9, 16, 17, 23, 25, 26, 32, 38 to 40, 43, 45	NC		No Connect. These pins are not internally connected, but are reserved for possible future use. Therefore, do not externally connect these pins. These pins can be grounded, if required.
10	TDO	O	JTAG Test Data Output. This data output pin is one of the standard 5-pin JTAG debug ports on the part. TDO is an output pin only. On power-on, this output is disabled and pulled high via an internal weak pull-up resistor. This pin can be left unconnected when not in use.
11	NTRST	I	JTAG Test Reset. This reset input pin is one of the standard 5-pin JTAG debug ports on the part. NTRST is an input pin only and has an internal weak pull-down resistor. This pin can be left unconnected when not in use. NTRST is also monitored by the on-chip kernel to enable LIN boot-load mode.
12	TMS	I	JTAG Test Mode Select. This mode select input pin is one of the standard 5-pin JTAG debug ports on the part. TMS is an input pin only and has an internal weak pull-up resistor. This pin can be left unconnected when not in use.
13	VBAT	I	Battery Voltage Input to Resistor Divider.
14	VREF	I	External Reference Input Terminal. When this input is not used, connect it directly to the AGND system ground.
15	GND_SW	I	Switch to Internal Analog Ground Reference. This pin is the negative input for the external temperature channel and external reference. When this input is not used, connect it directly to the AGND system ground.
18	VTEMP	I	External Pin for NTC/PTC Temperature Measurement.
19	IIN+	I	Positive Differential Input for Current Channel.
20	IIN–	I	Negative Differential Input for Current Channel.
21, 22	AGND	S	Ground Reference for On-Chip Precision Analog Circuits.
24	REG_AVDD	S	Nominal 2.6 V Output from On-Chip Regulator.
27	GPIO_0/IRQ0/$\overline{SS}$	I/O	General Purpose Digital Input/Output 0, External Interrupt Request 0, or SPI Interface. This is a multi-function pin. By default and after power-on-reset, this pin is configured as an input. The pin has an internal weak pull-up resistor and if not being used it can be left unconnected. This multifunction pin can be configured in one of three states, namely: General purpose digital I/O 0 External Interrupt Request 0, active high SPI interface, slave select input
28	GPIO_1/SCLK	I/O	General Purpose Digital Input/Output 1, SPI Interface. This is a multi-function pin. By default and after power-on-reset, this pin is configured as an input. The pin has an internal weak pull-up resistor and if not being used it can be left unconnected. This multi-function pin can be configured in one of 2 states, namely: General purpose digital I/O 1 SPI interface, serial clock input
29	GPIO_2/MIS0	I/O	General Purpose Digital Input/Output 2 is a Multi-Function Pin. By default and after Power-On-Reset, this pin is configured as an input. The pin has an internal weak pull-up resistor and if not being used it can be left unconnected. This multi-function pin can be configured in one of 2 states, namely: General Purpose Digital I/O 2 SPI Interface, Master Input/Slave Output Pin
30	GPIO_3/MOSI	I/O	General Purpose Digital Input/Output 3 is a Multi-Function Pin. By default and after Power-On-Reset, this pin is configured as an input. The pin has an internal weak pull-up resistor and if not being used it can be left unconnected. This multi-function pin can be configured in one of 2 states, namely: General Purpose Digital I/O 3 SPI Interface, Master Output/Slave Input Pin

Pin No.	Mnemonic	Type[1]	Function
31	GPIO_4/ECLK	I/O	General Purpose Digital Input/Output 4 is a Multi-Function Pin. By default and after Power-On-Reset, this pin is configured as an input. The pin has an internal weak pull-up resistor and if not being used it can be left unconnected. This multi-function pin can be configured in one of 2 states, namely: General Purpose Digital I/O 4 Output a 2.56MHz clock
33	REG_DVDD	S	Nominal 2.6V output from the on-chip regulator
36	XTAL1	O	Crystal Oscillator Output. If an external Crystal is not being used, this pin can be left unconnected.
37	XTAL2	I	Crystal Oscillator Input. If an external Crystal is not being used, this pin should be connected to the DGND system ground.
41	WU	I/O	High Voltage Wake-Up pin. This high voltage I/O pin has an internal 10 kΩ pull-down resistor and a high-side driver to VDD. If this pin is not being used, it should not be connected externally.
42	VDD	S	Battery Power Supply to on-chip regulator
44	VSS	S	Ground Reference for the internal Voltage Regulators
46	STI	I/O	Serial Test Interface Output Pin. If this pin is not being used it should be connected externally to the IO_VSS ground reference
47	IO_VSS	S	Ground Reference for High Voltage I/O Pins
48	LIN/BSD	I/O	LIN Serial Interface Input/Output Pin

[1] I = Input, O = Output, S = Supply

THEORY OF OPERATION

The ADuC7030/ADuC7033 are each a complete system solution for battery monitoring in 12 V automotive applications. The device integrates all of the required features to precisely and intelligently monitor, process, and diagnose 12 V battery parameters including battery current, voltage, and temperature over a wide range of operating conditions.

Minimizing external system components, the device is powered directly from the 12 V battery. An on-chip, low drop-out regulator generates the supply voltage for three integrated 16-bit Σ-Δ ADCs. The ADCs precisely measure battery current, voltage, and temperature to characterize the car battery's state of health and charge.

A Flash/EE memory-based ARM7® microcontroller (MCU) is also integrated on-chip and is used both to pre-process the acquired battery variables, and to manage communications from the ADuC7030/ADuC7033 to the main electronic control unit (ECU) via a local interconnect network (LIN) interface that is integrated on-chip.

Both the MCU and the ADC subsystem can be individually configured to operate in normal or flexible power saving modes of operation.

In its normal operating mode, the MCU is clocked indirectly from an on-chip oscillator via the phase locked loop (PLL) at a maximum clock rate of 20.48 MHz. In its power saving operating modes, the MCU can be totally powered down, waking up only in response to an ADC conversion result ready, digital comparators, the wake-up timer, a POR, or an external serial communication event.

The ADC can be configured to operate in a normal (full power) mode of operation, interrupting the MCU after various sample conversion events. The current channel features two low power modes, low power and low power plus, generating conversion results to a lower performance specification.

On-chip factory firmware supports in-circuit Flash/EE reprogramming via the LIN or JTAG serial interface ports, and nonintrusive emulation is also supported via the JTAG interface. These features are incorporated into a low-cost QuickStart™ Development System supporting the ADuC7030/ADuC7033.

The ADuC7030/ADuC7033 operate directly from the 12 V battery supply and are fully specified over a temperature range of −40°C to +115°C. The ADuC7030/ADuC7033 are functional, but with degraded performance, at temperatures from 115°C to 125°C.

OVERVIEW OF THE ARM7TDMI CORE

The ARM7 core is a 32-bit reduced instruction set computer (RISC), developed by ARM Ltd. The ARM7TDMI is a Von Neumann-based architecture, meaning that it uses a single 32-bit bus for instruction and data. The length of the data can be 8, 16 or 32 bits and the length of the instruction word is either 16 bits or 32 bits, depending on the mode in which the core is operating.

The ARM7TDMI is an ARM7 core with four additional features as listed in Table 8.

Table 8. ARM7TDMI

Feature	Description
T	Support for the thumb (16-bit) instruction set
D	Support for debug
M	Enhanced multiplier
I	Includes the EmbeddedICE module to support embedded system debugging

Thumb Mode (T)

An ARM instruction is 32 bits long. The ARM7TDMI processor supports a second instruction set compressed into 16 bits, the thumb instruction set. Faster code execution from 16-bit memory and greater code density can be achieved by using the thumb instruction set, which makes the ARM7TDMI core particularly suited for embedded applications.

However, the thumb mode has three limitations:

- Relative to ARM, the thumb code usually requires more instructions to perform that same task. Therefore, ARM code is best for maximizing the performance of time-critical code in most applications.
- The thumb instruction set does not include some instructions that are needed for exception handling, so ARM code can be required for exception handling.
- When an interrupt occurs, the core vectors to the interrupt location in memory and executes the code present at this address. The first command is required to be in ARM code.

Multiplier (M)

The ARM7TDMI instruction set includes an enhanced multiplier, with four extra instructions to perform 32-bit by 32-bit multiplication with 64-bit result, and 32-bit by 32-bit multiplication-accumulation (MAC) with 64-bit result.

EmbeddedICE (I)

The EmbeddedICE module provides integrated on-chip debug support for the ARM7TDMI. The EmbeddedICE module contains the breakpoint and watchpoint registers that allow nonintrusive user code debugging. These registers are controlled through the JTAG test port. When a breakpoint or watchpoint is encountered, the processor halts and enters debug state. Once in a debug state, the processor registers can be interrogated, as well as the Flash/EE, the SRAM, and the memory mapped registers.

ARM7 Exceptions

The ARM7 supports five types of exceptions, with a privileged processing mode associated with each type. The five types of exceptions are

- Normal interrupt or IRQ. It is provided to service general-purpose interrupt handling of internal and external events.
- Fast interrupt or FIQ. It is provided to service data transfer or communication channel with low latency. FIQ has priority over IRQ.
- Memory abort (prefetch and data).
- Attempted execution of an undefined instruction.
- Software interrupt (SWI) instruction that can be used to make a call to an operating system.

Typically, the programmer defines interrupts as IRQ, but for higher priority interrupts, the programmer can define interrupts as of type FIQ.

The priority of these exceptions and vector address are listed in Table 9.

Table 9.

Priority	Exception	Address
1	Hardware Reset	0x00
2	Memory Abort (Data)	0x10
3	FIQ	0x1C
4	IRQ	0x18
5	Memory Abort (Prefetch)	0x0C
6	Software Interrupt[1]	0x08
6	Undefined Instruction[1]	0x04

[1] A software interrupt and an undefined instruction exception have the same priority and are mutually exclusive.

The list of exceptions in Table 9.are located from 0x00 to 0x1C, with a reserved location at 0x14. This location is required to be written with either 0x27011970 or the checksum of Page Zero, excluding location 0x14. If this is not done, user code does not execute and LIN download mode is entered. For more information, refer to the relevant LIN download Technote.

ARM Registers

The ARM7TDMI has 16 standard registers. R0-R12 are used for data manipulation, R13 is the stack pointer, R14 is the link register and R15 is the program counter which indicates the instruction currently being executed. The link register contains the address from which the user has branched, if the branch and link command was used, or the command during which an exception occurred.

The stack pointer contains the current location of the stack. As a general rule of thumb on an ARM7TDMI, the stack starts at the top of the available RAM area, and descends, using the area as required. A separate stack is defined for each of the exceptions. The size of each stack is user configurable and is dependent on the target application. On the ADuC7030/ ADuC7033 the stack begins at 0x00040FFC and descends. When programming using high level languages, such as C, it can be possible to ensure that the stack does not overflow. This is dependent on the compiler used.

When an exception occurs, some of the standard register are replaced with registers specific to the exception mode. All exception modes have replacement banked registers for the stack pointer (R13) and the link register (R14) as represented in Figure 2. The FIQ mode has more registers (R8 to R12) supporting faster interrupt processing. With the increased number of non-critical registers, the interrupt may be processed without the need to save or restore these registers, which reduces the response time of the interrupt handling process.

More information relative to the programmer's model and the ARM7TDMI core architecture can be found in the following documents available from ARM Ltd.:
DDI0029G, ARM7TDMI Technical Reference Manual.
DDI0100E, ARM Architecture Reference Manual.

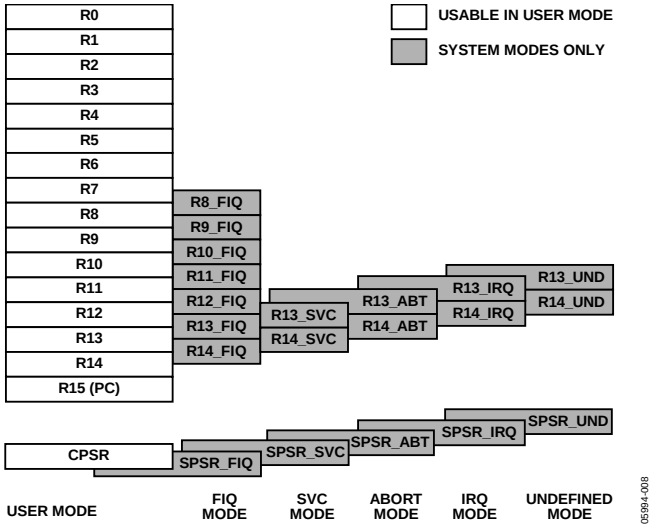


Figure 8. ADuC7030/ADuC7033 Register Organization

Interrupt Latency

The worst case latency for an FIQ consists of the longest time the request can take to pass through the synchronizer, plus the time for the longest instruction to complete (the longest instruction is an LDM) which loads all the registers including the PC, plus the time for the data abort entry, plus the time for FIQ entry. At the end of this time, the ARM7TDMI will be executing the instruction at 0x1C (FIQ interrupt vector address). The maximum total time is 50 processor cycles, which is just over 2.44 μS in a system using a continuous 20.48 MHz processor clock. The maximum IRQ latency calculation is similar, but must allow for the fact that FIQ has higher priority and could delay entry into the IRQ handling routine for an arbitrary length of time. This time may be reduced to 42 cycles if the LDM command is not used, some compilers have an option to compile without using this command. Another option is to run the part in THUMB mode where this is reduced to 22 cycles.

The minimum latency for FIQ or IRQ interrupts is five cycles. This consists of the shortest time the request can take through the synchronizer plus the time to enter the exception mode.

Note that the ARM7TDMI will initially (1st instruction) run in ARM (32-bit) mode when an exception occurs. The user may immediately switch from ARM mode to Thumb mode if required, e.g. when executing interrupt service routines.

MEMORY ORGANISATION

The ARM7, a Von Neumann architecture, MCU core sees memory as a linear array of 2^{32} byte locations. As shown in Figure 10 and Figure 11, the ADuC7030 and the ADuC7033 both map this into 4 distinct user areas namely, a re-mappable memory area, an SRAM area, a Flash/EE area and a Memory Mapped Register (MMR) area.

1. For the ADuC7030,the first 30kBytes of this memory space is used as an area into which the on-chip Flash/EE or SRAM can be remapped.
   For the ADuC7033, the first 94kBytes of this memory space is used as an area into which the on-chip Flash/EE or SRAM can be remapped.

2. Both the ADuC7030 and the ADuC7033, feature a second 4kByte area at the top of the memory map used to locate the Memory Mapped Registers (MMR), through which all on-chip peripherals are configured and monitored.

3. The ADuC7030 features a SRAM size of 4 kByte
   The ADuC7033 features a SRAM size of 6 kByte

4. The ADuC7030 features 32 kByte of On-Chip Flash/EE memory. 30kByte of On-Chip Flash/EE memory are available to the user.
   The ADuC7033 features 96 kByte of On-Chip Flash/EE memory. 94kByte of On-Chip Flash/EE memory are available to the user
   For both the ADuC7030 and ADuC7033, 2 kBytes are reserved for the on-chip Kernel.

Any access, either reading or writing, to an area not defined in the memory map will result in a Data Abort exception.

Memory Format

The ADuC7030/ADuC7033 memory organization is configured in little endian format: the least significant byte is located in the lowest byte address and the most significant byte in the highest byte address.

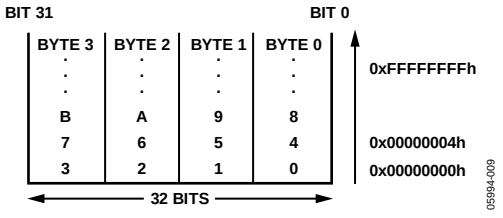


Figure 9. Little Endian Format

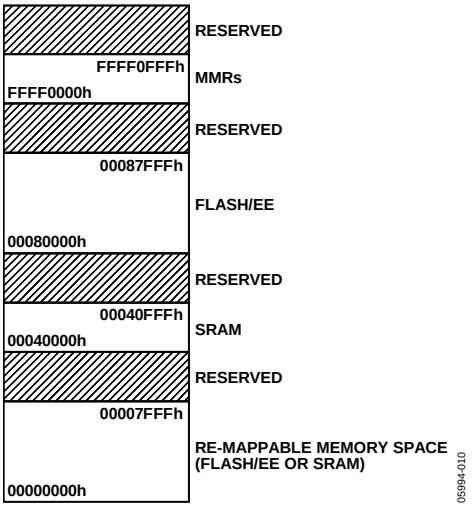


Figure 10. ADuC7030 Memory Map

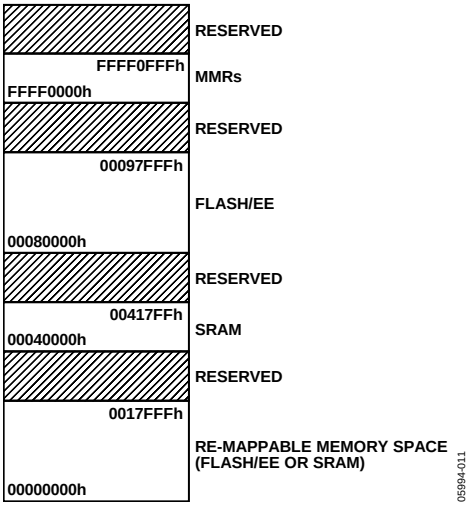


Figure 11. ADuC7033 Memory Map

SRAM

The ADuC7030 features 4kBytes of SRAM, organized as 1024 X 32 bits, i.e. 1024 Words, which is located at 0x40000.
The ADuC7033 features 6kBytes of SRAM, organized as 1536 X 32 bits, i.e. 1536 Words, which is located at 0x40000.

The RAM space can be used as data memory and also as a volatile program space.

ARM code can run directly from SRAM at full clock speed given that the SRAM array is configured as a 32-bit wide memory array.

SRAM is read/writeable in 8-, 16-, 32-bit segments.

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. This is done by a setting bit zero of the SYSMAP0 MMR, which is located at 0xFFFF0220. To revert Flash/EE to 0x00000000, bit zero of SYSMAP0 is cleared.

It may be desirable to remap RAM to 0x00000000 to optimize the interrupt latency of the ADuC7030/ADuC7033, as code may 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 will default to ARM mode.

Remap Operation

When a reset occurs on the ADuC7030/ADuC7033, 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 ADuC7030/ADuC7033 is in normal mode, it will execute the power-on configuration routine of the kernel and then jump to the reset vector address, 0x00000000, to execute the users reset exception routine. Since the Flash/EE is mirrored at the bottom of the memory array at reset, the reset routine must always be written in Flash/EE.

The REMAP command must be executed from the absolute Flash/EE address, and not from the mirrored, remapped segment of memory, as this may be replaced by SRAM. If a remap operation is executed whilst operating code from the mirrored location, Prefetch/Data aborts may occur or the user may observe abnormal program operation.

Any kind of reset will logically remap 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 Access

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 location 0x00000000.

Table 10. 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.

ADUC7030/ADUC7033 RESET

There are four kinds of reset: external reset, Power-on-reset, watchdog reset and software reset. The RSTSTA register indicates the source of the last reset and can also be written by user code to initiate a software reset event. The bits in this register can be cleared to '0' by writing to the RSTCLR MMR at 0xFFFF0234. The bit designations in RSTCLR mirror those of RSTSTA. These registers can be used during a reset exception service routine to identify the source of the reset. The implications of all four kinds of reset event are tabulated in Table 11 below.

Table 11. Device RESET Implications

IMPACT / RESET	Reset External Pins to Default State	Kernel Executed	Reset All External MMRs(excluding RSTSTA	Reset All HV Indirect Registers	Peripherals Reset	Watchdog Timer Reset	RAM Valid	RSTSTA (Status after Reset Event)
POR	✓	✓	✓	✓	✓	✓	Note 1	RSTSTA[0] =1
Watchdog Reset	✓	✓	✓	✓	✓	×	✓	RSTSTA[1] =1
Software Reset	✓	✓	✓	✓	✓	×	✓	RSTSTA[2] =1
External Reset Pin	✓	✓	✓	✓	✓	×	✓	RSTSTA[3] =1

Note 1: If LVF is enabled(HVCFG0[2]), RAM has not been corrupted by the POR reset mechanism if LVF Status bit HVSTA[6] is '1'.

RSTSTA Register:

Name: RSTSTA

Address: 0xFFFF0230

Default Value: Depends on type of reset

Access: Read/Write Access

Function: This 8-bit register indicates the source of the last reset event and can also be written by user code to initiate a software reset.

RSTCLR Register:

Name: RSTCLR

Address: 0xFFFF0234

Access: Write Only

Function: This 8-bit write only register clears the corresponding bit in RSTSTA.

Table 12. RSTSTA/RSTCLR MMR Bit Designations

Bit	Description
7 to 4	Not Used These bits are not used and will always read as '0'
3	External Reset Set to 1 automatically when an external reset occurs Cleared by setting the corresponding bit in RSTCLR
2	Software Reset Set to '1' by user code to generate a software reset. Cleared by setting the corresponding bit in RSTCLR
1	Watchdog timeout Set to 1 automatically when a watchdog timeout occurs Cleared by setting the corresponding bit in RSTCLR
0	Power-on-reset Set automatically when a power-on-reset occurs Cleared by setting the corresponding bit in RSTCLR

Note: If the "Software Reset" bit in RSTSTA is set, any write to RSTCLR that does not clear this bit will generate a software reset

FLASH/EE MEMORY AND THE ADUC7030/ADUC7033

The ADuC7030/ADuC7033 incorporate Flash/EE memory technology on-chip to provide the user with nonvolatile, in-circuit reprogrammable memory space.

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.

Overall, Flash/EE memory represents a step closer to the ideal memory device that includes nonvolatility, in-circuit program-mability, high density, and low cost. Incorporated in the ADuC7030/ADuC7033, 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.

The Flash/EE memory is physically located at 0x80000. Upon a hard reset, it logically maps to 0x00000000. The factory default contents of all Flash/EE memory locations is 0xFF. Flash/EE can be read in 8/16/32 bit segments, and written in segments of 16 bits. The Flash/EE is rated for 10K endurance cycles. This rating is based on the number of times that each individual byte is cycled i.e. erased and programmed. A redundancy scheme may be implemented in software to ensure greater than 10K cycles endurance.

The user can also write data variables to the Flash/EE memory during run-time code execution, for example, for storing diagnostic battery parameter data.

The entire Flash/EE is available to the user as code and non-volatile data memory. There is no distinction between data and program, as ARM code shares the same space. The real width of the Flash/EE memory is 16 bits, meaning that in ARM mode (32-bit instruction), two accesses to the Flash/EE are necessary for each instruction fetch. When operating at speeds less than 20.48 MHz the Flash/EE memory controller can transparently fetch the second 16-bit half word (part of the 32-bit ARM op-code) within a single core clock period. It is, therefore, recommended that for speeds less than 20.48 MHz, that is, CD > 0, that ARM mode is used. For 20.48MHz operation, that is, CD = 0 , it is recommended to operate in thumb mode.

The page size of this Flash/EE memory is 512 bytes. Typically, it takes the Flash/EE controller 20 ms to erase a page, irrespective of CD. To write a 16-bit word at CD = 0, 1, 2, 3 requires 50 µs; 70 µs at CD=4, 5; 80 µs at CD=6; and 105 µs at CD=7.

It is possible to write to a single 16 bit location only twice between erases, that is, it is possible to walk bytes, not bits. If a location is written to more than twice, then it is possible that the contents of the Flash/EE page may be corrupted.

The Flash/EE memory can be programmed in-circuit, using a serial download mode via the LIN interface or the integrated JTAG port.

(1) Serial Downloading (In-Circuit Programming)

The ADuC7030/ADuC7033 facilitate code download via the LIN pin. For more information please refer to the ADuC7030/ADuC7033 LIN download protocol technote.

(2) JTAG access

The ADuC7030/ADuC7033 feature an on-chip JTAG debug port to facilitate code download and debug.

ADuC7030 Flash/EE Memory

The total 32kBytes of Flash/EE are organized as 15k X 16 bits. 30kBytes are user space and 2kBytes are reserved for boot loader/kernel space.

ADuC7033 Flash/EE Memory

The total 96kBytes of Flash/EE are organized as 47k X 16 bits. 94kBytes user space and 2kBytes reserved for boot loader/kernel space.

ADUC7030 FLASH/EE CONTROL INTERFACE

The access to and control of the Flash/EE memory on the ADuC7030 is managed by an on-chip memory controller. The controller manages the Flash/EE memory as single block of 32 kbytes.

It should be noted that MCU core is halted until the command is completed. User software must ensure that the Flash/EE controller has completed any erase or write cycle before the PLL is powered down. If the PLL is powered down before an erase or write cycle is completed, the Flash/EE page can be corrupted. User Code, LIN and JTAG programming use the Flash/EE Control Interface, consists of the following MMRs:

FEE0STA: read only register, reflects the status of the Flash/EE control interface

FEE0MOD: sets the operating mode of the Flash/EE control interface

FEE0CON: 8-bit command register. The commands are interpreted as described in Table 13.

FEE0DAT: 16-bit data register.

FEE0ADR: 16-bit address register.

FEE0SIG: Holds the 24-bit code signature as a result of the signature command being initiated.

FEE0HID: Protection MMR. Controls read and write protection of the Flash/EE memory code space. If previously configured via the FEE0PRO register, FEE0HID can require a software key to enable access.

FEE0PRO: A buffer of the FEE0HID register that stores the FEE0HID value, so it is automatically downloaded to the FEE0HID registers on subsequent reset and power-on events

The following sections describe in detail the bit designations of each of Flash/EE control MMRs.

FEE0CON Register:

Name: FEE0CON

Address: 0xFFFF0E08

Default Value: 0x07

Access: Read/Write Access

Function: This 8-bit register is written by user code to control the operating modes of the Flash/EE memory controller.

Table 13. Command Codes in FEE0CON

Code	Command	Description
0x00*	Reserved	Reserved, this command should not be written by user code
0x01[1]	Single Read	Load FEE0DAT with the 16-bit data indexed by FEE0ADR
0x02[1]	Single Write	Write FEE0DAT at the address pointed by FEE0ADR. This operation takes 50μs.
0x03[1]	Erase-Write	Erase the page indexed by FEE0ADR and write FEE0DAT at the location pointed by FEE0ADR. This operation takes 20ms
0x04[1]	Single Verify	Compare the contents of the location pointed by FEE0ADR to the data in FEE0DAT. The result of the comparison is returned in FEE0STA bit 1
0x05[1]	Single Erase	Erase the page indexed by FEE0ADR
0x06[1]	Mass erase	Erase 30kBytes of user space. The 2kByte Kernel is protected. This operation takes 1.2s To prevent accidental execution, a command sequence is required to execute this instruction, this is described below.
0x07	Idle	Default command.
0x08	Reserved	Reserved, this command should not be written by user code
0x09	Reserved	Reserved, this command should not be written by user code
0x0A	Reserved	Reserved, this command should not be written by user code
0x0B	Signature	This command will result in a 24-bit LFSR based signature been generated and loaded into FEE0SIG. If FEE0ADR is less than 0x87800, this command will result in a 24-bit LFSR based signature of the user code space from the page specified in FEE0ADR upwards, including the Kernel, security bits and Flash/EE key. If FEE0ADR is greater than 0x87800, the Kernel and manufacturing data is signed. This operation takes 120us.
0x0C	Protect	This command can be run only once. The value of FEE0PRO is saved and can be removed only with a mass erase (0x06) or with the key
0x0D	Reserved	Reserved, this command should not be written by user code
0x0E	Reserved	Reserved, this command should not be written by user code
0x0F	Ping	No operation, interrupt generated

[1] The FEE0CON will always read 0x07 immediately after execution of any of these commands.

Command Sequence for Executing a Mass Erase

Giving the significance of the 'Mass Erase' command, a specific code sequence must be executed to initiate this operation.

1. Set bit 3 in FEE0MOD.
2. Write 0xFFC3 in FEE0ADR
3. Write 0x3CFF in FEE0DAT
4. Run the Mass Erase command 0x06 in FEE0CON

This sequence is illustrated in the following example:

```
FEE0MOD= 0x08
FEE0ADR= 0xFFC3
FEE0DAT= 0x3CFF
FEE0CON= 0x06;              // Mass-Erase command
while (FEE0STA & 0x04){} //Wait for command to finish
```

FEE0STA Register:

Name: FEE0STA

Address: 0xFFFF0E00

Default Value: 0x20

Access: Read Only

Function: This 8-bit read only register can be read by user code and reflect the current status of the Flash/EE memory controller.

Table 14. FEE0STA MMR bit designation

Bit	Description
15 to 4	Not Used These bits are not used and will always read as 0.
3	Flash/EE Interrupt Status Bit *Set* automatically when an interrupt occurs, i.e. when a command is complete and the Flash/EE interrupt enable bit in the FEE0MOD register is set *Cleared* automatically when the FEE0STA register is read by user code
2	Flash/EE controller busy *Set* automatically when the Flash/EE controller is busy *Cleared* automatically when the controller is not busy
1	Command Fail *Set* automatically when a command written to FEE0CONcompletes unsuccessfully *Cleared* automatically when the FEE0STA register is read by user code
0	Command Successful *Set* automatically by MCU when a command is completed successfully *Cleared* automatically when the FEE0STA register is read by user code

FEE0MOD Register:

Name: FEE0MOD

Address: 0xFFFF0E04

Default Value: 0x00

Access: Read/Write Access

Function: This register is written by user code to configure the mode of operation of the Flash/EE memory controller.

Table 15. FEE0MOD MMR Bit Designation

Bit	Description
15 to 7	Not Used These bits are reserved for future functionality and should be written as 0 by user code
6, 5	Flash/EE Security Lock Bits These bits must be written as [6,5] = 1,0 to complete the Flash/EE security protect sequence
4	Flash/EE Controller Command Complete Interrupt Enable This bit is set to 1 by user code to enable the Flash/EE controller to generate an interrupt upon completion of a Flash/EE command. This bit is 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 FEE0CON Cleared by user code to disable the Flash/EE erase and write access via FEE0CON
2	Reserved
1	Flash/EE Controller Abort Enable This bit is set to 1 by user code to enable the Flash/EE controller abort functionality.
0	Reserved

FEE0ADR Registers:

Name: FEE0ADR

Address: 0xFFFF0E10

Default Value: Non Zero. Please see the system identification section

Access: Read/Write Access

Function: This 16-bit register dictates the address upon which any Flash/EE command executed via FEE0CON acts upon.

FEE0DAT Registers:

Name: FEE0DAT

Address: 0xFFFF0E0C

Default Value: 0x0000

Access: Read/Write Access

Function: This 16-bit register contains the data either read from or to be written to the Flash/EE memory.

ADUC7030 FLASH/EE MEMORY SECURITY

The 30 kByte of Flash/EE memory available to the user can be read and write protected using the FFE0HID register.

The FEE0HID MMR protects the 30 kBytes. Bits 0-28 of this register protect Page 0 to Page 57 from writing. Each bit protects 2 pages, that is, 1 kBytes. 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 (Bit31) protects the entire Flash/EE from been 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.

Flash/EE Memory Protection Registers:

Name: FEE0HID and FEE0PRO

Address: 0xFFFF0E20 (for FEE0HID) and 0xFFFF0E1C (for FEE0PRO)

Default Value: 0xFFFFFFFF (for FEE0HID) and 0x00000000 (for FEE0PRO)

Access: Read/Write Access

Function: These registers are written by user code to configure the protection of the Flash/EE memory.

Table 16. FEE0HID and FEE0PRO MMR Bit Designations

Bit	Description
31	Read protection Cleared by user to read protect the 32-kbyte Flash/EE Block code. Set by user to allow read access to the 32-kbyte Flash/EE Block via JTAG.
30	Write Protection Bit This bit is set by user code to unprotect protect Page 59. This bit is cleared by user code write protect Page 59.
29	Write Protection Bit This bit is set by user code to unprotect Page 58. This bit is cleared by user code write protect Page 58.
28-0	Write Protection Bits When set by user code these bits unprotect Page 0 to Page 57 of the 30 kB Flash/EE code memory. Each bit write protects 2 pages and each pages consists of 512 bytes. When cleared by user code these bits will write protect pages 0-57 of the 30KB Flash/EE code memory. Each bit write protects two pages and each page consists of 512 bytes.

In Summary, there are three levels of protection:

1. Temporary protection can be set and removed by writing directly into FEE0HID MMR. This register is volatile and therefore protection is only in place while the part remains powered on. This protection is not reloaded after a power cycle.

2. Keyed permanent protection can be set via FEE0PRO to lock the protection configuration. The software key used at the start of the required FEE0PRO write sequence is saved once and must subsequently be used for any subsequent access of the FEE0HID or FEE0PRO MMRs. A mass erase sets the key back to 0xFFFF but also erases the entire user code space.

3. Permanent Protection can be set via FEE0PRO, similarly to Keyed Permanent Protection, the only difference been that the software key used is 0xDEADDEAD. Once the FEE0PRO write sequence is saved, only a mass erase sets the key back to 0xFFFFFFFF. This also erases the entire user code space.

Sequence to Write the Key and Set Permanent Protection

1. Write in FEE0PRO corresponding to the pages to be protected.
2. Write the new (user defined) 32 bit key in FEE0ADR [Bits 31-16] and FEE0DAT [Bits 15-0].
3. Write 1,0 in FEE0MOD[6:5] and set FEE0MOD[3].
4. Run the write key command 0x0C in FEE0CON.

To remove or modify the protection the same sequence can be used with a modified value of FEE0PRO.

The sequence above is illustrated in the following example, this protects writing Page 4 and Page 5 of the Flash/EE:

```
Int a = FEE0STA;                // Ensure FEE0STA is cleared
FEE0PRO=0xFFFFFFFB;             // Protect Pages 4 and 5
FEE0ADR=0x66BB;                 // 32 bit key value [Bits 31-16]
FEE0DAT=0xAA55;                 // 32 bit key value [Bits 15-0]
FEE0MOD = 0x0048                // Lock Security Sequence
FEE0CON= 0x0C;                  // Write key command
while (FEE0STA & 0x04){}        // Wait for command to finish
```

ADUC7033 FLASH/EE CONTROL INTERFACE

The access to and control of the Flash/EE memory on the ADuC7033 is managed by an on-chip memory controller. The controller manages the Flash/EE memory as two separate blocks (0 and 1).

Block 0 consists of the 32 KB Flash/EE memory mapped from 0x00090000 to 0x00097FFF (including the 2 KB kernel space which is reserved at the top of this block).

Block 1 consists of the 64 KB Flash/EE memory mapped from 0x0008 0000 to 0x0008 FFFF.

It should be noted that MCU core can continue to execute code from one memory block while an active erase or program cycle is being carried out on the other block. If a command operates on the same block as the code currently executing, the core is halted until the command is completed, this also applies to code execution.

User Code, LIN and JTAG programming use the Flash/EE Control Interface, which consists of the following MMRs :

FEExSTA (x= 0 or 1): read only register, reflects the status of the Flash/EE Control Interface

FEExMOD (x= 0 or 1): sets the operating mode of the Flash/EE Control Interface

FEExCON (x= 0 or 1): 8-bit command register. The commands are interpreted as described in Table 13.

FEExDAT (x= 0 or 1): 16-bit data register.

FEExADR (x= 0 or 1): 16-bit address register.

FEExSIG (x= 0 or 1): Holds the 24-bit code signature as a result of the signature command being initiated.

FEExHID (x= 0 or 1): Protection MMR. Controls read and write protection of the Flash/EE memory code space. If previously configured via the FEExPRO register, FEExHID may require a software key to enable access.

FEExPRO (x= 0 or 1): A buffer of the FEExHID register, which is used to store the FEExHID value, so it is automatically downloaded to the FEExHID registers on subsequent reset and power-on events.

NOTE: User Software must ensure that the Flash/EE controller has completed any Erase or Write cycle before the PLL is powered down. If the PLL is powered down before an Erase or Write cycle is completed, the Flash/EE page or byte may be corrupted.

The following sections describe in detail the bit designations of each of Flash/EE control MMRs.

FEE0CON and FEE1CON Registers:

Name: FEE0CON and FEE1CON

Address: 0xFFFF0E08 and 0xFFFF0E88

Default Value: 0x07

Access: Read/Write Access

Function: These 8-bit registers are written by user code to control the operating modes of the Flash/EE memory controllers for Block0 (32 KB) and Block1 (64 KB).

Table 17. Command Codes in FEE0CON and FEE1CON

Code	Command	Description (note x is 0 or 1 to designate Flash/EE Block 0 or 1)
0x00*	Reserved	Reserved, this command should not be written by user code
0x01*	Single Read	Load FEExDAT with the 16-bit data indexed by FEExADR
0x02*	Single Write	Write FEExDAT at the address pointed by FEExADR. This operation takes 50 μs.
0x03*	Erase-Write	Erase the page indexed by FEExADR and write FEExDAT at the location pointed by FEExADR. This operation takes 20ms
0x04*	Single Verify	Compare the contents of the location pointed by FEExADR to the data in FEExDAT. The result of the comparison is returned in FEExSTA bit 1
0x05*	Single Erase	Erase the page indexed by FEExADR
0x06*	Mass erase	Erase Block0 (30kByte) or Block1 (64kByte) of user space. The 2 kByte Kernel is protected. This operation takes 1.2 s To prevent accidental execution, a command sequence is required to execute this instruction, this is described below.
0x07		Default command.
0x08	Reserved	Reserved, this command should not be written by user code.
0x09	Reserved	Reserved, this command should not be written by user code.
0x0A	Reserved	Reserved, this command should not be written by user code.
0x0B	Signature	FEE0CON: This command will result in a 24-bit LFSR based signature been generated and loaded into FEE0SIG. If FEE0ADR is less than 0x97800, this command will result in a 24-bit LFSR based signature of the user code space from the page specified in FEE0ADR upwards, including the Kernel, security bits and Flash/EE key. If FEE0ADR is greater than 0x97800, the Kernel and manufacturing data is signed. This operation takes 120us. FEE1CON: This command will result in a 24-bit LFSR based signature been generated, beginning at FEE1ADR and ending at the end of the 63.5 k Block, and loaded into FEE1SIG. The last page of this block is not included in the Sign generation.
0x0C	Protect	This command can be run only once. The value of FEExPRO is saved and can be removed only with a mass erase (0x06) or with the key
0x0D	Reserved	Reserved, this command should not be written by user code.
0x0E	Reserved	Reserved, this command should not be written by user code.
0x0F	Ping	No operation, interrupt generated.

* The FEExCON will always read 0x07 immediately after execution of any of these commands.

Command Sequence for executing a Mass Erase

Giving the significance of the 'Mass Erase' command, a specific code sequence must be executed to initiate this operation.

5. Set bit 3 in FEExMOD.
6. Write 0xFFC3 in FEExADR
7. Write 0x3CFF in FEExDAT
8. Run the Mass Erase command 0x06 in FEExCON

This sequence is illustrated in the following example:

```
Int a = FEExSTA;          // Ensure FEExSTA is cleared
FEExMOD = 0x08
FEExADR = 0xFFC3
FEExDAT = 0x3CFF
FEExCON = 0x06;           // Mass-Erase command
while (FEExSTA & 0x04){} //Wait for command to finish
```

Note: To run the mass erase command via FEE0CON, Write protection on the lower 64kbytes must be disabled, that is, FEE1HID/FEE1PRO are set to 0xFFFFFFFF. This can be done by first removing the protection or erasing the lower 64 kBytes first.

FEE0STA and FEE1STA Registers:

Name: FEE0STA and FEE1STA

Address: 0xFFFF0E00 and 0xFFFF0E80

Default Value: 0x20

Access: Read Only

Function: These 8-bit read only registers can be read by user code and reflect the current status of the Flash/EE memory controllers.

Table 18. FEE0STA and FEE1STA MMR bit designations

Bit	Description (Note x is 0 or 1 to designate Flash/EE Block 0 or 1)
7-4	Not Used These bits are not used and always read as 0.
3	Flash/EE Interrupt Status Bit Set automatically when an interrupt occurs, that is, when a command is complete and the Flash/EE interrupt enable bit in the FEExMOD register is set Cleared automatically when the FEExSTA register is read by user code
2	Flash/EE controller busy Set automatically when the Flash/EE controller is busy Cleared automatically when the controller is not busy
1	Command fail Set automatically when a command written to FEExCON completes unsuccessfully Cleared automatically when the FEExSTA register is read by user code
0	Command Successful Set automatically by MCU when a command is completed successfully. Cleared automatically when the FEE0STA register is read by user code

FEE0ADR and FEE1ADR Registers:

Name: FEE0ADR and FEE1ADR

Address: 0xFFFF0E10 and 0xFFFF0E90

Default Value: 0x0000 (FEE1ADR). For FEE0ADR, please see Page 148

Access: Read/Write Access

Function: This 16-bit register dictates the address upon which any Flash/EE command executed via FEExCON acts upon.

FEE0DAT and FEE1DAT Registers:

Name: FEE0DAT and FEE1DAT

Address: 0xFFFF0E0C and 0xFFFF0E8C

Default Value: 0x0000

Access: Read/Write Access

Function: This 16-bit register contains the data either read from or to be written to the Flash/EE memory

FEE0MOD and FEE1MOD Registers:

Name: FEE0MOD and FEE1MOD

Address: 0xFFFF0E04 and 0xFFFF0E84

Default Value: 0x00

Access: Read/Write Access

Function: These registers are written by user code to configure the mode of operation of the Flash/EE memory controllers.

Table 19. FEE0MOD and FEE1MOD MMR bit designations

Bit	Description (note: x is 0 or 1 to designate Flash/EE Block 0 or 1)
15-7	Not Used These bits are reserved for future functionality and should be written as 0 by user code
6, 5	Flash/EE Security Lock Bits These bits must be written as [6,5] = 1,0 to complete the Flash/EE security protect sequence
4	Flash/EE Controller Command Complete Interrupt Enable This bit is set to 1 by user code to enable the Flash/EE controller to generate an interrupt upon completion of a Flash/EE command. This bit is 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 and should be written as zero
1	Flash/EE Controller Abort Enable This bit is set to 1 by user code to enable the Flash/EE controller abort functionality.
0	Reserved and should be written as zero

ADUC7033 FLASH/EE MEMORY SECURITY

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

In Block0, the FEE0HID MMR protects the 30kBytes. Bits 0-28 of this register protect pages 0-57 from writing. Each bit protects 2 pages, that is, 1 kBytes. Bits 29-30 protect pages 58 and 59 respectively, i.e. each bit write protects a single page of 512 bytes. The MSB of this register (Bit31) protects Block0 from been read via 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 Block1 (64 K), the FEE1HID MMR protects the 64kBytes. Bits 0-29 of this register protect pages 0-119 from writing. Each bit protects 4 pages, i.e. 2 kBytes. Bit30 protect pages 120-127, i.e. bit 30 write protects eight pages of 512 bytes. The MSB of this register (Bit31) protects Flash/EE Block1, from been read via JTAG.

As with Block0, FEE1PRO register mirrors the bit definitions of the FEE1HID MMR. The FEE1PRO MMR is 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.

Block0, Flash/EE Memory Protection Registers:

Name: FEE0HID and FEE0PRO

Address: 0xFFFF0E20 (for FEE0HID) and 0xFFFF0E1C (for FEE0PRO)

Default Value: 0xFFFFFFFF (for FEE0HID) and 0x00000000 (for FEE0PRO)

Access: Read/Write Access

Function: These registers are written by user code to configure the protection of the Flash/EE memory.

Table 20. FEE0HID and FEE0PRO MMR Bit Designations

Bit	Description (note: x is 0 or 1 to designate Flash/EE Block 0 or 1)
31	Read protection Cleared by user to protect the 32kbyte Flash/EE Block code via JTAG read access Set by user to allow reading the 32kbyte Flash/EE Block code via JTAG read access
30	Write Protection Bit This bit is set by user code to unprotect protect page 59 This bit is cleared by user code write protect page 59
29	Write Protection Bit This bit is set by user code to unprotect page 58 This bit is cleared by user code write protect page 58
28-0	Write Protection Bits When set by user code these bits will unprotect pages 0-57 of the 30-kByte Flash/EE code memory. Each bit write protects 2 pages and each page consists of 512 bytes. When cleared by user code these bits will write protect pages 0-57 of the 30-kByte Flash/EE code memory. Each bit write protects 2 pages and each page consists of 512 bytes.

Block1, Flash/EE Memory Protection Registers:

Name: FEE1HID and FEE1PRO

Address: 0xFFFF0EA0 (for FEE1HID) and 0xFFFF0E9C (for FEE1PRO)

Default Value: 0xFFFFFFFF (for FEE1HID) and 0x00000000 (for FEE1PRO)

Access: Read/Write Access

Function: These registers are written by user code to configure the protection of the Flash/EE memory.

Table 21. FEE1HID and FEE1PRO MMR Bit Designations

Bit	Description
31	Read protection Cleared by user to protect the 64kbyte Flash/EE Block code via JTAG read access Set by user to allow reading the 64kbyte Flash/EE Block code via JTAG read access
30	Read protection This bit write protects 8 pages and each page consists of 512 bytes. When set by user code these bits will unprotect pages 120-127 of the 64-kByte Flash/EE code memory. When cleared by user code these bits will write protect pages 120-127 of the 64-kByte Flash/EE code memory.
29 to 0	Write Protection Bits When set by user code these bits will unprotect pages 0-119 of the 64-kByte Flash/EE code memory. Each bit write protects 4 pages and each page consists of 512 bytes. When cleared by user code these bits will write protect pages 0-119 of the 64-kByte Flash/EE code memory. Each bit write protects 2 pages and each page consists of 512 bytes.

In Summary, there are three levels of protection:

1. Temporary Protection can be set and removed by writing directly into FEExHID MMR. This register is volatile and therefore protection will only be in place while the part remains powered on. This protection is not reloaded after a power cycle.

2. Keyed Permanent Protection can be set via FEExPRO which is used to lock the protection configuration. The software key used at the start of the required FEExPRO write sequence is saved once and MUST subsequently be used for any subsequent access of the FEExHID or FEExPRO MMRs. A mass erase will set the key back to 0xFFFF but will also erase the entire user code space.

3. Permanent Protection can be set via FEExPRO, similarily to Keyed Permanent Protection, the only difference being that the software key used is 0xDEADDEAD. Once the FEExPRO write sequence is saved, only a mass erase will set the key back to 0xFFFFFFFF. This will also erase the entire user code space.

Sequence to Write the Key and Set Permanent Protection:

1. Write in FEExPRO corresponding to the pages to be protected.
2. Write the new (user defined) 32 bit key in FEExADR [Bits 31-16] and FEExDAT [Bits 15-0].
3. Write 1,0 in FEExMOD[6:5] and set FEExMOD[3].
4. Run the write key command 0x0C in FEExCON.

To remove or modify the protection the same sequence can be used with a modified value of FEExPRO.

The sequence above is illustrated in the following example, this protects writing pages 4 and 5of the FLASH/EE:

```
FEExPRO=0xFFFFFFFB;    //Protect pages 4 and 5
FEExADR=0x66BB;        //32 bit key value [Bits 31-16]
FEExDAT=0xAA55;        //32 bit key value [Bits 15-0]
FEExMOD = 0x0048       // Lock Security Sequence
FEExCON= 0x0C;         // Write key command
while (FEExSTA & 0x04){}//Wait for command to finish
```

FLASH/EE MEMORY RELIABILITY

The Flash/EE memory array on the part is 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. A single endurance cycle is composed of four independent, sequential events, defined as:

1. Initial page erase sequence.
2. Read/verify sequence
3. Byte program sequence
4. Second read/verify sequence.

In reliability qualification, every half word (16-bit wide) location of the three pages (top, middle and bottom) in the Flash/EE memory is cycled 10,000 times from 0x0000 to 0xFFFF. As indicated in Table 1, the parts' Flash/EE memory endurance qualification is carried out in accordance with JEDEC Retention Lifetime Specification A117 the results allow the specification of a minimum endurance figure over supply, temperature of 10,000 cycles.

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. Again, the parts is qualified in accordance with the formal JEDEC Retention Lifetime Specification (A117) at a specific junction temperature (T_J = 85°C). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit, described previously, before data retention is characterized. This means that the Flash/EE memory is guaranteed to retain its data for its fully specified retention lifetime every time the Flash/EE memory is reprogrammed. Also note that retention lifetime, based on an activation energy of 0.6 eV, derates with T_J as shown in Figure 12.

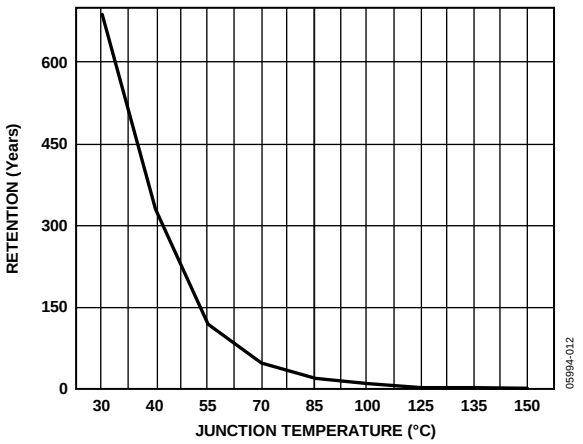


Figure 12. Flash/EE Memory Data Retention

CODE EXECUTION TIME FROM SRAM AND FLASH/EE

This chapter describes SRAM and Flash/EE access times during execution for applications where execution time is critical.

Execution from SRAM

Fetching instructions from SRAM takes one clock cycle as the access time of the SRAM is 2ns and a clock cycle is 49ns minimum. However, if the instruction involves reading or writing data to memory, one extra cycle must be added if the data is in SRAM, or three cycle if the data is in Flash/EE, one cycle to execute the instruction and two cycles to get the 32-bit data from Flash/EE. A control flow instruction, for example a branch instruction will take one cycle to fetch but also two cycles to fill the pipeline with the new instructions.

Execution from Flash/EE

Because the Flash/EE width is 16-bit, execution from Flash/EE cannot be done in one cycle, as from SRAM, when CD bit =0. Also some dead time is needed before accessing data for any value of CD bits.

In ARM mode, where instructions are 32 bits, two extra cycles are needed to fetch any instruction when CD = 0 and in Thumb mode, where instructions are 16 bits, one extra cycle is needed to fetch any instruction.

Timing is identical in both modes when executing instructions that involve using the Flash/EE for data memory. If the instruction to be executed is a control flow instruction, an extra cycle is needed to decode the new address of the program counter and then four cycles are needed to fill the pipeline. A data processing instruction involving only core register doesn't require any extra clock cycle but if it involves data in Flash/EE, an extra clock cycle is needed to decode the address of the data and two cycles to get the 32-bit data from Flash/EE. An extra cycle must also be added before fetching another instruction. Data transfer instruction are more complex and are summarized Table 22.

Table 22. Typical execution cycles in ARM/Thumb mode

Instructions	Fetch cycles	Dead time	Data access
LD	2/1	1	2
LDH	2/1	1	1
LDM/PUSH	2/1	N	$2 \times N$
STR	2/1	1	$2 \times 50\ \mu s$
STRH	2/1	1	50µs
STRM/POP	2/1	N	$2 \times N \times 50\ \mu s$

With $1 < N \leq 16$, N number of data to load or store in the multiple load/store instruction.

By default, Flash/EE code execution will be suspended during any Flash/EE erase or write cycle. A page (512 Bytes) erase cycle takes 20 ms and a write (16 bits) word command takes 50µs. However, the Flash/EE controller allows Erase/Write cycles to be aborted, if the ARM core receives an enabled interrupt during the current Flash/EE Erase/Write cycle. The ARM7 can therefore immediately service the interrupt and then return to repeat the Flash/EE command. The Abort operation will typically take 10 clock cycles. If the abort operation is not feasible, it is possible to run Flash/EE programming code and the relevant interrupt routines from SRAM, allowing the core to service the Interrupt immediately.

ADUC7030/ADUC7033 KERNEL

The ADuC7030/ADuC7033 features an on-chip Kernel resident in the top 2kBytes of the Flash/EE Code space. After any reset event, this kernel copies the factory calibrated data from the manufacturing data space, into the various on-chip peripherals. The peripherals calibrated by the Kernel are as follows:

Power Supply Monitor (PSM)
Precision, Oscillator
Low Power, Oscillator
REG_AVDD/ REG_DVDD
Low Power Voltage Reference
Normal Mode Voltage Reference
Current ADC (Offset and Gain)
Voltage/ Temperature ADC (Offset and Gain)

User MMRs that can be modified by the kernel and differ from their POR default values are as follows:

R0-R15
GP0CON/GP2CON
SYSCHK
ADCMDE/ADC0CON
FEE0ADR/FEE0CON/FEESIG
HVDAT/HVCON
HVCFG0/1
T3LD

The ADuC7030/ADuC7033 also features an On-Chip LIN downloader. The operation of this download is detailed in *ADuC7030/ADuC7033 Series Flash/EE Programming via LIN* Technote.

A flow chart showing the execution of the kernel is shown in Figure 13.

The current revision of the Kernel may be derived from SYSSER1, as described in Table 93.

For the duration of Kernel execution, the Watchdog Timer is active with a timeout period of 30ms. This ensures that if an error occurs in the Kernel, the ADuC7030/ADuC7033 will be reset. After a POR reset, the Watchdog timer is disabled once the Kernel code is exited. After any other reset, the Watchdog timer maintains user code configuration for the period of the Kernel, and is refreshed just prior to Kernel exit. A minimum watchdog period of 30ms is required. If LIN download mode is entered the Watchdog is periodically refreshed.

Normal Kernel execution time, excluding LIN Download, is approximately 5ms.

It is only possible to leave LIN download mode via a Reset.

SRAM is not modified during normal Kernel execution. SRAM is modified during LIN download Kernel execution.

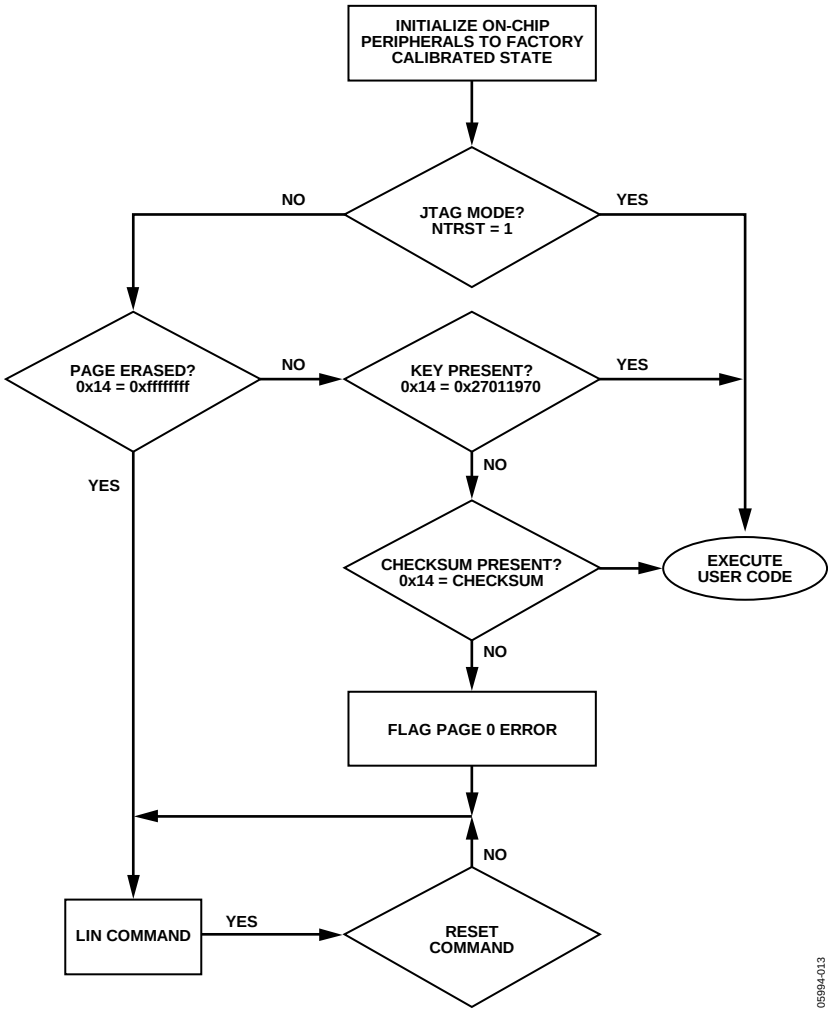


Figure 13. ADuC7030/ADuC7033 Kernel Flowchart

MEMORY MAPPED REGISTERS

The memory mapped register (MMR) space is mapped into the top 4 kBytes of the MCU memory space and accessed by indirect addressing, load and store commands, through the ARM7 banked registers. An outline of the ADuC7030/ADuC7033's Memory Mapped Register Bank 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 ARM Registers) reside in the MMR area.

As can be seen from the detailed MMR map in Table 23, the MMR data widths vary from 1 Byte (8 bits) to 4 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, will be aligned per 'little endian' format described earlier. However, errors will 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 will be written as 0's. 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.

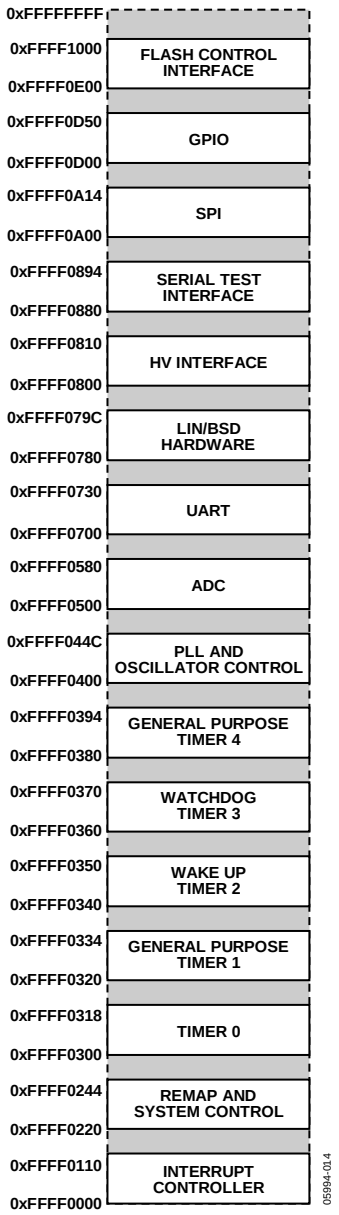


Figure 14. Top Level MMR Map

Table 23. Complete MMR List

Address	Name	Byte	Access Type	Default Value	Page	Description
IRQ address base = 0xFFFF0000						
0x0000	IRQSTA	4	R	0x00000000	80	Active IRQ Source
0x0004	IRQSIG[1]	4	R		80	Current State of all IRQ sources (Enabled and Disabled)
0x0008	IRQEN	4	RW	0x00000000	80	Enabled IRQ sources
0x000C	IRQCLR	4	W		80	MMR used to disabled IRQ Sources
0x0010	SWICFG	4	W		81	Software Interrupt Configuration MMR
0x0100	FIQSTA	4	R	0x00000000	80	Active IRQ Source
0x0104	FIQSIG[1]	4	R		80	Current State of all IRQ sources (Enabled and Disabled)
0x0108	FIQEN	4	RW	0x00000000	80	Enabled IRQ sources
0x010C	FIQCLR	4	W		80	MMR used to disabled IRQ Sources
System Control address base = 0xFFFF0200						
0x0220	SYSMAP0	1	RW		25	REMAP control Register
0x0230	RSTSTA	1	RW		26	Reset Status MMR
0x0234	RSTCLR	1	W		26	RSTSTA clear MMR
0x0238	SYSSER0[2]	4	RW		146	SYSTEM Serial Number 0
0x023C	SYSSER1[2]	4	RW		147	SYSTEM Serial Number 1
0x0240	SYSCHK[2]	4	RW		147	Kernel Checksum
Timer address base = 0xFFFF0300						
0x0300	T0LD	2	RW	0x0000	85	Timer 0 Load Register
0x0304	T0VAL0	2	R	0x0000	83	Timer 0 Value Register 0
0x0308	T0VAL1	4	R	0x00000000	83	Timer 0 Value Register 1
0x030C	T0CON	4	RW	0x00000000	84	Timer 0 Control MMR
0x0310	T0CLRI	1	W		85	Timer 0 Interrupt Clear Register
0x0314	T0CAP	2	RW	0x0000	83	Timer 0 Capture Register
0x0320	T1LD	4	RW	0x00000000	86	Timer 1 Load Register
0x0324	T1VAL	4	R	0xFFFFFFFF	86	Timer 1 Value Register
0x0328	T1CON	4	RW	0x01000000	87	Timer 1 Control MMR
0x032C	T1CLRI	1	W		86	Timer 1 Interrupt Clear Register
0x0330	T1CAP	4	R	0x00000000	87	Timer 1 Capture Register

Address	Name	Byte	Access Type	Default Value	Page	Description
0x0340	T2LD	4	RW	0x00000000	88	Timer 2 Load Register
0x0344	T2VAL	4	R	0xFFFFFFFF	88	Timer 2 Value Register
0x0348	T2CON	2	RW	0x0000	89	Timer 2 Control MMR
0x034C	T2CLRI	1	W		88	Timer 2 Interrupt Clear Register
0x0360	T3LD	2	RW	0x0040	90	Timer 3 Load Register
0x0364	T3VAL	2	R	0x0040	91	Timer 3 Value Register
0x0368	T3CON	2	RW	0x0000	91	Timer 3 Control MMR
0x036C	T3CLRI[2]	1	W		91	Timer 3 Interrupt Clear Register
0x0380	T4LD	2	RW	0x0000	92	Timer 4 Load Register
0x0384	T4VAL	2	R	0xFFFF	92	Timer 4 Value Register
0x0388	T4CON	4	RW	0x00000000	93	Timer 4 Control MMR
0x038C	T4CLRI	1	W		92	Timer 4 Interrupt Clear Register
0x0390	T4CAP	2	R	0x0000	92	Timer 4 Capture Register
PLL base address = 0xFFFF0400						
0X0400	PLLSTA	4	R		74	PLL Status MMR
0x0404	POWKEY0	4	W		75	POWCON Pre Write Key
0x0408	POWCON	1	RW	0x79	76	Power Control and Core speed Control Register
0x040C	POWKEY1	4	W		75	POWCON Post Write Key
0x0410	PLLKEY0	4	W		75	PLLCON Pre Write Key
0x0414	PLLCON	1	RW	0x00	75	PLL clock source selection MMR
0x0418	PLLKEY1	4	W		75	PLLCON Post Write Key
0x042C	OSC0TRM	1	RW	0xX8	78	Low Power Oscillator trim bits MMR.
0x0440	OSC0CON	1	RW	0x00	78	Low Power Oscillator Calibration Control MMR
0x0444	OSC0STA	1	R	0x00	79	Low Power Oscillator Calibration Status MMR
0x0448	0SC0VAL0	2	R	0x0000	79	Low Power Oscillator Calibration Counter 0 MMR
0x044C	OSC0VAL1	2	R	0x0000	79	Low Power Oscillator Calibration Counter 1 MMR
ADC address base = 0xFFFF0500						
0x0500	ADCSTA	2	R	0x0000	53	ADC Status MMR
0x0504	ADCMSKI	1	RW	0x00	55	ADC Interrupt Source Enable MMR
0x0508	ADCMDE	1	RW	0x00	55	ADC Mode Register

Address	Name	Byte	Access Type	Default Value	Page	Description
0x050C	ADC0CON	2	RW	0x0000	57	Current ADC Control MMR
0x0510	ADC1CON	2	RW	0x0000	58	V/T ADC Control MMR
0x0518	ADCFLT	2	RW	0x0007	59	ADC Filter Control MMR
0x051C	ADCCFG	1	RW	0x00	61	ADC Configuration MMR
0x0520	ADC0DAT	2	R	0x0000	62	Current ADC Result MMR
0x0524	ADC1DAT	2	R	0x0000	62	V ADC Result MMR
0x0528	ADC2DAT	2	R	0x0000	62	T ADC Result MMR
0x0530	ADC0OF[2]	2	RW		62	Current ADC Offset MMR
0x0534	ADC1OF[2]	2	RW		63	Voltage ADC Offset MMR
0x0538	ADC2OF[2]	2	RW		63	Temperature ADC Offset MMR
0x053C	ADC0GN[2]	2	RW		63	Current ADC Gain MMR
0x0540	ADC1GN[2]	2	RW		64	Voltage ADC Gain MMR
0x0544	ADC2GN[2]	2	RW		64	Temperature ADC Gain MMR
0x0548	ADC0RCL	2	RW	0x0001	64	Current ADC Result Count Limit
0x054C	ADC0RCV	2	R	0x0000	64	Current ADC Result Count Value
0x0550	ADC0TH	2	RW	0x0000	65	Current ADC Result Threshold
0x0554	ADC0TCL	1	RW	0x01	65	Current ADC Result Threshold Count Limit
0x0558	ADC0THV	1	R	0x00	65	Current ADC Result Threshold Count Limit Value
0x055C	ADC0ACC	4	R	0x00000000	65	Current ADC Result Accumulator
0x057C	ADCREF[2]	2	RW		66	Low Power Mode Voltage Reference Scaling Factor
UART base address = 0XFFFF0700						
0x0700	COMTX	1	W		118	UART Transmit Register
	COMRX	1	R	0x00	118	UART Receive Register
	COMDIV0	1	RW	0x00	118	UART Standard Baud Rate Generator Divisor Value 0
0x0704	COMIEN0	1	RW	0x00	122	UART Interrupt Enable MMR 0
	COMDIV1	1	R/W	0x00	118	UART Standard Baud Rate Generator Divisor Value 1
0x0708	COMIID0	1	R	0x01	122	UART Interrupt Identification 0
0x070C	COMCON0	1	RW	0x00	119	UART Control Register 0
0x0710	COMCON1	1	RW	0x00	120	UART Control Register 1
0x0714	COMSTA0	1	R	0x60	121	UART Status Register 0

Address	Name	Byte	Access Type	Default Value	Page	Description
0X072C	COMDIV2	2	RW	0x0000	123	UART Fractional Divider MMR
LIN Hardware Sync base address = 0XFFFF0780						
0x0780	LHSSTA	1	R	0x00	133	LHS Status MMR
0x0784	LHSCON0	2	R/W	0x0000	134	LHS Control MMR 0
0x0788	LHSVAL0	2	R/W	0x0000	136	LHS Timer 0 MMR
0x078C	LHSCON1	1	R/W	0x32	136	LHS Control MMR 1
0x0790	LHSVAL1	2	R/W	0x0000	137	LHS Timer 1 MMR
0x0794	LHSCAP	1	R	0x0000	142	LHS Capture MMR
0x0798	LHSCMP	2	R/W	0x0000	142	LHS Compare MMR
High Voltage Interface base address = 0xFFFF0800						
0x0804	HVCON	1	RW		111	High Voltage Interface Control MMR
0x080C	HVDAT	1	RW		110	High Voltage Interface Data MMR
STI base address = 0xFFFF0880						
0x0880	STIKEY0	4	W		128	STICON Pre Write Key
0x0884	STICON	2	RW	0x0000	129	Serial Test Interface Control MMR
0x0888	STIKEY1	4	W		128	STICON Post Write Key
0x088C	STIDAT0	2	RW	0x0000	128	STI Data MMR 0
0x0890	STIDAT1	2	RW	0x0000	129	STI Data MMR 1
0x0894	STIDAT2	2	RW	0x0000	129	STI Data MMR 2
SPI base address = 0xFFFF0A00						
0x0A00	SPISTA	1	R	0x00	126	SPI Status MMR
0x0A04	SPIRX	1	R	0x00	126	SPI Receive MMR
0x0A08	SPITX	1	W		126	SPI Transmit MMR
0x0A0C	SPIDIV	1	RW	0x1B	127	SPI Baud Rate Select MMR
0x0A10	SPICON	2	RW	0x00	125	SPI Control MMR
GPIO base address = 0xFFFF0D00						
0x0D 00	GP0CON	4	RW	0x11100000	96	GPIO Port 0 Control MMR
0x0D 04	GP1CON	4	RW	0x10000000	97	GPIO Port 1 Control MMR
0x0D 08	GP2CON	4	RW	0x01000000	98	GPIO Port 2 Control MMR
0x0D 20	GP0DAT[3]	4	RW	0x000000XX	99	GPIO Port 0 Data Control MMR

Address	Name	Byte	Access Type	Default Value	Page	Description
0x0D 24	GP0SET	4	W		102	GPIO Port 0 Data Set MMR
0x0D 28	GP0CLR	4	W		105	GPIO Port 0 Data Clear MMR
0x0D 30	GP1DAT[3]	4	RW	0x000000XX	100	GPIO Port 1 Data Control MMR
0x0D 34	GP1SET	4	W		103	GPIO Port 1 Data Set MMR
0x0D 38	GP1CLR	4	W		106	GPIO Port 1 Data Clear MMR
0x0D 40	GP2DAT[3]	4	RW	0x000000XX	101	GPIO Port 2 Data Control MMR
0x0D 44	GP2SET	4	W		104	GPIO Port 2 Data Set MMR
0x0D 48	GP2CLR	4	W		107	GPIO Port 2 Data Clear MMR
Flash/EE base address = 0xFFFF0E00						
0x0E00	FEE0STA	1	R	0x20	34	Flash/EE Status MMR
0x0E04	FEE0MOD	1	RW	0x00	35	Flash/EE Control MMR
0x0E08	FEE0CON	1	RW	0x07	33	Flash/EE Control MMR
0x0E0C	FEE0DAT	2	RW	0x0000	35	Flash/EE Data MMR
0x0E10	FEE0ADR	2	RW		35	Flash/EE Address MMR
0x0E18	FEE0SIG	3	R	0xFFFFFF		Flash/EE LFSR MMR
0x0E1C	FEE0PRO	4	RW	0x00000000	36	Flash/EE Protection MMR
0x0E20	FEE0HID	4	RW	0xFFFFFFFF	36	Flash/EE Protection MMR
0x0E80	FEE1STA[4]	1	R	0x20	34	Flash/EE Status MMR
0x0E84	FEE1MOD[4]	1	RW	0x00	35	Flash/EE Control MMR
0x0E88	FEE1CON[4]	1	RW	0x07	33	Flash/EE Control MMR
0x0E8C	FEE1DAT[4]	2	RW	0x0000	35	Flash/EE Data MMR
0x0E90	FEE1ADR[4]	2	RW	0x0000	35	Flash/EE Address MMR
0x0E98	FEE1SIG[4]	3	R	0xFFFFFF		Flash/EE LFSR MMR
0x0E9C	FEE1PRO[4]	4	RW	0x00000000	37	Flash/EE Protection MMR
0x0EA0	FEE1HID[4]	4	RW	0xFFFFFFFF	37	Flash/EE Protection MMR

[1] Depends on the level on the external interrupt pins GP0, GP5, GP7 and GP8
[2] Updated by Kernel
[3] Depends on the level on the external GPIO pins
[4] Only available on the ADuC7033

16-BIT Σ–Δ ANALOG TO DIGITAL CONVERTERS

The ADuC7030/ADuC7033 incorporate two independent sigma-delta ADCs namely, the current channel ADC (I-ADC) and the voltage/temperature channel ADC (V/T-ADC). These precision measurement channels integrate on-chip buffering, programmable gain amplifier, 16-bit sigma-delta 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 mean the I-ADC can be configured to accommodate battery current levels from ±1A to ±1500A

As shown in Figure 15, the I-ADC employs a sigma-delta conversion technique to realize 16 bits of no missing codes performance. The sigma-delta 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 kKHz 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 have elapsed or if the I-ADC result exceeds a programmable threshold value. A fast ADC-Over-Range 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.

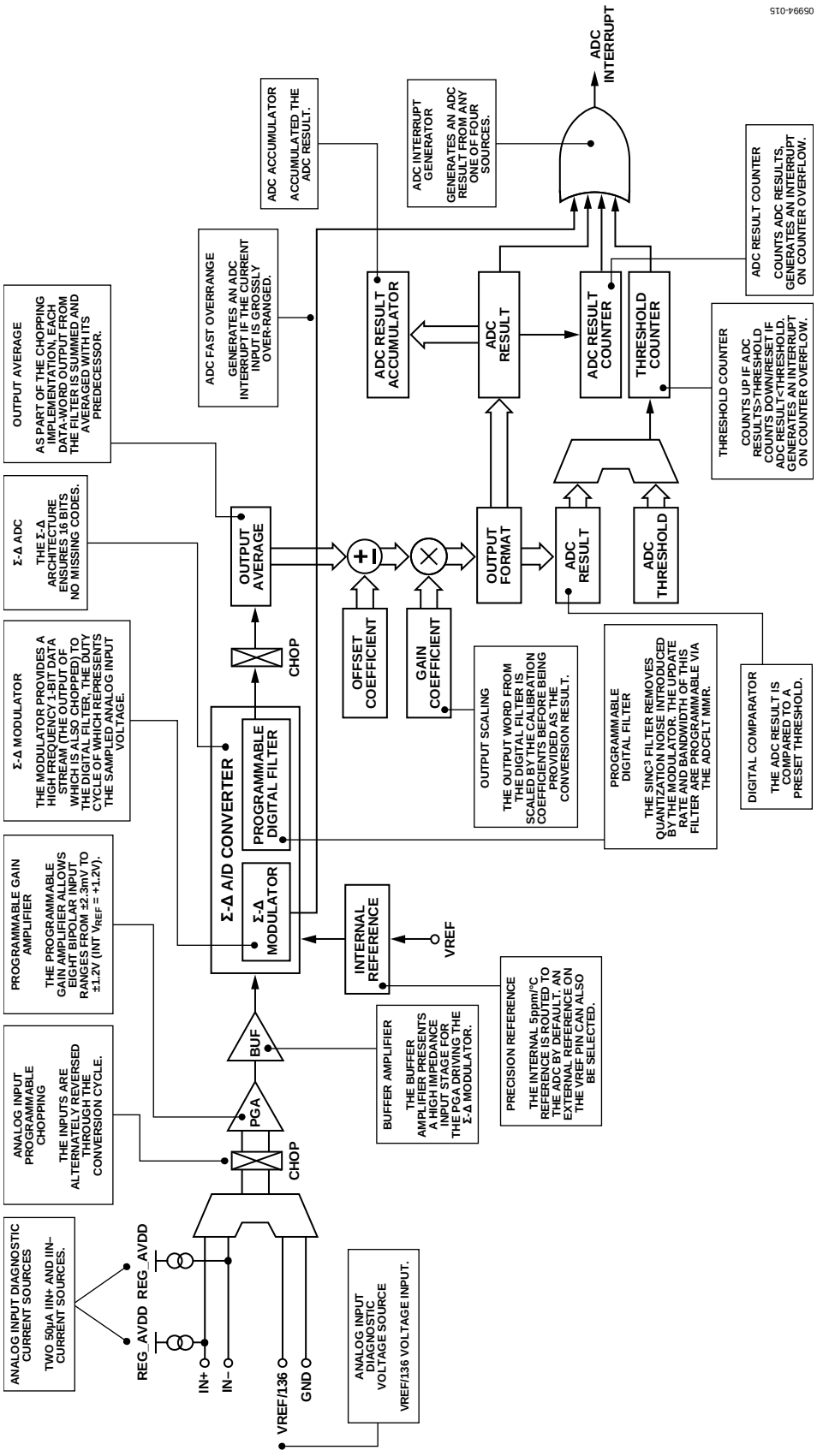


Figure 15. Current ADC, Top Level Overview

VOLTAGE/TEMPERATURE CHANNEL ADC (V/T-ADC)

This voltage/temperature channel ADC (V/T-ADC) is intended to convert additional battery parameters such as voltage and temperature. The input to this channel can be multiplexed from one of three input sources, namely external voltage, external temperature sensor circuit and on-chip temperature sensor.

As with the Current Channel ADC described previously, the V/T-ADC employs an identical sigma-delta conversion technique, including a modified Sinc3 low-pass filter to give a valid 16-Bit data conversion result at programmable output rates from 4Hz to 8 KHz. An external RC filter network is not required as this is implemented internally in the voltage channel.

The external battery voltage (VBAT) is routed to the ADC input via an on-chip high voltage, (divide by 24) resistive attenuator. This must be enabled/disabled via HVCFG1[7].

The battery temperature can be derived via the on chip temperature sensor or an external temperature sensor input. The time to a first valid (fully settled) result after an input channel switch on the voltage/temperature channel is three ADC conversion cycles with chop mode turned off.

This ADC is again buffered but unlike the current channel has a fixed input range of 0V to V_{REF} on VTEMP+ and 0V to 28.8V on VBAT (assuming an internal 1.2V reference). A top level overview of this ADC signal chain is shown in Figure 16.

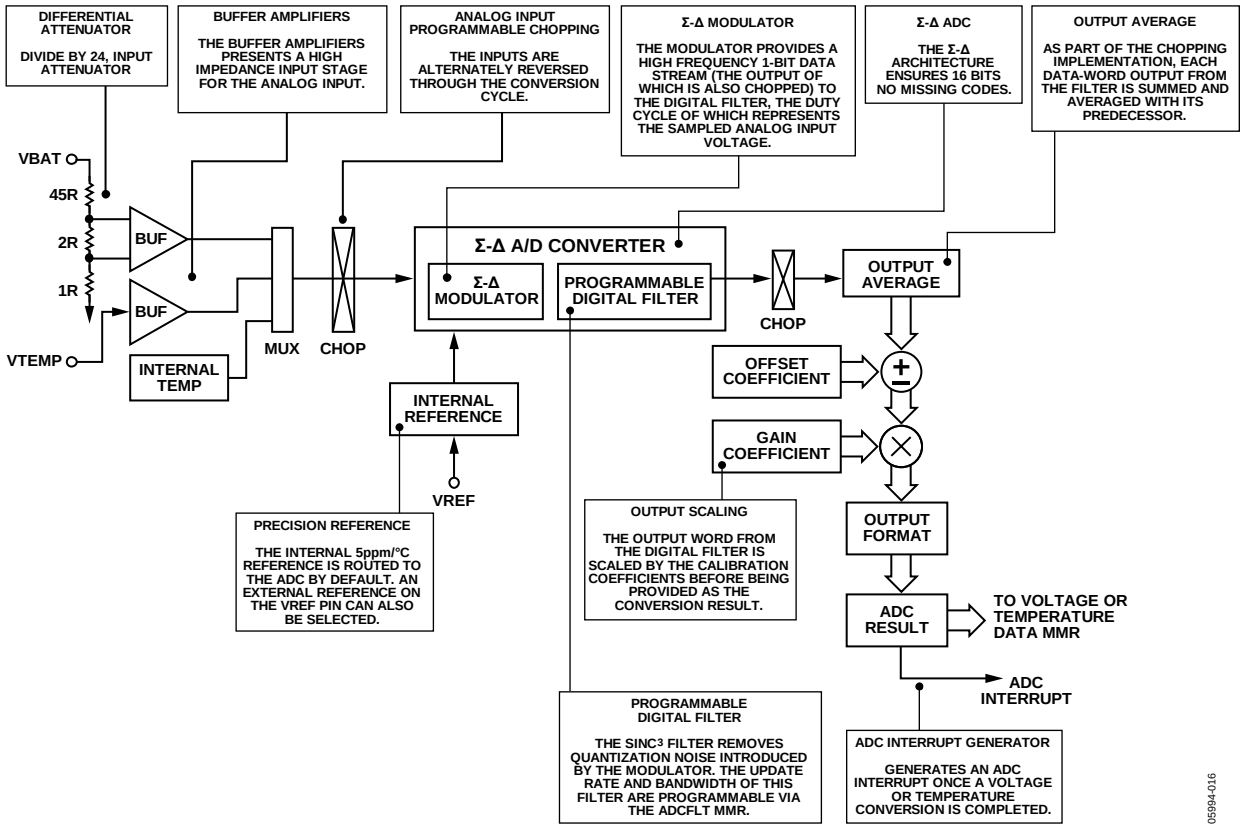


Figure 16. Voltage/ Temperature ADC, Top Level Overview

ADC GROUND SWITCH

The ADuC7030/ADuC7033 features an integrated ground switch pin, GND_SW, pin15. This switch allows the user to dynamically disconnect ground from external devices. It allows either a direct connection to ground, or a connection to ground via a 20 kΩ. This additional resistor may be used to reduce the number of external components required for an NTC circuit.

The ground switch feature may be used for reducing power consumption on application specific boards.

An example application is shown in Figure 17. This diagram shows an external NTC used in two modes, one using the internal 20 kΩ resistor, and the second showing a direct connection to ground, via the GND_SW. ADCCFG[7] controls the connection of the ground switch to ground and ADCMDE[6] controls the GND_SW resistance.

The possible combinations are shown in Table 24.

Table 24. GND_SW Configuration

ADCCFG[7]	ADCMDE[6]	GND_SW
0	0	Floating
0	1	Floating
1	0	Direct connection to Ground
1	1	Connected to ground via 20 kΩ resistor

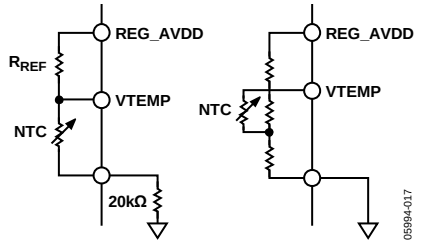


Figure 17. Example External Temperature Sensor Circuits

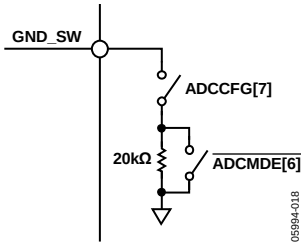


Figure 18. Internal Ground Switch Configuration

ADC NOISE PERFORMANCE TABLES

Table 25, Table 26 and Table 27 below show the output RMS noise in μV for some typical output update rates on the I and V/T ADCs. The numbers are typical and are generated at a differential input voltage of 0 V. The output RMS noise is specified as the standard deviation (or 1 X Sigma) of the distribution of ADC output codes collected when the ADC input voltage is at a dc voltage. It is expressed as V RMS.

Table 25. Current Channel ADC, Normal Power Mode, Typical Output RMS Noise (μV)

		ADC Input Range									
ADCFLT	Data Update Rate	±2.3mV (512)	±4.6mV (256)	±4.68mV (128)	±18.75mV (64)	±37.5mV (32)	±75mV (16)	±150mV (8)	±300mV (4*)	±600mV (2*)	±1.2V (1*)
0xBF1D	4 Hz	0.040	0.040	0.043	0.045	0.087	0.175	0.35	0.7	1.4	2.8
0x961F	10 Hz	0.060	0.060	0.060	0.065	0.087	0.175	0.35	0.7	1.4	2.8
0x007F	50 Hz	0.142	0.142	0.144	0.145	0.170	0.305	0.380	0.7	2.3	2.8
0x0007	1 kHz	0.620	0.620	0.625	0.625	0.770	1.310	1.650	2.520	7.600	7.600
0x0000	8 kHz	2.000	2.000	2.000	2.000	2.650	4.960	8.020	15.0	55.0	55.0

**Please note that the maximum absolute input voltage allowed is -200mV to 300mV relative to ground*

Table 26. Voltage Channel ADC, Typical Output RMS Noise (referred to ADC Voltage attenuator Input)(μV)

ADCFLT	Data Update Rate	28.8V ADC Input Range
0xBF1D	4Hz	65
0x961F	10Hz	65
0x0007	1KHz	180
0x0000	8KHz	1600

Table 27. Temperature Channel ADC, Typical Output RMS Noise (μV)

ADCFLT	Data Update Rate	0 to 1.2 V ADC Input Range
0xBF1D	4 Hz	2.8
0x961F	10 Hz	2.8
0x0007	1 kHz	7.5
0x0000	8 kHz	55

ADC MMR INTERFACE

The ADC is controlled and configured via a number of MMRs that are described in detail in the following pages:

ADC Status Register:

Name: ADCSTA

Address: 0xFFFF0500

Default Value: 0x0000

Access: Read Only

Function: This read only register holds general status information related to the mode of operation or current status of the ADuC7030/ADuC7033 ADCs.

Table 28. ADCSTA MMR Bit Designations

Bit	Description
15	ADC Calibration Status This bit is set automatically in hardware to indicate an ADC calibration cycle has been completed. This bit is cleared after ADCMDE is written to.
14	ADC Temperature Conversion Error This bit is set automatically in hardware to indicate that a temperature conversion over-range or under-range has occurred. The conversion result will be clamped to negative full-scale (under-range error) or positive full-scale (over-range error) in this case. This bit will be cleared when a valid (in-range) temperature conversion result is written to the ADC2DAT register.
13	ADC Voltage Conversion Error This bit is set automatically in hardware to indicate that a voltage conversion over-range or under-range has occurred. The conversion result will be clamped to negative full-scale (under-range error) or positive full-scale (over-range error) in this case. This bit will be cleared when a valid (in-range) voltage conversion result is written to the ADC1DAT register.
12	ADC Current Conversion Error This bit is set automatically in hardware to indicate that a current conversion over-range or under-range has occurred. The conversion result will be clamped to negative full-scale (under-range error) or positive full-scale (over-range error) in this case. This bit will be cleared when a valid (in-range) current conversion result is written to the ADC0DAT register.
11-5	Not Used This bit is reserved for future functionality and should not be monitored by user code
4	Current Channel ADC Comparator Threshold This bit is only valid if the Current Channel ADC comparator is enabled via the ADCCFG MMR. This bit is set by hardware if the absolute value of the I-ADC conversion result exceeds the value written in the ADC0TH MMR. If the ADC threshold counter is used (ADC0TCL), this bit is only set once the specified number of I-ADC conversions equals the value in the ADC0THV MMR.
3	Current Channel ADC Over-Range Bit If the Over-Range Detect function is enabled via the ADCCFG MMR, this bit is set by hardware if the I-ADC input is grossly (>30% approx.) over-ranged. This bit is updated every 125µsecs. Once set, this bit can only be cleared by software when ADCCFG[2] is cleared to disable the function, or the ADC gain is changed via the ADC0CON MMR.
2	Temperature Conversion Result Ready Bit If the Temperature Channel ADC is enabled, this bit is set by hardware as soon as a valid temperature conversion result is written in the temperature data register (ADC2DAT MMR). It is also set at the end of a calibration This bit is cleared by reading either ADC2DAT or ADC0DAT.
1	Voltage Conversion Result Ready Bit If the Voltage Channel ADC is enabled, this bit is set by hardware as soon as a valid voltage conversion result is written in the voltage data register (ADC1DAT MMR) It is also set at the end of a calibration This bit is cleared by reading either ADC1DAT or ADC0DAT.
0	Current Conversion Result Ready Bit If the Current Channel ADC is enabled, this bit is set by hardware as soon as a valid current conversion result is written in the current data register (ADC0DAT MMR) It is also set at the end of a calibration This bit is cleared by reading ADC0DAT.

NOTES

1. All bits defined in the top 8 MSBs (bits 8–15) of the MMR are used as flags only and will not generate interrupts

2. All bits defined in the lower 8 LSBs (bits 0-7) of this MMR are logic OR'ed to produce a single ADC interrupt to the MCU core.

3. In response to an ADC interrupt, user code should interrogate the ADCSTA MMR to determine the source of the interrupt.

4. Each ADC interrupt source can be individually masked via the ADCMSKI MMR described below

5. All ADC Result Ready bits are cleared by a read of the ADC0DAT MMR. If the Current Channel ADC is not enabled, all ADC Result Ready bits are cleared by a read of the ADC1DAT or ADC2DAT MMRs.

6. To ensure that I-ADC and V/T-ADC conversion data are synchronous, user code should first read the ADC1DAT MMR and then ADC0DAT MMR.

7. New ADC conversion results will not be written to the ADCxDAT MMRs unless the respective ADC Result Ready bits are first cleared. The only exception to this rule is data conversion result updates when the ARM core is powered down. In this mode, ADCxDAT registers will always contain the most recent ADC conversion result even though the Ready bits have not been cleared.

ADC Interrupt Mask Register:

Name: ADCMSKI

Address: 0xFFFF0504

Default Value: 0x00

Access: Read/Write

Function: This register allows the ADC interrupt sources to be enabled individually. The bits position in this register are the same as the lower 8-bits in the ADCSTA MMR. If a bit is set by user code to a '1', the respective interrupt is enabled. By default all bits are '0' meaning all ADC interrupt sources are disabled.

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 sub-system.

Table 29. ADCMDE MMR Bit Designations

Bit	Description
7	Not Used This bit is reserved for future functionality and be written as 0 by user code
6	20KΩ resistor select: This bit is set to 1 to select the 20 KΩ resistor as shown in Figure 18 This bit is set to 0 to select the direct path to ground as shown in Figure 18 (Default).
5	Low Power Mode Reference Select: This bit is set to 1 to enable the Precision Voltage Reference in either Low Power Mode or Low Power Plus Mode. This will increase current consumption. This bit is set to 0 to enable the Low Power Voltage Reference in either Low Power Mode or Low Power Plus Mode (Default).
4-3	ADC Power Mode Configuration 0, 0 ADC Normal Mode If enabled, the ADC will operate with normal current consumption yielding optimum electrical performance 0, 1 ADC Low Power Mode If enabled, the I-ADC will operate with reduced current consumption. This limitation is current consumption is achieved, (at the expense of ADC noise performance) by fixing the gain to 128 and using the on-chip low power (131kHz) oscillator to drive the ADC circuits directly. 1, 0 ADC Low Power-Plus Mode If enabled, the ADC will again operate with reduced current consumption. In this mode, the gain is fixed to 512 and the current consumed is 200uA (approx.) more than ADC low Power Mode above. The additional current consumed also ensures ADC noise performance is better than that achieved in ADC Low Power Mode. 1, 1 Not Defined
2-0	ADC Operation Mode Configuration 0, 0, 0 ADC Power-Down Mode All ADC circuits (including internal reference) are powered-down 0, 0, 1 ADC Continuous Conversion Mode In this mode, any enabled ADC will continuously convert. 0, 1, 0 ADC Single Conversion Mode In this mode, any enabled ADC will perform a single conversion. The ADC will enter Idle Mode once the single shot conversion is complete. A single conversion will take 2/3 ADC clock cycles depending on the CHOP mode. 0, 1, 1 ADC IDLE Mode In this Mode, the ADC is fully powered on but is held in RESET 1, 0, 0 ADC Self-Offset Calibration In this mode, an offset calibration is performed on any enabled ADC using an internally generated 0V. The calibration is carried out at the user programmed ADC settings, therefore, as with a normal single ADC conversion, it will take 2/3 ADC conversion cycles before a fully settled calibration result is ready. The calibration result is automatically written to the

Bit	Description	
		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.
	1, 0, 1	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 2-stage process and 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 pre-programmed, factory calibration coefficients (downloaded automatically from internal Flash/EE) should be used for voltage temperature measurements. If an external NTC is used, an ADC Self Calibration should be done on the temperature channel.
	1, 1, 0	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 will take 3 ADC conversion cycles before a fully settled calibration result is ready.
	1, 1, 1	ADC System Full-Scale Calibration In this mode, a full-scale calibration is performed on enabled ADC channels against an external full-scale voltage driven at the ADC input pins.

Current Channel ADC Control Register:

Name: ADC0CON

Address: 0xFFFF050C

Default Value: 0x0000

Access: Read/Write

Function: The Current Channel ADC Control MMR is a 16-bit register that is used to configure the I-ADC.

Note: If the Current ADC is reconfigured via ADC0CON, the Voltage and Temperature ADCs are also reset.

Table 30. ADC0CON MMR Bit Designations

Bit	Description
15	Current Channel ADC Enable This bit is set to 1 by user code to enable the I-ADC Clearing this bit to 0, powers down the I-ADC and resets the respective ADC READY bit in the ADCSTA MMR to 0
14, 13	IIN Current Source Enable 0, 0 Current Sources Off 0, 1 Enable 50 µA current source on IIN+ 1, 0 Enable 50 µ A current source on IIN- 1, 1 Enable 50 µ A current source on both IIN- and IIN+
12– 10	Not Used These bits are reserved for future functionality and should be written as zero
9	Current Channel ADC Output Coding This bit is set to 1 by user code to configure I-ADC output coding as unipolar This bit is cleared to 0 by user code to configure I-ADC output coding as 2's complement
8	Not Used These bits are reserved for future functionality and should be written as zero
7, 6	Current Channel ADC Input Select 0, 0 IIN+, IIN- 0, 1 IIN-, IIN- Diagnostic, internal short configuration 1, 0 V_{REF}/136, 0V Diagnostic, test voltage for gain settings <= 128 Note: If (REG_AVDD, AGND) divided by 2 Reference is selected, REG_AVDD is used for V_{REF} in this mode. This will lead to ADC0DAT scaled by two 1, 1 Not Defined
5, 4	Current Channel ADC Reference Select 0, 0 Internal, 1.2V precision reference selected. In ADC Low Power Mode, the Voltage Reference selection is controlled by ADCMDE[5] 0, 1 External reference inputs (VREF, GND_SW) selected 1, 0 External reference inputs divided by 2 (VREF, GND_SW)/2 selected, this allows an external reference up to REG_AVDD 1, 1 (REG_AVDD, AGND) divided by 2 selected
3 - 0	Current Channel ADC Gain Select (note, nominal I-ADC Full-scale Input Voltage = (V_{REF}/GAIN) 0, 0, 0, 0 I-ADC Gain =1 0, 0, 0, 1 I-ADC Gain =2 0, 0, 1, 0 I-ADC Gain =4 0, 0, 1, 1 I-ADC Gain =8 0, 1, 0, 0 I-ADC Gain =16 0, 1, 0, 1 I-ADC Gain =32 0, 1, 1, 0 I-ADC Gain =64 0, 1, 1, 1 I-ADC Gain =128 1, 0, 0, 0 I-ADC Gain =256 1, 0, 0, 1 I-ADC Gain =512 1, x, x, x I-ADC Gain is undefined

Voltage/Temperature Channel ADC Control Register:

Name: ADC1CON

Address: 0xFFFF0510

Default Value: 0x0000

Access: Read/Write

Function: The Voltage/Temperature Channel ADC Control MMR is a 16-bit register that is used to configure the V/T-ADC.

Note: When enabling/disabling the Voltage/Temperature ADC, the Voltage Attenuator must also be enabled/disabled via HVCFG1[7].

Table 31. ADC1CON MMR Bit Designations

Bit	Description
15	Voltage/Temperature Channel ADC Enable This bit is set to 1 by user code to enable the V/T-ADC. When enabling/disabling the Voltage channel, the Voltage Attenuator must also be enabled/disabled via HVCFG1[7] if measuring battery voltage. Clearing this bit to 0, powers down the V/T-ADC.
14, 13	VTEMP Current Source Enable 0, 0 Current Sources Off 0, 1 Enable 50µA current source on VTEMP+ 1, 0 Enable 50µA current source on GND_SW 1, 1 Enable 50µA current source on both VTEMP+ and GND_SW
12– 10	Not Used. These bits are reserved for future functionality and should not be modified by user code
9	Voltage/Temperature Channel ADC Output Coding This bit is set to 1 by user code to configure V/T-ADC output coding as unipolar This bit is cleared to 0 by user code to configure V/T -ADC output coding as 2's complement
8	Not Used. This bit is reserved for future functionality and should be written as 0 by user code
7, 6	Voltage/Temperature Channel ADC Input Select 0, 0 VBAT/24, AGND VBAT attenuator selected 0, 1 VTEMP, GND_SW External Temperature Input selected, conversion result written to ADC2DAT 1, 0 Internal Sensor Internal Temperature Sensor Input selected, conversion result written to ADC2DAT 1, 1 Internal Short Shorted Input
5, 4	Voltage Temperature Channel ADC Reference Select 0, 0 Internal, 1.2V precision reference selected. 0, 1 External reference inputs (VREF, GND_SW) selected. 1, 0 External reference inputs divided by 2 (VREF, GND_SW)/2 selected. This allows an external reference up to REG_AVDD 1, 1 (REG_AVDD, AGND) / 2 selected for the Voltage Channel. (REG_AVDD, GND_SW) / 2 selected for the Temperature Channel.
3 – 0	Not Used. These bits are reserved for future functionality and should not be written as 0 by user code

ADC Filter Register:

Name: ADCFLT

Address: 0xFFFF0518

Default Value: 0x0007

Access: Read/Write

Function: The ADC Filter MMR is an 16-bit register that controls the speed and resolution of the on-chip ADCs.

Note: If ADCFLT is modified, the Current and Voltage/Temperature ADCs are reset.

Table 32. 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 will have very low offset errors and drift but the ADC output rate will be reduced by a factor of 3 if AF=0 (see Sinc3 Decimation Factor bits below). If AF > 0, then ADC output update rate will be the same with chop on or off. When chop is enabled, the settling time is 2 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) does not reduce ADC output rate but will increase the settling time by 1 conversion period. Cleared by user to disable the running average function.
13 - 8	Averaging Factor (AF) The value written to these bits is use to implement a programmable 1st order Sinc post filter. The averaging factor can further reduce ADC noise at the expense of output rate as described in Sinc Decimation Factor bits below.
7	Sinc3 Modify Set by user to modify the standard Sinc3 frequency response to increase the filter stopband rejection by 5dBs approx. This is achieved by inserting a second notch (NOTCH2) at F_{NOTCH2} = 1.333 * F_{NOTCH} where F_{NOTCH} is the location of the 1st notch in the response.
6 – 0	Sinc3 Decimation Factor (SF) The value (SF) written in these bits controls the over sampling (decimation factor) of the Sinc3 filter. The output rate from the Sinc3 filter is given by F_{ADC} = (512,000 / ([SF+1] X 64)) Hz when the CHOP bit (bit#15 above) = 0 and AF=0 (note AF = Averaging Factor) Note : this is valid for all SF values <= 125 For SF= 126, F_{ADC} is forced to 60Hz For SF= 127, F_{ADC} is forced to 50Hz For information on calculating the F_{ADC} for SF (other than 126 and 127) and AF values please refer to Table 33. Notes: - Due to limitations on the digital filter internal data-path, 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 in Normal Power Mode to 4Hz or 1Hz in Low Power Mode. - In low power mode and low power-plus mode, the ADC is driven directly by the low power oscillator (131KHz) and not 512KHz. All F_{ADC} calculations should be divided by 4 (approx).

Table 33. ADC Conversion Rates and Settling Times

Chop Enabled	Averaging Factor	Running Average	F_{ADC}	*$T_{Settling}$
No	No	No	$\frac{512000}{[SF+1]*64}$	$\frac{3}{F_{ADC}}$
No	No	Yes	$\frac{512000}{[SF+1]*64}$	$\frac{4}{F_{ADC}}$
No	Yes	No	$\frac{512000}{[SF+1]*64*[3+AF]}$	$\frac{1}{F_{ADC}}$
No	Yes	Yes	$\frac{512000}{[SF+1]*64*[3+AF]}$	$\frac{2}{F_{ADC}}$
Yes	N/A	N/A	$\frac{512000}{[SF+1]*64*[3+AF]}$	$\frac{2}{F_{ADC}}$

*An additional time of approximately 60µs per ADC is required before the first ADC is available.

Table 34. Allowable Combinations of SF and AF

AF Range / SF	0	1 to 7	8 to63
0-31	✓	✓	✓
32-63	✓	✓	✗
64-127	✓	✗	✗

ADC Configuration Register:

Name: ADCCFG

Address: 0xFFFF051C

Default Value: 0x00

Access: Read/Write

Function: The 8-bit ADC Configuration MMR controls extended functionality related to the on-chip ADCs.

Table 35. ADCCFG MMR Bit Designations

Bit	Description
7	Analog Ground Switch Enable This bit is set to '1' by user software to connect the external GND_SW pin (pin#15) to an internal analog ground reference point. This bit can be used to connect and disconnect external circuits and components to ground under program control and thereby minimize dc current consumption when the external circuit or component is not being used. This bit is used in conjunction with ADCMDE[6] to select a 20KΩ resistor to ground.
6, 5	Current Channel (32-bit) Accumulator Enable 0, 0 Accumulator Disabled and reset to 0 0, 1 Accumulator Active Positive current values are added to accumulator total, accumulator can overflow if allowed run for > 65535 conversions Negative current values are subtracted from accumulator total, accumulator is clamped to a minimum value of 0 1, 0 Accumulator Active Positive current values are added to accumulator total, accumulator can overflow if allowed run for > 65535 conversions The absolute values of Negative current are subtracted from accumulator total, accumulator in this mode will continue to accumulate negatively, below 0 1, 1 Not Defined.
4, 3	Current Channel ADC Comparator Enable 0, 0 Comparator Disabled 0, 1 Comparator Active, Interrupt asserted if absolute value of I-ADC conversion result |I| >= ADC0TH 1, 0 Comparator-Count Mode Active, Interrupt asserted if absolute value of an I-ADC conversion result |I| >= ADC0TH for #ADC0TCL conversions. A conversion value |I| < ADC0TH will reset the threshold counter value (ADC0THV) to 0 1, 1 Comparator-Count Mode Active, Interrupt asserted if absolute value of an I-ADC conversion result |I| >= ADC0TH for #ADC0TCL conversions. A conversion value |I| < ADC0TH will decrement the threshold counter value (ADC0THV) towards 0.
2	Current Channel ADC Over Range Enable Set by user to enable a 'coarse' comparator on the Current Channel ADC. If the current reading is grossly (>30% approx.) over-ranged for the active gain setting, then the over range bit in the ADCSTA MMR is set. The current must be outside this range for greater than 125µsecs for the flag to be set. This feature should not be used in ADC Low Power Mode
1	Not Used This bit is reserved for future functionality and be written as 0 by user code
0	Current Channel ADC, Result Counter Enable Set by user to enable the result count mode. In this mode an I-ADC interrupt will only be generated when ADC0RCV=ADC0RCL. This allows the I-ADC to continuously monitor current but only interrupt the MCU core after a defined number of conversions. The Voltage/Temperature ADC will also continue to convert if enabled but again only the last conversion result will be available (intermediate V/T-ADC conversion results are not stored) when the ADC counter interrupt occurs

Current Channel ADC Data Register:

Name: ADC0DAT

Address: 0xFFFF0520

Default Value: 0x0000

Access: Read Only

Function: This ADC Data MMR holds the 16-bit conversion result from the I-ADC. The ADC will not update this MMR if the ADC0 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 ADC Data MMR holds the 16-bit Voltage conversion result from the V/T-ADC. The ADC will 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 ADC Data MMR holds the 16-bit temperature conversion result from the V/T-ADC. The ADC will not update this MMR if the Temperature Conversion Result READY bit (ADCSTA[2]) is set. If I-and V ADC is not active, a read of this MMR by the MCU clears all asserted READY flags (ADCSTA[2]). A ready of this MMR will clear ADCSTA[2].

Current Channel ADC Offset Calibration Register:

Name: ADC0OF

Address: 0xFFFF0530

Default Value: Part Specific, factory programmed

Access: Read/Write Access

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 will be automatically overwritten if an offset calibration of the I-ADC is initiated by the user via bits in the ADCMDE MMR. User code can only write to this calibration register if the ADC is in idle mode. An ADC must be enabled and in idle mode before written to any Offset or Gain Register. The ADC must be in idle mode for at least 23µs.

Voltage Channel Offset Calibration Register:

Name: ADC1OF

Address: 0xFFFF0534

Default Value: Part Specific, factory programmed

Access: Read/Write Access

Function: This 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 will be automatically overwritten if an offset calibration of the voltage channel is initiated by the user via bits in the ADCMDE MMR. User code can only write to this calibration register if the ADC is in idle mode. An ADC must be enabled and in idle mode before written to any Offset or Gain Register. The ADC must be in idle mode for at least 23µs.

Temperature Channel Offset Calibration Register:

Name: ADC2OF

Address: 0xFFFF0538

Default Value: Part Specific, factory programmed

Access: Read/Write

Function: This 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 will be automatically overwritten if an offset calibration of the temperature channel is initiated by the user via bits in the ADCMDE MMR. User code can only write to this calibration register if the ADC is in idle mode. An ADC must be enabled and in idle mode before written to any Offset or Gain Register. The ADC must be in idle mode for at least 23µs.

Current Channel ADC Gain Calibration Register:

Name: ADC0GN

Address: 0xFFFF053C

Default Value: Part Specific, factory programmed

Access: Read/Write

Function: This Gain MMR holds a 16-bit gain calibration coefficient for scaling the I-ADC conversion result. The register is configured at power-on with a factory default value. However, this register will be automatically overwritten if a gain calibration of the I-ADC is initiated by the user via bits in the ADCMDE MMR. User code can only write to this calibration register if the ADC is in idle mode. An ADC must be enabled and in idle mode before written to any Offset or Gain Register. The ADC must be in idle mode for at least 23µs.

Voltage Channel Gain Calibration Register:

Name: ADC1GN

Address: 0xFFFF0540

Default Value: Part Specific, factory programmed

Access: Read/Write

Function: This Gain MMR holds a 16-bit gain calibration coefficient for scaling a voltage channel conversion result. The register is configured at power-on with a factory default value. However, this register will be automatically overwritten if a gain calibration of the voltage channel is initiated by the user via bits in the ADCMDE MMR. User code can only write to this calibration register if the ADC is in idle mode. An ADC must be enabled and in idle mode before written to any Offset or Gain Register. The ADC must be in idle mode for at least 23µs.

Temperature Channel Gain Calibration Register:

Name: ADC2GN

Address: 0xFFFF0544

Default Value: Part Specific, factory programmed

Access: Read/Write

Function: This Gain MMR holds a 16-bit gain calibration coefficient for scaling a temperature channel conversion result. The register is configured at power-on with a factory default value. However, this register is automatically overwritten if a gain calibration of the temperature channel is initiated by the user via bits in the ADCMDE MMR. User code can only write to this calibration register if the ADC is in idle mode. An ADC must be enabled and in idle mode before written to any Offset or Gain Register. The ADC must be in idle mode for at least 23µs.

Current Channel ADC Result Counter Limit Register:

Name: ADC0RCL

Address: 0xFFFF0548

Default Value: 0x0001

Access: Read/Write

Function: This 16-bit MMR sets the number of conversions required before an ADC interrupt is generated. By default this register is set to 0x01. The ADC counter function must be enabled via the ADC Result Counter Enable bit in the ADCCFG MMR.

Current Channel ADC Result Count Register:

Name: ADC0RCV

Address: 0xFFFF054C

Default Value: 0x0000

Access: Read Only

Function: This 16-bit, Read Only MMR holds the current number of I-ADC conversion results. It is used in conjunction with ADC0RCL to mask I-ADC interrupts, generating a lower interrupt rate. Once ADC0RCV=ADC0RCL, the value is ADC0RCV resets to 0 and recommences counting. It can also be used in conjunction with the Accumulator (ADC0ACC) to allow an average current calculation to be undertaken. The result counter is enabled via ADCCFG[0]. This MMR is also reset to 0 when the I-ADC is reconfigured i.e. when the ADC0CON or ADCMDE are written.

Current Channel ADC Threshold Register:

Name: ADC0TH

Address: 0xFFFF0550

Default Value: 0x0000

Access: Read/Write

Function: This 16-bit MMR sets the threshold against which the absolute value of the I-ADC conversion result is compared. In Unipolar mode ADC0TH [15:0] are compared and in 2's complement mode ADC0TH[14:0] are compared.

Current Channel ADC Threshold Count Limit Register:

Name: ADC0TCL

Address: 0xFFFF0554

Default Value: 0x01

Access: Read/Write

Function: This 8-bit MMR determines how many cumulative (given values below the threshold will decrement or reset the count to 0) I-ADC conversion result readings above ADC0TH must occur before the I-ADC Comparator Threshold bit is set in the ADCSTA MMR generating an ADC interrupt. The I-ADC Comparator Threshold bit is asserted as soon as the ADC0THV=ADC0TCL.

Current Channel ADC Threshold Count Register:

Name: ADC0THV

Address: 0xFFFF0558

Default Value: 0x00

Access: Read Only

Function: This 8-bit MMR is incremented every time the absolute value of an I-ADC conversion result |I| >= ADC0TH. This register is decremented or reset to 0 every time the absolute value of an I-ADC conversion result |I| < ADC0TH. The configuration of this function is enabled via the Current Channel ADC Comparator bits in the ADCCFG MMR.

Current Channel ADC Accumulator Register:

Name: ADC0ACC

Address: 0xFFFF055C

Default Value: 0x00000000

Access: Read Only

Function: This 32-bit MMR holds the current accumulator value. The I-ADC READY bit in the ADCSTA MMR should be used to determine when it is safe to read this MMR. The MMR value is reset to 0 by disabling the accumulator in the ADCCFG MMR or reconfiguring the Current Channel ADC.

Low Power Voltage Reference Scaling Factor

Name: ADCREF

Address: 0xFFFF057C

Default Value: Part Specific, factory programmed

Access: Read/Write. Care should be taken not to write to this register.

Function: This allows user code to correct for the initial error of the LPM reference. 0x8000 corresponds to no error when compared to the Normal Mode Reference. The magnitude of the ADC result should be multiplied by the value in ADCREF and divided by 0x8000 to compensate for the actual value of the low power reference

If the LPM Voltage Reference is 1% below 1.200V, then the value of ADCREF will be approximately 0x7EB9

If the LPM Voltage Reference is 1% above 1.200V, then the value of ADCREF will be approximately 0x8147

This register corrects the effective value of the LPM reference at the temperature the reference is measured at during ADI's production flow, which is 25°C. There is no change to the Temperature-Coefficient of the LPM reference when using the ADCREF MMR..

This register should not be used if the Precision Reference is being used in Low power mode (if ADCMDE[5] is set.)

ADC POWER MODES OF OPERATION

The ADCs can be configured into various reduced or full 'power' modes of operation by configuring ADCMDE[4:3] as appropriate. The ARM7 MCU can itself also be configured in Low Power modes of operation (POWCON[5:3]). The core power modes are independently controlled and are not related to the ADC power modes described here. The ADC power modes of operation are described in more detail below.

ADC Startup Procedure

Prior to beginning converting, the following procedure should be followed.

1. Configure the Current ADC, ADC0, into Low-Power-Mode (ADC0CON = 0x8007; ADCMDE = 0x09)
2. Delay for 200µs.
3. Switch the Current ADC, ADC0, into Idle-Mode (ADCMDE = 0x03), keeping ADC0CON unchanged. If the Voltage or Temperature channels are to be used, they should be enabled here.
4. Delay for 1ms
5. Switch ADCMDE to desired mode, for example. ADCMDE = 0x1.

ADC Normal Power Mode

In Normal Mode, the Current and Voltage/Temperature channels are fully enabled. The ADC modulator clock is 512KHz and enables the ADCs to provide regular conversion results at a rate of between 4Hz and 8KHz (see ADCFLT). Both channels are under full control of the MCU and can be reconfigured at any time. The default ADC update rate for all channels in this mode is 1.0kHz

It is worth emphasizing that I-ADC and V/T-ADC channels can be configured to initiate periodic, normal power mode, high accuracy, single conversion cycles before returning to ADC full power-down mode. This flexibility is facilitated under full MCU control via the ADCMDE MMR and ensures that continuous periodic monitoring of battery current, voltage and temperature settings is feasible while ensuring the average dc current consumption is minimized.

In ADC Normal Mode, the PLL must not be powered down.

ADC Low Power Mode

In ADC Low Power mode, the I-ADC is enabled in a reduced power and reduced accuracy configuration. The ADC modulator clock is now driven directly from the on-chip 131kHz low power oscillator, which allows the ADC to be configured at update rates as low as 1Hz (ADCFLT). The gain of the ADC in this mode is fixed at 128.

All of the ADC peripheral functions (result counter, digital comparator and accumulator) described earlier in normal power mode can still be enabled in low power mode.

Typically, in Low Power Mode, the I-ADC only, is configured to run at a low update rate, continuously monitoring battery current. The MCU will be in power-down mode and will only be woken up when the I-ADC interrupts the MCU. This would happen after the I-ADC detects a current conversion beyond a pre-programmed threshold, set-point or a set number of conversions.

It is also possible to select either the ADC Precision Voltage Reference or the ADC Low Power Mode Voltage Reference via ADCMDE[5].

ADC Low Power-Plus Mode

In Low Power-Plus mode, the I-ADC channel is enabled in a mode almost identical to low-power mode (ADCMDE[4:3]). However, in this mode, the I-ADC gain is fixed at 512 and the ADC consumes an additional 200μA (approx.) to yield improved noise performance relative to the low-power mode setting.

Again, all of the ADC peripheral functions (result counter, digital comparator and accumulator) described earlier in normal power mode can still be enabled in Low Power-Plus mode.

As in Low Power Mode, the I-ADC only, is configured to run at a low update rate, continuously monitoring battery current. The MCU will be in power-down mode and will only be woken up when the I-ADC interrupts the MCU. This would happen after the I-ADC detects a current conversion result beyond a pre-programmed threshold or set-point.

It is also possible to select either the ADC Precision Voltage Reference or the ADC Low Power Mode Voltage Reference via ADCMDE[5].

ADC comparator and accumulator

Every I-ADC result can also compared to a pre-set threshold level (ADC0TH) as configured via ADCCFG[4:3]. An MCU interrupt is generated if the absolute (sign-independent) value of the ADC result is greater than the pre-programmed comparator threshold level. An extended function of this comparator function allows user code to configure a threshold counter (ADC0THV) which monitors the number of I-ADC results that have occurred above or below the pre-set threshold level. Again, an ADC interrupt is generated once the threshold counter reaches a pre-set value (ADC0TCL).

Finally, a 32-bit accumulator(ADC0ACC) function can be configured(ADCCFG[6:5]) allowing the I-ADC to add(or subtract) multiple I-ADC sample results. User code to read the accumulated value directly(ADC0ACC) without any further software processing.

ADC Sinc3 Digital Filter Response

The overall frequency response on all ADuC7030/ADuC7033's ADCs is dominated by the low pass filter response of the on-chip Sinc3 digital filters. The Sinc3 filters are used to decimate the ADC sigma-delta modulator output data bit-stream to generate a valid 16-bit data result. The digital filter response is identical for all ADCs and is configured via the 16-bit ADC Filter (ADCFLT) register, which determines the overall throughput rate of the ADCs. The noise resolution of the ADCs is determined by the programmed ADC throughput rate. In the case of the Current Channel ADC, the noise resolution will be determined by throughput rate and selected gain.

The overall frequency response and the ADC through-put is dominated by the configuration of the Sinc3 Filter Decimation Factor (SF) bits (ADCFLT[6:0]) and the Averaging Factor (AF) bits(ADCFLT[13:8]). Due to limitations on the digital filter internal data-path, there are some limitations on the allowable 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 in Normal Power Mode to 4Hz or 1Hz in Low Power Mode. The calculation of the ADC throughput rate is detailed in the ADCFLT bit designations table and the restrictions on allowable combinations of AF and SF values are outlined again in Table 36.

Table 36. Allowable Combinations of SF and AF

AF Range / SF	0	1 to 7	8-63
0-31	✓	✓	✓
32-63	✓	✓	✗
64-127	✓	✗	✗

By default the ADCFLT = 0x07 which configures the ADCs for a through-put of 1.0KHz with all other filtering options (Chop, Running Average, Averaging Factor and Sin3 Modify) being disabled. A typical filter response based on this default configuration is shown in Figure 19 below.

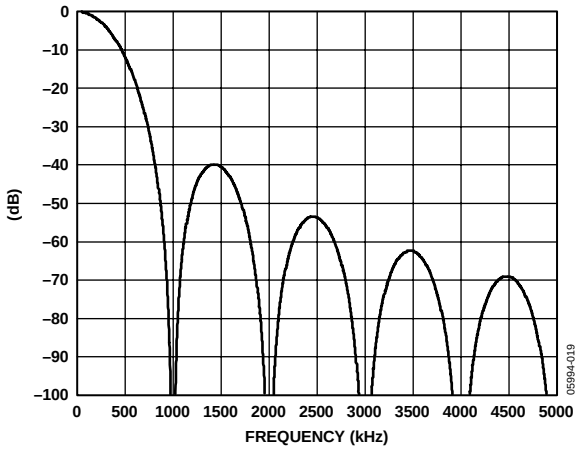


Figure 19. Typical Digital Filter Response at FADC=1.0kHz (ADCFLT = 0x0007)

An additional 'Sinc3 Modify' bit (ADCFLT[7]) is also available in the ADCFLT register. This bit is set by user code to modify the standard Sinc3 frequency response increasing the filter stop-band rejection by 5dBs approx. This is achieved by inserting a second notch (NOTCH2) at FNOTCH2 = 1.333 X FNOTCH where FNOTCH is the location of the 1st notch in the response. There is a slight increase in ADC noise if this bit is active. Figure 20 shows the modified 1KHz filter response when the Sinc3 modify bit is active. The 'new' notch is clearly visible at

1.33KHz as is the improvement in stop-band rejection when compared to the standard 1KHz response above.

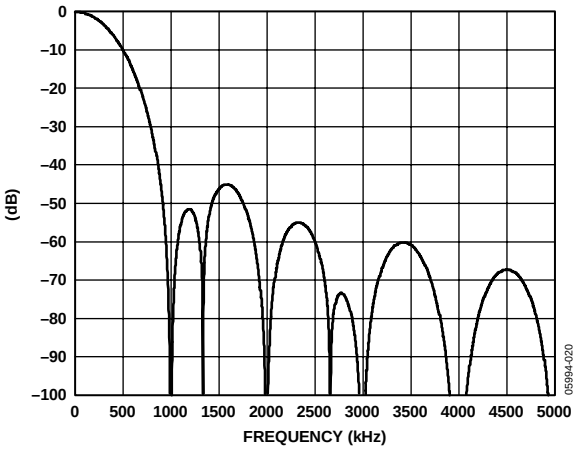


Figure 20. ModifiedSinc3 Digital Filter Response at FADC=1.0kHz (ADCFLT = 0x0087)

In ADC Normal Power Mode, the maximum ADC throughput rate is 8kHz which is configured by setting the SF and AF bits in the ADCFLT MMR to 0, with all other filtering options disabled. This results in 0x0000 written to ADCFLT and a typical 8KHz filter response based on these settings is shown below in Figure 21.

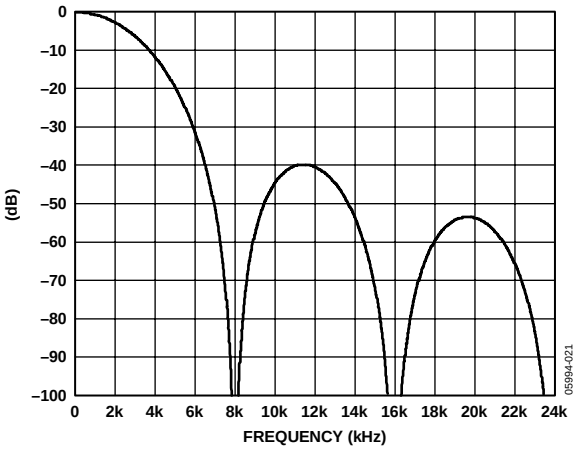


Figure 21. Typical Digital Filter Response at FADC=8 kHz, (ADCFLT = 0x0000)

A modified version of the 8KHz filter response can be configured by setting the 'Running Average' bit (ADCFLT[14]). This has the effect of introducing an additional running average by 2 filter on all ADC output samples. This further reduces the ADC output noise and while maintaining an 8kHz ADC through-put rate the ADC settling time is increased by 1 full conversion period. The modified frequency response for this configuration is shown below in Figure 22.

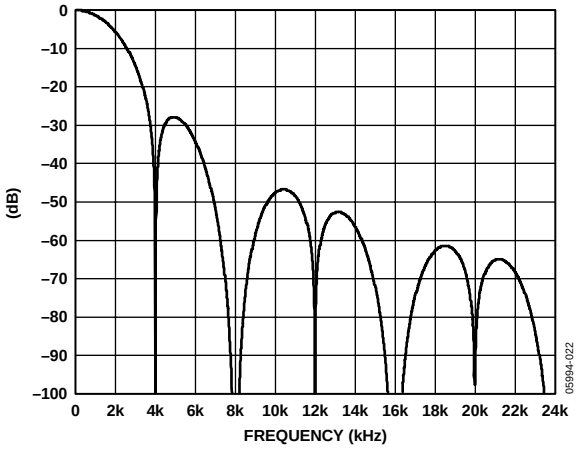


Figure 22. Typical Digital Filter Response at FADC=8KHz, (ADCFLT = 0x4000)

At very low throughput rates, the chop bit in the ADCFLT register can be enabled to minimize offset errors and more importantly temperature drift in the ADC offset error. With Chop enabled, there are again 2 primary variables (Sinc3 decimation factor and averaging factor) available to allow the user select an optimum filter response trading off filter bandwidth against ADC noise.

For example, with the CHOP bit ADCFLT[15] set to 1, increasing the SF value (ADCFLT[6:0]) to 0x1F (31dec) and selecting an AF value (ADCFLT[13:8]) of 0x16 (22dec) results in an ADC through-put of 10Hz. The frequency response in this case is shown in Figure 23.

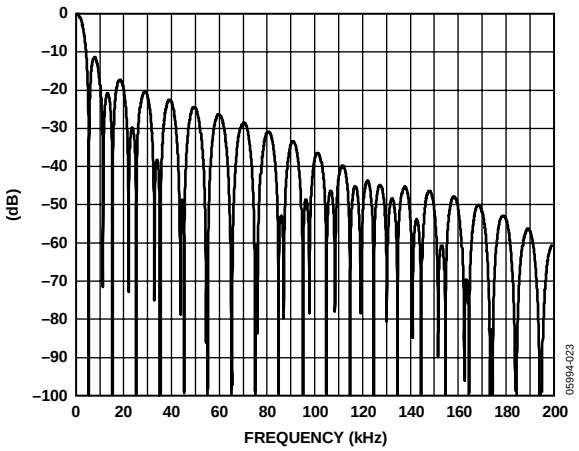


Figure 23 Typical Digital Filter Response at FADC=8 kHz, (ADCFLT = 0x961F)

Changing SF to 0x1D and setting AF to 0x3F, again with the Chop bit enabled, configures the ADC into its minimum through-put rate in Normal Mode of 4Hz. The digital filter frequency response with this configuration is shown below in Figure 24.

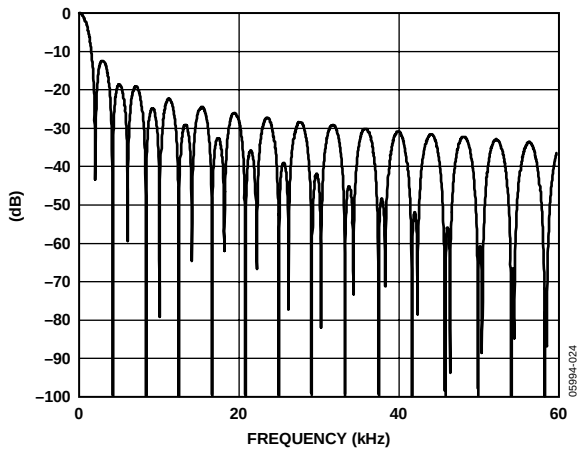


Figure 24. Typical Digital Filter Response at FADC=4Hz, (ADCFLT = 0xBF1D)

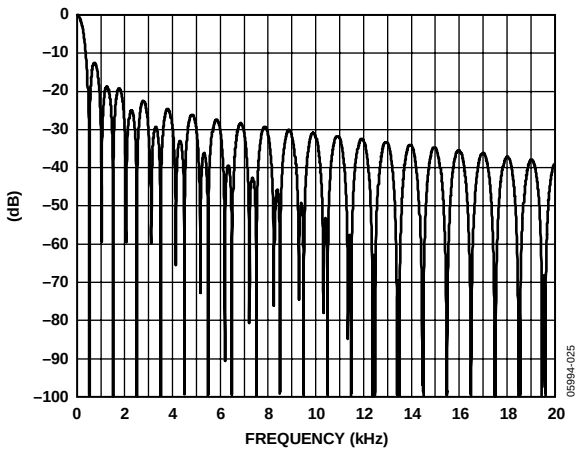


Figure 25. Typical Digital Filter Response at FADC=1Hz, (ADCFLT = 0xBD1F

In ADC Low Power Mode, the ADC, Sigma-Delta modulator clock no longer driven at 512kHz but is driven directly from the on-chip low power (131kHz) oscillator. Subsequently, for the same ADCFLT configurations in Normal Mode, all filter values should be scaled by a factor of approximately four. This means that it is possible to configure the ADC for 1Hz throughput in Low Power Mode. The filter frequency response for this configuration is shown below in Figure 25.

In general, it should be noted that it is possible to program different values of SF and AF in the ADCFLT register and achieve the same ADC update rate. In practical terms, the trade-off with any value of ADCFLT will be frequency response versus ADC noise. For optimum filter response and ADC noise when using combinations of SF and AF, a good rule of thumb to use would be to first choose an SF in the range of 16 – 40 (dec) or 0x10 to 0x28 and then increase the AF value to achieve the required ADC through-put. Table 37 shows some common ADCFLT configurations.

Table 37. Common ADCFLT Configurations

ADC Mode	SF	AF	Other Config	ADCFLT	FADC	TSETTLE
Normal	0x1D	0x3F	Chop On	0xBF1D	4Hz	0.5secs
Normal	0x1F	0x16	Chop On	0x961F	10Hz	0.2secs
Normal	0x07	0x00	None	0x0007	1KHz	3msecs
Normal	0x07	0x00	Sinc 3 Modifiy	0x0087	1KHz	3msec
Normal	0x03	0x00	Running Average	0x4003	2KHz	2msec
Normal	0x00	0x00	Running Average	0x4000	8KHz	0.5ms
Low Power	0x10	0x03	Chop On	0x8310	20Hz	100ms
Low Power	0x10	0x09	Chop On	0x8910	10Hz	200ms
Low Power	0x1F	0x3D	Chop On	0xBD1F	1Hz	2sec

ADC Calibration

As described in detail in the top-level diagrams at the start of this section, the signal flow through all ADC Channels can be described in simple terms as:

An Input-voltage is applied through an input buffer (and PGA in the case of the I-ADC) to the Sigma-Delta Modulator.

The Modulator-output is applied to a programmable Digital Decimation Filter.

The filter output result is then averaged if chopping is used.

An Offset value (ADCxOF) is subtracted from the result.

This result is scaled by a Gain value (ADCxGN).

Finally, the result is formatted as

- 2's Complement. / Offset-Binary,
- Rounded to 16 bits
- Clamped to +/-Full-Scale

Each ADC has a specific Offset and Gain correction or Calibration coefficient associated with it that are stored in MMR based Offset and Gain registers(ADCxOF and ADCxGN). The offset and gain registers can be used to remove offsets and gain errors arising within the part as well as System-level offset and gain errors external to the part.

These registers are configured at power-on with a factory programmed calibration value. These factory calibration values will vary from part to part reflecting the manufacturing variability of internal ADC circuits. However, these registers can also be overwritten by user code (only if the ADC is in idle mode) and will be automatically overwritten if an offset or gain

calibration cycle is initiated by user via the mode bits in the ADCMDE[2:0] MMR. 2 types of automatic calibration are available to the user, namely:

Self (Offset or Gain) Calibration, where the ADC generates its calibration coefficient based on an internally generated 0V in the case of Self-Offset calibration and full-scale voltage in the case of Self-Gain calibration. It should be emphasized that ADC Self-Calibrations correct for offset and gain errors within the ADC. Self-calibrations cannot compensate for other external errors in the system, e.g. Shunt-Resistor tolerance/drift, external offset voltages etc.

System (Offset or Gain) Calibration, where the ADC generates its calibration coefficient based on an externally generated zero-scale voltage in the case of System-Offset calibration and Full-scale voltage in the case of System-Gain calibration, which are applied to the external ADC input for the duration of the calibration cycle.

The duration of an Offset calibration is 1 single conversion cycle (3/FADC Chop off, 2/FADC Chop on) before returning the ADC to idle mode. A Gain calibration is a 2-stage process and subsequently takes twice as long as an offset calibration cycle. Once a calibration cycle is initiated, any ongoing ADC conversion is immediately halted, the calibration is carried out automatically at an ADC update rate programmed into ADCFLT and the ADC is always returned to idle after any calibration cycle. It is strongly recommended that ADC calibration is initiated at as low an ADC update rate as possible (high SF value in ADCFLT) in order to minimize the impact of ADC noise during calibration.

NOTE: in self-calibration mode, ADC0GN must first contain the values for PGA = 1 before a calibration scheme is started

Using the Offset and Gain Calibration

If the Chop bit (ADCFLT[15]) is enabled, then internal ADC offset errors will be minimized and an Offset calibration may not be required. If chopping is disabled however, an initial Offset calibration will be required and may need to be repeated particularly after a large change in temperature.

A Gain calibration, particularly in the context of the I-ADC (with internal PGA) may need to be carried out at all relevant system gain ranges depending on system accuracy requirements. If it is not possible to apply an external full-scale current on all gain ranges then it is possible to apply a lower current, and scale the result produced by the calibration. e.g Apply a 50% current and then divide the ADC0GN value produced by 2 and write this value back into ADC0GN. It should be noted that there is a lower limit to the input signal that can be applied for a System-Calibration because the ADC0GN register is only 16-Bit. The input span (difference between the System Zero-Scale value and System Full-Scale value) should be greater than 40% of the nominal Full-Scale-Input range, that is, > 40% of V_{REF}/Gain.

The on-chip Flash/EE memory can be used to store multiple calibration coefficients, which can be copied by user code directly into the relevant calibration registers as appropriate based on system configuration. In general, the simplest way to use the calibration registers is to let the ADC calculate the values required as part of the ADC automatic calibration modes.

A factory or end-of-line calibration for the I-ADC would be a 2-step procedure:

1. Apply 0A current.

Configure the ADC in the required PGA setting etc. and write to ADCMDE[2:0] to perform a System Zero-Scale Calibration. This writes a new offset calibration value into ADC0OF.

2. Apply a Full-Scale current for the selected PGA setting.

Write to ADCMDE to perform a System Full-Scale Calibration. This writes a new gain calibration value into ADC0GN.

Understanding the Offset and Gain Calibration Registers

The output of the average block in the ADC signal flow described earlier after the digital filter and before the Offset and Gain scaling can be considered to be a fractional number with a span, for a +/- Full-Scale input, of approx +/-0.75. The span is less than +/-1.0 because there is attenuation in the modulator to accommodate some over-range capacity on the input signal. The exact value of the attenuation will vary slightly from part-to-part, because of manufacturing tolerances.

The Offset Coefficient is read from the ADC0OF calibration register. This value is a 16-Bit 2's complement number. The range of this number, in terms of the signal chain, is effectively +/-1.0. 1 LSB of the ADC0OF register is therefore not the same as 1LSB of ADC0DAT.

A positive value of ADC0OF indicates that offset is subtracted from the output of the filter, a negative value is added. The nominal value of this register is 0x0000, indicating zero offset is to be removed. The actual offset of the ADC may vary slightly from part-to-part and at different PGA gains. The offset within the ADC is minimized if the Chopping mode is active (ADCFLT[15]=1).

The Gain Coefficient is a unit less scaling factor. The 16-Bit value in this register is divided by 16384, and then multiplied by the offset-corrected value. The nominal value of this register equals 0x5555, which corresponds to a multiplication factor of 1.3333. This scales the nominal +/-0.75 signal to produce a full-scale output signal of +/-1.0 which is checked for Overflow/ Underflow and converted to Two's Complement or Unipolar mode as appropriate, before being output to the Data register.

The actual gain, and the required scaling coefficient for zero gain error, varies slightly from part to part, and at different PGA settings and in Normal / Low-Power-Mode. The value

downloaded into ADC0GN at power-on/reset represents the scaling factor for a PGA Gain=1. There will be some level of gain error if this value is used at different PGA settings. User code can over-write the calibration coefficients or run ADC calibrations to correct the gain error at the current PGA setting.

In Summary, the simplified ADC transfer function can be described as:

$$ADCOUT = \left[\frac{VIN * PGA}{VREF} - ADCOF \right] * \frac{ADCGN}{ADCGNNOM}$$

This equation is valid for Voltage/Temperature channel ADC.

For the Current Channel ADC,

$$ADCOUT = \left[\frac{VIN * PGA}{VREF} - K * ADCOF \right] * \frac{ADCGN}{ADCGNNOM}$$

where *K* is dependent on PGA gain setting and ADC mode.

Normal Mode:

For PGA gains of 1,4,8,16,32 and 64 the K factor is 1. For PGA gains of 2 and 128 the K factor is 2. For PGA gain of 256 the K Factor is 4. For PGA gain of 512, the K factor is 8.

Low Power Mode:

The PGA gain is set to 128 and the K factor is 32.

Low Power Plus Mode:

The PGA gain is set to 512 and the K factor is 8.

In Low-Power and Low-Power-Plus Mode, the K factor doubles if (REG_AVDD)/2 is used as the reference.

ADC DIAGNOSTICS

The ADuC7030/ADuC7033 features diagnostic capability on both ADCs.

Current ADC Diagnostics

The ADuC7030/ADuC7033 features the capability to detect Open Circuit conditions on the application board. This is accomplished using the two current sources on IIN+ and IIN-, which is controlled via ADC0CON[14,13].

NOTE: These current sources have a tolerance of +-30%. A PGA gain equal to or greater than 2 (ADC0CON [3-0] ≥ 0001) must be used when current sources are enabled.

Voltage/Temperature ADC Diagnostics

The ADuC7030/ADuC7033 features the capability to detect Open Circuit conditions on the Voltage/Temperature Channel inputs. This is accomplished using the two current sources on VTEMP+ and GND_SW, which is controlled via ADC1CON[14,13].

These functions are described in details in the application note *ADC diagnostic on the ADuC703x series*.

POWER SUPPLY SUPPORT CIRCUITS

The ADuC7030/ADuC7033 incorporates an on-chip Low Drop-Out (LDO) regulator, which is driven directly from the battery voltage to generate a 2.6V internal supply. This 2.6V supply is then used as the supply voltage for the ARM7 MCU and peripherals including the precision analog circuits on-chip.

Power on Reset (POR), Power Supply Monitor (PSM) and Low Voltage Flag (LVF) functions are also integrated to ensure safe operation of the MCU as well as continuously monitoring the battery power supply.

The POR circuit is designed to handle all battery ramp rates and guarantee full functional operation of the Flash/EE memory based MCU during power-on and power-down cycles.

As shown in Figure 26, once the supply voltage, VDD, reaches a minimum operating voltage of 3V, a POR signal keeps the ARM core in reset for 20ms. This ensures that the regulated power supply voltage. REG_DVDD, supplied to the ARM core and associated peripherals is above the minimum operational voltage to guarantee full functionality. A POR flag is set in the RSTSTA MMR to indicate a POR reset event has occurred.

The ADuC7030/ADuC7033 also features a PSM, or Power Supply Monitor function. Once enabled via HVCFG0[3], the PSM continuously monitors the voltage at the VDD pin. If this voltage drops below 6.0V typically, the PSM flag is automatically asserted and can, if the high voltage IRQ is enabled via IRQ/FIQEN[16], generate a system interrupt. An example of this operation is shown in Figure 26.

At voltages below the POR level, an additional Low Voltage Flag can be enabled (HVCFG0[2]). It can be used to indicate that the contents of the SRAM are still valid after a reset event. The operation of the low voltage flag is shown in Figure 26. Once enabled, the status of this bit may be monitored via HVMON[3]. If this bit is set, then the SRAM contents are valid. If this bit is cleared, then the SRAM contents may have been corrupted.

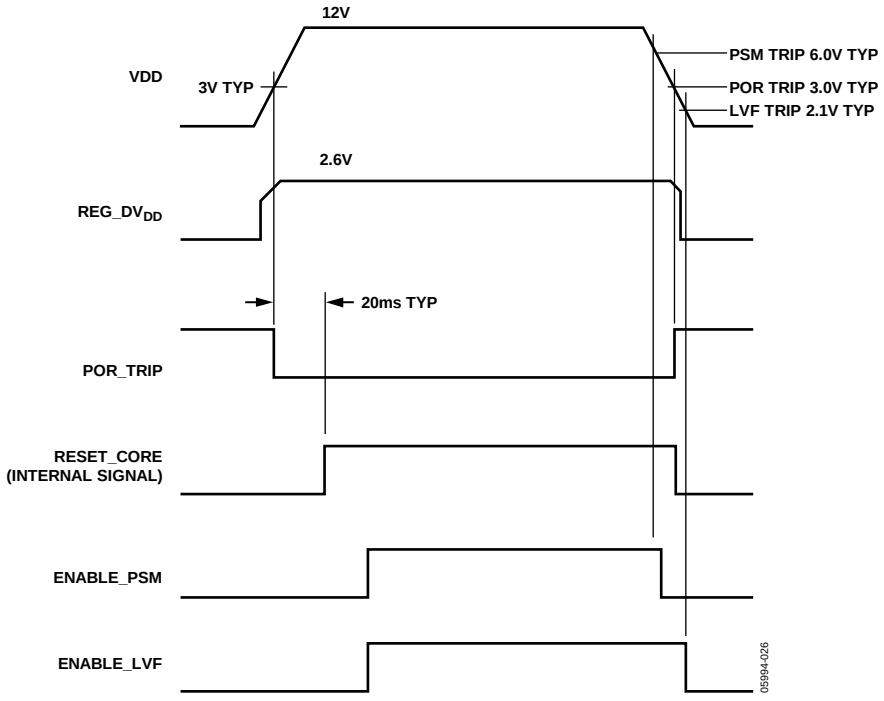


Figure 26. Typical Power-On Cycle

ADUC7030/ADUC7033 SYSTEM CLOCKS

The ADuC7030/ADuC7033 integrates a highly flexible clocking system, which may be clocked from one of three sources:

- An integrated on-chip precision oscillator

- An integrated on-chip low power oscillator

- An external watch crystal

These three options are shown in Figure 27.

Each of the internal oscillators are divided by 4 to generate a clock frequency of 32.768kHz. The PLL locks onto a multiple (625) of 32.768kHz, supplied by either of the internal oscillators or the external crystal, to provide a stable 20.48MHz clock for the system. The core can operate at this frequency, or at binary submultiples of it, which allows power saving if peak performance is not required.

By default, the PLL is driven by the Low Power oscillator which generates a 20.48MHz 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 2, which generates a core clock of 10.24MHz. The divide factor may be modified to generate a binary weighted divider factor from 1 to 128, which may be altered dynamically by user code.

The ADC is driven by the output of the PLL, divided to give an ADC clock source of 512kHz. In low-power mode the ADC clock source is switched from the standard 512kHz to the Low Power 131kHz 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 ADuC7030/ADuC7033 clocking system is shown in Figure 27.

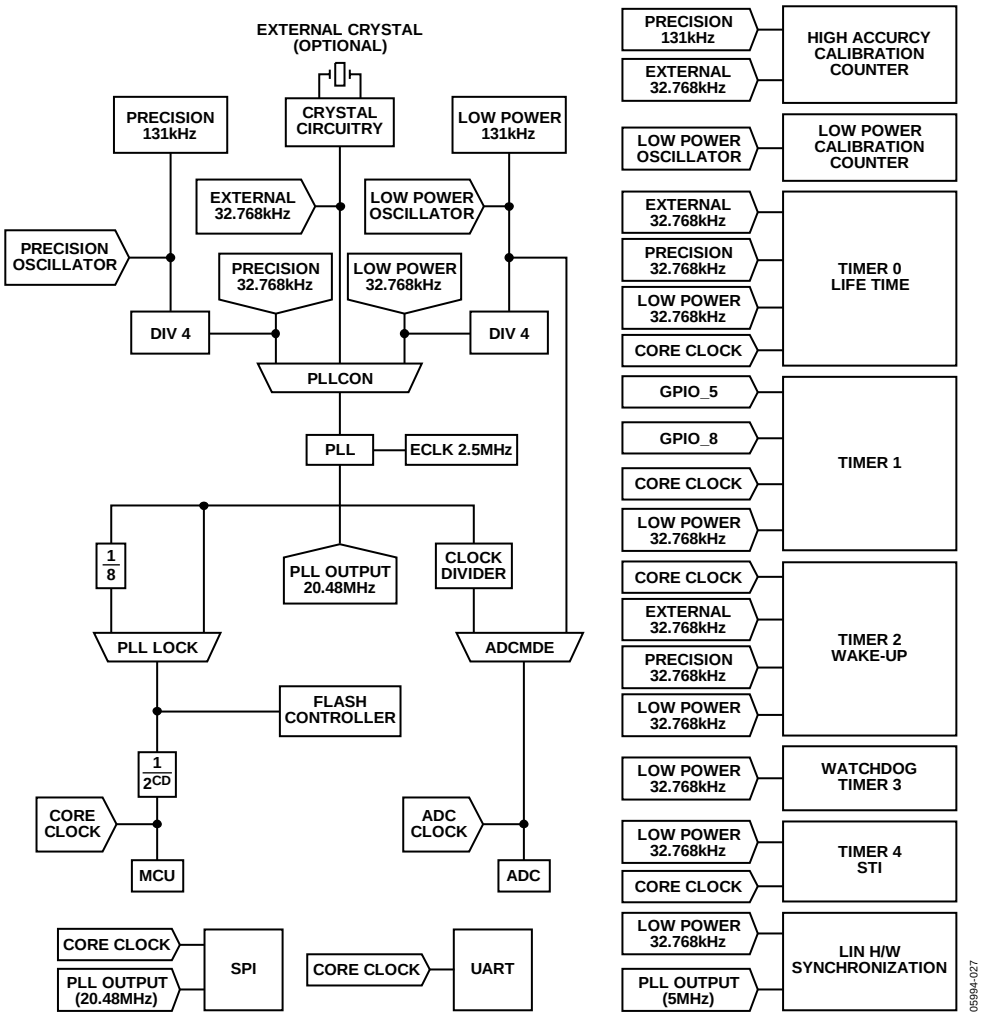


Figure 27. ADuC7030/ADuC7033 System Clock Generation

The operating mode, clocking mode and programmable clock divider are controlled via two MMRs, PLLCON and POWCON, and the status of the PLL is indicated by PLLSTA. PLLCON controls the operating mode of the clock system while POWCON controls the core clock frequency and the power-down mode. PLLSTA indicates the presence of an oscillator on the XTAL1 pin, the PLL Lock status, and the PLL Interrupt.

It is recommended that before the ADuC7030/ADuC7033 is powered down, that the clock source for the PLL is switched to the Low Power 131kHz oscillator to reduce wake up time. The Low Power Oscillator is always active.

When the ADuC7030/ADuC7033 wakes up from power down, the MCU core will begin executing code once the PLL begins oscillating. This occurs before the PLL has locked to a frequency of 20.48MHz. To ensure the Flash/EE memory controller is executing with a valid clock, the controller is driven with a PLL-Output/8 clock source while the PLL is locking. Once the PLL locks, the PLL's output is switched from the PLL-Output/8 to the locked PLL-Output.

If user code requires an accurate PLL output, user code must poll the Lock bit (PLLSTA[1]) after wake-up before resuming normal code execution.

The PLL will be locked within 2ms, if the PLL is clocked from an active clock source, e.g. Low Power 131kHz oscillator after waking up.

PLLCON is a protected MMR with two 32-bit keys PLLKEY0, a pre write key, and PLLKEY1, a post write key.

PLLKEY0 = 0x000000AA

PLLKEY1 = 0x00000055

POWCON is a protected MMR with two 32-bit keys POWKEY0, a pre write key, and POWKEY1, a post write key.

POWKEY0 = 0x00000001

POWKEY1 = 0x000000F4

An example of writing to both MMRs is shown below:

```
POWKEY0 =  0x01   //POWCON KEY
POWCON  =  0x00   //Full Power-down
POWKEY1 =  0xF4   //POWCON KEY
iA1*iA2           //dummy cycle to clear the pipe line,
where iA1 and iA2 are defined as longs and are not 0

PLLKEY0  =  0xAA  //PLLCON KEY
PLLCON   =  0x0   //Switch to LP Osc.
PLLKEY1  =  0x55  //PLLCON KEY
iA1*iA2           //dummy cycle to prevent Flash/EE
access during clock change
```

PLLSTA Register:

Name: PLLSTA

Address: 0xFFFF0400

Default Value:

Access: Read Only

Function: This 8-bit register allows user code to monitor the lock state of the PLL and the status of the external crystal.

Table 38. PLLSTA MMR Bit Description

Bit	Description
31 to 3	Reserved
2	XTAL Clock, Read Only This is a live representation of the current logic level on XTAL1. This allows the user to check to see if an external clock source is present. If present, this bit will alternate high and low at a frequency of 32.768kHz.
1	PLL Lock Status Bit, Read Only Set when the PLL is locked and outputting 20.48MHz. Clear when the PLL is not locked and outputting a Fcore/8 clock source
0	PLL Interrupt: Set if the PLL Lock status bit signal goes low. Cleared by writing 1 to this bit

PLLCON Pre-write Key PLLKEY0:

Name: PLLKEY0

Address: 0xFFFF0410

Access: Write Only

Key: 0x000000AA

Function: PLLCON is a keyed register that requires a 32-Bit key value to be written before and after PLLCON. PLLKEY0 is the Pre-Write Key.

PLLCON Pre-write Key PLLKEY1:

Name: PLLKEY1

Address: 0xFFFF0418

Access: Write Only

Key: 0x00000055

Function: PLLCON is a keyed register that requires a 32-Bit key value to be written before and after PLLCON. PLLKEY1 is the Post-Write Key.

PLLCON Register:

Name: PLLCON

Address: 0xFFFF0414

Default Value: 0x00

Access: Read/Write

Function: This 8-bit register allows user code dynamically select the PLL source clock from three different oscillator sources.

Table 39. PLLCON MMR Bit description

Bit	Description	
31-2	Reserved, these bits should be written as 0 by user code	
1-0	PLL Clock Source[1]	
	00	Low Power 131kHz oscillator
	01	Precision 131kHz oscillator[2]
	10	External 32.768kHz Crystal
	11	Reserved

[1] If user code switches MCU clock sources, a dummy MCU cycle should be included after the clock switch is written to PLLCON
[2] POWCON[7] must be enabled prior to switching to this clock source

POWCON Pre-write Key POWKEY0:

Name: POWKEY0

Address: 0xFFFF0404

Access: Write Only

Key: 0x00000001

Function: POWCON is a keyed register that requires a 32-Bit key value to be written before and after POWCON. POWKEY0 is the Pre-Write Key.

POWCON Pre-write Key POWKEY1:

Name: POWKEY1

Address: 0xFFFF040C

Access: Write Only

Key: 0x000000F4

Function: POWCON is a keyed register that requires a 32-Bit key value to be written before and after POWCON. POWKEY1 is the Post-Write Key.

POWCON Register:

Name: POWCON

Address: 0xFFFF0408

Default Value: 0x079

Access: Read/Write

Function: This 8-bit register allows user code dynamically enter various Low Power modes and modify the CD divider which controls the speed of the ARM7TDMI Core.

Table 40. POWCON MMR bit designations

Bit	Description		
31-8	Reserved		
7	Precision 131kHz Input Enable: Cleared by the user to Power down the Precision 131kHz Input Enable. Set by the user to enable the Precision 131kHz Input Enable. The Precision 131kHz oscillator must also be enabled via HVCFG0[6]. Setting this bit increases current consumption by approximately 50uA and should be disabled when not in use.		
6	XTAL Power Down: Cleared by the user to Power down the external crystal circuitry. Set by the user to enable the external crystal circuitry.		
5	PLL Power Down[1]: This bit is cleared to 0 to power down the PLL. The PLL can not be powered down if either the core or peripherals are enabled: Bits 3, 4 and 5 must be cleared simultaneously. Set by default, and set by hardware on a wake up event		
4	Peripherals[2,3,4]Power Down: Cleared to power down the peripherals. The peripherals cannot be powered down if the core is enabled: bits 3 and 4 must be cleared simultaneously. Set by default, or and by hardware on a wake up event		
3	Core Power Down:[5] Cleared to power down the ARM Core Set by default, and set by hardware on a wake up event		
2-0	CD Core clock divider bits:		
	000	20.48 MHz	48.83ns
	001	10.24 MHz	97.66ns
	010	5.12 MHz	195.31ns
	011	2.56 MHz	390.63ns
	100	1.28 MHz	781.25ns
	101	640 kHz	1.56µs
	110	320 kHz	3.125µs
	111	160 kHz	6.25µs

[1] Timer peripherals will be powered down if driven from the PLL Output clock. Timers driven from an active clock source will stay in normal power mode.
[2] The peripherals that are powered down by this bit are as follows:
SRAM, Flash/EE Memory and GPIO Interfaces
SPI and UART Serial Ports
[3] LIN can still respond to wake-up events even if this bit is cleared.
[4] Wake-Up Timer (Timer2) can still be active if driven from low power oscillator even if this bit is set.
[5] If user code powers down the MCU, a dummy MCU cycle should be included after the power-down command is written to POWCON.

ADUC7030/ ADUC7033 LOW POWER CLOCK CALIBRATION

The low power 131 kHz oscillator may be calibrated using either the precision 131kHz oscillator, or an external 32.768 kHz watch crystal. Two dedicated calibration counters and an oscillator trim register are used to implement this feature.

One counter, 9-bits wide, is clocked by the accurate clock oscillator, either the Precision oscillator or external watch crystal. The second counter, 10-bits wide, is clocked by the low power oscillator, either directly at 131kHz or via a divide by 4 block generating 32.768kHz. The source for each calibration counter should be of the same frequency. The trim register (OSC0TRM) is an 8-bit wide register, the lower 4-bits of which are user accessible trim bits. Increasing the value in OSC0TRM will decrease the frequency of the low power oscillator, decreasing the value will increase the frequency. Based on a nominal frequency of 131kHz, the typical trim range is between 127kHz to 135kHz.

The clock calibration mode is configured and controlled by the following MMRs:

OSC0CON: Control bits for calibration.

OSC0STA: Calibration Status Register

OSC0VAL0: 9-Bit counter. Counter 0.

OSC0VAL1: 10-Bit counter. Counter 1.

OSC0TRM: Oscillator Trim Register.

An example calibration routine is shown in Figure 28. User code configures and enables the calibration sequence via OSC0CON. When the precision oscillator calibration counter, OSC0VAL0, reaches 0x1FF, both counters are disabled.

User code then reads back the value of the low power oscillator calibration counter. There are three possible scenarios:

OSC0VAL0 = OSC0VAL1. No Further Action is required.

OSC0VAL0 > OSC0VAL1. The Low Power Oscillator is running slow. OSC0TRM must be decreased.

OSC0VAL0 < OSC0VAL1. The Low Power Oscillator is running fast. OSC0TRM must be increased.

When the OSC0TRM has been changed the routine should be re-run and the new frequency checked.

Using the internal precision 131kHz oscillator, it will take approximately 4milliseconds to execute the calibration routine. If the external 32.768kHz crystal is used, this time increases to 16milliseconds.

NOTE: Prior to the clock calibration routine been started, it is required that the user switch to either the precision 131kHz oscillator or the external 32.768KHz watch crystal as the PLL Clock Source. If this is not done, it is possible that the PLL will lose lock each time OSC0TRM is modified. This will increase the length of time it takes to calibrate the Low Power, Oscillator.

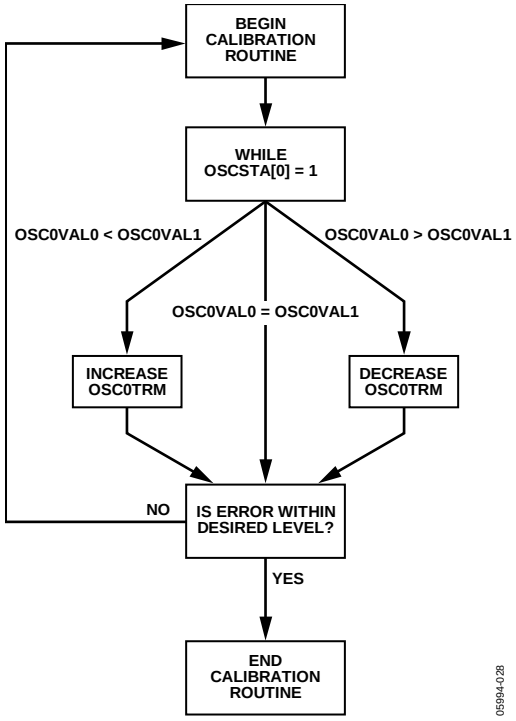


Figure 28. Example OSC0TRM Calibration Routine

OSC0TRM Register:

Name: OSC0TRM

Address: 0xFFFF042C

Default Value: 0xX8

Access: Read/Write

Function: This 8-bit register controls the Low Power Oscillator Trim

Table 41. OSCTRM MMR Bit Definition

Bit	Description
7 to 4	Reserved and should be written as zeros
3 to 0	User trim bits

OSC0CON Register:

Name: OSC0CON

Address: 0xFFFF0440

Default Value: 0x00

Access: Read/Write

Function: This 8-bit register controls the Low Power Oscillator Calibration routine

Table 42. OSCCON MMR Bit Definition

Bit	Description
7-5	Reserved. Should be written as 0
4	Calibration Source Set to select external 32.768kHz crystal Cleared to select internal precision 131kHz Oscillator.
3	Calibration Reset Set to reset the calibration counters and disable the Calibration logic
2	Set to clear OSCVAL1
1	Set to clear OSCVAL0
0	Calibration Enable Set to begin calibration Cleared to abort calibration

OSC0STA Register:

Name: OSC0STA

Address: 0xFFFF0444

Default Value: 0x00

Access: Read Access only

Function: This 8-bit register gives the status of the Low Power Oscillator Calibration routine

Table 43. OSCSTA MMR Bit Definition

Bit	Description
7-2	Reserved
1	Calibration complete Set by hardware on full completion of a calibration cycle Cleared by a read of OSC1VAL1
0	Set if calibration is in progress. Cleared if calibration completed

OSC0VAL0 Register:

Name: OSC0VAL0

Address: 0xFFFF0448

Default Value: 0x00

Access: Read Access only

Function: This 9-bit counter is clocked from either the 131kHz Precision Oscillator or the 32.768kHz external crystal.

OSC0VAL1 Register:

Name: OSC0VAL1

Address: 0xFFFF044C

Default Value: 0x00

Access: Read Access only

Function: This 10-bit counter is clocked from the Low Power, 131kHz oscillator.

PROCESSOR REFERENCE PERIPHERALS

INTERRUPT SYSTEM

There are 16 interrupt sources on the ADuC7030/ADuC7033, which are controlled by the Interrupt Controller. Most interrupts are generated from the on-chip peripherals such as the ADC, UART, etc.. The ARM7TDMI CPU core will only recognize interrupts as one of two types, a normal interrupt request IRQ and a fast interrupt request FIQ. All the interrupts can be masked separately.

The control and configuration of the interrupt system is managed through nine interrupt-related registers, four dedicated to IRQ, four dedicated to FIQ. An additional MMR is used to select the programmed interrupt source. The bits in each IRQ and FIQ registers represent the same interrupt source as described in Table 44.

IRQSTA/FIQSTA should be saved immediately upon entering the ISR (Interrupt Service Routine) to ensure that all valid interrupt sources are serviced.

The interrupt generation route through the ARM7TDMI core is shown in Figure 29.

Consider the example of Timer0, which is configured to generate a timeout every 1ms.
After the first 1ms timeout, FIQSIG/IRQSIG[2] will be set and will only be cleared by writing to T0CLRI.
If Timer0 is not enabled in either IRQEN or FIQEN, then FIQSTA/IRQSTA[2] will not be set and an interrupt will not occur.
If Timer0 is enabled in either IRQEN or FIQEN, then either FIQSTA/IRQSTA[2] will be set and either an FIQ or an IRQ interrupt will occur.

Please note that the IRQ and FIQ interrupt bit definitions in the CPSR only control interrupt recognition by the ARM Core, not by the peripherals.
For example, if Timer2 is configured to generate an IRQ via IRQEN, the IRQ interrupt bit is set (Disabled) in the CPSR and the ADuC7030 is powered down. When an interrupt occurs, the peripherals will be woken, but the ARM core will remain powered down. This is equivalent to POWCON = 0x71. The ARM Core can only be powered up by a reset event if this occurs.

Table 44. IRQ/FIQ MMRs bit description

Bit	Description	For more information please refer to:
0	All interrupts OR'ed	
1	SWI: not used in IRQEN/CLR and FIQEN/CLR	
2	Timer 0	Timer0—Life-Time timer
3	Timer 1	Timer1
4	Timer 2 - Wake Up timer	Timer2 - Wake-Up Timer
5	Timer 3 - Watchdog Timer	Timer3 - Watchdog Timer
6	Timer 4 - STI Timer	Timer4 - STI Timer
7	LIN Hardware	LIN (Local Interconnect NETWORK) INTERFACE
8	Flash/EE Interrupt	ADuC7030 Flash/EE Control Interface
9	PLL Lock	ADuC7030/ADuC7033 System Clocks
10	ADC	16-Bit Σ–Δ Analog to Digital Converters
11	UART	UART SERIAL INTERFACE
12	SPI	SERIAL PERIPHERAL INTERFACE
13	XIRQ0 (GPIO IRQ 0)	
14	XIRQ1 (GPIO IRQ 1)	
15	Reserved	
16	IRQ3 High Voltage IRQ	High Voltage Interrupt
17	Reserved	
18	XIRQ4 (GPIO IRQ 4)	
19	XIRQ5 (GPIO IRQ 5)	

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.

The four 32-bit registers dedicated to IRQ are:

IRQSIG, reflects the status of the different IRQ sources. If a peripheral generates an IRQ signal, the corresponding bit in the IRQSIG will be 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.

IRQEN, provides the value of the current enable mask. When bit is set to 1, the source request is enabled to create an IRQ exception. When bit is set to 0, the source request is disabled or masked which will not create an IRQ exception. IRQEN register cannot be used to disable an interrupt.

IRQCLR, (write-only register) allows clearing the IRQEN register in order to mask an interrupt source. Each bit set to 1 will clear the corresponding bit in the IRQEN register without affecting the remaining bits. The pair of registers IRQEN and IRQCLR allows independent manipulation of the enable mask without requiring an atomic read-modify-write.

IRQSTA, (read-only register) provides the current enabled IRQ source status (effectively a logic AND of the IRQSIG and IRQEN bits). When set to 1 that source will generate 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 FIQ (Fast Interrupt reQuest) 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 1 of FIQSTA are logically OR'ed to create the FIQ signal to the core and the bit 0 of both the FIQ and IRQ registers (FIQ source).

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

Programmed Interrupts

As the programmed interrupts are non-maskable, they are controlled by another register, SWICFG, which write into both IRQSTA and IRQSIG registers or/and FIQSTA and FIQSIG registers at the same time.

The 32-bit register dedicated to software interrupt is SWICFG described in Table 45. This MMR allows the control of programmed source interrupt.

Table 45. SWICFG MMR Bit Descriptions

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

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

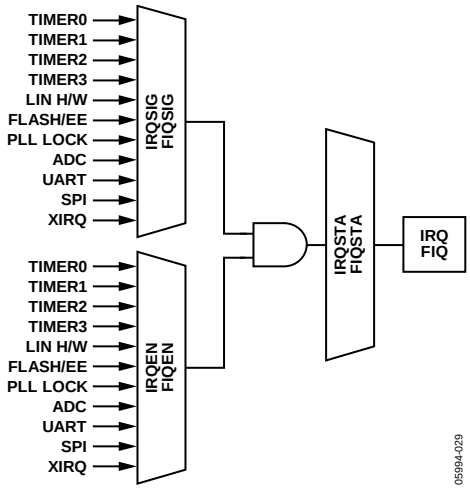


Figure 29. Interrupt Structure

TIMERS

The ADuC7030/AduC7033 features five general purpose Timer/Counters:

- Timer0, or Life-Time Timer

- Timer1,

- Timer2 or Wake-up Timer,

- Timer3 or Watchdog Timer.

- Timer4 or STI Timer.

The five timers in their normal mode of operation may either be free-running or periodic.

In free-running mode, the counter decrements/increments from the maximum/minimum value until zero/full scale and starts again at the maximum /minimum value.

In periodic mode the counter decrements/increments from the value in the Load Register(TxLD MMR,) until zero/full scale and starts again at the value stored in the Load Register. Note that the TxLD MMR should be configured before the TxCON MMR.

The value of a counter can be read at any time by accessing its value register (TxVAL). Timers are started by writing in the Control register of the corresponding timer (TxCON).

In normal mode, an IRQ is generated each time the value of the counter reaches zero, if counting down, or full-scale, if counting up. An IRQ can be cleared by writing any value to Clear register of the particular timer (TxCLRI).

Table 46. Timer Event Capture

Bit	Description
0	Timer 0 – Life Time timer
1	Timer 1
2	Timer 2 - Wake Up timer
3	Timer 3 - Watchdog Timer
4	Timer 4 - STI Timer
5	LIN Hardware
6	Flash/EE Interrupt
7	PLL Lock
8	ADC
9	UART
10	SPI
11	XIRQ0 - (GPIO_0)
12	XIRQ1 - (GPIO_5)
13	Reserved
14	IRQ3 High Voltage Interrupt
15	Reserved
16	XIRQ4 -(GPIO_7)
17	XIRQ5 - (GPIO_8)

TIMER0—LIFE-TIME TIMER

Timer0 is a general purpose 48-bit count-up, or a 16-bit count up/down timer with a programmable prescalar. Timer0 may be clocked from either the Core clock or the Low Power 32.768kHz Oscillator, with a prescalar of 1, 16, 256 or 32768. This gives a minimum resolution of 48.83ns when the core is operating at 20.48MHz, and with a prescalar of 1.

In 48-bit mode, Timer0 counts up from zero. The current counter value may be read from T0VAL0 and T0VAL1.

In 16-Bit mode,Timer0 may count up or count down. A 16-bit value may be written to T0LD, which will be loaded into the counter. The current counter value may be read from T0VAL0. Timer0 has a capture register (T0CAP), which may be triggered by a selected IRQ's source initial assertion. Once triggered, the current timer value is copied to T0CAP, and the timer keeps running. This feature can be used to determine the assertion of an event with more accuracy than by servicing an interrupt alone.

Timer0 reloads the value from T0LD either when timer0 overflows, or immediately when T0CLRI is written.

Timer0 interface consists of six MMRS:

- T0LD is a 16-bit register, which holds the 16-bit value that is loaded into the counter. T0LD is only available in 16-bit mode.

- T0CAP is a 16-bit register, which holds the 16-bit value captured by an enabled IRQ event. T0CAP is only available in 16-bit mode.

- T0VAL0/T0VAL1 are 16-bit and 32-bit registers which hold the 16 least significant bits and 32 most significant bits respectively. T0VAL0 and T0VAL1 is read-only. In 16-bit mode, 16-bit T0VAL0 is used. In 48-bit mode, both 16-bit T0VAL0 and 32-bit T0VAL1 are used.

- T0CLRI is an 8-bit register. Writing any value to this register will clear the interrupt. T0CLRI is only available in 16-bit mode.

- T0CON is the configuration MMR described in Table 47.

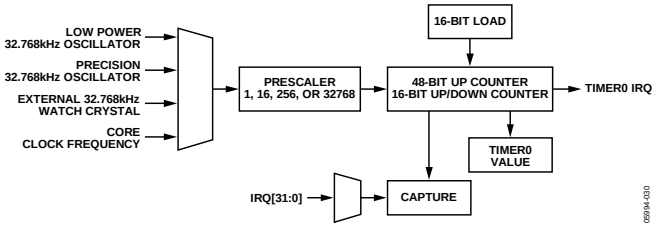


Figure 30. Timer 0 block diagram

Timer0 Value Register:

Name: T0VAL0/T0VAL1

Address: 0xFFFF0304, 0xFFFF0308

Default Value: 0x0000, 0x00000000

Access: Read Access only

Function: T0VAL0 and T0VAL1 are 16-bit and 32-bit registers, which hold the 16 least significant bits and 32 most significant bits respectively. T0VAL0 and T0VAL1 is read-only. In 16-bit mode, 16-bit T0VAL0 is used. In 48-bit mode, both 16-bit T0VAL0 and 32-bit T0VAL1 are used.

Timer0 Capture Register:

Name: T0CAP

Address: 0xFFFF0314

Default Value: 0x0000

Access: Read Access only

Function: This is a 16-bit register, which holds the 16-bit value captured by an enabled IRQ event. Only available in 16-bit mode.

Timer0 Control Register:

Name: T0CON

Address: 0xFFFF030C

Default Value: 0x00000000

Access: Read/Write

Function: The 32-bit MMR configures the mode of operation of Timer0

Table 47. T0CON MMR Bit Descriptions

Bit	Description
31-18	*Reserved*
17	Event Select bit: *Set* by user to enable time capture of an event *Cleared* by user to disable time capture of an event
16-12	Event select range, 0 to 31: The events are as described in Table 46. Timer Event Capture.
11	*Reserved*
10-9	Clock Select: 00 Core Clock (Default) 01 Low Power 32.768kHz Oscillator 10 External 32.768kHz Watch Crystal 11 Precision 32.768kHz Oscillator
8	Count up: (Only available in 16Bit Mode) *Set* by user for timer 0 to count up *Cleared* by user for timer 0 to count down (Default)
7	Timer0 enable bit: Set by user to enable timer 0 Cleared by user to disable timer 0 (Default)
6	Timer 0 mode: Set by user to operate in periodic mode Cleared by user to operate in free-running mode (Default)
5	Reserved
4	Timer0 Mode of Operation: 0 16 Bit operation (Default) 1 48 Bit Operation
3-0	Prescalar: 0000 Source clock / 1 (Default) 0100 Source clock / 16 1000 Source clock / 256 1111 Source clock / 32768

Timer0 Load Registers:

Name: T0LD

Address: 0xFFFF0300

Default Value: 0x0000

Access: Read/Write

Function: T0LD0 is a 16-bit register, which holds the 16 bit value that is loaded into the counter. Only available in 16-bit mode.

Timer0 Clear Register:

Name: T0CLRI

Address: 0xFFFF0310

Access: Write Only

Function: This 16-bit, write-only MMR is written (with any value) by user code to refresh(reload) Timer0.

TIMER1

Timer1 is a 32-bit general purpose timer, count-down or count-up, with a programmable pre-scalar. The pre-scalar source can be the Low Power 32.768kHz Oscillator, the core clock, or from one of two external GPIO. This source can be scaled by a factor of 1, 16, 256 or 32768. This gives a minimum resolution of 48.83ns when operating at CD zero, the core is operating at 20.48MHz, and with a pre-scalar of 1 (Ignoring external GPIO).

The counter can be formatted as a standard 32-bit value or as Hours:Minutes:Seconds:Hundreths.

Timer1 has a capture register (T1CAP), which can be triggered by a selected IRQ's source initial assertion. Once triggered, the current timer value is copied to T1CAP, and the timer keeps running. This feature can be used to determine the assertion of an event with increased accuracy.

Timer1 interface consists of five MMRS:

- T1LD, T1VAL and T1CAP are 32-bit registers and hold 32-bit unsigned integers. T1VAL and T1CAP are read-only.

- T1CLRI is an 8-bit register. Writing any value to this register will clear the timer1 interrupt.

- T1CON is the configuration MMR described in below.

Timer1 features a post-scalar. This allows the user to count between1 and 256 the number of timer1 timeouts. To activate the post-scalar, the user sets bit 18 and writes the desired number to count into bits 24-31 of T1CON. Once that number of timeouts has reached, Timer1 may generate an interrupt if T1CON[18] is set.

NOTE: If the part is in a low power mode, and Timer1 is clocked from the GPIO or low power oscillator source then, Timer1 will continue to operate.

Timer1 reloads the value from T1LD either when Timer1 overflows, or immediately when T1CLRI is written.

Timer1 Load Registers:

Name: T1LD

Address: 0xFFFF0320

Default Value: 0x00000000

Access: Read/Write

Function: T1LD is a 32-bit register, which holds the 32-bit value that is loaded into the counter.

Timer1 Clear Register:

Name: T1CLRI

Address: 0xFFFF032C

Access: Write Only

Function: This 32-bit, write-only MMR is written (with any value) by user code to refresh (reload) Timer1.

Timer1 Value Register:

Name: T1VAL

Address: 0xFFFF0324

Default Value: 0xFFFFFFFF

Access: Read Only

Function: T1VAL is a 32-bit register, which holds the current value of Timer1.

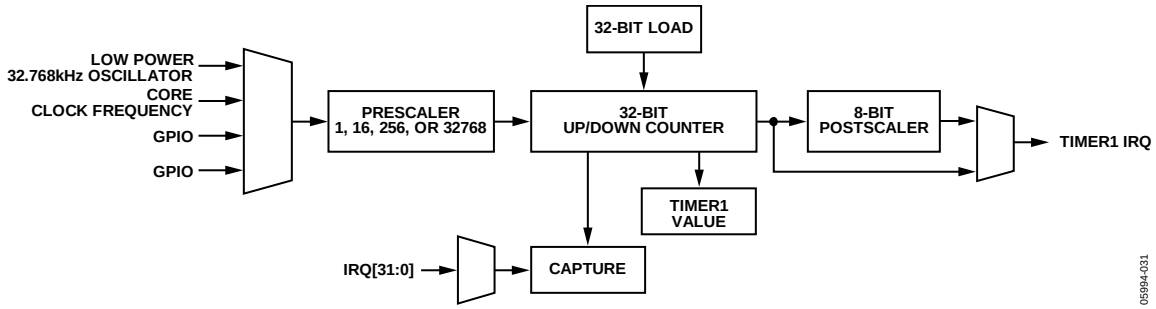


Figure 31. Timer 1 Block Diagram

Timer1 Capture Register:

Name: T1CAP

Address: 0xFFFF0330

Default Value: 0x00000000

Access: Read Only

Function: This is a 32-bit register, which 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 48. T1CON MMR Bit Descriptions

Bit	Description
31-24	8 Bit Post-Scalar By writing to these 8 bits, a value is written to the post-scalar. Writing 0 is interpreted as a 1. By reading these 8 bits, the current value of the counter is read.
23	Timer 1 Enable Post-Scalar: *Set* To enable Timer1 Post Scalar. If enabled, an interrupts will be generated after T1CON[31-24] periods as defined by T1LD. *Cleared* To disable Timer1 Post Scalar.
22-20	These bits are reserved and should be written as 0 by user code
19	Post-Scalar Compare Flag (read only). Set if the number of Timer1 overflows is equal to the number written to the post-scalar
18	Timer 1 Interrupt Source *Set* To select interrupt generation from post-scalar 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
16-12	Event select range, 0 to 31. The events are as described in Table 46. Timer Event Capture
11-9	Clock select: 000 Core clock (Default) 001 Low Power 32.768kHz Oscillator 010 GPIO_8 011 GPIO_5
8	Count up: *Set* by user for timer 1 to count up *Cleared* by user for timer 1 to count down (Default)
7	Timer1 enable bit: Set by user to enable timer 1 Cleared by user to disable timer 1 (Default)
6	Timer 1 mode: Set by user to operate in periodic mode. Cleared by user to operate in free-running mode (Default)
5-4	Format: 00 Binary (Default) 01 *Reserved* 10 Hr:Min:Sec:Hundredths – 23 hours to 0 hour 11 Hr:Min:Sec:Hundredths – 255 hours to 0 hour
3-0	Pre-Scalar: 0000 Source clock / 1 (Default) 0100 Source clock / 16 1000 Source clock / 256 1111 Source clock / 32768

TIMER2 - WAKE-UP TIMER

Timer2 is a 32-bit wake-up timer, count-down or count-up, with a programmable prescalar. The pre-scalar is clocked directly from 1 of 4 clock sources, namely, the Core Clock (default selection), the Low Power 32.768kHz Oscillator, External 32.768kHz Watch Crystal, or the Precision 32.768kHz Oscillator. The selected clock source can be scaled by a factor of 1, 16, 256 or 32768. The wake-up timer will continue to run when the core clock is disabled. This gives a minimum resolution of 48.83ns when operating at CD zero, the core is operating at 20.48MHz, and with a prescalar of 1.

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.

Timer2 interface consists of four MMRS:

- T2LD and T2VAL are 32-bit registers and hold 32-bit unsigned integers. T2VAL is read-only.

- T2CLRI is an 8-bit register. Writing any value to this register will clear the timer2 interrupt.

- T2CON is the configuration MMR described in Table 36 below.

Timer2 Load Registers:

Name: T2LD

Address: 0xFFFF0340

Default Value: 0x00000000

Access: Read/Write

Function: T2LD is a 32-bit register, which holds the 32 bit value that is loaded into the counter.

Timer2 Clear Register:

Name: T2CLRI

Address: 0xFFFF034C

Access: Write Only

Function: This 32-bit, write-only MMR is written (with any value) by user code to refresh (reload) Timer2.

Timer2 Value Register:

Name: T2VAL

Address: 0xFFFF0344

Default Value: 0xFFFFFFFF

Access: Read Only

Function: T2VAL is a 32-bit register which holds the current value of Timer2.

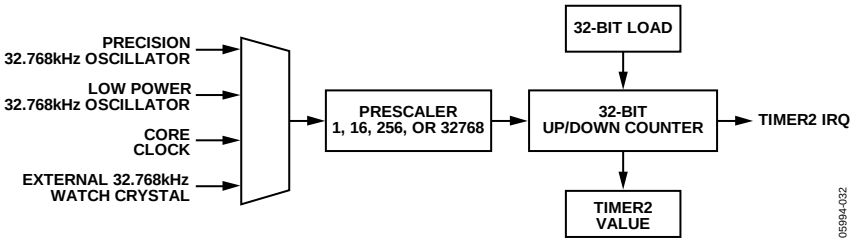


Figure 32. Timer 2 Block Diagram

Timer2 Control Register:

Name: T2CON

Address: 0xFFFF0348

Default Value: 0x0000

Access: Read/Write

Function: This 16-bit MMR configures the mode of operation of Timer2

Table 49. T2CON MMR Bit Descriptions

Bit	Description
15-11	*Reserved*
10-9	Clock Source Select: 00 Core Clock (Default) 01 Low Power 32.768kHz Oscillator 10 External 32.768kHz Watch Crystal 11 Precision 32.768kHz Oscillator
8	Count up: Set by user for timer 2 to count up Cleared by user for timer 2 to count down (Default)
7	Timer2 enable bit: Set by user to enable timer 2 Cleared by user to disable timer 2 (Default)
6	Timer 2 mode: Set by user to operate in periodic mode Cleared by user to operate in free-running mode (Default)
5-4	Format: 00 Binary (Default) 01 *Reserved* 10 Hr:Min:Sec:Hundredths – 23 hours to 0 hour (only valid with a 32kHz clock) 11 Hr:Min:Sec:Hundredths – 255 hours to 0 hour (only valid with a 32kHz clock)
3-0	Prescalar: 0000 Source clock / 1 (Default) 0100 Source clock / 16 1000 Source clock / 256 (This setting should be used in conjunction Timer2 Formats 1,0 and 1,1) 1111 Source clock / 32768

TIMER3 - WATCHDOG TIMER

Timer3 has two modes of operation, normal mode and watchdog mode. The Watchdog timer is used to recover from an illegal software state. Once enabled it requires periodic servicing to prevent it from forcing a reset of the processor.

Timer3 reloads the value from T3LD either when TIMER3 overflows, or immediately when T3CLRI is written.

Normal mode:

The Timer3 in normal mode is identical to Timer0, in 16-bit mode of operation, except for the clock source. The clock source is the Low Power 32.768kHz oscillator and can be scaled by a factor of 1, 16, or 256.

Watchdog mode:

Watchdog mode is entered by setting T3CON[5]. Timer3 decrements from the timeout value present in T3LD Register until zero. The maximum timeout is 512 seconds, using the maximum pre-scalar /256 and full-scale in T3LD.

User software should only configure a minimum timeout period of 30msecs. This is to avoid any conflict with Flash/EE memory page erase cycles, which require 20ms to complete a single page erase cycle, and Kernel Execution.

If T3VAL reaches 0, a reset or an interrupt occurs, depending on T3CON[1]. To avoid a reset or an interrupt event, any value must be written to T3CLRI before T3VAL reaches zero. This reloads the counter with T3LD and begins a new timeout period.

Once watchdog mode is entered, T3LD and T3CON are write-protected. These two registers can not be modified until a Power On Reset event, resets the Watchdog Timer, after any other reset event, the Watchdog Timer continues to count. The Watchdog Timer should be configured in the initial lines of user code to avoid an infinite loop of Watchdog Resets. User software should only configure a minimum timeout period of 30msecs.

Timer3 is automatically halted during JTAG debug access and will only recommence counting once JTAG has relinquished control of the ARM7 core. By default, Timer3 continues to count during power-down. This may be disabled by setting bit zero in T3CON. It is recommended that the default value is used, i.e. that the Watchdog Timer continues to count during power-down.

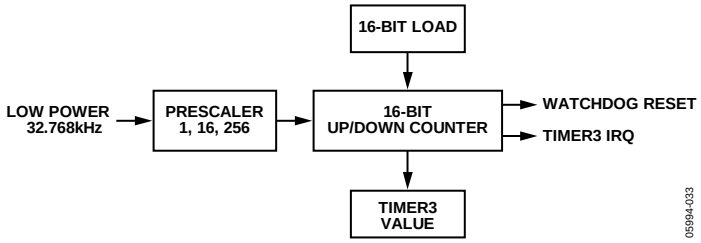


Figure 33. Timer3 Block Diagram

Timer3 Interface:

Timer3 interface consists of four MMRS:

- T3CON is the configuration MMR described in Table 37

- T3LD and T3VAL are 16-bit registers (bit 0 to 15) and hold 16-bit unsigned integers. T3VAL is read-only.

- T3CLRI is an 8-bit register. Writing any value to this register will clear the Timer3 interrupt in normal mode or will reset a new timeout period in watchdog mode

Timer3 Load Register:

Name: T3LD

Address: 0xFFFF0360

Default Value: 0x0040

Access: Read/Write

Function: This 16-bit MMR holds the Timer3 reload value

Timer3 Value Register:

Name: T3VAL

Address: 0xFFFF0364

Default Value: 0x0040

Access: Read Only

Function: This 16-bit, read-only MMR holds the currentTimer3 count value.

Timer3 Clear Register:

Name: T3CLRI

Address: 0xFFFF036C

Access: Write Only

Function: This 16-bit, write-only MMR is written (with any value) by user code to refresh(reload) Timer3 in watchdog mode to prevent a watchdog timer reset event.

Timer3 Control Register:

Name: T3CON

Address: 0xFFFF0368

Default Value: 0x0000

Access: Read/Write

Function: The 16-bit MMR configures the mode of operation of Timer3 as is described in detail in Table 50.

Table 50. T3CON MMR Bit Definition

Bit	Description
15-9	These bits are reserved and should be written as 0 by user code
8	Count Up/Down Enable Set by user code to configure Timer3 to count up Cleared by user code to configure Timer3 to count down
7	Timer3 Enable Set by user code to enable Timer 3 Cleared by user code to disable Timer 3
6	Timer3 Operating Mode Set by user code to configure Timer3 to operate in periodic mode Cleared by user to configure Timer3 to operate in free-running mode
5	Watchdog Timer Mode Enable Set by user code to enable watchdog mode Cleared by user code to disable watchdog mode
4	Reserved This bit is reserved and should be written as 0 by user code
3-2	Timer3 Clock(32.768kHz) Pre-Scalar 00 Source clock / 1 (Default) 01 Source clock / 16 10 Source clock / 256 11 *Reserved*
1	Watchdog Timer IRQ Enable Set by user code to produce an IRQ instead of a reset when the watchdog reaches 0 Cleared by user code to disable the IRQ option
0	PD_OFF Set by the user code to stop Timer3 when the peripherals are powered down via bit 4 in the POWCON MMR. Cleared by the user code to enable Timer3 when the peripherals are powered down via bit 4 in the POWCON MMR.

TIMER4 - STI TIMER

Timer4 is a general purpose 16-bit count-up/down timer with a programmable prescalar. Timer4 may be clocked from the Core clock, or the Low Power 32.768kHz Oscillator, with a prescalar of 1, 16, 256 or 32,768.

Timer4 has a capture register (T4CAP), which can be triggered by a selected IRQ's source initial assertion. Once triggered, the current timer value is copied to T4CAP, and the timer keeps running. This feature can be used to determine the assertion of an event with increased accuracy.

Timer4 may also be used to drive the STI (Serial Test Interface) peripheral.

Timer4 interface consists of five MMRS:

- T4LD, T4VAL and T4CAP are 16-bit registers and hold 16-bit unsigned integers. T4VAL and T4CAP are read-only.

- T4CLRI is an 8-bit register. Writing any value to this register will clear the interrupt

- T4CON is the configuration MMR described in Table 35

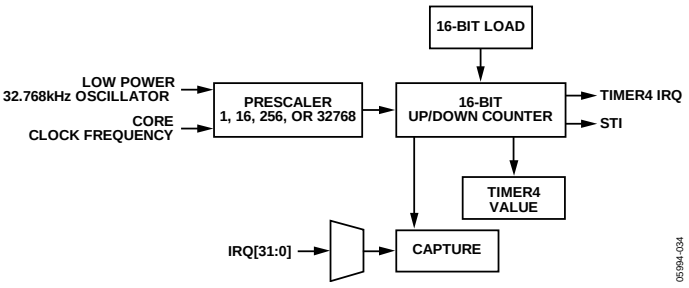


Figure 34.. Timer 4 block diagram

Timer4 Load Registers:

Name: T4LD

Address: 0xFFFF0380

Default Value: 0x00000

Access: Read/Write

Function: T4LD is a 16-bit register, which holds the 16-bit value that is loaded into the counter.

Timer4 Clear Register:

Name: T4CLRI

Address: 0xFFFF038C

Access: Write Only

Function: This 8-bit, write-only MMR is written (with any value) by user code to refresh(reload) Timer4.

Timer4 Value Register:

Name: T4VAL

Address: 0xFFFF0384

Default Value: 0xFFFF

Access: Read Only

Function: T4VAL is a 16-bit register which holds the current value of Timer4.

Timer4 Capture Register:

Name: T4CAP

Address: 0xFFFF0390

Default Value: 0x0000

Access: Read Only

Function: This is a 16-bit register, which holds the 32-bit value captured by an enabled IRQ event.

Timer4 Control Register:

Name: T4CON

Address: 0xFFFF0388

Default Value: 0x00000000

Access: Read/Write

Function: This 32-bit MMR configures the mode of operation of Timer4.

Table 51. T4CON MMR Bit Description

Bit	Description	
31-18	*Reserved*	
17	Event Select bit: *Set* by user to enable time capture of an event *Cleared* by user to disable time capture of an event	
16-12	Event select range, 0 to 31 The events are as described in Table 46. Timer Event Capture	
11-10	*Reserved*	
9	Clock Select 0 Core clock (Default) 1 Low Power 32.768kHz Oscillator	
8	Count up: *Set* by user for timer 4 to count up *Cleared* by user for timer 4 to count down (Default)	
7	Timer4 enable bit: Set by user to enable timer 0 Cleared by user to disable timer 0 (Default)	
6	Timer 4 mode: Set by user to operate in periodic mode Cleared by user to operate in free-running mode. Default mode	
5-4	*Reserved*	
3-0	Prescalar: 0000 0100 1000 1111	 Source clock / 1 (Default) Source clock / 16 Source clock / 256 Source clock / 32768

GENERAL PURPOSE I/O

The ADuC7030/ADuC7033 features 9 General Purpose bi-directional I/O pins (GPIO). In general, many of the GPIO pins have multiple functions which 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 and their sink capability is 0.8mA and they can source 0.1mA.

The 9 GPIO are grouped into three ports, Port0, Port1 and Port2. Port0 is 5 bits wide. Port1 and Port2 are both 2 bits wide. The GPIO assignment within each port is detailed in Table 52.

A typical GPIO structure is shown Figure 35.

External Interrupts are present on GP0, GP5, GP7 and GP8. This 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 are interrogated. The Interrupt source must be active for at least 1 CD divided core clock to guarantee recognition.

All port pins are configured and controlled by 4 sets (1 set for each port) of four port specific MMRs:

- GPxCON: Port x Control Register

- GPxDAT: Port x Configuration and Data Register

- GPxSET: Data set port x

- GPxCLR: Data clear port x

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 will retain their external (high or low) during power-down (POWCON) mode.

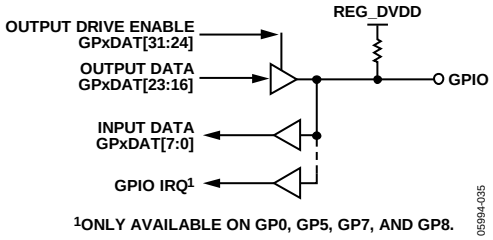


Figure 35. ADuC7030/ADuC7033 GPIO

Table 52. External GPIO Pin to Internal Port Signal Assignments

	GPIO PIN	*PORT SIGNAL*	*Functionality (Defined by GPxCON)*
Port 0	GPIO_0	P0.0 IRQ0	General Purpose I/O
			$\overline{SS}$, Slave Select I/O for SPI
	GPIO_1	P0.1	General Purpose I/O
			SCLK, Serial Clock I/O for SPI
	GPIO_2	P0.2	General Purpose I/O
			MISO, Master Input, Slave Output for SPI
	GPIO_3	P0.3	General Purpose I/O
			MOSI, Master Output, Slave Input for SPI
	GPIO_4	P0.4	General Purpose I/O
			ECLK , a 2.56MHz clock output
		P0.5[1]	High Voltage Serial Interface
		P0.6[1]	High Voltage Serial Interface
Port 1	GPIO_5	P1.0 IRQ1	General Purpose I/O
			RxD Pin for UART
	GPIO_6	P1.1	General Purpose I/O
			TxD Pin for UART
Port 2	GPIO_7	P2.0 IRQ4	General Purpose I/O
			LIN/BSD Output Pin. Used to read directly from LIN Pin for conformance testing.
	GPIO_8	P2.1 IRQ5	General Purpose I/O
			LIN/BSD HV Input Pin. Used to directly drive LIN Pin for conformance testing.
	GPIO_11[2]	P2.4[2]	General Purpose I/O
			LIN/BSD Input Pin
	GPIO_12[2]	P2.5[2]	General Purpose I/O
			LIN/BSD Output Pin
	GPIO_13[1]	P2.6[1]	General Purpose I/O
			STI data output

[1] These signals are internal signals only and do not appear on an external pin. These pins are used along with HVCON as the 2-wire interface to the high voltage interface circuits.
[2] These pins/signals are internal signals only and do not appear on an external pin. Both signls are used to provide external pin diagnostic write (GPIO_12) and read-back (GPIO_11) capability.

GPIO Port0 Control Register:

Name: GP0CON

Address: 0xFFFF0D00

Default Value: 0x11100000

Access: Read/Write

Function: The 32-bit MMR selects the pin function for each Port0 pin.

Table 53. GP0CON MMR Bit Designations

Bit	Description
31-29	Reserved These bits are reserved and should be written as 0 by user code
28	Reserved This bit is reserved and should be written as 1 by user code
27-25	Reserved These bits are reserved and should be written as 0 by user code
24	Internal P0.6 Enable Bit This bit must be set to 1 by user software to enable the High Voltage Serial Interface before using the HVCON and HVDAT registered high voltage interface
23-21	Reserved These bits are reserved and should be written as 0 by user code
20	Internal P0.5 Enable Bit This bit must be set to 1 by user software to enable the High Voltage Serial Interface before using the HVCON and HVDAT registered high voltage interface
19-17	Reserved These bits are reserved and should be written as 0 by user code
16	GPIO_4 Function Select Bit This bit is cleared by user code to 0 to configure the GPIO_4 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO_4 pin as ECLK enabling a 2.56MHz clock output on this pin
15-13	Reserved These bits are reserved and should be written as 0 by user code
12	GPIO_3 Function Select Bit This bit is cleared by user code to 0 to configure the GPIO_3 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO_3 pin as MOSI, Master Output, Slave Input Data for the SPI Port
11-9	Reserved These bits are reserved and should be written as 0 by user code
8	GPIO_2 Function Select Bit This bit is cleared to 0 by user code to configure the GPIO_2 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO_2 pin as MISO, Master Input, Slave Output Data for the SPI Port
7-5	Reserved These bits are reserved and should be written as 0 by user code
4	GPIO_1 Function Select Bit This bit is cleared to 0 by user code to configure the GPIO_1 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO_1 pin as SCLK, Serial Clock I/O for the SPI Port
3-1	Reserved These bits are reserved and should be written as 0 by user code
0	GPIO_0 Function Select Bit This bit is cleared to 0 by user code to configure the GPIO_0 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO_0 pin as $\overline{SS}$, Serial Clock I/O for the SPI Port

GPIO Port1 Control Register:

Name: GP1CON

Address: 0xFFFF0D04

Default Value: 0x10000000

Access: Read/Write

Function: The 32-bit MMR selects the pin function for each Port1 pin.

Table 54. GP1CON MMR Bit Designations

Bit	Description
31-5	Reserved These bits are reserved and should be written as 0 by user code
4	GPIO_6 Function Select Bit This bit is cleared by user code to 0 to configure the GPIO_6 pin as a General Purpose I/O (GPIO) pin This bit is set to 1 by user code to configure the GPIO_6 pin as TxD, Transmit Data for UART Serial Port
3-1	Reserved These bits are reserved and should be written as 0 by user code
0	GPIO_5 Function Select Bit This bit is cleared by user code to 0 to configure the GPIO_5 pin as a General Purpose I/O (GPIO) pin This bit is set by user code to 1 to configure the GPIO_5 RxD. Receive Data for UART Serial Port

GPIO Port2 Control Register:

Name: GP2CON

Address: 0xFFFF0D08

Default Value: 0x01000000

Access: Read/Write

Function: The 32-bit MMR selects the pin function for each Port2 pin.

Table 55. GP2CON MMR Bit Designations

Bit	Description
31-25	Reserved These bits are reserved and should be written as 0 by user code
24	GPIO_13 Function Select Bit This bit is set to 1 by user code to route the STI data output to the STI pin. If this bit is clear to 0 by user code, then the STI data will not be routed to the external STI pin even if the STI interface itself is enabled correctly.
23-21	Reserved These bits are reserved and should be written as 0 by user code
20	GPIO_12 Function Select Bit This bit is cleared to 0 by user code to route the LIN/BSD transmit data to an internal General Purpose I/O (GPIO_12) pad which can then be written via the GP2DAT MMR. This configuration is used in BSD mode to allow user code write output data to the BSD interface and can also be used to support diagnostic write capability to the high-voltage I/O pins(see HVCFG1[2:0]). This bit is set to 1 by user code to route the UART TxD (transmit data) to the LIN/BSD data pin. This configuration is used in LIN mode.
19-17	Reserved These bits are reserved and should be written as 0 by user code
16	GPIO_11 Function Select Bit This bit is cleared to 0 by user code to internally disable the LIN/BSD input data path. In this configuration GPIO_11 is used to support diagnostic read-back on all external high-voltage I/O pins (see HVCFG1[2:0]) This bit is set to 1 by user code to route input data from the LIN/BSD interface to both the LIN/BSD hardware timing/synchronization logic and to the UART RxD (receive data). This mode must be configured by user code when using LIN or BSD modes.
15-5	Reserved These bits are reserved and should be written as 0 by user code
4	GPIO_8 Function Select Bit This bit is cleared by user code to 0 to configure the GPIO_8 pin as a General Purpose I/O (GPIO) pin This bit is set by user code to 1 to route the LIN/BSD input data to the GPIO_8 pin. This mode can be used to drive the LIN transceiver interface as a standalone component without any interaction from MCU or UART.
3-1	Reserved These bits are reserved and should be written as 0 by user code
0	GPIO_7 Function Select Bit This bit is cleared by user code to 0 to configure the GPIO_7 pin as a General Purpose I/O (GPIO) pin This bit is set by user code to 1 to route data driven into the GPIO_7 pin through the on-chip LIN transceiver to be output at the LIN/BSD pin. This mode can be used to drive the LIN transceiver interface as a standalone component without any interaction from MCU or UART.

GPIO Port0 Data Register:

Name: GP0DAT

Address: 0xFFFF0D20

Default Value: 0x000000XX

Access: Read/Write

Function: This 32-bit MMR configures the direction of the GPIO pins assigned to Port0 (see Table 52). This register also sets the output value for GPIO pins configured as outputs and reads the status of GPIO pins configured as inputs.

Table 56. GP0DAT MMR Bit Descriptions

Bit	Description
31-29	Reserved These bits are reserved and should be written as 0 by user code
28	Port 0.4 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P0.4 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P0.4 as an output.
27	Port 0.3 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P0.3 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P0.3 as an output.
26	Port 0.2 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P0.2 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P0.2 as an output.
25	Port 0.1 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P0.1 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P0.1 as an output.
24	Port 0.0 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P0.0 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P0.0 as an output.
23-21	Reserved These bits are reserved and should be written as 0 by user code
20	Port 0.4 Data Output The value written to this bit appears directly on the GPIO pin assigned to P0.4.
19	Port 0.3 Data Output The value written to this bit appears directly on the GPIO pin assigned to P0.3.
18	Port 0.2 Data Output The value written to this bit appears directly on the GPIO pin assigned to P0.2.
17	Port 0.1 Data Output The value written to this bit appears directly on the GPIO pin assigned to P0.1.
16	Port 0.0 Data Output The value written to this bit appears directly on the GPIO pin assigned to P0.0.
15-5	Reserved These bits are reserved and should be written as 0 by user code
4	Port 0.4 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P0.4. User code should write 0 to this bit.
3	Port 0.3 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P0.3. User code should write 0 to this bit.
2	Port 0.2 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P0.2. User code should write 0 to this bit.
1	Port 0.1 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P0.1. User code should write 0 to this bit.
0	Port 0.0 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P0.0. User code should write 0 to this bit.

GPIO Port1 Data Register:

Name: GP1DAT

Address: 0xFFFF0D30

Default Value: 0x000000XX

Access: Read/Write

Function: This 32-bit MMR configures the direction of the GPIO pins assigned to Port1 (see Table 52). This register also sets the output value for GPIO pins configured as outputs and reads the status of GPIO pins configured as inputs.

Table 57. GP1DAT MMR Bit Descriptions

Bit	Description
31-26	Reserved These bits are reserved and should be written as 0 by user code
25	Port 1.1 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P1.1 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P1.1 as an output.
24	Port 1.0 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P1.0 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P1.0 as an output.
23-18	Reserved These bits are reserved and should be written as 0 by user code
17	Port 1.1 Data Output The value written to this bit appears directly on the GPIO pin assigned to P1.1.
16	Port 1.0 Data Output The value written to this bit appears directly on the GPIO pin assigned to P1.0.
15-2	Reserved These bits are reserved and should be written as 0 by user code
1	Port 1.1 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P1.1. User code should write 0 to this bit.
0	Port 1.0 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P1.0. User code should write 0 to this bit.

GPIO Port2 Data Register:

Name: GP2DAT

Address: 0xFFFF0D40

Default Value: 0x000000XX

Access: Read/Write

Function: This 32-bit MMR configures the direction of the GPIO pins assigned to Port2 (see Table 52). This register also sets the output value for GPIO pins configured as outputs and reads the status of GPIO pins configured as inputs.

Table 58 :GP2DAT MMR Bit Descriptions

Bit	Description
31	Reserved, this bit is reserved and should be written as 0 by user code
30	Port 2.6 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P2.6 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P2.6 as an output.
29	Port 2.5 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P2.5 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P2.5 as an output. This configuration is used to support diagnostic write capability to the high-voltage I/O pins.
28	Port 2.4 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P2.4 as an input. This configuration is used to support diagnostic read-back capability from the high-voltage I/O pins(see HVCFG1[2:0]). This bit is set to 1 by user code to configure the GPIO pin assigned to P2.4 as an output.
27-26	Reserved These bits are reserved and should be written as 0 by user code
25	Port 2.1 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P2.1 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P2.1 as an output.
24	Port 2.0 Direction Select Bit This bit is cleared to 0 by user code to configure the GPIO pin assigned to P2.0 as an input. This bit is set to 1 by user code to configure the GPIO pin assigned to P2.0 as an output.
23	Reserved, this bit is reserved and should be written as 0 by user code
22	Port 2.6 Data Output The value written to this bit appears directly on the GPIO pin assigned to P2.6
21	Port 2.5 Data Output The value written to this bit appears directly on the GPIO pin assigned to P2.5.
20-18	Reserved These bits are reserved and should be written as 0 by user code
17	Port 2.1 Data Output The value written to this bit appears directly on the GPIO pin assigned to P2.1.
16	Port 2.0 Data Output The value written to this bit appears directly on the GPIO pin assigned to P2.0.
15-7	Reserved, these bits are reserved and should be written as 0 by user code
6	Port 2.6 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P2.6. User code should write 0 to this bit.
5	Port 2.5 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P2.5. User code should write 0 to this bit.
4	Port 2.4 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P2.4. User code should write 0 to this bit.
3-2	Reserved, these bits are reserved and should be written as 0 by user code
1	Port 2.1 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P2.1. User code should write 0 to this bit.
0	Port 2.0 Data Input This bit is a read-only bit that reflects the current status of the GPIO pin assigned to P2.0. User code should write 0 to this bit.

GPIO Port0 Set Register:

Name: GP0SET

Address: 0xFFFF0D24

Access: Write only

Function: This 32-bit MMR allow user code to individually bit address external GPIO pins to set them high only. User code can do this via the GP0SET MMR without having to modify or maintain the status of any other GPIO pins as user code would need to do when using GP0DAT.

Table 59. GP0SET MMR Bit Descriptions

Bit	Description
31-21	Reserved These bits are reserved and should be written as 0 by user code
20	Port 0.4 Set Bit This bit is set to 1 by user code to set the external GPIO_4 pin high. If user software clears this bit to 0, this will have no effect on the external GPIO_4 pin.
19	Port 0.3 Set Bit This bit is set to 1 by user code to set the external GPIO_3 pin high. If user software clears this bit to 0, this will have no effect on the external GPIO_3 pin.
18	Port 0.2 Set Bit This bit is set to 1 by user code to set the external GPIO_2 pin high. If user software clears this bit to 0, this will have no effect on the external GPIO_2 pin.
17	Port 0.1 Set Bit This bit is set to 1 by user code to set the external GPIO_1 pin high. If user software clears this bit to 0, this will have no effect on the external GPIO_1 pin.
16	Port 0.0 Set Bit This bit is set to 1 by user code to set the external GPIO_0 pin high If user software clears this bit to 0, this will have no effect on the external GPIO_0 pin.
15-0	Reserved These bits are reserved and should be written as 0 by user code

GPIO Port1 Set Register:

Name: GP1SET

Address: 0xFFFF0D34

Access: Write only

Function: This 32-bit MMR allow user code to individually bit address external GPIO pins to set them high only. User code can do this via the GP1SET MMR without having to modify or maintain the status of any other GPIO pins as user code would need to do when using GP1DAT.

Table 60. GP1SET MMR Bit Descriptions

Bit	Description
31-18	Reserved These bits are reserved and should be written as 0 by user code
17	Port 1.1 Set Bit This bit is set to 1 by user code to set the external GPIO_6 pin high. If user software clears this bit to 0, this will have no effect on the external GPIO_6 pin.
16	Port 1.0 Set Bit This bit is set to 1 by user code to set the external GPIO_5 pin high If user software clears this bit to 0, this will have no effect on the external GPIO_5 pin.
15-0	Reserved These bits are reserved and should be written as 0 by user code

GPIO Port2 Set Register:

Name: GP2SET

Address: 0xFFFF0D44

Access: Write only

Function: This 32-bit MMR allow user code to individually bit address external GPIO pins to set them high only. User code can do this via the GP2SET MMR without having to modify or maintain the status of any other GPIO pins as user code would need to do when using GP2DAT.

Table 61. GP2SET MMR Bit Descriptions

Bit	Description
31-23	Reserved These bits are reserved and should be written as 0 by user code
22	Port 2.6 Set Bit This bit is set to 1 by user code to set the external GPIO_13 pin high. If user software clears this bit to 0, this will have no effect on the external GPIO_13 pin.
21	Port 2.5 Set Bit This bit is set to 1 by user code to set the external GPIO_12 pin high If user software clears this bit to 0, this will have no effect on the external GPIO_12 pin.
20-18	Reserved These bits are reserved and should be written as 0 by user code
17	Port 2.1 Set Bit This bit is set to 1 by user code to set the external GPIO_8 pin high. If user software clears this bit to 0, this will have no effect on the external GPIO_8 pin.
16	Port 2.0 Set Bit This bit is set to 1 by user code to set the external GPIO_7 pin high If user software clears this bit to 0, this will have no effect on the external GPIO_7 pin.
15-0	Reserved These bits are reserved and should be written as 0 by user code

GPIO Port0 Clear Register:

Name: GP0CLR

Address: 0xFFFF0D28

Access: Write only

Function: This 32-bit MMR allows user code to individually bit address external GPIO pins to clear them low only. User code can do this via the GP0CLR MMR without having to modify or maintain the status of any other GPIO pins as user code would need to do when using GP0DAT

Table 62. GP0CLR MMR Bit Descriptions

Bit	Description
31-21	Reserved These bits are reserved and should be written as 0 by user code
20	Port 0.4 Clear Bit This bit is set to 1 by user code to clear the external GPIO_4 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_4 pin.
19	Port 0.3 Clear Bit This bit is set to 1 by user code to clear the external GPIO_3 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_3 pin.
18	Port 0.2 Clear Bit This bit is set to 1 by user code to clear the external GPIO_2 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_2 pin.
17	Port 0.1 Clear Bit This bit is set to 1 by user code to clear the external GPIO_1 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_1 pin.
16	Port 0.0 Clear Bit This bit is set to 1 by user code to clear the external GPIO_0 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_0 pin.
15-0	Reserved These bits are reserved and should be written as 0 by user code

GPIO Port1 Clear Register:

Name: GP1CLR

Address: 0xFFFF0D38

Access: Write only

Function: This 32-bit MMR allows user code to individually bit address external GPIO pins to clear them low only. User code can do this via the GP1CLR MMR without having to modify or maintain the status of any other GPIO pins as user code would need to do when using GP1DAT.

Table 63. GP1CLR MMR Bit Descriptions

Bit	Description
31-18	Reserved These bits are reserved and should be written as 0 by user code
17	Port 1.1 Clear Bit This bit is set to 1 by user code to clear the external GPIO_6 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_6 pin.
16	Port 1.0 Clear Bit This bit is set to 1 by user code to clear the external GPIO_5 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_5 pin.
15-0	Reserved These bits are reserved and should be written as 0 by user code

GPIO Port2 Clear Register:

Name: GP2CLR

Address: 0xFFFF0D48

Access: Write only

Function: This 32-bit MMR allows user code to individually bit address external GPIO pins to clear them low only. User code can do this via the GP2CLR MMR without having to modify or maintain the status of any other GPIO pins as user code would need to do when using GP2DAT.

Table 64. GP2CLR MMR Bit Descriptions

Bit	Description
31-23	Reserved These bits are reserved and should be written as 0 by user code
22	Port 2.6 Clear Bit This bit is set to 1 by user code to clear the external GPIO_13 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_8 pin.
21	Port 2.5 Clear Bit This bit is set to 1 by user code to clear the external GPIO_12 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_7 pin.
20-18	Reserved These bits are reserved and should be written as 0 by user code
17	Port 2.1 Clear Bit This bit is set to 1 by user code to clear the external GPIO_8 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_8 pin.
16	Port 2.0 Clear Bit This bit is set to 1 by user code to clear the external GPIO_7 pin low. If user software clears this bit to 0, this will have no effect on the external GPIO_7 pin.
15-0	Reserved These bits are reserved and should be written as 0 by user code

HIGH VOLTAGE PERIPHERAL CONTROL INTERFACE

The ADuC7030/ADuC7033 integrates a number of high voltage circuit functions, which are controlled and monitored via a registered interface consisting of 2 MMRs, namely, HVCON and HVDAT. The HVCON register acts as a command byte interpreter allowing the microcontroller to indirectly read or write 8-bit data (the value in HVDAT) from/to one of 4 High voltage status/configuration registers. It should be noted that these high voltage registers are not MMRs but are so called 'indirect' registers that can only be accessed (as the name suggests) indirectly via the HVCON and HVDAT MMRs.

The physical interface between the HVCON register and the indirect high voltage registers is a 2-wire (data and clock) serial interface based on a 2.56MHz serial clock. Therefore, there is a finite, 10µsecs (maximum) latency between the MCU core writing a command into HVCON and that command or data reaching the indirect high voltage registers. There is also a finite 10µsecs latency between the MCU core writing a command into HVCON and indirect register data being read back into the HVDAT register. A busy bit (Bit0 of the HVCON when read by MCU) can be polled by the MCU to confirm when a read/write command has completed.

The following high voltage circuit functions are controlled and monitored via this interface and Figure 36 below describes the top-level architecture of the high voltage interface and related circuits.

Precision Oscillator
Wake-Up pin functionality
Power Supply Monitor
Low Voltage Flag
LIN Operating Modes
STI Diagnostics
High Voltage Diagnostics
High Voltage Attenuator/Buffer Circuit
High Voltage Temperature Monitor

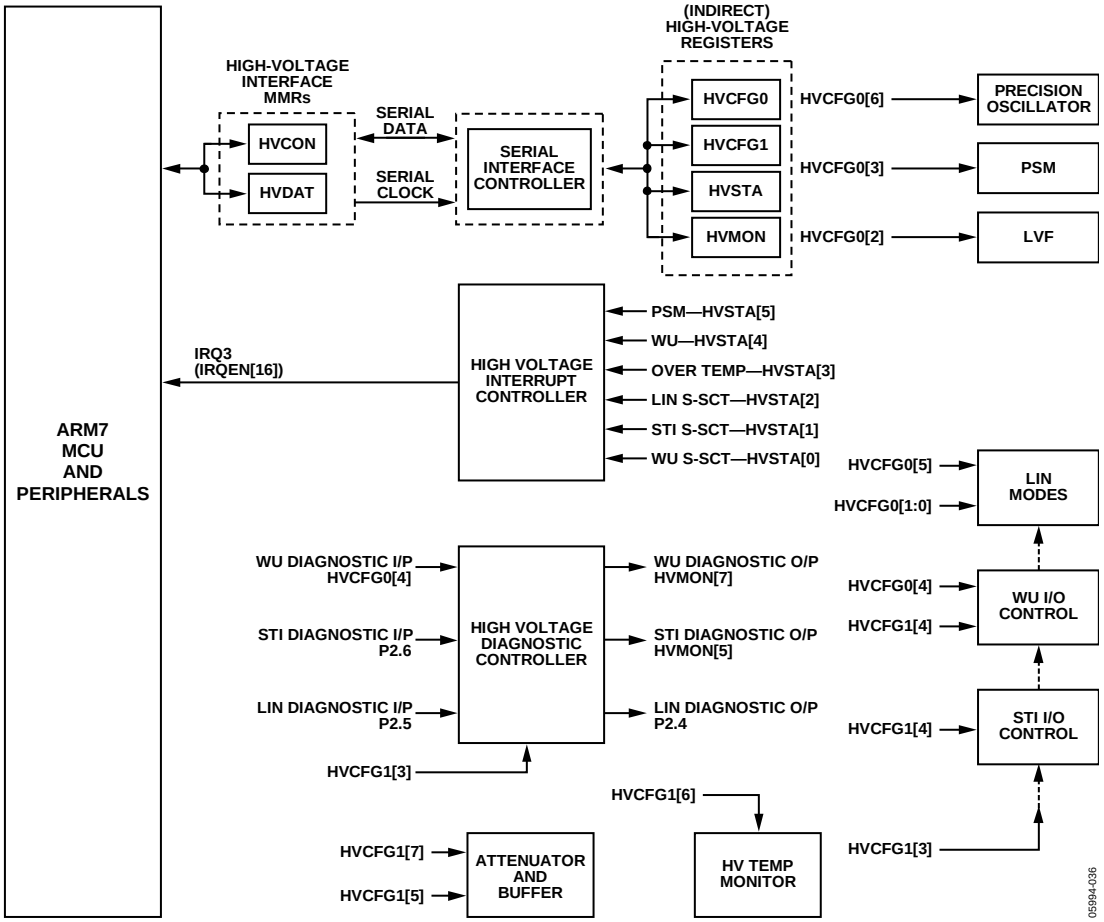


Figure 36. High Voltage Interface, Top Level Block Diagram

High Voltage Interface Control Register:

Name: HVCON

Address: 0xFFFF0804

Default Value: Updated by kernel

Access: Read/Write

Function: This 8-bit register acts as a command byte interpreter for the high voltage control interface. Bytes written to this register are interpreted as read or write commands to a set of 4 indirect registers related to the high voltage circuits. The HVDAT register is used to store data to be written to or read back from the indirect registers

Table 65. HVCON MMR Write Bit Designations

Bit	Description	
7-0	Command Byte	Interpreted as
	0x00	Read back high voltage register HVCFG0 into HVDAT
	0x01	Read back high voltage register HVCFG1 into HVDAT
	0x02	Read back high voltage status register HVSTA into HVDAT
	0x03	Read back high voltage status register HVMON into HVDAT
	0x08	Write the value in HVDAT to the high voltage register HVCFG0
	0x09	Write the value in HVDAT to the high voltage register HVCFG1

Table 66. HVCON MMR Read Bit Designations

Bit	Description
7-3	Reserved
2	Transmit Command to High Voltage Die Status: 1 Command Completed Successfully 0 Command Failed
1	Read Command from High Voltage Die Status: 1 Command Completed Successfully 0 Command Failed
0	Bit 0 (Read Only) BUSY Bit When user code reads this register, Bit0 should be interpreted as the BUSY signal for the high-voltage interface. This bit can be used to determine if a read request has completed. High voltage (read/write) commands as described above should not be written to HVCON unless BUSY=0. BUSY = 1, High voltage interface is busy and has not completed the previous command written to HVCON. Bit 1 and bit 2 are not valid. BUSY = 0, High voltage interface is not busy and has completed the command written to HVCON. Bit 1 and bit 2 are valid.

High Voltage Data Register:

Name: HVDAT

Address: 0xFFFF080C

Default Value: Updated by kernel

Access: Read/Write

Function: HVDAT is a 12-bit register that is used to hold data to be written indirectly to and read indirectly from the following high voltage interface registers.

Table 67. HVDAT MMR Bit Designations

Bit	Description
11-8	Command to which High Voltage Data, HVDAT[7-0], is associated with. These bits are read only and should be written as zeros. 0x00 Read back high voltage register HVCFG0 into HVDAT 0x01 Read back high voltage register HVCFG1 into HVDAT 0x02 Read back high voltage status register HVSTA into HVDAT 0x03 Read back high voltage status register HVMON into HVDAT 0x08 Write the value in HVDAT to the high voltage register HVCFG0 0x09 Write the value in HVDAT to the high voltage register HVCFG1
7-0	High Voltage Data to Read/Write

High Voltage Configuration0 Register:

Name: HVCFG0

Address: Indirectly addressed via the HVCON high voltage interface

Default Value: 0x00

Access: Read/Write

Function: This 8-bit register controls the function of high voltage circuits on the ADuC7030/ADuC7033. This register is not an MMR and does not appear in the MMR memory map. It is accessed via the HVCON registered interface. Data to be written to this register is loaded via the HVDAT MMR and data is read back from this register via the HVDAT MMR.

Table 68. HVCFG0 Bit Designations

Bit	Description		
7	Wake/STI Thermal Shutdown Disable: This bit is set to 1 to disable the automatic shutdown of the Wake/STI driver when a thermal event occurs. This bit is cleared to 0 to enable the automatic shutdown of the Wake/STI driver when a thermal event occurs.		
6	Precision Oscillator Enable Bit This bit is set to 1 to enable the Precision, 131kHz oscillator. The oscillator start-up time is typically 70µsecs (including HV interface latency of 10µsecs) This bit is cleared to 0 to power down the Precision, 131kHz oscillator		
5	BSD Mode Enable Bit This bit is cleared to 0 to enable an internal (LIN) pull-up resistor on the LIN/BSD pin This bit is set to 1 to disable the internal (LIN) pull-up and configure the LIN/BSD pin for BSD operation		
4	WU Assert Bit This bit is set to 1 to assert the external WU pin high. This bit is cleared to 0 to pull the external WU pin low via an internal 10KΩ pull-down resistor.		
3	PSM Enable Bit This bit is cleared to 0 to disable the Power Supply (Voltage at the VDD pin) Monitor This bit is set to 1 to enable the Power Supply (Voltage at the VDD pin) Monitor. If IRQ3 (IRQEN[16] is enabled the PSM will generate an interrupt if the voltage at the VDD pin drops below 6.0V.		
2	Low Voltage Flag Enable Bit This bit is cleared to 0 to disable the Low Voltage Flag function This bit is set to 1 to enable the Low Voltage Flag function. The Low Voltage Flag can be interrogated via HVMON[3] after power up to determine if the REG_DVDD voltage previously dropped below 2.1V		
1-0	LIN Operating Mode These bits enable/disable the LIN driver.		
	0	0	LIN Disabled
	0	1	Reserved – (not LIN V2.0 compliant)
	1	0	LIN Enabled
	1	1	LIN enabled, fast mode (>20kBd) - not LIN compliant

High Voltage Configuration1 Register:

Name: HVCFG1

Address: Indirectly addressed via the HVCON high voltage interface

Default Value: 0x00

Access: Read/Write

Function: This 8-bit register controls the function of high voltage circuits on the ADuC7030/ADuC7033. This register is not an MMR and does not appear in the MMR memory map. It is accessed via the HVCON registered interface, data to be written to this register is loaded via HVDAT and data is read back from this register via HVDAT.

Table 69. HVCFG1 Bit Designations

Bit	Description
7	Attenuator Enable Bit This bit is cleared to 0 to disable the internal voltage attenuator and attenuator buffer. This bit is set to 1 to enable the internal voltage attenuator and attenuator buffer.
6	High Voltage Temperature Monitor The high voltage temperature monitor is an un-calibrated temperature monitor located on-chip close to the high voltage circuits. This monitor is completely separate to the on-chip, precision temperature sensor (controlled via ADC1CON[7,6]) and allows user code to monitor die temperature change close the hottest part of the ADuC7030/ADuC7033 die. The monitor generates a typical output voltage of 600mV at 25°C and has a negative temperature coefficient of typically -2.1mV/°C This bit is set to 1 to enable the on-chip, high voltage temperature monitor. Once enabled this voltage out temperature monitor is routed directly to the voltage channel ADC. This bit is cleared to 0 to disable the on-chip, high voltage temperature monitor.
5	Voltage Channel Short Enable Bit This bit is set to 1 to enable an internal short (at the attenuator, before ADC input buffer) on the voltage channel ADC and allow noise be measured as a self-diagnostic test. This bit is cleared to 0 to disable an internal short on the voltage channel.
4	WU and STI Read Back Enable Bit This bit is cleared to 0 to disable input capability on the external WU/STI pin This bit is set to 1 to enable input capability on the external WU/STI pin. In this mode, a rising or falling edge transition on the WU/STI pin will generate a high voltage interrupt. Once this bit is set, the state of the WU/STI pin can be monitored via the HVMON register (HVMON[7] and HVMON[5]).
3	HV-IO Driver Enable Bit This bit is set to 1 to re-enable any High Voltage-IO pins (LIN/BSD/STI/WU) that have been disabled as a result of an short circuit current event(event must last longer than 20µsecs for LIN/BSD/STI Pins and 400usecs for WU Pin). This bit must also be set to 1 to re-enable the WU/STI pins if disabled by a thermal event. It should be noted that this bit must be set to clear any pending interrupt generated by the short circuit event (even if the event has passed) as well as re-enabling the High-Voltage IO pins. This bit is cleared to 0 automatically.
2	Enable/Disable Short Circuit Protection (LIN/BSD & STI) This bit is set to 1 to enable ***'passive'*** short circuit protection on LIN pin. In this mode, a short circuit event on the LIN/BSD pin will generate a HV interrupt (IRQ3-IRQEN[16]), assert the appropriate status bit in HVSTA but will NOT disable the short circuiting pin. This bit is cleared to 0 to enable ***'active'*** short circuit protection on LIN/BSD pin. In this mode, a short circuit event the LIN/BSD pin will generate a HV interrupt (IRQ3-IRQEN[16]), assert the appropriate status bit in HVSTA and automatically disable the short circuiting pin. Once disabled, the I/O pin can only be re-enabled by writing to HVCFG1[3].
1	WU Pin Time-Out (MonoFlop) Counter Enable/Disable This bit is set to disable the WU I/O time-out counter. This bit is cleared to enable a time-out counter which automatically de-asserts the WU pin 1.3 seconds after user code has asserted the WU pin via HVCFG0[4].
0	WU O/C Diagnostic Enable This bit is set to enable an internal WU I/O diagnostic pull-up resistor to the VDD pin thus allowing detection of an O/C condition on the WU pin. This bit is cleared to disable an internal WU I/O diagnostic pull-up resistor

High Voltage Monitor Register:

Name: HVMON

Address: Indirectly addressed via the HVCON high voltage interface

Default Value: 0x00

Access: Read Only

Function: This 8-bit read only register reflects the current status of enabled high voltage related circuits and functions on the ADuC7030/33. This register is not an MMR and does not appear in the MMR memory map. It is accessed via the HVCON registered interface, and data is read back from this register via HVDAT.

Table 70. HVMON Bit Designations

Bit	Description
7	WU Pin Diagnostic Read-back Once enabled via HVCFG1[4], this read only bit will reflect the state of the external WU pin.
6	Over Temperature This bit will be 0 if a thermal shutdown event has not occurred. This bit will be 1 if a thermal shutdown event has occurred.
5	STI Pin Diagnostic Read-back Once enabled via HVCFG1[4], this read only bit will reflect the state of the external STI pin.
4	Buffer Enabled This bit will be 0 if the Voltage Channel ADC input buffer is disabled This bit will be 1 if the Voltage Channel ADC input buffer is enabled
3	Low Voltage Flag Status Bit (Only valid If enabled via HVCFG0[2]) This bit will be 0 on power-on if REG_DVDD had dropped below 2.1V. In this state, RAM contents can be deemed corrupt. This bit will be 1 on power-on if REG_DVDD had not dropped below 2.1V. In this state, RAM contents can be deemed valid. It will only be cleared by re-enabling the Low Voltage Flag in HVCFG0[2]
2	LIN/BSD Short Circuit Status Flag: This bit will be 0 if the LIN/BSD driver is operating normally. This bit will be 1 if the LIN/BSD driver has experienced a short circuit condition and will be cleared automatically by writing to HVCFG1[3].
1	STI Short Circuit Status Flag: This bit will be 0 if the STI driver is operating normally. This bit will be 1 if the STI driver has experienced a short circuit condition and is cleared automatically by writing to HVCFG1[3].
0	Wake Short Circuit Status Flag: This bit will be 0 if the Wake driver is operating normally. This bit will be 1 if the Wake driver has experienced a short circuit condition.

High Voltage Status Register:

Name: HVSTA

Address: Indirectly addressed via the HVCON high voltage interface

Default Value: 0x00

Access: Read Only. This register should only be read on a high voltage interrupt.

Function: This 8-bit read only register reflects a change of state of all the corresponding bit in the HVMON register. This register is not an MMR and does not appear in the MMR memory map. It is accessed via the HVCON registered interface, and data is read back from this register via HVDAT. It should be noted that in response to a high voltage interrupt event, the high voltage interrupt controller simultaneously and automatically loads the current value of the high voltage Status register (HVSTA) into the HVDAT register.

Table 71. HVSTA Bit Designations

Bit	Description
7-6	Reserved These bits should not be used and are reserved for future use.
5	PSM Status Bit (Only valid If enabled via HVCFG0[3]) This bit is 0 if the voltage at the VDD pin stays above 6.0V This bit will be 1 if the voltage at the VDD pin drops below 6.0V. Please note that this bit is not latched and the IRQ needs to be enabled to detect it.
4	WU Request Status Bit (Only valid If enabled via HVCFG1[4]) Once enabled via HVCFG1[4], this bit will be set to 1 to indicate that a rising or falling edge transition on the WU pin generated a high voltage interrupt.
3	Over Temperature (Always enabled) This bit will be 0 if a thermal shutdown event has not occurred. This bit will be 1 if a thermal shutdown event has occurred. All high voltage (LIN/BSD, WU & STI) pin drivers will be automatically disabled once a thermal shutdown has occurred.
2	LIN/BSD Short Circuit Status Flag This bit will be 0 during normal LIN/BSD operation and is cleared automatically by reading the HVSTA register. This bit will be 1 if a LIN/BSD short circuit is detected. In this condition, the LIN driver will be automatically disabled.
1	STI Short Circuit Status Flag: This bit will be 0 if the STI driver is operating normally and is cleared automatically by reading the HVSTA register. This bit will be 1 if the STI driver has experienced a short circuit condition.
0	Wake Short Circuit Status Flag This bit will be 0 during normal Wake operation This bit will be 1 if a Wake short circuit is detected.

WAKE-UP (WU)

The WU pin is a high voltage GPIO controlled via HVCON and HVDAT.

Wake-Up (WU) Pin Circuit Description

The WU pin is configured by default as an output with an internal 10KΩ pull-down resistor and high side FET driver. The WU pin in its default mode of operation is specified to generate an active high system wake-up request by forcing the external system WU bus high. User code can assert the WU output by writing directly to HVCFG0[4].

It should be noted the output will only respond after the 10μsecs latency through the (serial communication based) high voltage interface.

The internal FET is capable of sourcing significant current and therefore a substantial on-chip self-heating can occur if this driver is asserted for a long time period. For this reason a Monoflop, a 1.3-second timeout timer, has been included. By default the Monoflop is enabled and will disable the Wake Driver after 1.3 seconds. It is possible to disable the Monoflop via HVCFG1[1]. If the Wake Monoflop is disabled, then the Wake driver should be disabled after 1.3s.

The WU Pin also features a short circuit detection feature. When the wake Pin sources more than 100mA typically for 400μS a high voltage interrupt will be generated with HVMON[0] set.

A thermal shutdown event disables the WU Driver. The WU driver must be re-enabled manually after a thermal event via HVCFG1[3].

The WU pin can be configured in I/O mode by writing a 1 to HVCFG1[4]. In this mode, a rising or falling edge will immediately generate a high voltage interrupt. HVMON[7] directly reflects the state of the external WU pin. This comparator has a trip level of $3V_{TYP}$.

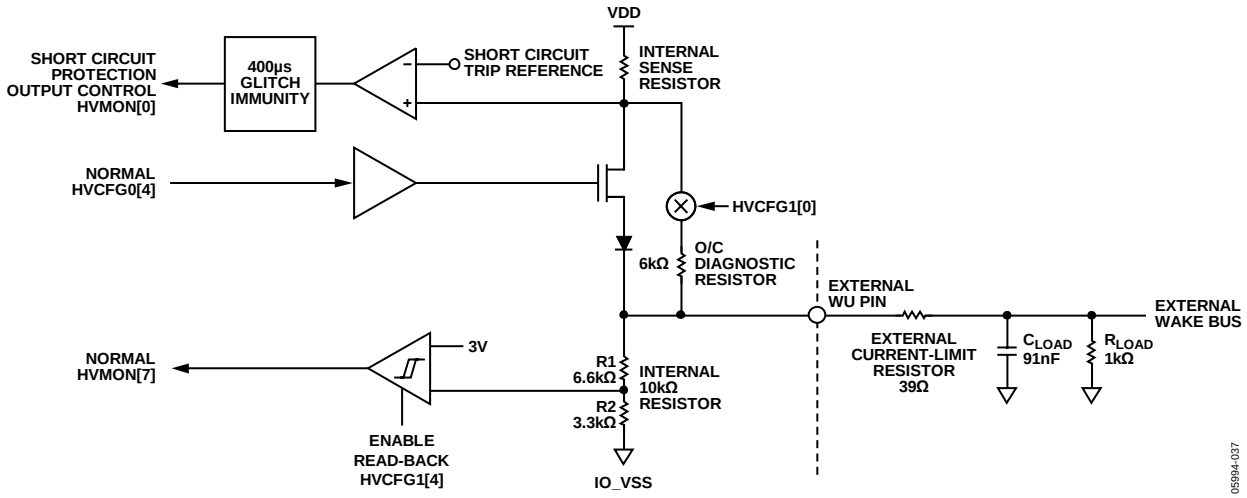


Figure 37. WU Circuit, Block Diagram

HANDLING INTERRUPTS FROM THE HIGH VOLTAGE PERIPHERAL CONTROL INTERFACE

An interrupt controller is also integrated with the high voltage circuits. If enabled via IRQEN[16], one of 6 high voltage sources can assert the high voltage interrupt (IRQ3) signal and interrupt the MCU core.

While the MCU response to this interrupt event is, as normal, to vector to the IRQ or FIQ interrupt vector address. The high voltage interrupt controller simultaneously and automatically loads the current value of the high voltage Status register (HVSTA) into the HVDAT register. During this time the BUSY bit in HVCON[0] is set to indicate the transfer is in progress and will be cleared after 10μsecs to indicate the HVSTA contents are now available in HVDAT.

The interrupt handler can therefore poll the busy bit in HVCON until it de-asserts. Once the busy bit is cleared, HVCON[1] must be checked to ensure the data was read correctly. Then the HVDAT register can be read. At this time HVDAT will then hold the value of the HVSTA register. The status flags can then be interrogated to determine the exact source of the high voltage interrupt and the appropriate action taken.

LOW VOLTAGE FLAG (LVF)

The ADuC7030 features a Low Voltage Flag (LVF), which when enabled allows the user to monitor REG_DVDD. When enabled via HVCFG0[2], the Low Voltage Flag may be monitored via HVMON[3]. If REG_DVDD drops below 2.1V, then HVMON[3] is cleared. If REG_DVDD drops below 2.1V the RAM contents are corrupted. Once the Low Voltage Flag is enabled, it is only reset by REG_DVDD dropping below 2.1V or the disabling of the LVF functionality via HVCFG0[2].

HIGH VOLTAGE DIAGNOSTICS

It is possible to diagnosis fault conditions on the Wake, LIN and STI bus as follows.

Table 72. High Voltage Diagnostics

High Voltage Pin	Fault Condition	Method	Result
LIN/STI	Short Between LIN/STI and VBAT	Drive LIN Low	LIN/STI Short Circuit interrupt will be generated after 20μS if more than 100mA is drawn continuously.
	Short Between LIN/STI and GND	Drive LIN High	LIN/STI Read back reads back low.
Wake	Short Between Wake and VBAT	Drive Wake Low	Read Back high in HVMON[7]
	Short Between Wake and GND	Drive Wake High	Wake Short Circuit interrupt will be generated after 400us if more than 100mA typically is sourced
	Open Circuit	Enable OC Diagnostic Resistor with Wake disabled.	HVMON[7] will be cleared if load is connected and set if wake is open circuited

UART SERIAL INTERFACE

The ADuC7030/ADuC7033 features a 16450 compatible UART. The UART is a full-duplex Universal Asynchronous Receiver/Transmitter. A UART performs serial-to-parallel conversion on data characters received from a peripheral device, and parallel-to-serial conversion on data characters received from the ARM7TDMI. The UART features a fractional divider, which facilitates high accuracy baud rate generation, and a network addressable mode. The UART functionality is made available on GPIO_5, RXD, and GPIO_6, TXD, of the ADuC7030/ADuC7033.

The serial communication adopts a asynchronous protocol that supports various word length, stop bits and parity generation options selectable in the configuration register.

BAUD RATE GENERATION

The ADuC7030/ADuC7033 features two methods of generating the UART Baud Rate:

1. Normal 450 UART baud rate generation.
2. ADuC7030/ADuC7033 Fractional Divider

These two methods are explained in detail below.

Normal 450 UART Baud Rate Generation

The baud rate is a divided version of the core clock using the value in COMDIV0 and COMDIV1 MMRs (16-bit value, DL). The standard baud rate generator formula is

$$Baudrate = \frac{20.48MHz}{2^{CD} \times 16 \times 2 \times DL} \quad (1)$$

Table 73 gives some common baud rate values:

Table 73. Baud rate using the standard Baud rate generator

Baud rate	CD	DL	Actual Baud rate	% error
9600	0	0x43	9552	0.50%
19200	0	0x21	19394	1.01%
115200	0	0x6	106667	7.41%
9600	3	0x8	10000	4.17%
19200	3	0x4	20000	4.17%
115200	3	0x1	80000	30.56%

ADuC7030/ADuC7033 Fractional divider:

The fractional divider combined with the normal baud-rate generator allows the generation of accurate, high speed, baud-rates.

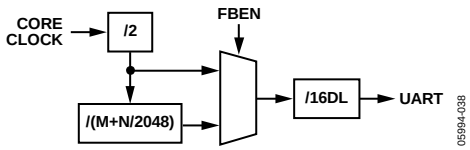


Figure 38. Fractional Divider Baud Rate generation

Calculation of the baud rate using fractional divider is as follows:

$$Baudrate = \frac{20.48MHz}{2^{CD} \times 16 \times DL \times 2 \times (M + \frac{N}{2048})}$$

$$M + \frac{N}{2048} = \frac{20.48MHz}{Baudrate \times 2^{CD} \times 16 \times DL \times 2}$$

Table 74 gives some common baud rate values.

Table 74. Baud rate using the Fractional Baud rate generator

Baud rate	CD	DL	M	N	Actual Baud rate	% error
9600	0	42h	1	21	9598.55	0.015%
19200	0	21h	1	21	19197.09	0.015%
115200	0	5h	1	228	115177.51	0.0195%

UART REGISTER DEFINITION

The UART interface consists of 9 registers namely:

- COMTX: 8-bit transmit register

- COMRX: 8-bit receive register

- COMDIV0: divisor latch (low byte)

COMTX, COMRX and COMDIV0 share the same address location. COMTX and COMTX can be accessed when bit 7 in COMCON0 register is cleared. COMDIV0 can be accessed when bit 7 of COMCON0 is set.

- COMDIV1: divisor latch (high byte)

- COMCON0: line control register

- COMSTA0: line status register

- COMIEN0: interrupt enable register

- COMIID0: interrupt identification register

- COMDIV2: 16-bit fractional baud divide register

UART TX Register:

Name: COMTX

Address: 0xFFFF0700

Access: Write Only

Function: This 8-bit register is written to, to transmit data via the UART.

UART RX Register:

Name: COMRX

Address: 0xFFFF0700

Default Value: 0x00

Access: Read Only

Function: This 8-bit register is read from to receive data transmitted via the UART.

UART Divisor Latch Register 0:

Name: COMDIV0

Address: 0xFFFF0700

Default Value: 0x00

Access: Read/Write

Function: This 8-bit register contains the least significant byte of the divisor latch witch controls the baud rate at which the UART operates.

UART Divisor Latch Register 1:

Name: COMDIV1

Address: 0xFFFF0704

Default Value: 0x00

Access: Read/Write

Function: This 8-bit register contains the most significant byte of the divisor latch witch controls the baud rate at which the UART operates.

UART Control Register 0:

Name: COMCON0

Address: 0xFFFF070C

Default Value: 0x00

Access: Read/Write

Function: This 8-bit register controls the operation of the UART in conjunction with COMCON1.

Table 75. COMCON0 MMR Bit Descriptions

Bit	Name	Description
7	DLAB	Divisor latch access Set by user to enable access to COMDIV0 and COMDIV1 registers Cleared by user to disable access to COMDIV0 and COMDIV1 and enable access to COMRX, COMTX and COMIEN0
6	BRK	Set break. Set by user to force TXD to 0 Cleared to operate in normal mode
5	SP	Stick parity Set by user to force parity to defined values: 1 if EPS = 1 and PEN = 1 0 if EPS = 0 and PEN = 1
4	EPS	Even parity select bit Set for even parity Cleared for odd parity
3	PEN	Parity enable bit: Set by user to transmit and check the parity bit Cleared by user for no parity transmission or checking
2	STOP	Stop bit Set by user to transmit 1.5 Stop bit if the Word Length is 5 bits or 2 Stop bits if the word length is 6, 7 or 8 bits. The receiver checks the first Stop bit only, regardless of the number of Stop bits selected Cleared by user to generate 1 Stop bit in the transmitted data
1-0	WLS	Word length select: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

UART Control Register 1:

Name: COMCON1

Address: 0xFFFF0710

Default Value: 0x00

Access: Read/Write

Function: This 8-bit register controls the operation of the UART in conjunction with COMCON0.

Table 76. COMCON1 MMR Bit Descriptions

Bit	Name	Description		
7-6		UART Input Mux		
		00	RxD driven by LIN Input	Required for LIN Communications via LIN pin
		01	*Reserved*	
		10	RxD driven by GP5	Required for Serial communications via GP5 pin (RxD)
		11	*Reserved*	
5		*Reserved – not used*		
4	LOOPBACK	Loop back Set by user to enable loop back mode. In loop-back mode, the TxD is forced high.		
3-0		*Reserved – not used*		

UART Status Register 0:

Name: COMSTA0

Address: 0xFFFF0714

Default Value: 0x60

Access: Read Only

Function: This 8-bit read only register reflects the current status on the UART.

Table 77. COMSTA0 MMR Bit Descriptions

Bit	Name	Description
7		*Reserved*
6	TEMT	COMTX and shift register empty status bit Set automatically if COMTX and the shift register are empty. This bit indicates that the data has been transmitted, i.e. is no more present in the shift register. Cleared automatically when writing to COMTX
5	THRE	COMTX empty status bit. Set automatically if COMTX is empty. COMTX can be written is soon as this bit gets set, the previous data might not have been transmitted yet and still be present in the shift register. Cleared automatically when writing to COMTX
4	BI	Break Indicator Set when SIN is held low for more than the maximum word length Cleared automatically
3	FE	Framing error Set when invalid stop bit Cleared automatically
2	PE	Parity error Set when a parity error occurs Cleared automatically
1	OE	Overrun error Set automatically if data are overwrite before been read Cleared automatically
0	DR	Data ready Set automatically when COMRX is full Cleared by reading COMRX

UART Interrupt Enable Register 0:

Name: COMIEN0

Address: 0xFFFF0704

Default Value: 0x00

Access: Read/Write

Function: The 8-bit register enables/disables the individual UART interrupt sources

Table 78. COMIEN0 MMR Bit Descriptions

Bit	Name	Description
7-4		Reserved – not used
3	EDSSI	Reserved – should be written as 0
2	ELSI	RX status interrupt enable bit Set by user to enable generation of an interrupt if any of COMSTA0[3:0] 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 i.e. 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 79. COMIID0 MMR Bit Descriptions

Bit 2-1 Status bits	Bit 0 NINT	Priority	Definition	Clearing operation
00	1		No interrupt	
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 register

UART Fractional Divider Register:

Name: COMDIV2

Address: 0xFFFF072C

Default Value: 0x0000

Access: Read/Write

Function: This 16-bit register controls the operation of the ADuC7030/ADuC7033's fractional divider

Table 80. COMDIV2 MMR Bit Descriptions

Bit	Name	Description
15	FBEN	Fractional baud rate generator enable bit Set by user to enable the fractional baudrate generator Cleared by user to generate baudrate using the standard 450 UART baudrate generator
14-13		*Reserved*
12-11	FBM[1-0]	M. if FBM = 0, M = 4
10-0	FBN[10-0]	N

SERIAL PERIPHERAL INTERFACE

The ADuC7030 features a complete hardware Serial Peripheral Interface (SPI) on-chip. SPI is an industry standard synchronous serial interface, which allows eight bits of data to be synchronously transmitted and received simultaneously, i.e., full duplex.

The SPI interface is only operational 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, which are multiplexed with four GPIO. The four SPI pins are MISO, MOSI, SCLK and $\overline{SS}$. The pins to which this signals are connected are shown in Table 81.

Table 81. SPI Output Pins

Pin	Signal	Description
GP0 (GPIO MODE 1)	$\overline{SS}$	Chip Select
GP1 (GPIO MODE 1)	SCLK	Serial Clock
GP2 (GPIO MODE 1)	MISO	Master Out, Slave In
GP3 (GPIO MODE 1)	MOSI	Master In, Slave Out

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 follow:

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

Equation 1. SPI Baud Rate Calculation

The maximum speed of the SPI clock is dependant on the clock divider bits and is summarized in Table 82.

Table 82. SPI speed vs. clock divider bits in master mode

CD bits	0	1
SPIDIV	0x05	0x0B
Max SCLK (MHz)	1.667	0.833

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 is as follow:

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

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 are configured the same 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 will then transmit and receive 8-bit data until the transfer is concluded by de-assertion of $\overline{SS}$. In slave mode $\overline{SS}$ is always an input.

SPI registers definition

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

- SPITX: 8-bit write only transmit register

- SPIRX: 8-bit read only receive register

SPI Control Register:

Name: SPICON

Address: 0xFFFF0A10

Default Value: 0x0000

Access: Read/Write

Function: The 16-bit MMR configures the Serial Peripheral Interface.

Table 83. SPICON MMR Bit Descriptions

Bit	Description
15-13	*Reserved*
12	Continuous transfer enable *Set* by user to enable continuous transfer. In master mode, the transfer will continue until no valid data is available in the TX register. $\overline{SS}$ will be asserted and remain asserted for the duration of each 8-bit serial transfer until TX is empty *Cleared* by user to disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPITX register then a new transfer is initiated after a stall period
11	Loop back enable *Set* by user to connect MISO to MOSI and test software *Cleared* by user to be in normal mode
10	Slave output enable *Set* by user to enable the slave output *Cleared* by user to disable slave output
9	Slave select input enable *Set* by user in master mode to enable the output
8	SPIRX overflow overwrite enable Set by user, the valid data in the RX register is overwritten by the new serial byte received *Cleared* by user, the new serial byte received is discarded
7	SPITX underflow mode *Set* by user to transmit the previous data *Cleared* by user to transmit 0
6	Transfer and interrupt mode (master mode) *Set* by user to initiate transfer with a write to the SPITX register. Interrupt will occur when TX is empty *Cleared* by user to initiate transfer with a read of the SPIRX register. Interrupt will occur when RX is full
5	LSB first transfer enable bit *Set* by user the LSB is transmitted first *Cleared* by user the MSB is transmitted first
4	*Reserved*
3	Serial clock polarity mode bit *Set* by user, the serial clock idles high *Cleared* by user the serial clock idles low
2	Serial clock phase mode bit *Set* by user, the serial clock pulses at the beginning of each serial bit transfer *Cleared* by user, the serial clock pulses eat end of each serial bit transfer
1	Master mode enable bit *Set* by user to enable master mode *Cleared* by user to enable slave mode
0	SPI enable bit *Set* by user to enable the SPI *Cleared* to disable the SPI

SPI Status Register:

Name: SPISTA

Address: 0xFFFF0A00

Default Value: 0x00

Access: Read Only

Function: The 8-bit MMR represents the current status of the Serial Peripheral Interface.

Table 84. SPISTA MMR Bit Descriptions

Bit	Description
7-6	*Reserved*
5	SPIRX data register overflow status bit *Set* if SPIRX is overflowing *Cleared* by reading SPISRX register
4	SPIRX data register IRQ *Set* automatically if bit 3 or 5 are set *Cleared* by reading SPIRX register
3	SPIRX data register full status bit *Set* automatically if a valid data is present in the SPIRX register *Cleared* by reading SPIRX register
2	SPITX data register underflow status bit *Set* automatically if SPITX is under flowing *Cleared* by writing in the SPITX register
1	SPITX data register IRQ *Set* automatically if bit 0 is clear or bit 2 is set *Cleared* by writing in the SPITX register or if finished transmission disabling the SPI
0	SPITX data register empty status bit *Set* by writing to SPITX to send data. This bit is set during transmission of data *Cleared* when SPITX is empty

SPI Receive Register:

Name: SPIRX

Address: 0xFFFF0A04

Default Value: 0x00

Access: Read Only

Function: This 8-bit MMR contains the data received via the Serial Peripheral Interface.

SPI Transmit Register:

Name: SPITX

Address: 0xFFFF0A08

Access: Write Only

Function: This 8-bit MMR is written to, to transmit data via the Serial Peripheral Interface.

SPI Divider Register:

Name: SPIDIV

Address: 0xFFFF0A0C

Default Value: 0x1B

Access: Read/Write

Function: The 8-bit MMR represents the frequency of the Serial Peripheral Interface is operating at. For more information on the calculation of the baud rate, please refer to Equation 1.

SERIAL TEST INTERFACE

The ADuC7030/ADuC7033 incorporate a single pin, serial test interface (STI) port that can be used for end-customer evaluation or diagnostics on finished production units.

The STI port transmits from 1 to 6 bytes of data in 12-bit packets. As shown in Figure 39 below, each transmission packet includes a start bit, the transmitted byte (8 bits), an even parity bit and two stop bits. The STI data is transmitted on the STI pin and the baud rate is determined by the overflow rate of Timer4.

The STI port is configured and controlled via six MMRs, namely:

- STIKEY0: Serial Test Interface key 0.
- STIKEY1: Serial Test Interface key 1.
- STIDAT0: Data (16-Bit) 0 – holds 2 bytes
- STIDAT1: Data (16-Bit) 1 - holds 2 bytes
- STIDAT2: Data (16-Bit) 2 - holds 2 bytes
- STICON: Controls the Serial Test Interface

STI Key0 Register:

Name: STIKEY0

Address: 0xFFFF0880

Access: Write Only

Function: The STIKEY0 MMR is used in conjunction with the STIKEY1 MMR to protect the STICON MMR. STIKEY0 must be written with 0x0007 immediately before any attempt is made to write to STICON. STIKEY1 must be written with 0x00B9 immediately after STICON is written to ensure the STICON write sequence is completed successfully. If STIKEY0 is not written, is written out of sequence or is written incorrectly, any subsequent write to the STICON MMR will be ignored.

STI Key1 Register:

Name: STIKEY1

Address: 0xFFFF0888

Access: Write Only

Function: The STIKEY01 MMR is used in conjunction with the STIKEY0 MMR to protect the STICON MMR. STIKEY1 must be written with 0x00B9 immediately after any attempt is made to write to STICON. STIKEY0 must be written with 0x0007 immediately before STICON is written to ensure the STICON write sequence is completed successfully. If STIKEY1 is not written, is written out of sequence or is written incorrectly, any previous write to the STICON MMR will be ignored.

STI DATA0 Register:

Name: STIDAT0

Address: 0xFFFF088C

Default Value: 0x0000

Access: Read/Write

Function: The STIDAT0 MMR is a 16-bit register which is used to hold the first and second data bytes that will be transmitted on the STI pin once the STI port is enabled. The first byte to be transmitted will occupy bits0-7 and the second byte will occupy bits 8-15.

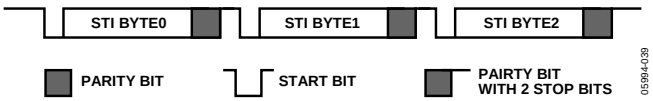


Figure 39. Serial ADC Test Interface Example- 3 Byte Transmission

STI DATA1 Register:

Name: STIDAT1

Address: 0xFFFF0890

Default Value: 0x0000

Access: Read/Write

Function: The STIDAT1 MMR is a 16-bit register which is used to hold the third and fourth data bytes that will be transmitted on the STI pin once the STI port is enabled. The third byte to be transmitted will occupy bits0-7 and the fourth byte will occupy bits8-15.

STI DATA2 Register:

Name: STIDAT1

Address: 0xFFFF0894

Default Value: 0x0000

Access: Read/Write

Function: The STIDAT2 MMR is a 16-bit register which is used to hold the fifth and sixth data bytes that will be transmitted on the STI pin once the STI port is enabled. The fifth byte to be transmitted will occupy bits0-7 and the sixth byte will occupy bits8-15.

STI Control Register:

Name: STICON

Address: 0xFFFF0884

Default Value: 0x0000

Access: Read/Write Access, write protected by 2 key registers (STIKEY0 and STIKEY1). A write access to STICON is only completed correctly if the following triple write sequence is followed

1. STIKEY0 MMR is written with 0x7
2. STICON is written
3. The sequence is completed by writing 0xB9 to STIKEY1

Function: The STI Control MMR is an 16-bit register that configures the mode of operation of the Serial Test Interface.

Note: GPIO_13 must be configured for STI operation in GP2CON for STI communications.

Table 85. STICON MMR Bit Descriptions

Bit	Description
16-9	Reserved These bits are reserved for future use and should be written as '0' by user code
8-5	State bits, read only If the interface is in the middle of a transmission, these bits will not be 0
4-2	Number of Bytes to Transmit These bits select the number of bytes that will be transmitted. User code must subsequently write the bytes to be transmitted into the STIDAT0,1 and 2 MMRs 0, 0, 0 1-byte Transmission 0, 0, 1 2-byte Transmission 0, 1, 0 3-byte Transmission 0, 1, 1 4-byte Transmission 1, 0, 0 5-byte Transmission 1, 0, 1 6-byte Transmission
1	Reset Serial Test Interface This bit is set to a 1 to reset the Serial Test Interface, a subsequent read of STICON will return all 0's This bit is set to a 0 by user code during normal Serial Test Interface operation, by power-on default
0	Serial Test Interface Enable This bit is set to a 1 by user code to enable the Serial Test Interface This bit is set to 0 by user code to disable the Serial Test Interface

Serial Test Interface Output Structure

The Serial Test Interface is a high voltage output, which incorporates a low side driver, short circuit protection and diagnostic pin read-back capability. The output driver circuit configuration is shown in Figure 40 below.

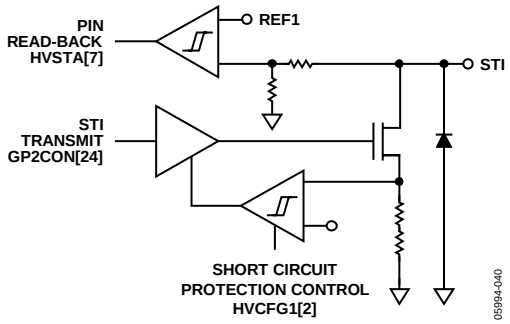


Figure 40. STI Output Structure

Using the Serial Test Interface

Data will only begin transmission once configuration of the STI port has been completed in the following sequence:

a. Configure Timer4 for baud rate generation

b. Correctly enable STICON using STIKEY0 and STIKEY1 for secure access

c. Required bytes to be transmitted are written into STIDAT0, STIDAT1 and STIDAT2.

Timer 4 is configured with the correct load value to generate an overflow at the required baud rate. If the STI port is being used to transmit ADC conversion results, then the baud rate must be sufficient to output each ADC result (16-bits) prior to next ADC conversion result being available.

For example, if the ADC is sampling at 1kHz, then the baud rate has to be sufficient to output 36 bits, 3 x 8 bits (16-bit ADC result and a checksum byte for example) + 3 x 1 Start bits + 3 x 1 Parity bits + 3 x 2Stop Bits = 36 bits. Therefore, the Serial Test Interface must transmit data at greater than 36kbps. The closest standard baud rate is 38.4kbps.

Therefore the reload value written to the Timer4 load MMR (T4LD) is 0106h(267d). This value is calculated as shown below and is based on a pre-scalar of 1, using a core clock of 10.24MHz.

$$T4LD = \frac{CoreClockFrequency}{DesiredBaudRate}$$

$$= 10.24MHz/\ 38.4kbps = 267$$

Once the Timer4 load value is written and the Timer itself is configured and enabled using the T4CON MMR, the STI port must be configured. The is done by writing to the STICON MMR in a specific sequence using the STIKEY0 and STIKEY1 MMRs as described earlier.

Finally, the STI port will not begin transmission until the required number of transmit bytes is written into the STIDATx MMRs. As soon as STI starts transmitting, the value in the STICON MMR will change from the value initially written to this register. User code can ensure that all data is transmitted by continuously polling the STICON MMR until it reverts back to the value originally written to it. To disable the serial interface, user code must write a 0 to STICON[0].

An example code segment configuring the STI port to transmit 5 bytes and then to transmit 2 bytes is shown below:

```
T4LD = 267;                     // Timer4 Reload Value
T4CON = 0xC0;                   // Enable T4, selecting core clock in periodic mode

STIKEY0 = 07;                   // STICON Start write sequence
STICON = 0x11;                  // Enable and transmit 5 Bytes
STIKEY1 = 0xb9;                 // STICON Complete write

STIDAT0 = 0xAABB;               // 5 bytes for
STIDAT1 = 0xCCDD;               // transmission
STIDAT2 = 0xFF;

while(STICON != 0x09)           // wait for transmission to complete
{}

STIKEY0 = 07;                   // STICON Start write sequence
STICON = 0x05;                  // Enable and transmit 2 Bytes
STIKEY1 = 0xb9;                 // STICON Complete write

STIDAT0 = 0xEEFF;               // 2 bytes for transmission

while(STICON != 0x09)           // wait for transmission to complete
{}
```

LIN (LOCAL INTERCONNECT NETWORK) INTERFACE

The ADuC7030/ADuC7033 features a high voltage physical interface between the ARM7 MCU core and an external LIN bus. The LIN interface operates as a slave only interface operating from 1-20KBaud, and is compatible with the LIN2.0 standard. The pull-up resistor required for a slave node is on-chip, reducing the need for external circuitry. The LIN protocol is emulated using the on-chip UART, an IRQ, a dedicated LIN Timer and the high voltage transceiver which is also incorporated on-chip. This is shown in Figure 41. The LIN is clocked from the Low Power Oscillator, for the Break Timer, and a 5MHz output from the PLL which is used for the synch byte timing.

LIN MMR DESCRIPTION

The LIN Hardware Synchronization (LHS) functionality is controlled via five MMRs. The function of each MMR is described below:.

LHSSTA: LHS Status Register. This MMR contains information flags which describe the current status on the interface.

LHSCON0: LHS Control Register 0. This MMR controls the configuration of the LHS Timer.

LHSCON1: LHS Start and Stop Edge Control Register Dictates which edge of the LIN Synchronization byte the LHS starts/stops counting.

LHSVAL0: LHS Synchronization 16Bit Timer, which is controlled by LHSCON0.

LHSVAL1: LHS Break Timer Register

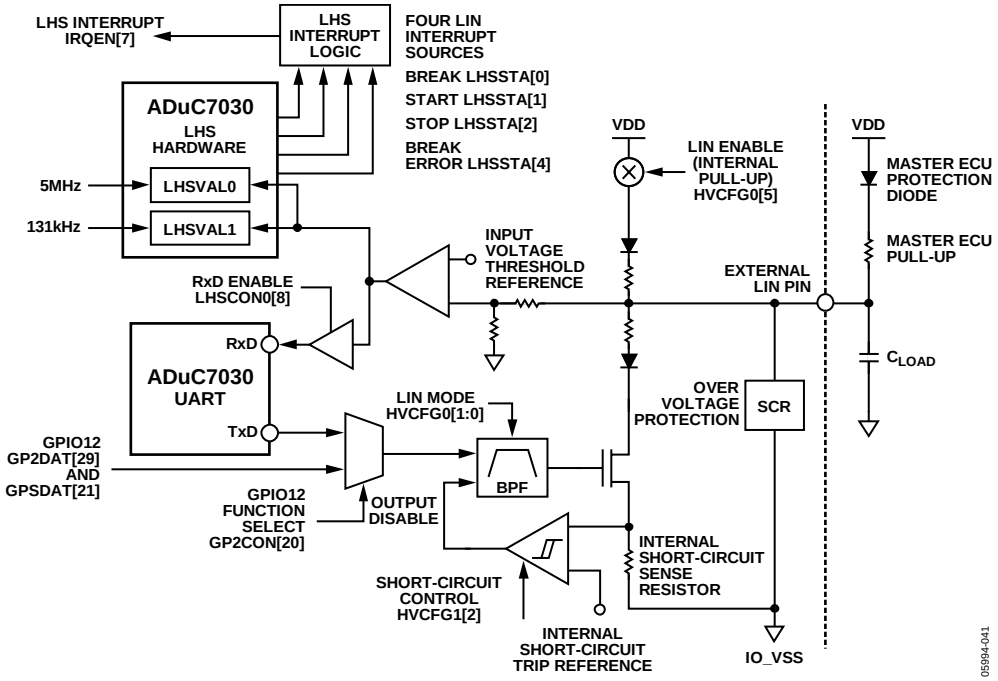


Figure 41. LIN I/O, Block Diagram

LIN Hardware Synchronization Status Register:

Name: LHSSTA

Address: 0xFFFF0780

Default Value: 0x00000000

Access: Read Only

Function: The LHS Status register is a 32-bit register whose bits reflect the current operating status of the ADuC7030/ADuC7033 LIN interface

Table 86. LHSSTA MMR Bit Descriptions

Bit	Description
31 - 7	Reserved These read-only bits are reserved for future use
6	Rising edge Detected (BSD Mode Only) This bit is set to 1 by hardware to indicate a rising edge has been detected on the BSD Bus. This bit is cleared to 0, after user code reads the LHSSTA MMR
5	LHS Reset Complete Flag This bit is set to 1 by hardware to indicate a LHS Reset Command has completed successfully. This bit is cleared to 0, after user code reads the LHSSTA MMR
4	Break Field Error This bit is set to 1 by hardware and generates an LHS Interrupt (IRQEN[7]) when the 12-bit, Break Timer (LHSVAL1) register overflows to indicate the LIN bus has stayed low too long thus indicating a possible LIN bus error. This bit is cleared to 0, after user code reads the LHSSTA MMR
3	LHS Compare Interrupt This bit is set to 1 by hardware when the value in LHSVAL0 (LIN Synchronization Bit Timer) = the value in the LHSCMP register. This bit is cleared to 0, after user code reads the LHSSTA MMR
2	Stop Condition Interrupt This bit is set to 1 by hardware when a stop condition is detected. This bit is cleared to 0, after user code reads LHSSTA MMR
1	Start Condition Interrupt This bit is set to 1 by hardware when a start condition is detected. This bit is cleared to 0, after user code reads LHSSTA MMR
0	Break Timer Compare Interrupt This bit is set to 1 by hardware when a valid LIN Break condition is detected. A LIN Break condition is generated when the LIN Break Timer value reaches the Break timer compare value (see LHSVAL1 description below). This bit is cleared to 0, after user code reads the LHSSTA MMR

LIN Hardware Synchronization Control Register 0:

Name: LHSCON0

Address: 0xFFFF0784

Default Value: 0x00000000

Access: Read/Write

Function: The LHS Control register is a 32-bit register that in conjunction with the LHSCON1 register is used to configure the LIN mode of operation

Table 87. LHSCON0 MMR Bit Descriptions

Bit	Description
31-13	Reserved These bits are reserved for future and should be written as 0 by user software.
12	Rising edge Detected Interrupt Disable BSD Mode: This bit is set to 1 to disable the Rising edge Detected interrupt. This bit is cleared to 0 to enable the Break Rising edge Detected Interrupt LIN Mode: This bit is set to 1 to enable the Rising edge Detected interrupt. This bit is cleared to 0 to disable the Break Rising edge Detected Interrupt
11	Break Timer Compare Interrupt Disable: This bit is set to 1 to disable the Break Timer Compare interrupt. This bit is cleared to 0 to enable the Break Timer Compare Interrupt
10	Break Timer Error Interrupt Disable: This bit is set to 1 to disable the Break Timer Error interrupt. This bit is cleared to 0 to enable the Break Timer Error Interrupt
9	LIN Transceiver, Stand-Alone Test Mode This bit is cleared to 0 by user code to operate the LIN in normal mode, driven directly from the on-chip UART. This bit is 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 bits <0 and 4> in the GP2CON register.
8	Gate UART/BSD R/$\overline{W}$ Bit In LIN Mode (LHSCON0[6] is cleared to 0): This bit is 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 will not receive any spurious serial data during Break or Sync field periods which will have to be flushed out of the UART before valid data fields can start to be received. This bit is 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. In BSD Mode (LHSCON0<6> is set to1): In BSD Read-Mode, this bit is set to 1 by user code to enable the generation of a Break Condition interrupt (LHSSTA[0]) on a rising edge of the BSD bus. In BSD read mode the Break Timer (LHSVAL1) starts counting on the falling edge and stops counting on the rising edge. The generate of an interrupt on this rising edge allows user code determine if a 0, 1 or Sync pulse width has been received. It should also be noted that the Break timer will also still generate an interrupt if the value in the LIN Break timer(LHSVAL1 read value) = the Break timer compare value (LHSVAL1 write value) and if the Break timer overflows. This configuration can be used in BSD read mode detect fault conditions on the BSD bus. In BSD Write Mode, this bit is cleared to 0 by user code to disable the generation of Break Condition interrupts on a rising edge of the BSD bus (as is required in BSD Read mode).In BSD Write Mode, the LHS Compare Interrupt (LHSSTA[3]) is used to determine when the MCU should release the BSD bus when transmitting data. If the Break Condition interrupt was still enabled it would generate an unwanted interrupt as soon as the BSD bus is de-asserted. As in BSD Read Mode, the Break timer will stop counting on a rising edge so the Break timer can also be used in this mode to allow user code confirm the pulse width in transmitted data bits. Note: Because of the finite propagation delay in the BSD transmit (from MCU to external pin) and receive (from external pin to MCU) paths, user code must not switch between BSD Write and Read modes until the MCU confirms the external BSD pin is de-asserted. Failure to adhere to this recommendation may result in the generation of an inadvertent Break Condition interrupt

Bit	Description
	after user code switches from BSD Write Mode to BSD Read Mode. A Stop Condition interrupt could be used to ensure that this scenario is avoided.
7	Sync Timer Stop Edge Type Bit This bit is cleared to 0 by user code to stop the sync timer on the falling edge count configured via the LHSCON1[7:4] register. This bit is set to 1 by user code to stop the sync timer on the rising edge count configured via the LHSCON1[7:4] register.
6	Mode of Operation Bit This bit is cleared to 0 by user code to select LIN mode of operation This bit is set to 1 by user code to select BSD mode of operation
5	Enable Compare Interrupt Bit This bit is cleared to 0 by user code to disable compare interrupts This bit is set to 1 by user code to generate an LHS Interrupt(IRQEN[7]) when the value in LHSVAL0 (LIN Synchronisation Bit Timer) = the value in the LHSCMP register. The LHS Compare Interrupt bit LHSSTA[3] is set when this interrupt occurs. This configuration is used in BSD write mode to allow user code correctly time the output pulse widths of BSD bits to be transmitted.
4	Enable Stop Interrupt This bit is cleared to 0 by user code to disable interrupts when a stop condition occurs This bit is set to 1 by user code to generate an interrupt when a stop condition occurs
3	Enable Start Interrupt This bit is cleared to 0 by user code to disable interrupts when a start condition occurs This bit is set to 1 by user code to generate an interrupt when a start condition occurs
2	LIN Sync Enable Bit This bit is cleared to 0 by user code to disable LHS functionality This bit is set to 1 by user code to enable LHS functionality
1	Edge Counter Clear Bit This bit is set to 1 by user code to clear the internal edge counters in the LHS peripheral. This bit is cleared to 0 automatically after a 15us delay.
0	LHS Reset Bit This bit is set to 1 by user code to reset all LHS logic to default conditions. This bit is cleared to 0 automatically after a 15us delay.

LIN Hardware Synchronization Control Register 1:

Name: LHSCON1

Address: 0xFFFF078C

Default Value: 0x00000032

Access: Read/Write

Function: The LHS Control register is a 32-bit register that in conjunction with the LHSCON0 register is used to configure the LIN mode of operation

Table 88. LHSCON1 MMR Bit Descriptions

Bit	Description
31-8	Reserved These bits are reserved for future and should be written as 0 by user software.
7-4	LIN STOP Edge Count These 4 bits are set by user code to the number of falling or rising edges on which to stop the internal LIN synchronization counter. The stop value of this counter can be read by user code via LHSVAL0. The type of edge, either rising or falling, is configured by LHSCON0[7]. The default value of these bits is 0x3 which configures the hardware to stop counting on the third falling edge. It should be noted that the first falling edge is taken as the falling edge at the start of the LIN break pulse.
3-0	LIN START Edge Count These 4 bits are set by user code to the number of falling edges after which the internal LIN synchronization timer will start counting. The stop value of this counter can be read by user code via LHSVAL0. The default value of these bits is 0x2 which configures the hardware to start counting on the second falling edge. It should be noted that the first falling edge is taken as the falling edge at the start of the LIN break pulse.

LIN Hardware Synchronization Timer0 Register:

Name: LHSVAL0

Address: 0xFFFF0788

Default Value: 0x0000

Access: Read Only

Function: The 16-bit read only LHSVAL0 register holds the value of the internal LIN synchronization timer. The LIN synchronization timer is clocked from an internal 5MHz clock which is independent of Core clock and baud-rate frequency. In LIN mode, the value read by user code from the LHSVAL0 register can be used calculate the master LIN baud-rate. This calculation can then be used to configure the internal UART baud-rate to ensure correct LIN communication via the UART from the ADuC7030/ADuC7033 slave to the LIN master node

LIN Hardware Break Timer1 Register:

Name: LHSVAL1

Address: 0xFFFF0790

Default Value: 0x000(read) or 0x047(write)

Access: Read/Write

Function: When user code reads this location, the 12-bit value returned is the value of the internal LIN break timer, which is clocked directly from the on-chip low power (131KHz) oscillator and times the LIN Break pulse. A negative edge on the LIN bus or user code reading the LHSVAL1 will result in the timer and the register contents being reset to 0. When user code writes to this location, the 12-bit value is actually written not to the LIN Break timer but to a LIN Break Compare register.
In LIN mode of operation the value in the compare register is continuously compared to the break timer value. A LIN Break interrupt (IRQEN[7] and LHSSTA[0]) is generated when the timer value reaches the compare value. After the Break Condition interrupt, the LIN Break timer continues to count until the rising edge of the Break signal. If a rising edge is not detected and the 12-bit timer overflows (4096 X 1/131KHz= 31msecs), a Break Field Error Interrupt (IRQEN[7] and LHSSTA[4]) will be generated. By default, the value in the compare register is 0x47, this corresponds to 11 X Bit periods i.e. the minimum pulse width for a LIN break pulse at 20kbps. For different baud rates, this value may be changed by writing to. It is also important to note that if a valid break interrupt is not received then subsequent Sync pulse timing via LHSVAL0 register will not occur.

LIN HARDWARE INTERFACE

LIN Frame Protocol

The LIN frame protocol is broken into 4 main categories:

- Break Symbol
- SYNC Byte
- Protected Identifier
- Data Bytes

The format of the frame header, Break, Synchronization Byte and Protected Identifier are shown in Figure 42. Essentially, the embedded UART, LIN Hardware Synchronization logic and the high voltage transceiver interface all combine on-chip to support and manage LIN based transmissions and receptions.

LIN Frame Break Symbol

As shown in Figure 43, the LIN "break" symbol is used to signal the start of a new frame. It lasts at least 13-bit periods and a slave must be able to detect a "break" symbol, even if it expects data or is in the process of receiving data. The ADuC7030/ADuC7033 accomplishes this by using the LHSVAL1 Break Condition and Break Error Detect functionality as described earlier. The "break" period does not have to be accurately measured, but if a bus fault condition (bus held low) occurs, it must be flagged.

LIN Frame Synchronization Byte

The baud rate of the communication via LIN is calculated from the SYNC Byte. This can be seen in Figure 44. The time between the first falling edge of the Sync Field and the fifth falling edge of the Sync Field is measured. This is then divided by eight to give the baud rate of the data that will be transmitted. The ADuC7030/ADuC7033 implements the timing of this Sync byte in hardware. For more information on this feature, please refer to LIN Hardware Synchronization.

LIN Frame Protected Identifier

After receiving the LIN synch field, the required baud rate for the UART is calculated. The UART is then configured, which allows the ADuC7030/ADuC7033 to receive the Protected Identifier, as shown in Figure 45. The Protected Identifier consists of two sub-fields, the identifier and the identifier parity. The six-bit identifier contains the identifier of the target for the frame. The identifier signifies the number of data bytes to be either received or transmitted. The number of bytes is user configurable at system level design. The parity is calculated on the identifier, and is dependent on the revision of LIN the system is designed for.

LIN Frame Data Byte

The data byte frame carries between one and eight bytes of data. The number of bytes contained in the frame will be dependent on the LIN Master. The data byte frame is split into data bytes as shown in Figure 46.

LIN Frame Data Transmission and Reception

Once the Break Symbol and Synchronization Byte have being correctly received, data is transmitted and received via the COMTX and COMRX MMRs, after configuration of the UART to the required Baud Rate. To configure the UART for use with LIN requires the use of the following UART MMRs:

- COMDIV0: divisor latch (low byte)

- COMDIV1: divisor latch (high byte)

- COMDIV2: 16-bit fractional baud divide register. The required values for COMDIV0, COMDIV1 and COMDIV2 are derived from the LHSVAL0, to generate the required baud rate.

- COMCON0: line control register. Once the UART is correctly configured, the LIN protocol for receiving and transmitting data is identical to the UART specification. To manage data on the LIN bus requires use of the following UART MMRs:

- COMTX: 8-bit transmit register

- COMRX: 8-bit receive register

- COMCON0: line control register

- COMSTA0: line status register

To transmit data on the LIN bus requires that the relevant data be placed into COMTX. To read data received on the LIN bus requires the monitoring of COMRX. To ensure that data is received or transmitted correctly COMSTA0 is monitored. For more information please refer to the UART section of the datasheet.

Under software control it is possible to multiplex the UART data lines (TxD and RxD) to external GPIO pins (GPIO_7 and GPIO_8). For more information please refer to the description of the GPIO Port1 Control Register (GP1CON).

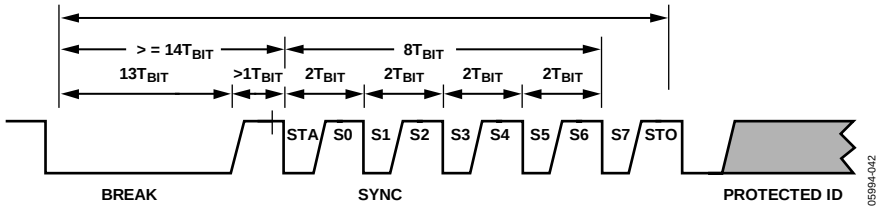


Figure 42. LIN Interface Timing

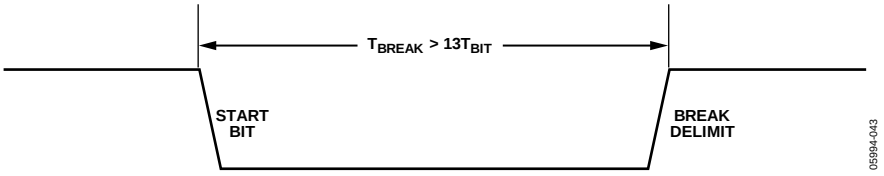


Figure 43. LIN Break Field

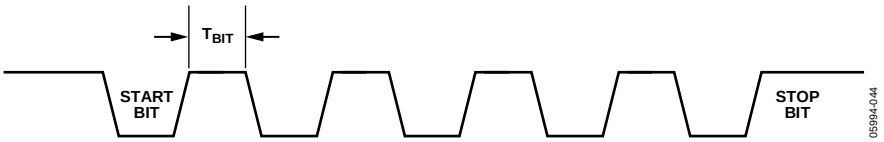


Figure 44. LIN Synch byte Field

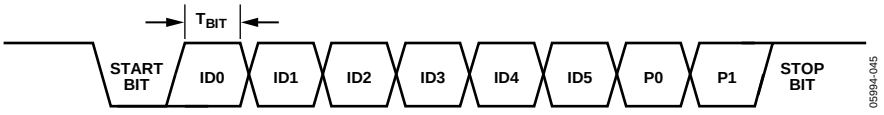


Figure 45. LIN Identifier Byte Field

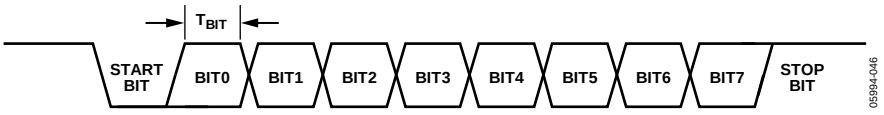


Figure 46. LIN Data Byte Field

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 131kHz periods to generate a Break Interrupt
                        // 0x3F / 131kHz ~ 480us 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 to the value written to LHSCON1[7:4], in this example the number of edges to monitor is the sixth falling edge of the LIN frame, or the fifth falling edge of the SYNC byte. Once this 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, via LHSCON0[8], before the LIN bus returns high. If this occurs, UART communication errors may occur. Example code to ensure this is shown below:

This process is shown in detail in Figure 47.

```
      while((GP2DAT & 0x10 ) == 0 )
      {}                // Wait until LIN Bus returns High
      LHSCON0 = 0x4;    // Enable LHS to detect Break Condition Ungate RX Line
                        // Disable all interrupts except Break Compare Interrupt
      IRQEN = 0x800;    // Enable UART Interrupt
                        // The UART is now configured and ready to be used for LIN
```

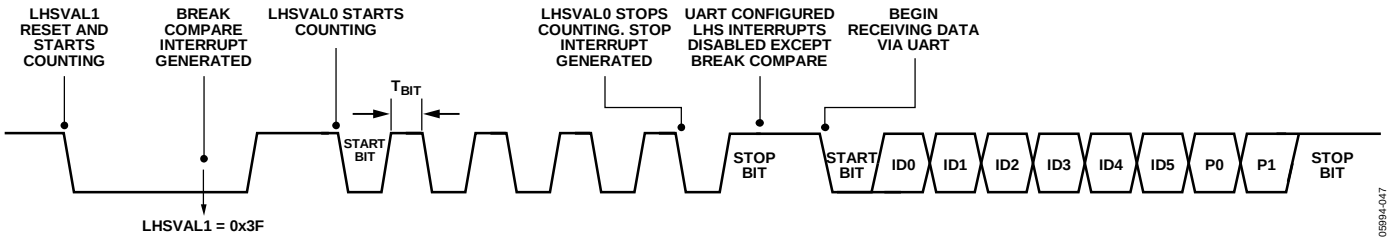


Figure 47. Example LIN Configuration

LIN Diagnostics

The ADuC7030/ADuC7033 features the capability to non-intrusively monitor the current state of the LIN pin. This read back functionality is implemented via GPIO_11. The current state of the LIN pin is contained in GP2DAT[4]

It is also possible to drive the LIN pin high and low via user software, allowing the user to detect open circuit conditions. This functionality is implemented via GPIO_12. To enable this functionality GPIO_12 must be configured as a GPIO via GP2CON[20]. Once configured, the LIN pin may be pulled high or low via GP2DAT.

The ADuC7030/ADuC7033 also features short circuit protection on the LIN pin. If a short circuit condition is detected on the LIN pin, HVSTA[2] is set. This bit is cleared by re-enabling the LIN driver via HVCFG1[3]. It is possible to disable this feature via HVCFG1[2].

LIN Operation during Thermal Shutdown

When a Thermal event occurs, i.e. HVSTA[3] is set, and LIN communications continues uninterrupted.

BIT SERIAL DEVICE (BSD) INTERFACE

BSD is a Pulse Width Modulated signal with 3 possible states: SYNC, ZERO and ONE. These are detailed, along with their associated tolerances, in Table 89. The frame length is 19 bits, and communications occurs at 1200bps ±3%.

Table 89. BSD Bit Level Description

Parameter	Name	Min	Typ	Max	Unit
TX Rate		1164	1200	1236	Bits/sec
Bit Encoding	T_{SYNC}	1/16	2/16	3/16	T_{Period}
	T_0	5/16	6/16	8/16	T_{Period}
	T_1	10/16	12/16	14/16	T_{Period}

BSD COMMUNICATION HARDWARE INTERFACE

The ADuC7030/ADuC7033 emulates the BSD communication protocol using a GPIO, an IRQ and the LIN Synchronization Hardware, all of which is under software control

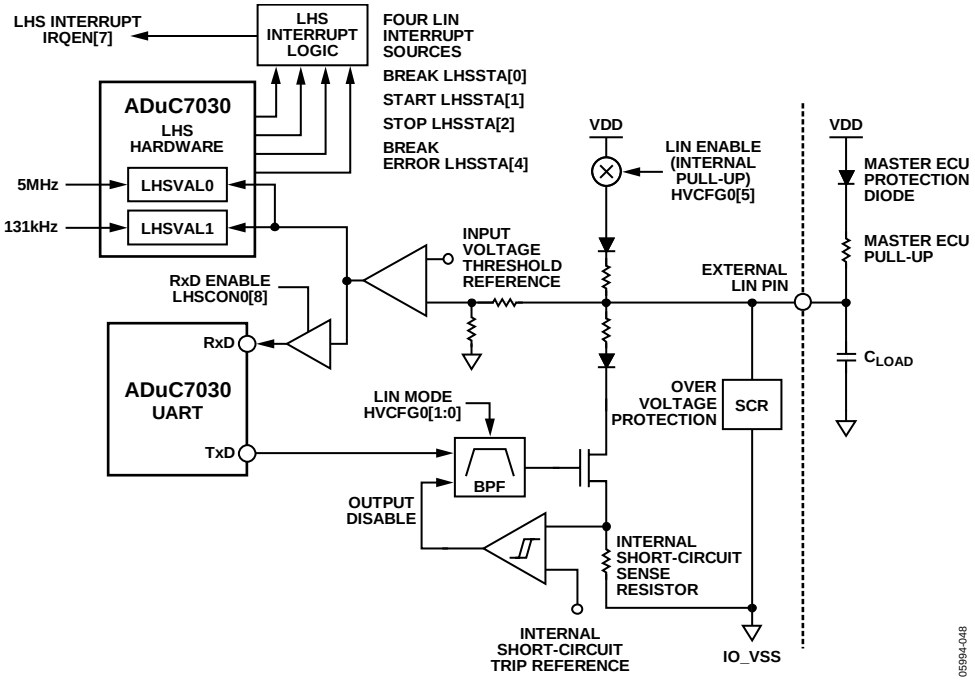


Figure 48. BSD I/O Hardware Interface

BSD RELATED MMRS

The ADuC7030/ADuC7033 emulates the BSD communication protocol using a software (bit bang) interface with some hardware assistance form LIN Hardware Synchronization logic. In effect, the ADuC7030/ADuC7033 BSD interface uses

- An internal GPIO signal (GPIO_12) which is routed to the external LIN/BSD pin and is controlled directly by software to generate 0's and 1's.
- When reading bits, the LIN Synchronization Hardware, uses LHSVAL1 to count the width of the incoming pulses so that user code can interpret the bits as sync, 0 or 1 bits.
- When writing bits, again user code toggles a GPIO pin and uses the LHSCAP and LHSCMP registers to time pulse widths and generate an interrupt when the BSD output pulse width has reached its required width.

The ADuC7030/ADuC7033 MMRs required for BSD communication are listed below.

LHSSTA:	LIN Hardware SYNC Status Register.
LHSCON0:	LIN Hardware SYNC Control Register.
LHSVAL0:	LIN Hardware SYNC Timer 0. 16-bit timer
LHSCON1:	LIN Hardware SYNC Edge Setup Register.
LHSVAL1:	LIN SYNC Break Timer.
LHSCAP:	LIN SYNC Capture Register.
LHSCMP:	LIN SYNC Compare Register.
IRQEN/CLR:	Enable Interrupt Register
FIQEN/CLR:	Enable Fast Interrupt Register
GP2DAT:	GPIO Data Register
GP2SET:	GPIO Set Register
GP2CLR:	GPIO Clear Register

Detailed bit definitions for most of these MMRs have been given previously. In addition to the registers described in the LIN section previously, LHSCAP and LHSCMP are new registers, which are required for the operation of the BSD interface. Details of these registers follow.

LIN Hardware Synchronization Capture Register:

Name: LHSCAP

Address: 0xFFFF0794

Default Value: 0x0000

Access: Read Only

Function: The 16-bit read only LHSCAP register holds the last captured value of the internal LIN synchronization timer (LHSVAL0). In BSD mode, the LHSVAL0 is clocked directly from an internal 5MHz clock, its value is loaded into the capture register on every falling edge of the BSD bus

LIN Hardware Synchronization Compare Register:

Name: LHSCMP

Address: 0xFFFF0798

Default Value: 0x0000

Access: Read/Write

Function: The LHSCMP register is used to time BSD output pulse widths. Once enabled via LHSCON0[5], a LIN interrupt is generated when the value in LHSCAP equals the value written in LHSCMP. This functionality allows user code determine how long a BSD transmission bit (SYNC, 0 or 1) should be asserted on the bus

BSD COMMUNICATIONS FRAME

To transfer data between a Master and Slave, or visa versa, requires the construction of a BSD frame. A BSD frame contains seven key components, Pause/Synch, Direction bit, the Slave Address, the Register Address, data, Parity bits one and two and the Acknowledge from the slave.

If the Master is transmitting data, then all bits, except the ACK, are transmitted by the Master.

If the Master is requesting data from the slave, the Master transmits the Pause/Synch, the direction bit, slave address, register address and P1 bits. The Slave then transmits the Data bytes, Parity bit 2 and the Ack.

PAUSE: >= 3 Synchronization Pulses

DIR: Signifies the direction of data transfer

Zero if master sends request

One if slave sends request

Slave Address

Register Address: Defines register to be read or written

Bit 3 is set to write, cleared to read

Data: 8-bit read only receive register

P1 and P2

P1 = "0" if even number of "1" in 8 previous bits

P1 = "1" if odd number of "1" in 8 previous bits

P2 = "0" if even number of "1" in DATA word

P2 = "1" if odd number of "1" in DATA word

ACK: Zero if transmission is successful.

The ACK is always transmitted by the slave to indicate if the information was received or transmitted.

Table 90. BSD Protocol Description

Pause	DIR	Slave Address	Register Address	P1	Data	P2	ACK
3 Bits	1 Bit	3 Bits	4 Bits	1 Bit	8 Bit	1 Bit	1 Bit

BSD Example Pulse Widths

An example of the different Pulse widths is shown in Figure 49. For each bit the period for which the bus is held low defines what type of bit it is. If the bit is a SYNCH bit, the pulse is held low for 1 bit. If the bit is a ZERO bit, the pulse is held low for 3 bits. If the bit is a ONE bit, the pulse is held low for 6 bits.

If the Master is transmitting data, the signal is held low for the duration of the signal by the Master. An example of a Master transmitting zero is shown in Figure 50. If the Slave is transmitting data, the Master pulls the bus low to begin communications. The slave must then pull the bus low before T_{SYNC} elapses and hold the bus low until either T_{ZERO} or T_{ONE} has elapsed, after which the bus is released by the Slave. An example of a Slave transmitting a zero is shown in Figure 51.

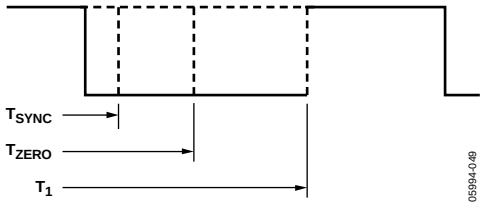


Figure 49. BSD Bit transmission

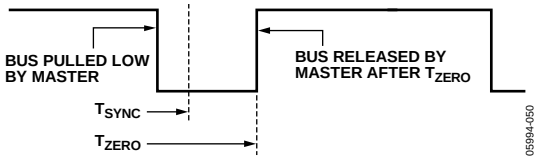


Figure 50. BSD Master Transmitting Zero

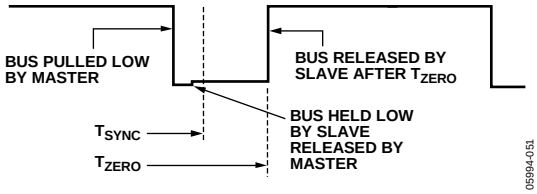


Figure 51. BSD Slave Transmitting Zero

Typical BSD Program Flow

As BSD is a PWM communications protocol controlled by software, it is necessary for the user to construct the required data from each bit. For example the slave address. The slave node receives the three bits and constructs the relevant address.

When BSD communication is initiated by the master, data is transmitted and received by the slave node. A flow diagram showing this process is shown in Figure 52.

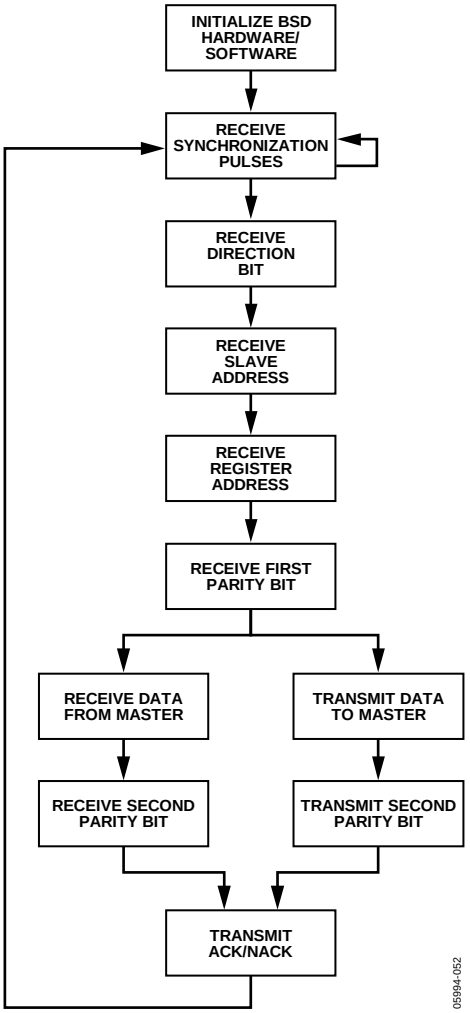


Figure 52. BSD Slave Node State Machine

BSD DATA RECEPTION

To receive data, the LIN/BSD peripheral must first be configured in BSD mode LHSCON[6]=1. In this mode LHSCON0[8] should be set to ensure the LHS break timer (see LHSVAL1) will generate an interrupt on the rising edge of the BSD bus.

The LHS break timer is cleared and starts counting on the falling edge of the BSD bus and is subsequently stopped and generates an interrupt on the rising edge of the BSD bus. Given the LHS break timer is clocked by the low power (131KHz) oscillator, the value in LHSVAL1 can be interpreted by user code to determine if the received data bit is a BSD Sync pulse, 0 or 1.

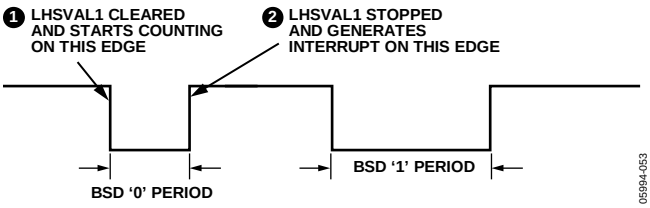


Figure 53. Master Transmit, Slave Read

BSD DATA TRANSMISSION

User code forces a GPIO signal (GPIO_12) low for a specified time to transmit data in BSD mode. In addition user code will also use the Sync Timer(LHSVAL0), LHS Sync Capture register(LHSCAP) and the LHS Sync Compare register(LHSCMP) to time how long the BSD bus should be held low for 0 or 1 bit transmissions.

As described earlier, even when the slave is transmitting, the master will always start the bit transmission period by pulling the BSD bus low. If BSD mode is selected (LHSCON0[6]=1), then the LIN Sync timer value will be captured in LHSCAP on every falling edge of the BSD bus. The LIN Sync timer is running continuously in BSD mode.

User code can then immediately force GPIO_12 low and reads the captured timer value from LHSCAP. A calculation of how many (5MHz) clock periods should elapse before the GPIO_12 should be driven high for a 0 or 1 pulse width can be made. This number can be added to the LHSCAP value and written into the LHSCMP register. If LHSCON0[5] is set, the Sync timer which continues to count (being clocked by a 5MHz clock) will eventually equal the LHSCMP value and generate an LHS Compare interrupt(LHSSTA[3]).

The response to this interrupt should be to force the GPIO_12 signal (and therefore, the BSD bus) high. The software control of the GPIO_12 signal along with the correct use of the LIN Synchronization timers ensures that valid 0 and 1 pulse widths can be transmitted from the ADuC7030/ADuC7033 as shown in figure 42 below. Again care needs to be taken if switching from BSD Write Mode to BSD Read Mode as described in LHSCON0[8].

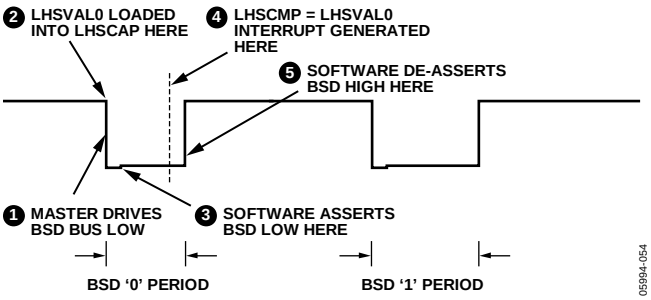


Figure 54. Master Read, Slave Transmit

WAKE-UP FROM BSD INTERFACE

The MCU core can be woken up from power-down via the BSD physical interface. Before entering power-down mode, user code should enable the Start Condition interrupt (LHSCON0[3]) Once this interrupt is enabled, a High to Low transition on the LIN/BSD pin will generate an interrupt event and wake up the MCU core.

ADUC7030/ADUC7033 ON-CHIP DIAGNOSTICS

The ADuC7030/ADuC7033 integrates multiple diagnostic support circuits on-chip. These circuits allow the device to test core digital functionality, 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 8.3mV test voltage. On any gain range the result should be within ±0.5% of the expected result.

Internal Short Mode

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

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 or open circuit conditions in both external shunt or temperature sensor configurations.

High Voltage I/O Diagnostics

High Voltage I/O Read back

All high voltage I/O pins will be supported with read back capability. This will allow detection of external short conditions.

High Voltage Current Detection

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

PART IDENTIFICATION

Two registers mapped into the MMR space are intended to allow user code identify and trace, manufacturing lot ID information, part ID number, silicon mask revision and kernel revision. This information is contained in the SYSSER0 and SYSSER1 MMR, which are described in detail below.

System Serial ID Register 0:

Name: SYSSER0

Address: 0xFFFF0238

Default Value: 0x00000000 (Updated by kernel at power-on)

Access: Read/Write

Function: At power-on, this 32-bit register will hold the value of the original manufacturing lot number from which this specific ADuC7030/ADuC7033 unit was manufactured (bottom die only). Used in conjunction with SYSSER1, this lot number will allow the full manufacturing history of this part to be traced (bottom die only)

Table 91. SYSSER0 MMR Bit Descriptions

Bit	Description
31-27	Wafer Number: The 5 bits read from this location will give the wafer number (1-24) from the Wafer Fabrication Lot ID which this device came from, and when used in conjunction with SYSSER0[26-0] provides individual wafer traceability.
26-22	Wafer Lot Fabrication Plant The 5 bits read from this location reflect the manufacturing plant associate with this Wafer Lot, and used in conjunction with SYSSER0[21-0] provides wafer lot traceability.
21-16	Wafer Lot fabrication ID The 6 bits read from this location form part of the Wafer Lot Fabrication ID, and used in conjunction with SYSSER0[26-22] and SYSSER0[15-0] provides wafer lot traceability.
15-0	Wafer Lot Fabrication ID These 16 LSBs will hold a 16-bit number, which should be interpreted as the Wafer Fabrication Lot ID number. When used in conjunction with the value in SYSSER1 i.e. the manufacturing lot ID, this number is a unique identifier for the part.

For full traceability, part assembly lot number, SYSSER0 and module number need to be recorded. The lot number is part of the branding on the package as shown Table 92:

Table 92. Branding example

	LFCSP	LFQFP
Line 1	ADuC7033	ADuC7033
Line 2	BCPZ 8W	BSTZ 8W
Line 3	A40 #DATE CODE	A40 #DATE CODE
Line 4	ASSEMBLY LOT NUMBER	ASSEMBLY LOT NUMBER

System Serial ID Register 1:

Name: SYSSER1

Address: 0xFFFF023C

Default Value: 0x00000000 (Updated by kernel at power-on)

Access: Read/Write

Function: At power-on, this 32-bit register will hold the values of the part ID number, silicon mask revision number and kernel revision number (bottom die only) as detailed below

Table 93. SYSSER1 MMR Bit Descriptions

Bit	Description
31-28	Silicon Mask Revision ID The 4 bits read from this nibble reflect the silicon mask ID number. Specifically, the hex value in this nibble should be decoded as the lower hex nibble in the hex numbers reflecting the ASCII characters in the range 'A' to 'O'. Examples: Bits 19-16 = 0001 = 1hex, therefore this value should be interpreted as 41 which is ASCII character A corresponding to silicon mask revision A Bits 19-16 = 1011 = Bhex, therefore the number is interpreted as 4B which is ASCII character K corresponding to silicon mask revision K The allowable range for this value is 1 to 15 which is interpreted as 41 to 4F or A to O)
27-20	Kernel Revision ID This byte contains the hex number, which should be interpreted as an ASCII character indicating the revision of the kernel firmware embedded in the on-chip Flash/EE memory. Example: Reading 0x41 from this byte should be interpreted as A indicating a revision A kernel is on-chip.
19-16	Kernel Minor Revision number For PreProduction Release, these bits refer to the Device's Kernel Minor Revision Number
15-0	Part ID These 16 LSBs will hold a 16-bit number, which should be interpreted as the part ID number. When used in conjunction with the value in SYSSER0 i.e. the manufacturing lot ID, this number is a unique identifier for the part.

System Kernel Checksum:

Name: SYSCHK

Address: 0xFFFF0240

Default Value: 0x00000000(Updated by kernel at power-on)

Access: Read/Write

Function: At power-on, this 32-bit register will hold the kernel checksum

System Identification FEE0ADR:

Name: FEE0ADR

Address: 0xFFFF0E10

Default Value: Non Zero

Access: Read/Write

Function: This 16-bit register dictates the address upon which any Flash/EE command executed via FEE0CON will act upon

Note: This MMR is also used to identify ADuC7030 Family member and pre-release silicon revision

Table 94. FEE0ADR System Identification MMR Bit Descriptions

Bit	Description	
15-12	Reserved	
11-8	Reserved	
7-4	Silicon Revision	
	0x0	Type6
	0x1	Type6X
	0x4	Type7Y
	0x5	Type7OP
	0x6	Type8
	0x7	Type7OP1
	0x8	Type7M
	0x9	Type7
	0xA	Type8W
	0xB	Type9
	0xC	Type7ML
	0XD	Type8V
	Others	Reserved
3-0	ADuC703X Family ID	
	0x0	ADuC7030
	0x1	ADuC7031
	0x2	ADuC7032
	0x3	ADuC7033
	Others	Reserved

ADUC7030/ADUC7033 EXAMPLE SCHEMATIC

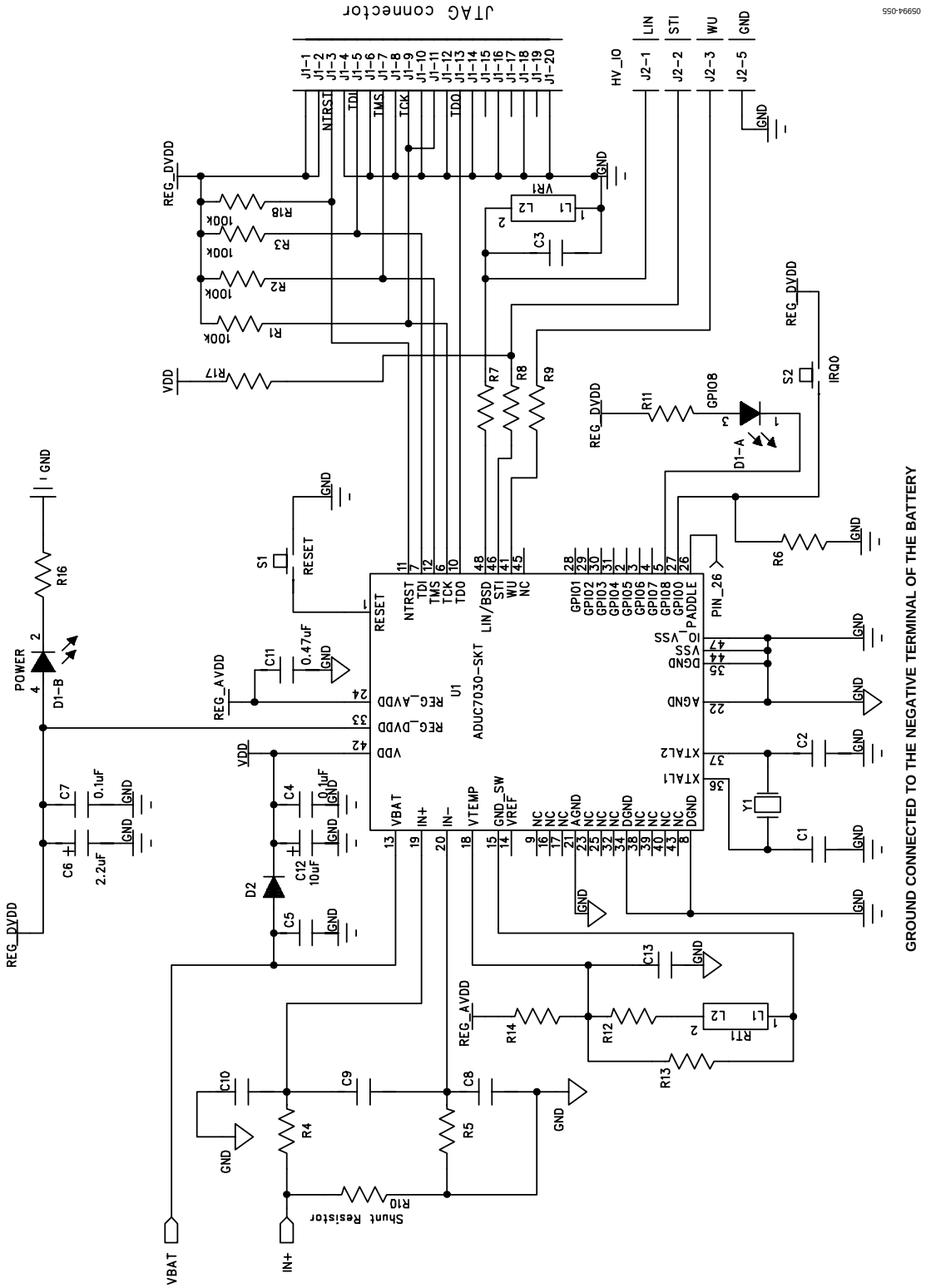
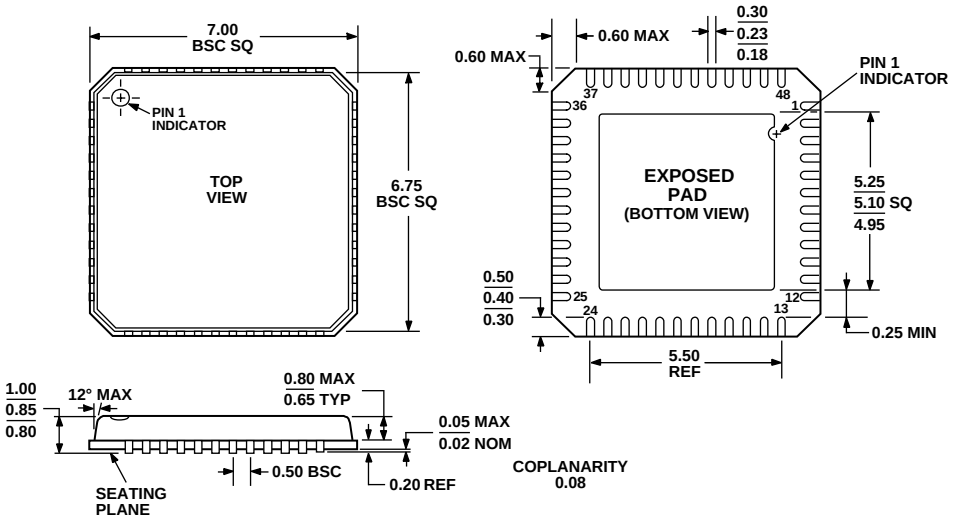


Figure 55. Schematic

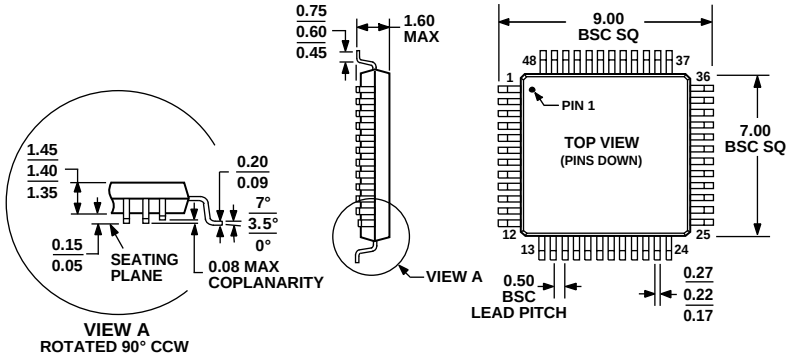
This example schematic represents a basic functional circuit implementation. Additional components may need to be added based on your use of the part. For a more detailed discussion on circuit implementation and layout please refer to the application note "*layout and board recommendation*".

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-220-VKKD-2

Figure 56. 48-Lead Lead Frame Chip Scale Package [LFCSP_VQ]
7 mm × 7 mm Body, Very Thin Quad
(CP-48-1)
Dimensions shown in millimeters



COMPLIANT TO JEDEC STANDARDS MS-026-BBC

Figure 57. 48-Lead Low Profile Quad Flat Package [LQFP]
(ST-48)
Dimensions shown in millimeters

ORDERING GUIDE

Model	Temperature Range	Package Description	Package Option

PR05994-0-10/06(PrE)