Precision Analog Microcontroller, 12-Bit Analog I/O, ARM7TDMI MCU

ADuC7122

FEATURES

Analog I/O
- 13-external channel, 12-bit, 1 MSPS ADC
  - 2 differential channels with programmable gain PGA (1 to 5) input range
  - IOVDD power monitor channel
  - On-chip temperature monitor
  - 11 general-purpose inputs
- Fully differential and single-ended modes
- 0 V to V_{REF} analog input range
- 12 × 12-bit voltage output DACs
- On-chip voltage reference: 1.2 V/2.5 V
  - Buffered output reference sources for use with external circuits

Microcontroller
- ARM7TDMI core, 16-bit/32-bit RISC architecture
- JTAG port supports code download and debug

Clocking options
- Trimmed on-chip oscillator (±3%)
- External watch crystal
- External clock source up to 41.78 MHz
- 41.78 MHz PLL with programmable divider

Memory
- 126 kB Flash/EE memory, 8 kB SRAM
- In-circuit download, JTAG-based debug
- Software-triggered in-circuit reprogrammability

On-chip peripherals
- UART, 2× I^2C and SPI serial I/Os
- 32-pin GPIO port
- 4 general-purpose timers
- Wake-up and watchdog timers (WDT)
- Power supply monitor

Vectored interrupt controller for FIQ and IRQ
- 8 priority levels for each interrupt type
- Interrupt on edge or level external pin inputs

Power
- Specified for 3 V operation
- Active mode: 11 mA at 5 MHz, 40 mA at 41.78 MHz

Packages and temperature range
- 7 mm × 7 mm 108-ball BGA
- Fully specified for −10°C to +95°C operation

Tools
- Low cost QuickStart development system
- Full third-party support

APPLICATIONS

- Optical networking, industrial control, and automation systems
- Smart sensors and precision instrumentation

FUNCTIONAL BLOCK DIAGRAM

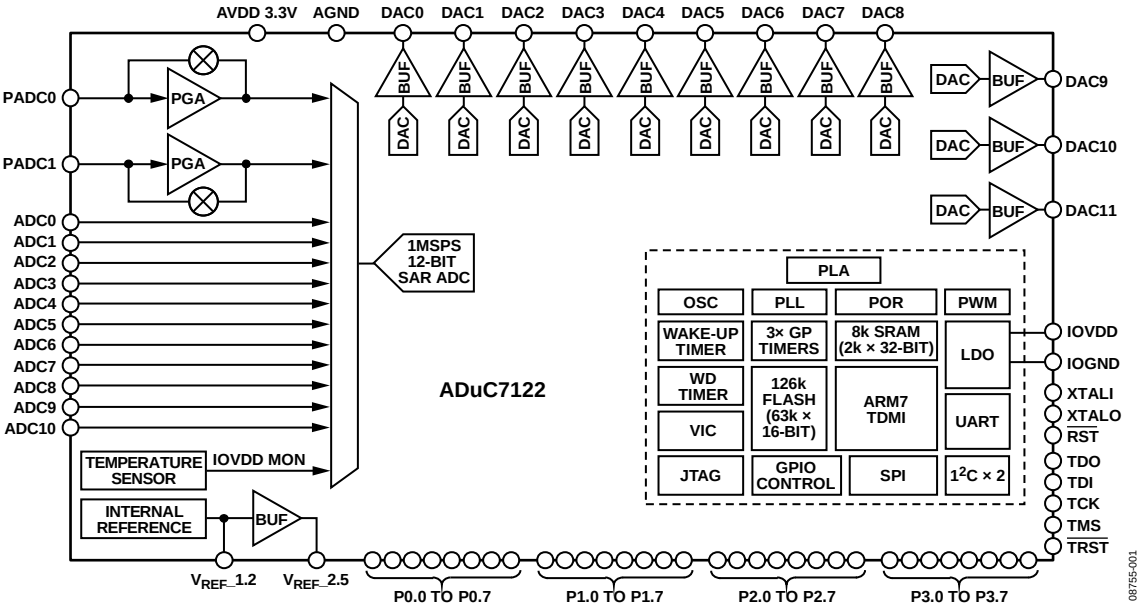


Figure 1.

Rev. 0

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

REVISION HISTORY

4/10—Revision 0: Initial Version

GENERAL DESCRIPTION

The ADuC7122 is a fully integrated, 1 MSPS, 12-bit data acquisition system, incorporating high performance multichannel ADCs, 12 voltage output DACs, 16-bit/32-bit MCUs, and Flash/EE memory on a single chip.

The ADC consists of up to 13 inputs. Four of these inputs can be configured as differential pairs with a programmable gain amplifier on their front end, providing a gain between 1 and 5. The ADC can operate in single-ended or differential input mode. The ADC input voltage is 0 V to V_{REF}. A low drift band gap reference, temperature sensor, and supply voltage monitor complete the ADC peripheral set.

The DAC output range is programmable to one of two voltage ranges. The DAC outputs have an enhanced feature of being able to retain their output voltage during a watchdog or software reset sequence.

The device operates from an on-chip oscillator and a PLL, generating an internal high frequency clock of 41.78 MHz. This clock is routed through a programmable clock divider from which the MCU core clock operating frequency is generated. The microcontroller core is an ARM7TDMI®, 16-bit/32-bit RISC machine that offers up to 41 MIPS peak performance. There are 8 kB of SRAM and 126 kB of nonvolatile Flash/EE memory provided on chip. The ARM7TDMI core views all memory and registers as a single linear array.

The ADuC7122 contains an advanced interrupt controller. The vectored interrupt controller (VIC) allows every interrupt to be assigned a priority level. It also supports nested interrupts to a maximum level of eight per IRQ and FIQ. When IRQ and FIQ interrupt sources are combined, a total of 16 nested interrupt levels are supported.

On-chip factory firmware supports in-circuit download via the I^2C serial interface port, and nonintrusive emulation is also supported via the JTAG interface. These features are incorporated into a low cost QuickStart™ development system supporting this MicroConverter® family. The part contains a 16-bit PWM with six output signals.

For communication purposes, the part contains 2× I^2C channels that can be individually configured for master or slave mode. An SPI interface supporting both master and slave modes is also provided.

The part operates from 3.0 V to 3.6 V and is specified over a temperature range of −10°C to +95°C. The ADuC7122 is available in a 108-ball BGA package.

SPECIFICATIONS

AV_{DD} = IOV_{DD} = 3.0 V to 3.6 V, V_{REF} = 2.5 V internal reference, f_{CORE} = 41.78 MHz, T_A = −10°C to +95°C, unless otherwise noted.

Table 1.

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
ADC CHANNEL SPECIFICATIONS					Eight acquisition clocks and $f_{ADC}/2$
ADC Power-Up Time		5		µs	
DC Accuracy[1, 2]					
Resolution	12			Bits	
Integral Nonlinearity		±0.6	±2	LSB	2.5 V internal reference, not production tested for PADC0/PADC1 channels
Differential Nonlinearity[3, 4]		±0.5	+1.4/−0.99	LSB	2.5 V internal reference, gauranteed monotonic
DC Code Distribution		1		LSB	ADC input is a dc voltage
ENDPOINT ERRORS[5]					Internally unbuffered channels
Offset Error		±2	±5	LSB	
Offset Error Match		±1		LSB	
Gain Error		±2	±5	LSB	
Gain Error Match		±1		LSB	
DYNAMIC PERFORMANCE					f_{IN} = 10 kHz sine wave, f_{SAMPLE} = 1 MSPS internally unbuffered channels
Signal-to-Noise Ratio (SNR)		69		dB	Includes distortion and noise components
Total Harmonic Distortion (THD)		−78		dB	
Peak Harmonic or Spurious Noise		−75		dB	
Channel-to-Channel Crosstalk		−80		dB	Measured on adjacent channels
ANALOG INPUT					
Input Voltage Ranges					
Differential Mode			V_{CM}[6] ± $V_{REF}/2$	V	See Table 35 and Table 36
Single-Ended Mode			0 to V_{REF}	V	Buffer bypassed
Single-Ended Mode	0.15		AV_{DD} − 1.5	V	Buffer enabled
Leakage Current		±0.2		µA	
Input Capacitance		20		pF	During ADC acquisition buffer bypassed
Input Capacitance		20		pF	During ADC acquisition buffer enabled
PADC0 INPUT					28.3 kΩ resistor, PGA gain = 3; acquisition time = 6 µs, pseudo differential mode
Full Scale Input Range	20		1000	µA	
Input Leakage at PADC0P[4]		0.15	2	nA	
Resolution	11			Bits	0.1% accuracy, 5 ppm external resistor for I to V
Gain Error[4]			1	%	
Gain Drift[4]			50	ppm/°C	
Offset[4]		3	6	nA	PGA offset not included
Offset Drift[4]		30	60	pA/°C	
PADC0P Compliant Range	0.1		AV_{DD} − 1.2	V	
PADC1 INPUT					53.5 kΩ resistor, PGA gain = 3; acquisition time = 6 µs, pseudo differential mode
Full Scale Input Range	10.6		700	µA	
Input Leakage at PADC1P[4]		0.15	2	nA	
Resolution	11			Bits	0.1% accuracy, 5 ppm external resistor for I to V
Gain Error[4]			1	%	
Gain Drift[4]			50	ppm/°C	
Offset[4]		3	6	nA	PGA offset not included
Offset Drift[4]		30	60	pA/°C	
PADC1P Compliant Range	0.1		AV_{DD} − 1.2	V	

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
ON-CHIP VOLTAGE REFERENCE					0.47 µF from V_{REF} to AGND
Output Voltage		2.5		V	
Accuracy[7]			±5	mV	T_A = 25°C
Reference Temperature Coefficient[4]		10	30	ppm/°C	
Power Supply Rejection Ratio		61		dB	
Output Impedance		10		Ω	T_A = 25°C
Internal V_{REF} Power-On Time		1		ms	
EXTERNAL REFERENCE INPUT					
Input Voltage Range	1.2		AV_{DD}	V	
BUF_VREF1, BUF_VREF2 OUTPUTS					
Accuracy		±5		mV	T_A = 25°C
Reference Temperature Coefficient		40		µV/°C	
Load Current			1.2	mA	
DAC CHANNEL SPECIFICATIONS					R_L = 5 kΩ, C_L = 100 pF
DC Accuracy[8]					Buffered
Resolution		12		Bits	
Relative Accuracy		±2		LSB	
Differential Nonlinearity		±0.2	±1	LSB	Guaranteed monotonic
Calculated Offset Error		±2		mV	2.5 V internal reference
Actual Offset Error		9		mV	Measured at Code 0
Gain Error[9]		±0.15	±0.8	%	
Gain Error Mismatch		0.1		%	% of full scale on DAC0
Settling Time		10		µs	
PSRR[4]					Buffered
DC	−59	−61		dB	
1 kHz	−57			dB	
10 kHz	−47			dB	
100 kHz	−19			dB	
OFFSET DRIFT[4]			10	µV/°C	
GAIN ERROR DRIFT[4]			10	µV/°C	
SHORT-CIRCUIT CURRENT		20		mA	
ANALOG OUTPUTS					
Output Range	0.1		V_{REF}/AV_{DD} − 0.1	V	Buffer on
DAC AC CHARACTERISTICS					
Slew Rate		2.49		V/µs	
Voltage Output Settling Time		10		µs	
Digital-to-Analog Glitch Energy		±20		nV-sec	1 LSB change at major carry (where maximum number of bits simultaneously change in the DACxDAT register)
TEMPERATURE SENSOR[10]					
Voltage Output at 25°C		707		mV	
Voltage TC		−1.25		mV/°C	
Accuracy		±3		°C	MCU in power-down or standby mode before measurement
POWER SUPPLY MONITOR (PSM)					
IOV_{DD} Trip Point Selection		2.79		V	Two selectable trip points
		3.07		V	
Power Supply Trip Point Accuracy		±2.5		%	Of the selected nominal trip point voltage
POWER-ON RESET		2.36		V	
WATCHDOG TIMER (WDT)					
Timeout Period	0		512	sec	

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
FLASH/EE MEMORY					
Endurance[11]	10,000			Cycles	
Data Retention[12]	20			Years	T_J = 85°C
DIGITAL INPUTS					All digital inputs excluding XCLKI and XTALO
Logic 1 Input Current		±0.2	±1	µA	$V_{IH} = V_{DD}$ or V_{IH} = 5 V
Logic 0 Input Current		−40	−60	µA	V_{IL} = 0 V; except TDI
Input Capacitance		10		pF	
LOGIC INPUTS[4]					All logic inputs excluding XTALI
V_{INL}, Input Low Voltage[4]			0.8	V	
V_{INH}, Input High Voltage[4]	2.0			V	
LOGIC OUTPUTS					All digital outputs excluding XTALO
V_{OH}, Output High Voltage	2.4			V	I_{SOURCE} = 1.6 mA
V_{OL}, Output Low Voltage[13]			0.4	V	I_{SINK} = 1.6 mA
CRYSTAL INPUTS XTALI and XTALO					
Logic Inputs, XTALI Only					
V_{INL}, Input Low Voltage		1.1		V	
V_{INH}, Input High Voltage		1.7		V	
XTALI Input Capacitance		20		pF	
XTALO Output Capacitance		20		pF	
INTERNAL OSCILLATOR		32.768		kHz	
			±3	%	
MCU CLOCK RATE					
From 32 kHz Internal Oscillator		326		kHz	CD = 7
From 32 kHz External Crystal		41.78		MHz	CD = 0
Using an External Clock	0.05		41.78	MHz	T_A = 95°C
START-UP TIME					Core clock = 41.78 MHz
At Power-On		70		ms	
From Pause/Nap Mode		24		ns	CD = 0
		3.06		µs	CD = 7
From Sleep Mode		1.58		ms	
From Stop Mode		1.7		ms	
PROGRAMMABLE LOGIC ARRAY (PLA)					
Pin Propagation Delay		12		ns	From input pin to output pin
Element Propagation Delay		2.5		ns	
POWER REQUIREMENTS[14, 15]					
Power Supply Voltage Range					
AV_{DD} to AGND and IOV_{DD} to IOGND	3.0		3.6	V	
Analog Power Supply Currents					
AV_{DD} Current		200		µA	ADC in idle mode
Digital Power Supply Current					
IOV_{DD} Current in Normal Mode					Code executing from Flash/EE
		7		mA	CD = 7
		11		mA	CD = 3
		30	40	mA	CD = 0 (41.78 MHz clock)
IOV_{DD} Current in Pause Mode[4]		25		mA	CD = 0 (41.78 MHz clock)
IOV_{DD} Current in Sleep Mode[4]		100		µA	T_A = 85°C
Additional Power Supply Currents					
ADC		2.7		mA	At 1 MSPS
DAC		250		µA	Per DAC

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
ESD TESTS					2.5 V reference, T_A = 25°C
HBM Passed Up To			4	kV	
FCIDM Passed Up To			0.5	kV	

[1] All ADC channel specifications are guaranteed during normal MicroConverter core operation.
[2] Applies to all ADC input channels.
[3] Measured using the factory-set default values in the ADC offset register (ADCOF) and gain coefficient register (ADCGN); see the the Calibration section.
[4] Not production tested but supported by design and/or characterization data on production release.
[5] Measured using the factory-set default values in ADCOF and ADCGN with an external AD845 op amp as an input buffer stage, as shown in Figure 23. Based on external ADC system components, the user may need to execute a system calibration to remove external endpoint errors and achieve these specifications (see the ADC Circuit Overview section).
[6] The input signal can be centered on any dc common-mode voltage (V_{CM}) as long as this value is within the ADC voltage input range specified.
[7] V_{REF} calibration and trimming are performed with core operating in normal mode (CD = 0), ADC on, and all DACs on. V_{REF} accuracy may vary under other operating conditions.
[8] DAC linearity is calculated using a reduced code range of 100 to 3995.
[9] DAC gain error is calculated using a reduced code range of 100 to internal 2.5 V V_{REF}.
[10] Die temperature.
[11] Endurance is qualified as per JEDEC Standard 22 Method A117 and measured at −40°C, +25°C, +85°C, and +125°C.
[12] Retention lifetime equivalent at junction temperature (T_J) = 85°C as per JEDEC Standard 22 Method A117. Retention lifetime derates with junction temperature.
[13] Test carried out with a maximum of eight I/Os set to a low output level.
[14] Power supply current consumption is measured in normal, pause, and sleep modes under the following conditions: normal mode with 3.6 V supply, pause mode with 3.6 V supply, and sleep mode with 3.6 V supply.
[15] IOV_{DD} power supply current decreases typically by 2 mA during a Flash/EE erase cycle.

TIMING SPECIFICATIONS

Table 2. I²C Timing in Fast Mode (400 kHz)

		Slave			Master			
Parameter	**Description**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
t_L	SCLx low pulse width	200				1360		ns
t_H	SCLx high pulse width	100				1140		ns
t_{SHD}	Start condition hold time	300						ns
t_{DSU}	Data setup time	100				740		ns
t_{DHD}	Data hold time	0				400		ns
t_{RSU}	Setup time for repeated start	100						ns
t_{PSU}	Stop condition setup time	100				800		ns
t_{BUF}	Bus-free time between a stop condition and a start condition	1.3						µs
t_R	Rise time for both SCLx and SDAx			300		200		ns
t_F	Fall time for both SCLx and SDAx			300				ns

Table 3. I²C Timing in Standard Mode (100 kHz)

		Slave			
Parameter	**Description**	**Min**	**Typ**	**Max**	**Unit**
t_L	SCLx low pulse width	4.7			µs
t_H	SCLx high pulse width	4.0			ns
t_{SHD}	Start condition hold time	4.0			µs
t_{DSU}	Data setup time	250			ns
t_{DHD}	Data hold time	0		3.45	µs
t_{RSU}	Setup time for repeated start	4.7			µs
t_{PSU}	Stop condition setup time	4.0			µs
t_{BUF}	Bus-free time between a stop condition and a start condition	4.7			µs
t_R	Rise time for both SCLx and SDAx			1	µs
t_F	Fall time for both SCLx and SDAx			300	ns

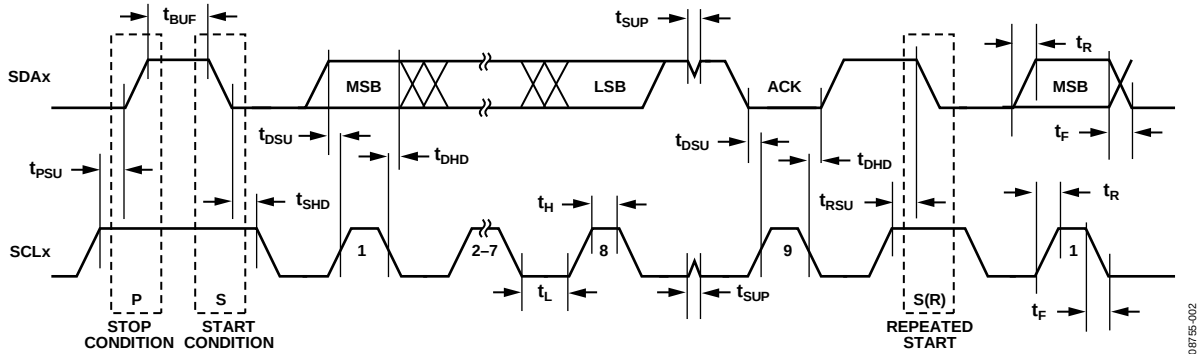


Figure 2. I²C-Compatible Interface Timing

Table 4. SPI Master Mode Timing (SPICPH = 1)

Parameter	Description	Min	Typ	Max	Unit
t_{SL}	SCLOCK low pulse width		(SPIDIV + 1) × t_{UCLK}		ns
t_{SH}	SCLOCK high pulse width		(SPIDIV + 1) × t_{UCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge			25	ns
t_{DSU}	Data input setup time before SCLOCK edge[1]	1 × t_{UCLK}			ns
t_{DHD}	Data input hold time after SCLOCK edge	2 × t_{UCLK}			ns
t_{DF}	Data output fall time		5	12.5	ns
t_{DR}	Data output rise time		5	12.5	ns
t_{SR}	SCLOCK rise time		5	12.5	ns
t_{SF}	SCLOCK fall time		5	12.5	ns

[1] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider.

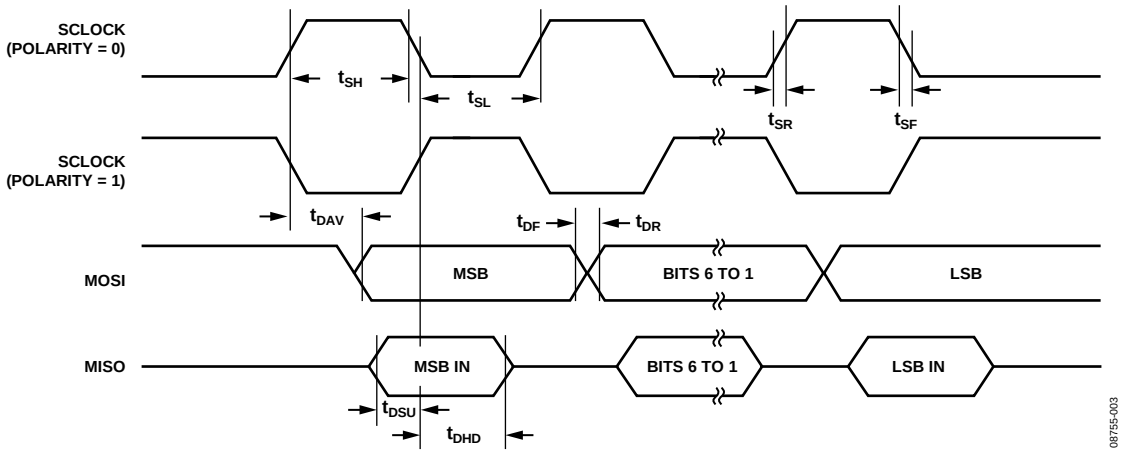


Figure 3. SPI Master Mode Timing (SPICPH = 1)

Table 5. SPI Master Mode Timing (SPICPH = 0)

Parameter	Description	Min	Typ	Max	Unit
t_{SL}	SCLOCK low pulse width		(SPIDIV + 1) × t_{UCLK}		ns
t_{SH}	SCLOCK high pulse width		(SPIDIV + 1) × t_{UCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge			25	ns
t_{DOSU}	Data output setup before SCLOCK edge			75	ns
t_{DSU}	Data input setup time before SCLOCK edge[1]	1 × t_{UCLK}			ns
t_{DHD}	Data input hold time after SCLOCK edge	2 × t_{UCLK}			ns
t_{DF}	Data output fall time		5	12.5	ns
t_{DR}	Data output rise time		5	12.5	ns
t_{SR}	SCLOCK rise time		5	12.5	ns
t_{SF}	SCLOCK fall time		5	12.5	ns

[1] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider.

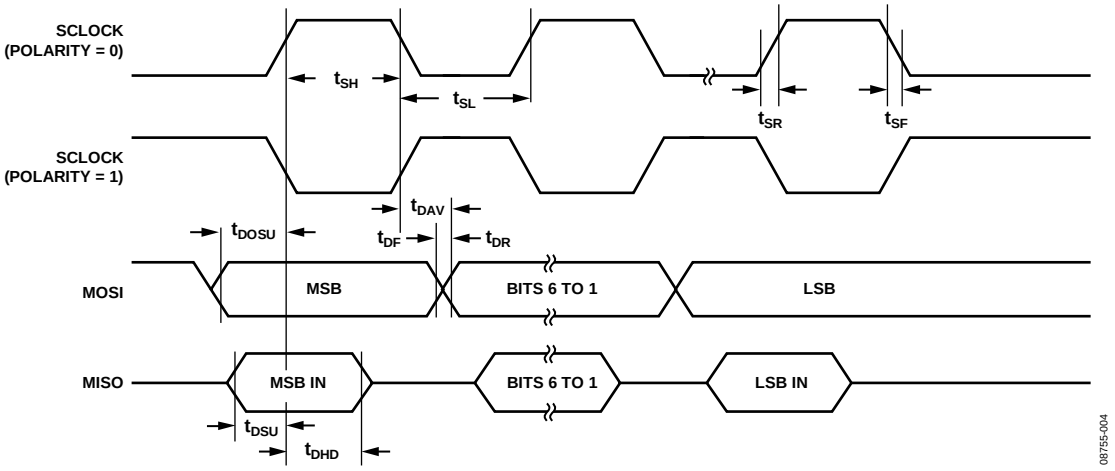


Figure 4. SPI Master Mode Timing (SPICPH = 0)

Table 6. SPI Slave Mode Timing (SPICPH = 1)

Parameter	Description	Min	Typ	Max	Unit
$t_{\overline{CS}}$	$\overline{CS}$ to SCLOCK edge	200			ns
t_{SL}	SCLOCK low pulse width[1]		(SPIDIV + 1) × t_{UCLK}		ns
t_{SH}	SCLOCK high pulse width[1]		(SPIDIV + 1) × t_{UCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge			25	ns
t_{DSU}	Data input setup time before SCLOCK edge	1 × t_{UCLK}			ns
t_{DHD}	Data input hold time after SCLOCK edge	2 × t_{UCLK}			ns
t_{DF}	Data output fall time		5	12.5	ns
t_{DR}	Data output rise time		5	12.5	ns
t_{SR}	SCLOCK rise time		5	12.5	ns
t_{SF}	SCLOCK fall time		5	12.5	ns
t_{SFS}	$\overline{CS}$ high after SCLOCK edge	0			ns

[1] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider.

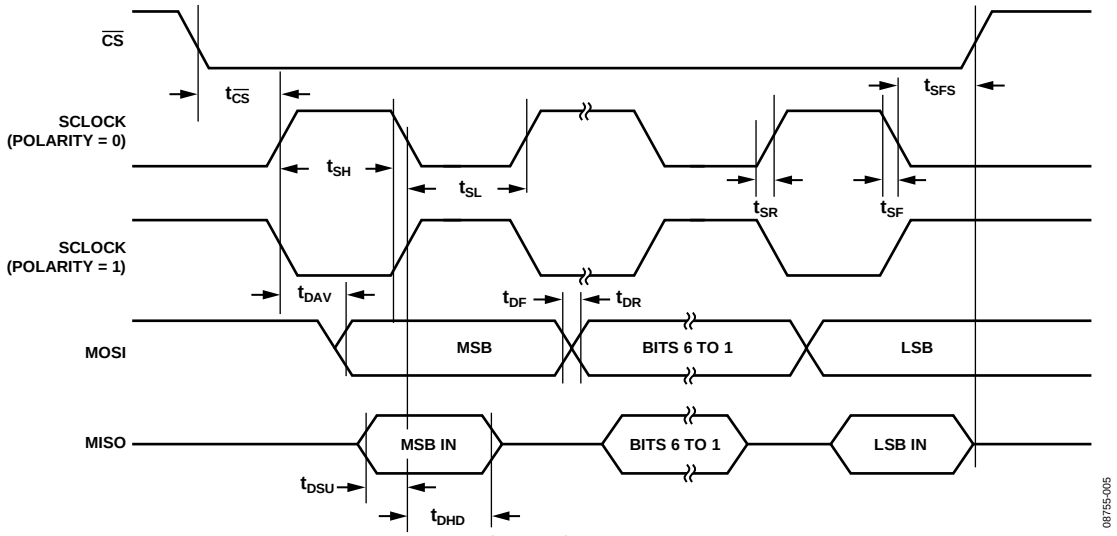


Figure 5. SPI Slave Mode Timing (SPICPH = 1)

Table 7. SPI Slave Mode Timing (SPICPH = 0)

Parameter	Description	Min	Typ	Max	Unit
$t_{\overline{CS}}$	$\overline{CS}$ to SCLOCK edge	200			ns
t_{SL}	SCLOCK low pulse width[1]		(SPIDIV + 1) × t_{UCLK}		ns
t_{SH}	SCLOCK high pulse width[1]		(SPIDIV + 1) × t_{UCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge			25	ns
t_{DSU}	Data input setup time before SCLOCK edge[1]	1 × t_{UCLK}			ns
t_{DHD}	Data input hold time after SCLOCK edge[1]	2 × t_{UCLK}			ns
t_{DF}	Data output fall time		5	12.5	ns
t_{DR}	Data output rise time		5	12.5	ns
t_{SR}	SCLOCK rise time		5	12.5	ns
t_{SF}	SCLOCK fall time		5	12.5	ns
t_{DOCS}	Data output valid after $\overline{CS}$ edge			25	ns
t_{SFS}	$\overline{CS}$ high after SCLOCK edge	0			ns

[1] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider.

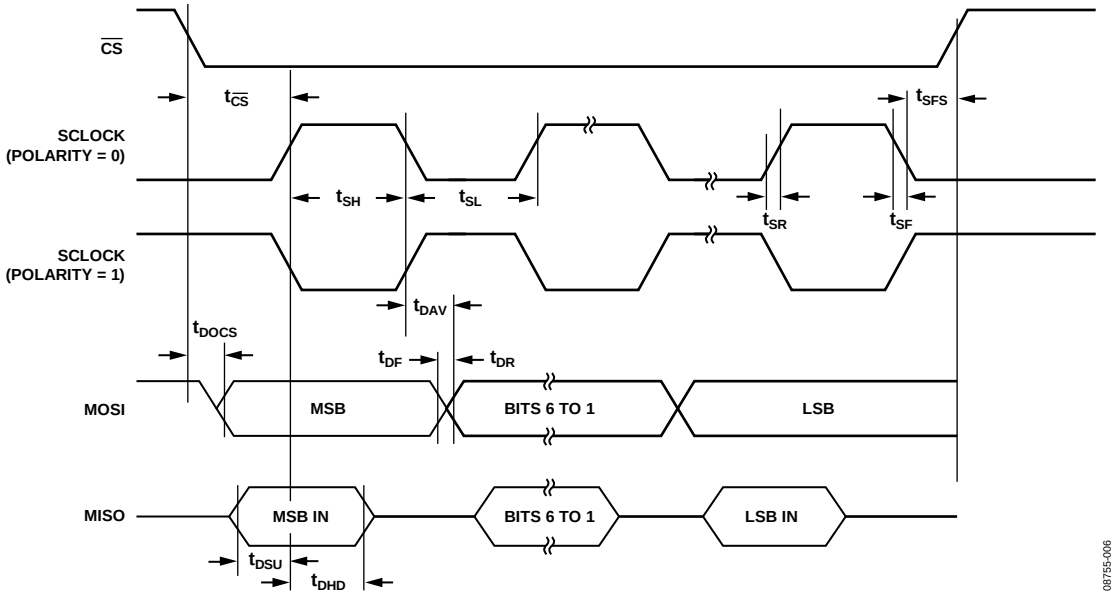


Figure 6. SPI Slave Mode Timing (SPICPH = 0)

ABSOLUTE MAXIMUM RATINGS

AGND = REFGND = DACGND = GND_{REF}, T_A = 25°C, unless otherwise noted.

Table 8.

Parameter	Rating
AV_{DD} to IOV_{DD}	−0.3 V to +0.3 V
AGND to DGND	−0.3 V to +0.3 V
IOV_{DD} to IOGND, AV_{DD} to AGND	−0.3 V to +6 V
Digital Input Voltage to IOGND	−0.3 V to +5.3 V
Digital Output Voltage to IOGND	−0.3 V to IOV_{DD} + 0.3 V
V_{REF}_2.5 and V_{REF}_1.2 to AGND	−0.3 V to AV_{DD} + 0.3 V
Analog Inputs to AGND	−0.3 V to AV_{DD} + 0.3 V
Analog Outputs to AGND	−0.3 V to AV_{DD} + 0.3 V
Operating Temperature Range	−10°C to +95°C
Storage Temperature Range	−65°C to +150°C
Junction Temperature	150°C
θ_{JA} Thermal Impedance	
108-Ball CSP_BGA	40°C/W
Peak Solder Reflow Temperature	
SnPb Assemblies (10 sec to 30 sec)	240°C
RoHS-Compliant Assemblies (20 sec to 40 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.

Only one absolute maximum rating can be applied at any one time.

ESD CAUTION

ESD (electrostatic discharge) sensitive device. Charged devices and circuit boards can discharge without detection. Although this product features patented or proprietary protection circuitry, damage may occur on devices subjected to high energy ESD. Therefore, proper ESD precautions should be taken to avoid performance degradation or loss of functionality.

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS

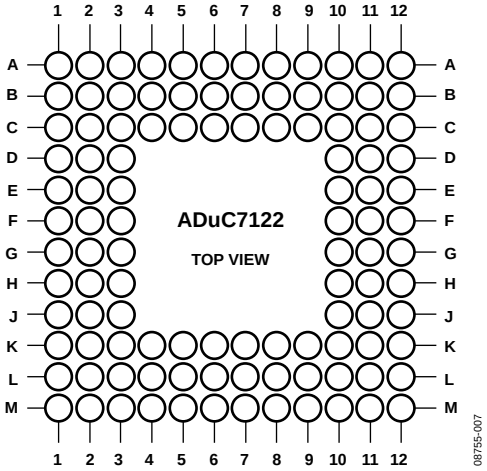


Figure 7. Pin Configuration

Table 9. Pin Function Descriptions

Pin No.	Mnemonic	Type[1]	Description
C12	$\overline{\text{RST}}$	I	Reset Input (Active Low).
D11	P0.0/SCL1/PLAI[5]	I/O	General-Purpose Input and Output Port 0.0 (P0.0). I^2C Interface SCLOCK for I2C0 (SCL1). Input to PLA Element 5 (PLAI[5]).
E11	P0.1/SDA1/PLAI[4]	I/O	General-Purpose Input and Output Port 0.1 (P0.1). I^2C Interface SDATA for I2C0 (SDA1). Input to PLA Element 4 (PLAI[4]).
C3	P0.2/SPICLK/ADC_{Busy}/PLAO[13]	I/O	General-Purpose Input and Output Port 0.2 (P0.2). SPI Clock (SPICLK). Status of the ADC (ADC_{Busy}). Output of PLA Element 13 (PLAO[13]).
D3	P0.3/SPIMISO/PLAO[12]/SYNC	I/O	General-Purpose Input and Output Port 0.3 (P0.3). SPI Master Input, Slave Output (SPIMISO). Output of PLA Element 12 (PLAO[12]). Input to Synchronously Reset PWM Counters Using an External Source (SYNC).
E3	P0.4/SPIMOSI/PLAI[11]/TRIP	I/O	General-Purpose Input and Output Port 0.4 (P0.4). SPI Master Out, Slave Input (SPIMOSI). Input to PLA Element 11 (PLAI[11]). Input that Allows the PWM Trip Interrupt to Be Triggered (TRIP).
F3	P0.5/SPI$\overline{\text{CS}}$/PLAI[10]/$\overline{\text{CONVST}}$	I/O	General-Purpose Input and Output Port 0.5 (P0.5). SPI Slave Select Input (SPI$\overline{\text{CS}}$). Input to PLA Element 10 (PLAI[10]). Initiates ADC Conversions Using PLA or Timer Output ($\overline{\text{CONVST}}$).
G3	P0.6/$\overline{\text{MRST}}$/PLAI[2]	I/O	General-Purpose Input and Output Port 0.6 (P0.6). Power-On Reset Output ($\overline{\text{MRST}}$). Input to PLA Element 2 (PLAI[2])
G10	P0.7/$\overline{\text{TRST}}$/PLAI[3]	I/O	General-Purpose Input and Output Port 0.7 (P0.7). JTAG Test Port Input, Test Reset ($\overline{\text{TRST}}$). Debug and download access. Input to PLA Element 3 (PLAI[3]).
C2	P1.0/SIN/SCL2/PLAI[7]	I/O	General-Purpose Input and Output Port 1.0 (P1.0). Serial Input, Receive Data (RxD), UART (SIN) I^2C Interface SCLOCK for I2C1 (SCL2). Input to PLA Element 7 (PLAI[7]).
D2	P1.1/SOUT/SDA2/PLAI[6]	I/O	General-Purpose Input and Output Port 1.1 (P1.1). Serial Output, Transmit Data (TxD), UART (SOUT) I^2C Interface SDATA for I2C1 (SDA2). Input to PLA Element 6 (PLAI[6]).

Pin No.	Mnemonic	Type[1]	Description
H3	P1.4/PWM1/PLAI[8]/ECLK/XCLK	I/O	General-Purpose Input and Output Port 1.4 (P1.4). PWM1 Output (PWM1). Input to PLA Element 8 (PLAI[8]). Base System Clock Output (ECLK). Base System Clock Input (XCLK).
J3	P1.5/PWM2/PLAI[9]	I/O	General-Purpose Input and Output Port 1.5 (P1.5). PWM2 Output (PWM2). Input to PLA Element 9 (PLAI[9]).
B3	P1.6/PLAO[5]	I/O	General-Purpose Input and Output Port 1.6 (P1.6). Output of PLA Element 5 (PLAO[5]).
B2	P1.7/PLAO[4]	I/O	General-Purpose Input and Output Port 1.7 (P1.7). Output of PLA Element 4 (PLAO[4]).
F11	P2.0/IRQ0/PLAI[13]	I/O	General-Purpose Input and Output Port 2.0 (P2.0). External Interrupt Request 0 (IRQ0). Input to PLA Element 13 (PLAI[13]).
G11	P2.1/IRQ1/PLAI[12]	I/O	General-Purpose Input and Output Port 2.1 (P2.1). External Interrupt Request 1 (IRQ1). Input to PLA Element 12 (PLAI[12]).
H11	P2.2/PLAI[1]	I/O	General-Purpose Input and Output Port 2.2 (P2.2). Input to PLA Element 1 (PLAI[1]).
J11	P2.3/IRQ2/PLAI[14]	I/O	General-Purpose Input and Output Port 2.3 (P2.3). External Interrupt Request 2 (IRQ2). Input to PLA Element 14 (PLAI[14]).
H10	P2.4/PWM5/PLAO[7]	I/O	General-Purpose Input and Output Port 2.4 (P2.4). PWM5 Output (PWM5). Output of PLA Element 7 (PLAO[7]).
J10	P2.5/PWM6/PLAO[6]	I/O	General-Purpose Input and Output Port 2.5 (P2.5). PWM6 Output (PWM6). Output of PLA Element 6 (PLAO[6]).
C1	P2.6/IRQ3/PLAI[15]	I/O	General-Purpose Input and Output Port 2.6 (P2.6). External Interrupt Request 3 (IRQ3). Input to PLA Element 15 (PLAI[15]).
C9	P2.7/PLAI[0]	I/O	General-Purpose Input and Output Port 2.7 (P2.7). Input to PLA Element 0 (PLAI[0]).
C4	P3.0/PLAO[0]	I/O	General-Purpose Input and Output Port 3.0 (P3.0). Output of PLA Element 0 (PLAO[0]).
C11	P3.1/PLAO[1]	I/O	General-Purpose Input and Output Port 3.1 (P3.1). Output of PLA Element 1 (PLAO[1]).
D1	P3.2/IRQ4/PWM3/PLAO[2]	I/O	General-Purpose Input and Output Port 3.2 (P3.2). External Interrupt Request 4 (IRQ4). PWM3 Output (PWM3). Output of PLA Element 2 (PLAO[2]).
E1	P3.3/IRQ5/PWM4/PLAO[3]	I/O	General-Purpose Input and Output Port 3.3 (P3.3). External Interrupt Request 5 (IRQ5). PWM4 Output (PWM4). Output of PLA Element 3 (PLAO[3]).
E2	P3.4/PLAO[8]	I/O	General-Purpose Input and Output Port 3.4 (P3.4). Output of PLA Element 8 (PLAO[8]).
F2	P3.5/PLAO[9]	I/O	General-Purpose Input and Output Port 3.5 (P3.5). Output of PLA Element 9 (PLAO[9]).
D12	P3.6/PLAO[10]	I/O	General-Purpose Input and Output Port 3.6 (P3.6). Output of PLA Element 10 (PLAO[10]).

Pin No.	Mnemonic	Type[1]	Description
E12	P3.7/$\overline{BM}$/PLAO[11]	I/O	General-Purpose Input and Output Port 3.7 (P3.7). Boot Mode ($\overline{BM}$). If $\overline{BM}$ is low and Address 0x00014 of Flash is 0xFFFFFFFF, then the part enters I^2C download after the next rest sequence. Output of PLA Element 11 (PLAO[11]).
L8	V_{REF}_2.5	AI/O	2.5 V Reference Output, External 2.5 V Reference Input. Can be used to drive the anode of a photo diode
L5	V_{REF}_1.2	AI/O	1.2 V Reference Output, External 1.2 V Reference Input. Cannot be used to source current externally.
B8	NC	NC	No Connect.
K6	BUF_VREF1	AO	Buffered 2.5 V Bias. Maximum load = 1.2 mA.
K7	BUF_VREF2	AO	Buffered 2.5 V Bias. Maximum load = 1.2 mA.
L6	PADC0P	AI	PADC0 Positive Input Channel. PGA-based ADC input channel.
M5	PADC0N	AI	PADC0 Negative Input Channel. PGA-based ADC input channel.
L7	PADC1P	AI	PADC1 Positive Input Channel. PGA-based ADC input channel.
M8	PADC1N	AI	PADC1 Negative Input Channel. PGA-based ADC input channel.
K5	ADC0	AI	Single-Ended or Differential Analog Input 0.
K4	ADC1	AI	Single-Ended or Differential Analog Input 1.
M4	ADC2	AI	Single-Ended or Differential Analog Input 2.
L4	ADC3	AI	Single-Ended or Differential Analog Input 3.
K3	ADC4	AI	Single-Ended or Differential Analog Input 4.
M3	ADC5	AI	Single-Ended or Differential Analog Input 5.
M10	ADC6	AI	Single-Ended or Differential Analog Input 6.
M9	ADC7	AI	Single-Ended or Differential Analog Input 7.
L9	ADC8	AI	Single-Ended or Differential Analog Input 8.
K9	ADC9	AI	Single-Ended or Differential Analog Input 9.
K8	ADC10/AINCM	AI	Single-Ended or Differential Analog Input 10. Common Mode for Pseudo Differential Input (AINCM).
K1	DAC0	AO	12-Bit DAC Output.
K2	DAC1	AO	12-Bit DAC Output.
J2	DAC2	AO	12-Bit DAC Output.
L2	DAC3	AO	12-Bit DAC Output.
M2	DAC4	AO	12-Bit DAC Output.
L3	DAC5	AO	12-Bit DAC Output.
M11	DAC6	AO	12-Bit DAC Output.
L11	DAC7	AO	12-Bit DAC Output.
L10	DAC8	AO	12-Bit DAC Output.
K10	DAC9	AO	12-Bit DAC Output.
K11	DAC10	AO	12-Bit DAC Output.
K12	DAC11	AO	12-Bit DAC Output.
B5	NC	NC	No Connect.
C6	NC	NC	No Connect.
A6	NC	NC	No Connect.
A8	NC	NC	No Connect.
A7	NC	NC	No Connect.
C8	NC	NC	No Connect.
A5	NC	NC	No Connect.
C5	NC	NC	No Connect.
B4	NC	NC	No Connect.
A4	NC	NC	No Connect.
A1	NC	NC	No Connect.
A3	NC	NC	No Connect.
A2	NC	NC	No Connect.
B1	NC	NC	No Connect.
A12	NC	NC	No Connect.

Pin No.	Mnemonic	Type[1]	Description
A9	NC	NC	No Connect.
A11	NC	NC	No Connect.
A10	NC	NC	No Connect.
B12	NC	NC	No Connect.
B11	NC	NC	No Connect.
B10	AGND	S	Analog Ground.
B9	AGND	S	Analog Ground.
M1	AGND	S	Analog Ground.
M6	AGND	S	Analog Ground.
L1	AVDD	S	Analog Supply (3.3 V).
M7	AVDD	S	Analog Supply (3.3 V).
M12	AGND	S	Analog Ground.
B6	AGND	S	Analog Ground.
L12	AVDD	S	Analog Supply (3.3 V).
C7	NC	NC	No Connect.
B7	REG_PWR	S	Output of 2.5 V On-Chip Regulator. A 470 nF capacitor to DGND must be connected to this pin.
G1	LVDD	S	Output of 2.6 V On-Chip LDO Regulator. A 470 nF capacitor to DGND must be connected to this pin.
G12	LVDD	S	Output of 2.6 V On-Chip LDO Regulator. A 470 nF capacitor to DGND must be connected to this pin.
F1	DGND	S	Digital ground.
F12	DGND	S	Digital ground.
H1	IOVDD	S	3.3 V GPIO Supply.
J1	IOGND	S	3.3 V GPIO Ground.
H12	IOVDD	S	3.3 V GPIO Supply.
J12	IOGND	S	3.3 V GPIO Ground.
G2	XTALO	DO	Output from the Crystal Oscillator Inverter. If an external crystal is not being used, this pin can be left unconnected.
H2	XTALI	DI	Input to the Crystal Oscillator Inverter and Input to the Internal Clock Generator Circuits. If an external crystal is not being used, this pin should be connected to the DGND system ground.
D10	TDO/P1.3/PLAO[14]	DO	JTAG Test Port Output, Test Data Out (TDO). Debug and download access. General-Purpose Input and Output Port 1.3 (P1.3). Output of PLA Element 14 (PLAO[14]). This pin should not be used as a GPIO when debugging via the JTAG interface.
C10	TDI/P1.2/PLAO[15]	DI	JTAG Test Port Input, Test Data In (TDI). Debug and download access. General-Purpose Input and Output Port 1.2 (P1.2). Output of PLA Element 15 (PLAO[15]). This pin should not be used as a GPIO when debugging via the JTAG interface.
F10	TCK	DI	JTAG Test Port Input, Test Clock. Debug and download access.
E10	TMS	DI	JTAG Test Port Input, Test Mode Select. Debug and download access.

[1] I = input, I/O = input/output, AI/O = analog input/output, NC = no connect, AO = analog output, AI = analog input, DI = digital input, DO = digital output, S = supply.

TERMINOLOGY

ADC SPECIFICATIONS

Integral Nonlinearity (INL)
The maximum deviation of any code from a straight line passing through the endpoints of the ADC transfer function. The endpoints of the transfer function are zero scale, a point ½ LSB below the first code transition, and full scale, a point ½ LSB above the last code transition.

Differential Nonlinearity (DNL)
The difference between the measured and the ideal 1 LSB change between any two adjacent codes in the ADC.

Offset Error
The deviation of the first code transition (0000…000) to (0000…001) from the ideal, that is, ½ LSB.

Gain Error
The deviation of the last code transition from the ideal AIN voltage (full scale − 1.5 LSB) after the offset error has been adjusted out.

Signal-to-(Noise + Distortion) Ratio
The measured ratio of signal to (noise + distortion) at the output of the ADC. The signal is the rms amplitude of the fundamental. Noise is the rms sum of all nonfundamental signals up to half the sampling frequency ($f_S/2$), excluding dc. The ratio is dependent upon the number of quantization levels in the digitization process; the more levels there are, the smaller the quantization noise becomes.

The theoretical signal-to-(noise + distortion) ratio for an ideal N-bit converter with a sine wave input is given by

$$Signal\ to\ (Noise + Distortion) = (6.02\ N + 1.76)\ dB$$

Thus, for a 12-bit converter, this is 74 dB.

Total Harmonic Distortion
The ratio of the rms sum of the harmonics to the fundamental.

DAC SPECIFICATIONS

Relative Accuracy
Otherwise known as endpoint linearity, relative accuracy is a measure of the maximum deviation from a straight line passing through the endpoints of the DAC transfer function. It is measured after adjusting for zero error and full-scale error.

Voltage Output Settling Time
The amount of time it takes the output to settle to within a 1 LSB level for a full-scale input change.

OVERVIEW OF THE ARM7TDMI CORE

The ARM7® core is a 32-bit reduced instruction set computer (RISC). It uses a single 32-bit bus for instruction and data. The length of the data can be eight bits, 16 bits, or 32 bits. The length of the instruction word is 32 bits.

The ARM7TDMI is an ARM7 core with four additional features:

- T support for the thumb (16-bit) instruction set
- D support for debugging
- M support for long multiplications
- 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 that has been compressed into 16 bits, called the thumb instruction set. Faster execution from 16-bit memory and greater code density can usually be achieved by using the thumb instruction set instead of the ARM instruction set, which makes the ARM7TDMI core particularly suitable for embedded applications.

However, the thumb mode has two limitations:

- Thumb code typically requires more instructions for the same job. As a result, ARM code is usually best for maximizing the performance of time-critical code.
- The thumb instruction set does not include some of the instructions needed for exception handling, which automatically switches the core to ARM code for exception handling.

See the ARM7TDMI user guide for details on the core architecture, the programming model, and both the ARM and ARM thumb instruction sets.

LONG MULTIPLY (M)

The ARM7TDMI instruction set includes four extra instructions that perform 32-bit by 32-bit multiplication with a 64-bit result, and 32-bit by 32-bit multiplication-accumulation (MAC) with a 64-bit result. These results are achieved in fewer cycles than required on a standard ARM7 core.

EmbeddedICE (I)

EmbeddedICE provides integrated on-chip support for the core. The EmbeddedICE module contains the breakpoint and watchpoint registers that allow code to be halted for debugging purposes. These registers are controlled through the JTAG test port.

When a breakpoint or watchpoint is encountered, the processor halts and enters a debug state. Once in a debug state, the processor registers can be inspected as well as the Flash/EE, SRAM, and memory mapped registers.

EXCEPTIONS

ARM supports five types of exceptions and a privileged processing mode for each type. The five types of exceptions are:

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

Typically, the programmer defines an interrupt as IRQ, but for a higher priority interrupt, that is, faster response time, the programmer can define the interrupt as FIQ.

ARM REGISTERS

ARM7TDMI has a total of 37 registers: 31 general-purpose registers and six status registers. Each operating mode has dedicated banked registers.

When writing user-level programs, 15 general-purpose, 32-bit registers (R0 to R14), the program counter (R15) and the current program status register (CPSR) are usable. The remaining registers are only used for system-level programming and exception handling.

When an exception occurs, some of the standard registers 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 8. The fast interrupt mode has more registers (R8 to R12) for fast interrupt processing. This means the interrupt processing can begin without the need to save or restore these registers, and thus save critical time in the interrupt handling process.

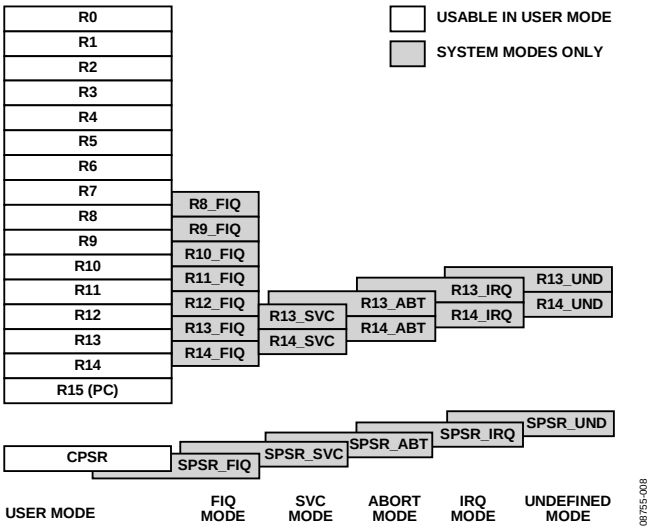


Figure 8. Register Organization

More information relative to the programmer's model and the ARM7TDMI core architecture can be found in the following materials from ARM:

- DDI 0029G, *ARM7TDMI Technical Reference Manual*
- DDI 0100, *ARM Architecture Reference Manual*

INTERRUPT LATENCY

The worst-case latency for a fast interrupt request (FIQ) consists of the following:

- The longest time the request can take to pass through the synchronizer
- The time for the longest instruction to complete (the longest instruction is an LDM) that loads all the registers including the PC
- The time for the data abort entry
- The time for FIQ entry

At the end of this time, the ARM7TDMI executes the instruction at 0x1C (FIQ interrupt vector address). The maximum total time is 50 processor cycles, which is just under 1.2 µs in a system using a continuous 41.78 MHz processor clock.

The maximum interrupt request (IRQ) latency calculation is similar but must allow for the fact that FIQ has higher priority and can delay entry into the IRQ handling routine for an arbitrary length of time. This time can 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 the time is reduced to 22 cycles.

The minimum latency for FIQ or IRQ interrupts is a total of five cycles, which consist of the shortest time the request can take through the synchronizer, plus the time to enter the exception mode.

Note that the ARM7TDMI always runs in ARM (32-bit) mode when in privileged mode, for example, when executing interrupt service routines.

MEMORY ORGANIZATION

The ADuC7122 incorporates three separate blocks of memory: 8 kB of SRAM and two 64 kB of on-chip Flash/EE memory. There are 126 kB of on-chip Flash/EE memory available to the user, and the remaining 2 kB are reserved for the factory-configured boot page. These two blocks are mapped as shown in Figure 9.

Note that by default, after a reset, the Flash/EE memory is mirrored at Address 0x00000000. It is possible to remap the SRAM at Address 0x00000000 by clearing Bit 0 of the REMAP MMR. This remap function is described in more detail in the Flash/EE Memory section.

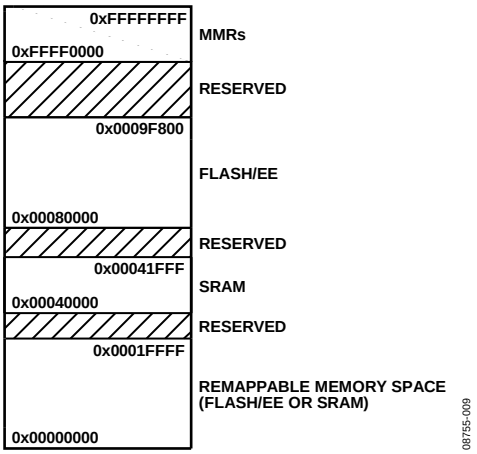


Figure 9. Physical Memory Map

Memory Access

The ARM7 core sees memory as a linear array of 2^{32} byte locations, where the different blocks of memory are mapped as outlined in Figure 9.

The ADuC7122 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.

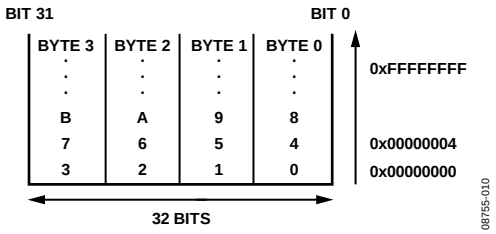


Figure 10. Little Endian Format

FLASH/EE MEMORY

The 128 kB of Flash/EE are organized as two banks of 32k × 16 bits. In the upper memory block, 31k × 16 bits are user space, and 1k × 16 bits are reserved for the factory-configured boot page. The page size of this Flash/EE memory is 512 bytes.

The lower 64 kB memory block is organized in a similar manner. It is arranged in 32k × 16 bits. All of this is available as user space.

The 126 kB of Flash/EE are available to the user as code and nonvolatile data memory. There is no distinction between data and program because 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. Therefore, it is recommended that Thumb mode be used when executing from Flash/EE memory for optimum access speed. The maximum access speed for the Flash/EE memory is 41.78 MHz in Thumb mode and 20.89 MHz in full ARM mode (see the Execution Time from SRAM and FLASH/EE section).

SRAM

The 8 kB of SRAM are available to the user, organized as 2k × 32 bits, that is, 2k words. ARM code can run directly from SRAM at 41.78 MHz, given that the SRAM array is configured as a 32-bit wide memory array (see the Execution Time from SRAM and FLASH/EE section).

MEMORY MAPPED REGISTERS

The memory mapped register (MMR) space is mapped into the upper two pages of the memory array and accessed by indirect addressing through the ARM7 banked registers.

The MMR space provides an interface between the CPU and all on-chip peripherals. All registers except the core registers reside in the MMR area. All shaded locations shown in Figure 11 are unoccupied or reserved locations and should not be accessed by user software. Table 10 to Table 26 show a full MMR memory map.

The access time reading or writing a MMR depends on the advanced microcontroller bus architecture (AMBA) bus used to access the peripheral. The processor has two AMBA buses: advanced high performance bus (AHB) used for system modules and advanced peripheral bus (APB) used for lower performance peripheral. Access to the AHB is one cycle, and access to the APB is two cycles. All peripherals on the ADuC7122 are on the APB except the Flash/EE memory and the GPIOs.

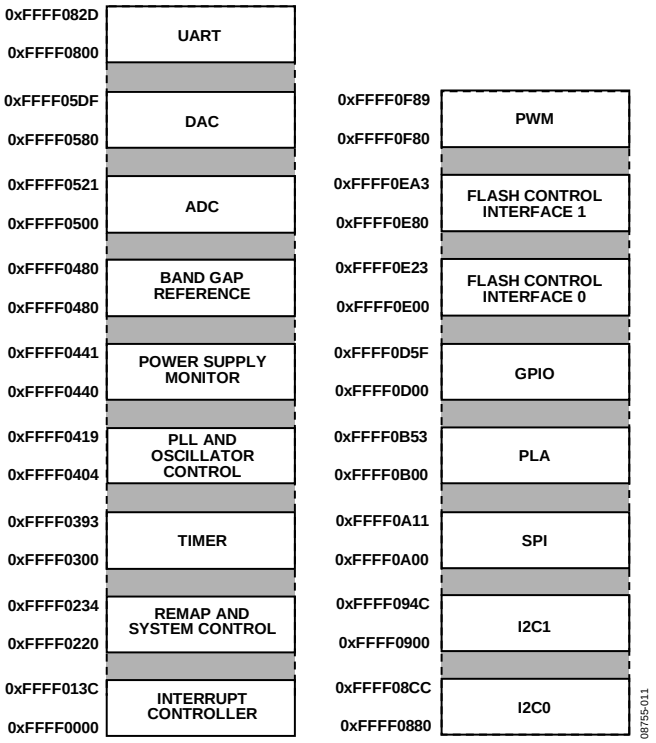


Figure 11. Memory Mapped Registers

COMPLETE MMR LISTING

Note that the access type column in Table 10 to Table 26 corresponds to the access time reading or writing an MMR. It depends on the AMBA bus used to access the peripheral. The processor has two AMBA buses: the AHB (advanced high performance bus) used for system modules and the APB (advanced peripheral bus) used for lower performance peripherals.

Table 10. IRQ Base Address = 0xFFFF0000

Address	Name	Byte	Access Type	Cycle
0x0000	IRQSTA	4	R	1
0x0004	IRQSIG	4	R	1
0x0008	IRQEN	4	R/W	1
0x000C	IRQCLR	4	W	1
0x0010	SWICFG	4	W	1
0x0014	IRQBASE	4	R/W	1
0x001C	IRQVEC	4	R	1
0x0020	IRQP0	4	R/W	1
0x0024	IRQP1	4	R/W	1
0x0028	IRQP2	4	R/W	1
0x002C	IRQP3	4	R/W	1
0x0030	IRQCONN	1	R/W	1
0x0034	IRQCONE	1	R/W	1
0x0038	IRQCLRE	1	W	1
0x003C	IRQSTAN	1	R/W	
0x0100	FIQSTA	4	R	1
0x0104	FIQSIG	4	R	1
0x0108	FIQEN	4	R/W	1
0x010C	FIQCLR	4	W	1
0x011C	FIQVEC	4	R	1
0x013C	FIQSTAN	1	R/W	1

Table 11. System Control Base Address = 0xFFFF0200

Address	Name	Byte	Access Type	Cycle
0x0220	REMAP	1	R/W	1
0x0230	RSTSTA	1	R	1
0x0234	RSTCLR	1	W	1
0x0248	RSTKEY1	1	W	N/A
0x024C	RSTCFG	1	R/W	0x00
0x0250	RSTKEY2	1	W	N/A

Table 12. Timer Base Address = 0xFFFF0300

Address	Name	Byte	Access Type	Cycle
0x0300	T0LD	2	R/W	2
0x0304	T0VAL0	2	R	2
0x0308	T0VAL1	4	R	2
0x030C	T0CON	4	R/W	2
0x0310	T0CLRI	1	W	2
0x0314	T0CAP	2	R	2
0x0320	T1LD	4	R/W	2
0x0324	T1VAL	4	R	2
0x0328	T1CON	4	R/W	2
0x032C	T1CLRI	1	W	2
0x0330	T1CAP	4	R	2
0x0340	T2LD	4	R/W	2
0x0344	T2VAL	4	R	2
0x0348	T2CON	4	R/W	2
0x034C	T2CLRI	1	W	2
0x0360	T3LD	2	R/W	2
0x0364	T3VAL	2	R	2
0x0368	T3CON	2	R/W	2
0x036C	T3CLRI	1	W	2
0x0380	T4LD	4	R/W	2
0x0384	T4VAL	4	R	2
0x0388	T4CON	4	R/W	2
0x038C	T4CLRI	1	W	2
0x0390	T4CAP	4	R	2

Table 13. PLL Base Address = 0xFFFF0400

Address	Name	Byte	Access Type	Cycle
0x0404	POWKEY1	2	W	2
0x0408	POWCON	1	R/W	2
0x040C	POWKEY2	2	W	2
0x0410	PLLKEY1	2	W	2
0x0414	PLLCON	1	R/W	2
0x0418	PLLKEY2	2	W	2

Table 14. PSM Base Address = 0xFFFF0440

Address	Name	Byte	Access Type	Cycle
0x0440	PSMCON	2	R/W	2

Table 15. Reference Base Address = 0xFFFF0480

Address	Name	Byte	Access Type	Cycle
0x0480	REFCON	1	R/W	2

Table 16. ADC Base Address = 0xFFFF0500

Address	Name	Byte	Access Type	Cycle
0x0500	ADCCON	4	R/W	2
0x0504	ADCCP	1	R/W	2
0x0508	ADCCN	1	R/W	2
0x050C	ADCSTA	1	R	2
0x0510	ADCDAT	4	R	2
0x0514	ADCRST	1	W	2
0x0520	PGA_GN	2	R/W	2

Table 17. DAC Base Address = 0xFFFF0580

Address	Name	Byte	Access Type	Cycle
0x0580	DAC0CON	2	R/W	2
0x0584	DAC0DAT	4	R/W	2
0x0588	DAC1CON	2	R/W	2
0x058C	DAC1DAT	4	R/W	2
0x0590	DAC2CON	2	R/W	2
0x0594	DAC2DAT	4	R/W	2
0x0598	DAC3CON	2	R/W	2
0x059C	DAC3DAT	4	R/W	2
0x05A0	DAC4CON	2	R/W	2
0x05A4	DAC4DAT	4	R/W	2
0x05A8	DAC5CON	2	R/W	2
0x05AC	DAC5DAT	4	R/W	2
0x05B0	DAC6CON	2	R/W	2
0x05B4	DAC6DAT	4	R/W	2
0x05B8	DAC7CON	2	R/W	2
0x05BC	DAC7DAT	4	R/W	2
0x05C0	DAC8CON	2	R/W	2
0x05C4	DAC8DAT	4	R/W	2
0x05C8	DAC9CON	2	R/W	2
0x05CC	DAC9DAT	4	R/W	2
0x05D0	DAC10CON	2	R/W	2
0x05D4	DAC10DAT	4	R/W	2
0x05D8	DAC11CON	2	R/W	2
0x05DC	DAC11DAT	4	R/W	2

Table 18. UART0 Base Address = 0xFFFF0800

Address	Name	Byte	Access Type	Cycle
0x0800	COMTX	1	W	2
0x0800	COMRX	1	R	2
0x0800	COMDIV0	1	R/W	2
0x0804	COMIEN0	1	R/W	2
0x0804	COMDIV1	1	R/W	2
0x0808	COMIID0	1	R	2
0x080C	COMCON0	1	R/W	2
0x0810	COMCON1	1	R/W	2
0x0814	COMSTA0	1	R	2
0x0818	COMSTA1	1	R	2
0x081C	COMSCR	1	R/W	2
0x0820	COMIEN1	1	R/W	2
0x0824	COMIID1	1	R	2
0x0828	COMADR	1	R/W	2
0x082C	COMDIV2	2	R/W	2

Table 19. I2C0 Base Address = 0xFFFF0880

Address	Name	Byte	Access Type	Cycle
0x0880	I2C0MCTL	2	R/W	2
0x0884	I2C0MSTA	2	R	2
0x0888	I2C0MRX	1	R	2
0x088C	I2C0MTX	2	R/W	2
0x0890	I2C0MCNT0	2	R/W	2
0x0894	I2C0MCNT1	1	R	2
0x0898	I2C0ADR0	1	R/W	2
0x089C	I2C0ADR1	1	R/W	2
0x08A0	I2C0SBYTE	1	R/W	2
0x08A4	I2C0DIV	2	R/W	2
0x08A8	I2C0SCTL	2	R/W	2
0x08AC	I2C0SSTA	2	R	2
0x08B0	I2C0SRX	1	R	2
0x08B4	I2C0STX	1	R/W	2
0x08B8	I2C0ALT	1	R/W	2
0x08BC	I2C0ID0	1	R/W	2
0x08C0	I2C0ID1	1	R/W	2
0x08C4	I2C0ID2	1	R/W	2
0x08C8	I2C0ID3	1	R/W	2
0x08CC	I2C0FSTA	1	R/W	2

Table 20. I2C1 Base Address = 0xFFFF0900

Address	Name	Byte	Access Type	Cycle
0x0900	I2C1MCTL	2	R/W	2
0x0904	I2C1MSTA	2	R	2
0x0908	I2C1MRX	1	R	2
0x090C	I2C1MTX	2	R/W	2
0x0910	I2C1MCNT0	2	R/W	2
0x0914	I2C1MCNT1	1	R	2
0x0918	I2C1ADR0	1	R/W	2
0x091C	I2C1ADR1	1	R/W	2
0x0920	I2C1SBYTE	1	R/W	2
0x0924	I2C1DIV	2	R/W	2
0x0928	I2C1SCTL	2	R/W	2
0x092C	I2C1SSTA	2	R	2
0x0930	I2C1SRX	1	R	2
0x0934	I2C1STX	1	R/W	2
0x0938	I2C1ALT	1	R/W	2
0x093C	I2C1ID0	1	R/W	2
0x0940	I2C1ID1	1	R/W	2
0x0944	I2C1ID2	1	R/W	2
0x0948	I2C1ID3	1	R/W	2
0x094C	I2C1FSTA	1	R/W	2

Table 21. SPI Base Address = 0xFFFF0A00

Address	Name	Byte	Access Type	Cycle
0x0A00	SPISTA	1	R	2
0x0A04	SPIRX	1	R	2
0x0A08	SPITX	1	W	2
0x0A0C	SPIDIV	1	R/W	2
0x0A10	SPICON	2	R/W	2

Table 22. PLA Base Address = 0xFFFF0B00

Address	Name	Byte	Access Type	Cycle
0x0B00	PLAELM0	2	R/W	2
0x0B04	PLAELM1	2	R/W	2
0x0B08	PLAELM2	2	R/W	2
0x0B0C	PLAELM3	2	R/W	2
0x0B10	PLAELM4	2	R/W	2
0x0B14	PLAELM5	2	R/W	2
0x0B18	PLAELM6	2	R/W	2
0x0B1C	PLAELM7	2	R/W	2
0x0B20	PLAELM8	2	R/W	2
0x0B24	PLAELM9	2	R/W	2
0x0B28	PLAELM10	2	R/W	2
0x0B2C	PLAELM11	2	R/W	2
0x0B30	PLAELM12	2	R/W	2
0x0B34	PLAELM13	2	R/W	2
0x0B38	PLAELM14	2	R/W	2
0x0B3C	PLAELM15	2	R/W	2
0x0B40	PLACLK	1	R/W	2
0x0B44	PLAIRQ	4	R/W	2
0x0B48	PLAADC	4	R/W	2
0x0B4C	PLADIN	4	R/W	2
0x0B50	PLADOUT	4	R	2

Table 23. GPIO Base Address = 0xFFFF0D00

Address	Name	Byte	Access Type	Cycle
0x0D00	GP0CON	4	R/W	1
0x0D04	GP1CON	4	R/W	1
0x0D08	GP2CON	4	R/W	1
0x0D0C	GP3CON	4	R/W	1
0x0D20	GP0DAT	4	R/W	1
0x0D24	GP0SET	1	W	1
0x0D28	GP0CLR	1	W	1
0x0D2C	GP0PAR	4	R/W	1
0x0D30	GP1DAT	4	R/W	1
0x0D34	GP1SET	1	W	1
0x0D38	GP1CLR	1	W	1
0x0D3C	GP1PAR	4	R/W	1
0x0D40	GP2DAT	4	R/W	1
0x0D44	GP2SET	1	W	1
0x0D48	GP2CLR	1	W	1
0x0D4C	GP2PAR	4	R/W	1
0x0D50	GP3DAT	4	R/W	1
0x0D54	GP3SET	1	W	1
0x0D58	GP3CLR	1	W	1
0x0D5C	GP3PAR	4	R/W	1
0x0D70	GP1OCE	1	W	1
0x0D74	GP2OCE	1	W	1
0x0D78	GP3OCE	1	W	1

Table 24. Flash/EE Block 0 Base Address = 0xFFFF0E00

Address	Name	Byte	Access Type	Cycle
0x0E00	FEE0STA	1	R	1
0x0E04	FEE0MOD	1	R/W	1
0x0E08	FEE0CON	1	R/W	1
0x0E0C	FEE0DAT	2	R/W	1
0x0E10	FEE0ADR	2	R/W	1
0x0E18	FEE0SGN	3	R	1
0x0E1C	FEE0PRO	4	R/W	1
0x0E20	FEE0HID	4	R/W	1

Table 25. Flash/EE Block 1 Base Address = 0xFFFF0E80

Address	Name	Byte	Access Type	Cycle
0x0E80	FEE1STA	1	R	1
0x0E84	FEE1MOD	1	R/W	1
0x0E88	FEE1CON	1	R/W	1
0x0E8C	FEE1DAT	2	R/W	1
0x0E90	FEE1ADR	2	R/W	1
0x0E98	FEE1SGN	3	R	1
0x0E9C	FEE1PRO	4	R/W	1
0x0EA0	FEE1HID	4	R/W	1

Table 26. PWM Base Address= 0xFFFF0F80

Address	Name	Byte	Access Type	Cycle
0x0F80	PWMCON1	2	R/W	2
0x0F84	PWM1COM1	2	R/W	2
0x0F88	PWM1COM2	2	R/W	2
0x0F8C	PWM1COM3	2	R/W	2
0x0F90	PWM1LEN	2	R/W	2
0x0F94	PWM2COM1	2	R/W	2
0x0F98	PWM2COM2	2	R/W	2
0x0F9C	PWM2COM3	2	R/W	2
0x0FA0	PWM2LEN	2	R/W	2
0x0FA4	PWM3COM1	2	R/W	2
0x0FA8	PWM3COM2	2	R/W	2
0x0FAC	PWM3COM3	2	R/W	2
0x0FB0	PWM3LEN	2	R/W	2
0x0FB4	PWMCON2	2	R/W	2
0x0FB8	PWMICLR	2	W	2

ADC CIRCUIT OVERVIEW

The analog-to-digital converter (ADC) incorporates a fast, multichannel, 12-bit ADC. It can operate from 3.0 V to 3.6 V supplies and is capable of providing a throughput of up to 1 MSPS when the clock source is 41.78 MHz. This block provides the user with a multichannel multiplexer, differential track-and-hold, on-chip reference, and ADC.

The ADC consists of a 12-bit successive approximation converter based around two capacitor DACs. Depending on the input signal configuration, the ADC can operate in one of the following three modes:

- Fully differential mode for small and balanced signals
- Single-ended mode for any single-ended signals
- Pseudo differential mode for any single-ended signals, taking advantage of the common-mode rejection offered by the pseudo differential input

The converter accepts an analog input range of 0 to V_{REF} when operating in single-ended mode or pseudo differential mode. In fully differential mode, the input signal must be balanced around a common-mode voltage, V_{CM}, in the range 0 V to AV_{DD}, and with maximum amplitude of 2 V_{REF} (see Figure 12).

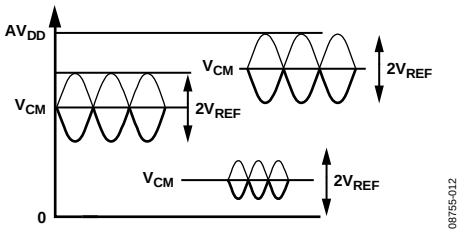


Figure 12. Examples of Balanced Signals for Fully Differential Mode

A high precision, low drift, and factory-calibrated 2.5 V reference is provided on chip. An external reference can also be connected as described in the Band Gap Reference section.

Single or continuous conversion modes can be initiated in software. An external $\overline{\text{CONVST}}$ pin, an output generated from the on-chip PLA, or a Timer0 or Timer1 overflow can also be used to generate a repetitive trigger for ADC conversions.

If the signal has not been deasserted by the time the ADC conversion is complete, a second conversion begins automatically.

A voltage output from an on-chip band gap reference proportional to absolute temperature can also be routed through the front-end ADC multiplexer, effectively an additional ADC channel input. This facilitates an internal temperature sensor channel, measuring die temperature to an accuracy of ±3°C.

For the ADuC7122, a number of modifications have been made to the ADC input structure that appears in the ADuC702x family.

The PADC0 and PADC1 inputs connect to a PGA in pseudo differential mode and allow for selectable gains from 1 to 5 with 32 steps. The remaining ADC channels can be configured as single, differential, or pseudo differential. A buffer is provided before the ADC for measuring internal channels.

ADC TRANSFER FUNCTION

Pseudo Differential and Single-Ended Modes

In pseudo differential or single-ended mode, the input range is 0 V to V_{REF}. The output coding is straight binary in pseudo differential and single-ended modes with

1 LSB = FS/4096 or
2.5 V/4096 = 0.61 mV or
610 µV when V_{REF} = 2.5 V

The ideal code transitions occur midway between successive integer LSB values (that is, 1/2 LSB, 3/2 LSBs, 5/2 LSBs, …, FS – 3/2 LSBs). The ideal input/output transfer characteristic is shown in Figure 13.

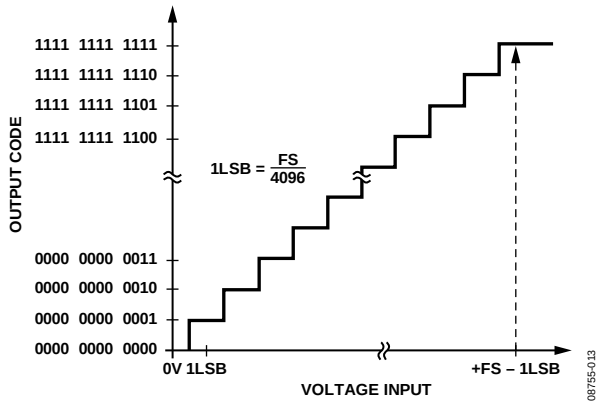


Figure 13. ADC Transfer Function in Pseudo Differential Mode or Single-Ended Mode

Fully Differential Mode

The amplitude of the differential signal is the difference between the signals applied to the V_{IN+} and V_{IN-} inputs (that is, V_{IN+} – V_{IN-}) of the currently enabled differential channel. The maximum amplitude of the differential signal is, therefore, $-V_{REF}$ to $+V_{REF}$ p-p ($2 \times V_{REF}$). This is regardless of the common mode (CM). The common mode is the average of the two signals (V_{IN+} + V_{IN-})/2, and is, therefore, the voltage that the two inputs are centered on. This results in the span of each input being CM ± V_{REF}/2. This voltage must be set up externally, and its range varies with V_{REF} (see the Driving the Analog Inputs section).

The output coding is twos complement in fully differential mode with 1 LSB = 2 V_{REF}/4096 or 2 × 2.5 V/4096 = 1.22 mV when V_{REF} = 2.5 V. The output result is ±11 bits, but this is shifted by one to the right, which allows the result in ADCDAT to be declared as a signed integer when writing C code. The designed code transitions occur midway between successive integer LSB values (that is, 1/2 LSB, 3/2 LSBs, 5/2 LSBs, …, FS – 3/2 LSBs). The ideal input/output transfer characteristic is shown in Figure 14.

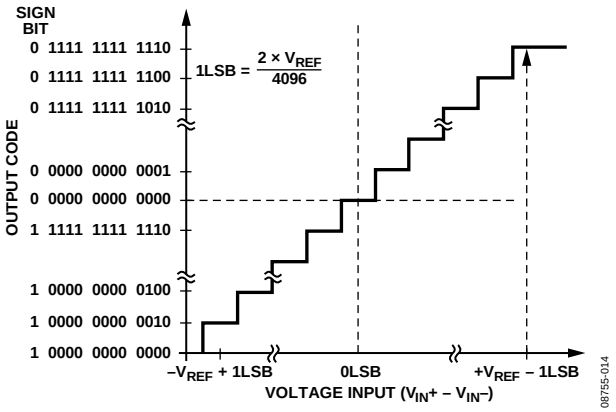


Figure 14. ADC Transfer Function in Differential Mode

ADC Input Channels

The ADuC7122 provides 11 fixed gain ADC input pins. Each of these pins can be separately configured as a differential input pair, single-ended input, or positive side pseudo differential input (the negative side must be the AINCM channel). The buffer and ADC are configured independently from input channel selection. Note that the input range of the ADC input buffer is from 0.15 V to AV_{DD} – 0.15 V. If the input signal range exceeds this range, the input buffer must be bypassed.

The ADC mux can be configured to select an internal channel like IOVDD_MON or the temperature sensor. When converting on an internal channel, the input buffer must be enabled.

In addition, an on-chip diode can be selected to provide chip temperature monitoring. The ADC can also select V_{REF} and AGND as the input for calibration purposes.

PGA and Input Buffer

The ADuC7122 contains two programmable gain channels that operate in pseudo differential mode. The PGA is a one-stage positive gain amplifier that is able to accept an input from 0.1 V to AV_{DD} – 1.2 V. The PGA output can swing up to 2.5 V. The PGA is designed to handle 10 mV minimum input.

The gain of the PGA is from 1 to 5 with 32 linear steps. The PGA cannot be bypassed for the PADC0 and PADC1 channels.

The PGAs use a PMOS input to minimize nonlinearity and noise. The input level for PGA is limited from AV_{DD} – 1.2 V to 0.1 V to make sure the amplifiers are not saturated. The input buffer is a rail-to-rail buffer. It can accept signals from 0.15 V to AV_{DD} – 0.15 V. Each of the input buffers can be bypassed independently.

To minimize noise, the PADC input buffer can be bypassed.

PADCxN is driven by a buffer to 0.15 V to keep the PGA from saturation when the input current drops to 0. The buffer can be disabled by setting ADCCON[14] so that the PADCxN can be connected to GND as well.

The PADCx channels are only specified to operate in pseudo differential mode and this assumes the negative input is close to ground.

All the controls are independently set through register bits to give maximum flexibility to the user. Typically, users must set the following:

1. Select PADCxP as the PGA input. Enable the PADCxN buffer and optionally disable the PADCxP input buffer.
2. Set the proper gain value for the PGA. Bypass the PADCxN buffer if a grounded signal is required.
3. Set the ADC to pseudo differential mode and start the conversion.

TYPICAL OPERATION

Once configured via the ADC control and channel selection registers, the ADC converts the analog input and provides a 12-bit result in the ADCDAT register.

The top four bits are the sign bits, and the 12-bit result is placed from Bit 16 to Bit 27, as shown in Figure 15. Note that in fully differential mode, the result is represented in twos complement format, and in pseudo differential and single-ended mode, the result is represented in straight binary format.

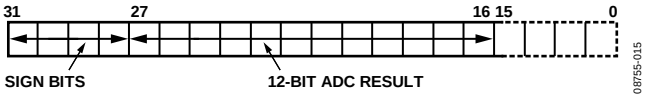


Figure 15. ADC Result Format

Calibration

By default, the factory-set values written to the ADC offset (ADCOF) and gain coefficient registers (ADCGN) yield optimum performance in terms of end-point errors and linearity for standalone operation of the part (see the General Description section). If system calibration is required, it is possible to modify the default offset and gain coefficients to improve end-point errors, but note that any modification to the factory-set ADCOF and ADCGN values can degrade ADC linearity performance.

For system offset error correction, the ADC channel input stage must be tied to AGND. A continuous software ADC conversion loop must be implemented by modifying the value in ADCOF until the ADC result (ADCDAT) reads Code 0 to Code 1. If the ADCDAT value is greater than 1, ADCOF should be decremented until ADCDAT reads Code 0 to Code 1. Offset error correction is performed digitally and has a resolution of 0.25 LSB and a range of ±3.125% of V_{REF}.

For system gain error correction, the ADC channel input stage must be tied to V_{REF}. A continuous software ADC conversion loop must be implemented to modify the value in ADCGN until ADCDAT reads Code 4094 to Code 4095. If the ADCDAT value is less than 4094, ADCGN should be incremented until ADCDAT reads Code 4094 to Code 4095. Similar to the offset calibration, the gain calibration resolution is 0.25 LSB with a range of ±3% of V_{REF}.

Current Consumption

The ADC in standby mode, that is, powered up but not converting, typically consumes 640 μA. The internal reference adds 140 μA. During conversion, the extra current is 0.3 μA, multiplied by the sampling frequency (in kHz).

Timing

Figure 16 gives details of the ADC timing. Users control the ADC clock speed and the number of acquisition clock in the ADCCON MMR. By default, the acquisition time is eight clocks and the clock divider is two. The number of extra clocks (such as bit trial or write) is set to 19, giving a sampling rate of 774 kSPS. For conversion on the temperature sensor, the ADC acquisition time is automatically set to 16 clocks and the ADC clock divider is set to 32. When using multiple channels, including the temperature sensor, the timing settings revert back to the user-defined settings after reading the temperature sensor channel.

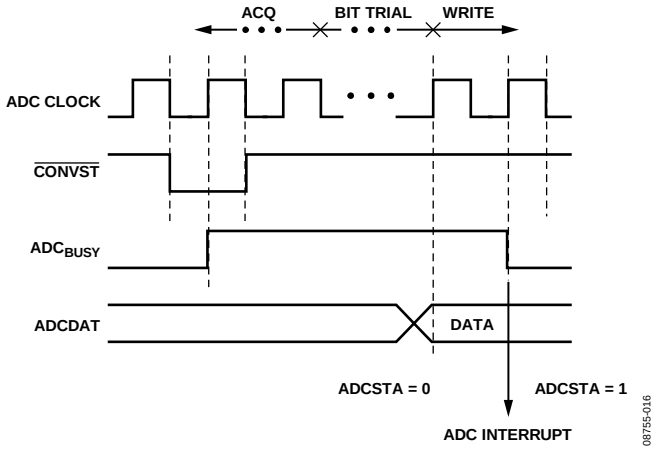


Figure 16. ADC Timing

TEMPERATURE SENSOR

The ADuC7122 provides a voltage output from an on-chip band gap reference proportional to absolute temperature. This voltage output can also be routed through the front-end ADC multiplexer (effectively, an additional ADC channel input), facilitating an internal temperature sensor channel that measures die temperature.

The internal temperature sensor is not designed for use as an absolute ambient temperature calculator. It is intended for use as an approximate indicator of the temperature of the ADuC7122 die.

The typical temperature coefficient is −1.25 mV/°C.

ADC MMRs Interface

The ADC is controlled and configured via a number of MMRs (see Table 27) that are described in detail in Table 28 to Table 34.

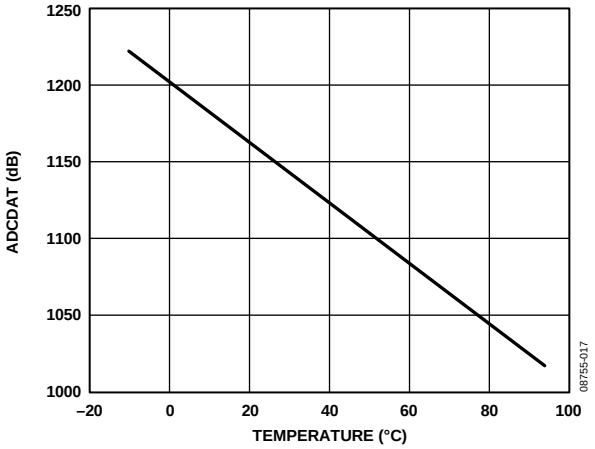


Figure 17. ADC Output vs. Temperature

Table 27. ADC MMRs

Name	Description
ADCCON	ADC control register. Allows the programmer to enable the ADC peripheral to select the mode of operation of the ADC (either single-ended, pseudo differential, or fully differential mode) and to select the conversion type (see Table 28).
ADCCP	ADC positive channel selection register.
ADCCN	ADC negative channel selection register.
ADCSTA	ADC status register. Indicates when an ADC conversion result is ready. The ADCSTA register contains only one bit, ADCREADY (Bit 0), representing the status of the ADC. This bit is set at the end of an ADC conversion generating an ADC interrupt. It is cleared automatically by reading the ADCDAT MMR. When the ADC is performing a conversion, the status of the ADC can be read externally via the ADC_{Busy} pin. This pin is high during a conversion. When the conversion is finished, ADC_{Busy} goes back low. This information is available on P0.2 (see the General-Purpose I/O section) if enabled in the GP0CON register.
ADCDAT	ADC data result register. Holds the 12-bit ADC result, as shown Table 32.
ADCRST	ADC reset register. Resets all the ADC registers to their default value.
PGA_GN	Gain of PADC0 and PADC1.

Table 28. ADCCON MMR Bit Designations (Address = 0xFFFF0500, Default Value = 0x00000A00)

Bit	Value	Description
31:16		These bits are reserved.
15		Positive ADC buffer bypass.
	0	Set to 0 by the user to enable the positive ADC buffer.
	1	Set to 1 by the user to bypass the positive ADC buffer.
14		Negative ADC buffer bypass.
	0	Set to 0 by the user to enable the negative ADC buffer.
	1	Set to 1 by the user to bypass the negative ADC buffer.
13:11		ADC clock speed ($f_{ADC} = f_{CORE}$, conversion = 19 ADC clocks + acquisition time).
	000	$f_{ADC}/1$. This divider is provided to obtain 1 MSPS ADC with an external clock <41.78 MHz.
	001	$f_{ADC}/2$ (default value).
	010	$f_{ADC}/4$.
	011	$f_{ADC}/8$.
	100	$f_{ADC}/16$.
	101	$f_{ADC}/32$.
10:8		ADC acquisition time (number of ADC clocks).
	000	2 clocks.
	001	4 clocks.
	010	8 clocks (default value).
	011	16 clocks.
	100	32 clocks.
	101	64 clocks.
7		Enable conversion.
	1	Set by user to 1 to enable conversion mode.
	0	Cleared by user to 0 to disable conversion mode.
6		Reserved. This bit should be set to 0 by the user.
5		ADC power control.
	1	Set by user to 1 to place the ADC in normal mode. The ADC must be powered up for at least 5 µs before it converts correctly.
	0	Cleared by user to 0 to place the ADC in power-down mode.
4:3		Conversion mode.
	00	Single-ended mode.
	01	Differential mode.
	10	Pseudo differential mode.
	11	Reserved.
2:0		Conversion type.
	000	Enable $\overline{\text{CONVST}}$ pin as a conversion input.
	001	Enable Timer1 as a conversion input.
	010	Enable Timer0 as a conversion input.
	011	Single software conversion. Automatically set to 000 after conversion.
	100	Continuous software conversion.
	101	PLA conversion.
	110	Reserved
	Other	Reserved.

Table 29. ADCCP MMR Bit Designations
(Address = 0xFFFF0504, Default Value = 0x00)

Bit	Value	Description
7:5		Reserved
4:0		Positive channel selection bits
	00000	PADC0P
	00001	PADC1P
	00010	ADC0
	00011	ADC1
	00100	ADC2
	00101	ADC3
	00110	ADC4
	00111	ADC5
	01000	ADC6
	01001	ADC7
	01010	ADC8
	01011	ADC9
	01100	ADC10/AINCM
	01101	Temperature sensor
	01110	Reserved
	01111	Reserved
	10000	Reserved
	10001	Reserved
	10010	Reserved
	10011	IOVDD_MON
	10100	Reserved
	10101	Reserved
	10110	V_{REF}
	10111	AGND
	Others	Reserved

Table 30. ADCCN MMR Bit Designations
(Address = 0xFFFF0508, Default Value = 0x00)

Bit	Value	Description
7:5		Reserved
4:0		Negative channel selection bits
	00000	PADC0N
	00001	PADC1N
	00010	ADC0
	00011	ADC1
	00100	ADC2
	00101	ADC3
	00110	ADC4
	00111	ADC5
	01000	ADC6
	01001	ADC7
	01010	ADC8
	01011	ADC9
	01100	ADC10/AINCM
	01101	Reserved
	01110	AGND
	01111	Reserved
	10000	IOGND
	Others	Reserved

Table 31. ADCSTA MMR Bit Designations
(Address = 0xFFFF050C, Default Value = 0x00)

Bit	Value	Description
0	1	Indicates that an ADC conversion is complete. It is set automatically when an ADC conversion completes.
0	0	Automatically cleared by reading the ADCDAT MMR.

Table 32. ADCDAT MMR Bit Designations
(Address = 0xFFFF0510, Default Value = 0x00000000)

Bit	Value	Description
27:16		Holds the ADC result (see Figure 15).

Table 33. ADCRST MMR Bit Designations
(Address = 0xFFFF0514, Default Value = 0x00)

Bit	Value	Description
0	1	Set to 1 by the user to reset all the ADC registers to their default values.

Table 34. PGA_GN MMR Bit Designations
(Address = 0xFFFF0520, Default Value = 0x0000)

Bit	Value	Description
15:12		Reserved. Set to 0.
11:6		Gain of PGA for PADC0 (PGA_PADC0_GN) = 1 + 4 × (PGA_ADC0_GN/32)
5:0		Gain of PGA for PADC1 (PGA_PADC1_GN) = 1 + 4 × (PGA_ADC1_GN/32)

Note that PGA_PADC0_GN and PGA_PADC1_GN must be ≤32.

CONVERTER OPERATION

The ADC incorporates a successive approximation (SAR) architecture involving a charge-sampled input stage. This architecture is described for the three different modes of operation: differential mode, pseudo differential mode, and single-ended mode.

Differential Mode

The ADuC7122 contains a successive approximation ADC based on two capacitive DACs. Figure 18 and Figure 19 show simplified schematics of the ADC in acquisition and conversion phase, respectively. The ADC comprises control logic, a SAR, and two capacitive DACs. In Figure 18 (the acquisition phase), SW3 is closed and SW1 and SW2 are in Position A. The comparator is held in a balanced condition, and the sampling capacitor arrays acquire the differential signal on the input.

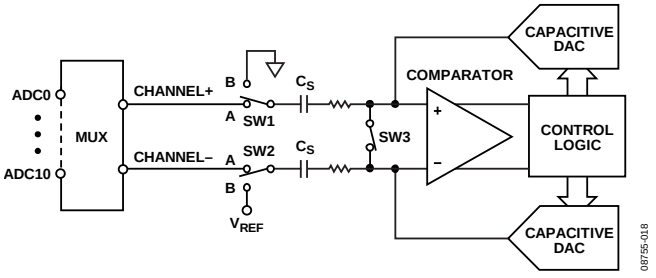


Figure 18. ADC Acquisition Phase

When the ADC starts a conversion (see Figure 19), SW3 opens and SW1 and SW2 move to Position B, causing the comparator to become unbalanced. Both inputs are disconnected when the conversion begins. The control logic and the charge redistribution DACs are used to add and subtract fixed amounts of charge from the sampling capacitor arrays to bring the comparator back into a balanced condition. When the comparator is rebalanced, the conversion is complete. The control logic generates the ADC output code. The output impedances of the sources driving the V_{IN+} and V_{IN-} inputs must be matched; otherwise, the two inputs have different settling times, resulting in errors. The input channel configuration for differential mode is set using the ADCCP and ADCCN registers.

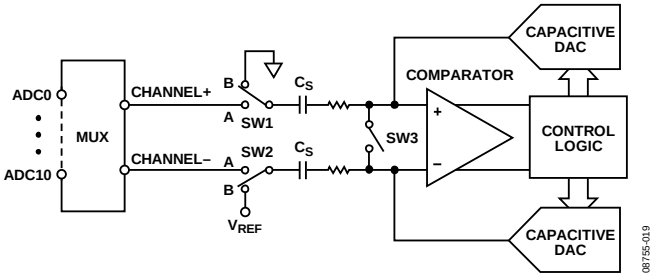


Figure 19. ADC Conversion Phase

Pseudo Differential Mode

In pseudo differential mode, Channel– is linked to the V_{IN-} input of the ADuC7122, and SW2 switches between A (Channel–) and B (V_{REF}). The V_{IN-} input must be connected to ground or a low voltage. The input signal on V_{IN+} can then vary from V_{IN-} to $V_{REF} + V_{IN-}$. Note that V_{IN-} must be selected so that $V_{REF} + V_{IN-}$ does not exceed AV_{DD}. In pseudo differential mode, only AINCM or PADCxN should be enabled for the V_{IN-} channel. The ADCCN register is used to set Channel– to AINCM or PADCxN, and the Channel+ can be selected using the ADCCP register.

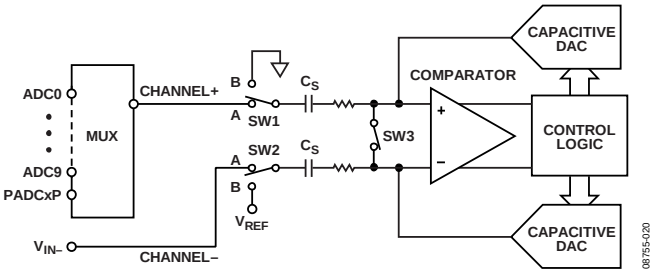


Figure 20. ADC in Pseudo Differential Mode

Single-Ended Mode

In single-ended mode, SW2 is always connected internally to ground. The V_{IN-} input can be floating. The input signal range on V_{IN+} is 0 V to V_{REF}. The ADuC7122 has 11 fixed gain ADC channels and two programmable gain ADC channels, which are enabled using the ADCCP register.

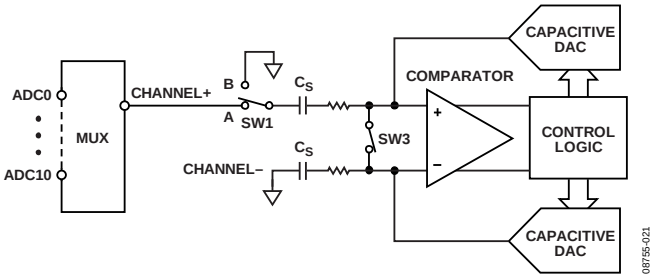


Figure 21. ADC in Single-Ended Mode

Analog Input Structure

Figure 22 shows the equivalent circuit of the analog input structure of the ADC. The four diodes provide ESD protection for the analog inputs. Care must be taken to ensure that the analog input signals never exceed the supply rails by more than 300 mV. Voltage in excess of 300 mV can cause these diodes to become forward biased and start conducting into the substrate. These diodes can conduct up to 10 mA without causing irreversible damage to the part.

The C1 capacitors in Figure 22 are typically 4 pF and can be primarily attributed to pin capacitance. The resistors are lumped components made up of the on resistance of the switches. The value of these resistors is typically about 100 Ω. The C2 capacitors are the ADC sampling capacitors and have a capacitance of 16 pF typical.

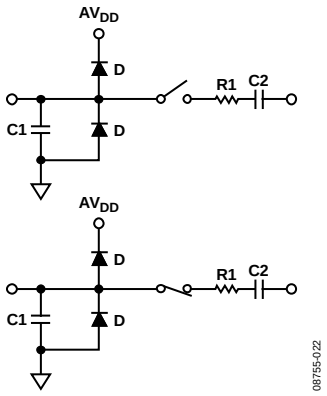


Figure 22. Equivalent Analog Input Circuit Conversion Phase: Switches Open, Track Phase: Switches Closed

For ac applications, removing high frequency components from the analog input signal is recommended through the use of an RC low-pass filter on the relevant analog input pins. In applications where harmonic distortion and signal-to-noise ratio are critical, the analog input should be driven from a low impedance source. Large source impedances significantly affect the ac performance of the ADC and can necessitate the use of an input buffer amplifier. The choice of the op amp is a function of the particular application. Figure 23 and Figure 24 give an example of an ADC front end.

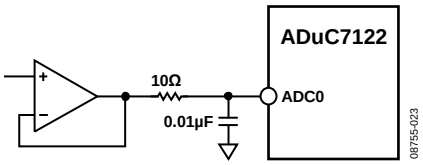


Figure 23. Buffering Single-Ended/Pseudo Differential Input

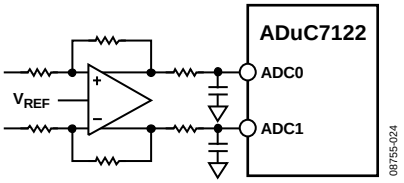


Figure 24. Buffering Differential Inputs

When no amplifier is used to drive the analog input, the source impedance should be limited to values lower than 1 kΩ. The maximum source impedance depends on the amount of total harmonic distortion (THD) that can be tolerated. The THD increases as the source impedance increases and the performance degrades.

DRIVING THE ANALOG INPUTS

Internal or external reference can be used for the ADC. In differential mode of operation, there are restrictions on the common-mode input signal (V_{CM}) that are dependent on reference value and supply voltage used to ensure that the signal remains within the supply rails.

Table 35 gives some calculated V_{CM} minimum and V_{CM} maximum values under various AV_{DD} and V_{REF} conditions.

Table 35. V_{CM} Ranges

AV_{DD}	V_{REF}	V_{CM} Min	V_{CM} Max	Peak-to-Peak Signal
3.3 V	2.5 V	1.25 V	2.05 V	2.5 V
	2.048 V	1.024 V	2.276 V	2.048 V
	1.25 V	0.75 V	2.55 V	1.25 V
3.0 V	2.5 V	1.25 V	1.75 V	2.5 V
	2.048 V	1.024 V	1.976 V	2.048 V
	1.25 V	0.75 V	2.25 V	1.25 V

BAND GAP REFERENCE

The ADuC7122 provides an on-chip band gap reference of 2.5 V that can be used for the ADC and for the DAC. This 2.5 V reference is generated from a 1.2 V reference.

This internal reference also appears on the V_{REF}_1.2 and V_{REF}_2.5 pins. When using the internal reference, a 470 nF capacitor must be connected between V_{REF}_1.2 and AGND and a 470 nF capacitor between V_{REF}_2.5 pin and AGND to ensure stability and fast response during ADC conversions.

The band gap reference also connects through buffers to the BUF_VREF1 and the BUF_VREF2 pins, which can be used as a reference for other circuits in the system. A minimum of 0.1 µF capacitor should be connected to these pins to damp noise.

The band gap reference interface consists of an 8-bit REFCON MMR, described in Table 36. It is recommended to enable REFCON Bit 0 and Bit 1 when performing an ADC or DAC conversion that uses the internal reference.

An external reference can be used for an ADC conversion. To perform an ADC conversion with an external 2.5 V reference, clear REFCON[1] and apply the external reference to the VREF_2.5 pin. To apply an external 1.2 V reference, clear REFCON[0] and apply the external reference to the V_{REF}_1.2 pin. Note that when applying an external reference to the V_{REF}_1.2 pin, this internally influences the 2.5 V reference as the 2.5 V reference is generated from the 1.2 V reference.

POWER SUPPLY MONITOR

The power supply monitor on the ADuC7122 indicates when the IOV_{DD} supply pin drops below one of two supply trip points. The monitor function is controlled via the PSMCON register. If enabled in the IRQEN or FIQEN register, the monitor interrupts the core using the PSMI bit in the PSMCON MMR. This bit is cleared immediately when CMP goes high. Note that if the interrupt generated is exited before CMP goes high (IOV_{DD} is above the trip point), no further interrupts are generated until CMP returns high. The user should ensure that code execution remains within the ISR until CMP returns high.

This monitor function allows the user to save working registers to avoid possible data loss due to the low supply or brownout conditions. It also ensures that normal code execution does not resume until a safe supply level has been established.

When the ADC channel selection bits are configured to IOVDD_MON (ADCCP[4:0] = 10011), this permits the ADC to convert the voltage available at the input of the power supply monitor comparator. When measuring an internal channel, the internal buffer must be enabled. The internal buffer should be enabled to isolate from external interference when sampling any of the internal channels. Before measuring this voltage, the following sequence is required:

1. Measure VREF using the ADC.
2. Set ADCCP = IOVDD_MON channel.
3. Set a typical delay of 60 µs.
4. Perform ADC conversion on the IOVDD_MON channel (use an ADCCON value of 0x2AA3 for optimum results).

The delay between the ADC mux select switching and the initiation of the conversion is required to allow the voltage on the ADC sampling capacitor to settle to the divided down supply voltage.

Table 36. REFCON MMR Bit Designations (Address = 0xFFFF0480, Default Value = 0x01)

Bit	Description
7:1	Reserved.
2	Reserved. Always set to 1. This bit outputs the buffered version of the internal 2.5 V reference onto BUF_VREF1 and BUF_VREF2. To disable this buffer, the user must disable the internal reference by clearing REFCON = 0x00.
1	Internal 2.5 V reference output enable. Set by the user to connect the internal 2.5 V reference to the V_{REF}_2.5 pin. Cleared by the user to disconnect the reference from the V_{REF}_2.5 pin. This pin should also be cleared to connect an external reference source to the V_{REF}_2.5 pin.
0	Internal 1.2 V reference output enable. Set by the user to connect the internal 1.2 V reference to the V_{REF}_1.2 pin. Cleared by the user to disconnect the reference from the V_{REF}_1.2 pin.

Table 37. PSMCON MMR Bit Designations (Address = 0xFFFF0440, Default Value = 0x08 or 0x00 (Dependent on Device Supply Level)

Bit	Name	Description
7:4	Reserved	Reserved bits. Clear to 0.
3	CMP	Comparator bit. This is a read-only bit that directly reflects the state of the comparator. Read 1 indicates the IOV_{DD} supply is above its selected trip point or the PSM is in power-down mode. Read 0 indicates the IOV_{DD} supply is below its selected trip point. This bit should be set before leaving the interrupt service routine.
2	TP	Trip point selection bit. 0 = 2.79 V 1 = 3.07 V
1	PSMEN	Power supply monitor enable bit. Set to 1 by the user to enable the power supply monitor circuit. Cleared to 0 by the user to disable the power supply monitor circuit.
0	PSMI	Power supply monitor interrupt bit. This bit is set high by the ADuC7122 if CMP is low, indicating low I/O supply. The PSMI bit can be used to interrupt the processor. When CMP returns high, the PSMI bit can be cleared by writing a 1 to this location. A write of 0 has no effect. There is no timeout delay. PSMI can be cleared immediately when CMP goes high.

NONVOLATILE FLASH/EE MEMORY

FLASH/EE MEMORY OVERVIEW

The ADuC7122 incorporates 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 is performed in page blocks. As a result, 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 no volatility, in-circuit programmability, high density, and low cost. Incorporated in the ADuC7122, Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one-time programmable (OTP) devices at remote operating nodes.

FLASH/EE MEMORY

The ADuC7122 contains two 64 kB arrays of Flash/EE memory. In the upper block of Flash/EE memory, the bottom 62 kB are available to the user and the top 2 kB of this Flash/EE program memory array contain permanently embedded firmware, allowing in-circuit serial download. The 2 kB of embedded firmware also contain a power-on configuration routine that downloads factory-calibrated coefficients to the various calibrated peripherals (band gap references and so on). This 2 kB embedded firmware is hidden from user code. It is not possible for the user to read, write, or erase this page. In the second block, all 64 kB of Flash/EE memory are available to the user.

The 126 kB of Flash/EE memory can be programmed in-circuit, using the serial download mode or the JTAG mode provided.

Flash/EE Memory Reliability

The Flash/EE memory arrays on the ADuC7122 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:

1. Initial page erase sequence
2. Read/verify a single Flash/EE sequence
3. Byte program memory sequence
4. Second read/verify endurance cycle sequence

In reliability qualification, three separate page blocks from each Flash/EE memory block is tested. An entire Flash/EE page at the top, middle, and bottom of each Flash/EE memory block is cycled 10,000 times from 0x0000 to 0xFFFF.

As indicated in the General Description section, the Flash/EE memory endurance qualification is carried out in accordance with JEDEC Retention Lifetime Specification A117 over the industrial temperature range of –10° to +95°C. The results allow the specification of a minimum endurance figure over a varying supply across the industrial temperature range for 10,000 cycles.

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. The parts are 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. Note, too, that retention lifetime, based on activation energy of 0.6 eV, derates with T_J, as shown in Figure 25.

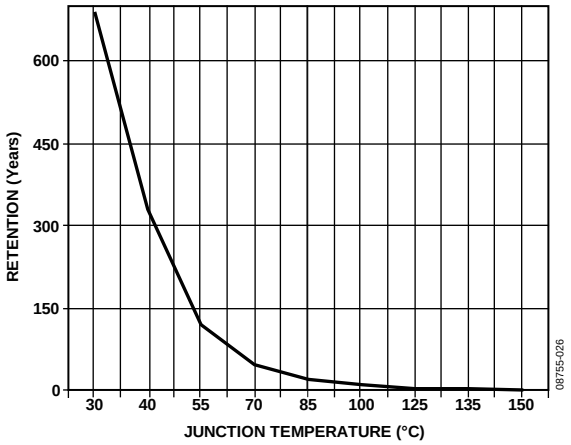


Figure 25. Flash/EE Memory Data Retention

Serial Downloading (In-Circuit Programming)

The ADuC7122 facilitates code download via the I²C serial port. The ADuC7122 enters serial download mode after a reset or power cycle if the $\overline{BM}$ pin is pulled low through an external 1 kΩ resistor. This is combined with the state of Address 0x00014 in Flash. If this address is 0xFFFFFFFF and the $\overline{BM}$ pin is pulled low, the part enters download mode; if this address contains any other value, user code is executed. When in serial download mode, the user can download code to the full 126 kB of Flash/EE memory while the device is in-circuit in its target application hardware. A PC executable serial download and hardware dongle are provided as part of the development system for serial downloads via the I²C port.

JTAG Access

The JTAG protocol uses the on-chip JTAG interface to facilitate code download and debug.

FLASH/EE MEMORY SECURITY

The 126 kB of Flash/EE memory available to the user can be read and write protected. Bit 31 of the FEE0PRO/FEE0HID MMR protects the 126 kB from being read through JTAG or the serial downloader. The other 31 bits of this register protect writing to the Flash/EE memory; each bit protects four pages, that is, 2 kB. Write protection is activated for all access types. FEE1PRO and FEE1HID similarly protect the second 64 kB block. All 32 bits of this are used to protect four pages at a time.

Three Levels of Protection

Protection can be set and removed by writing directly into FEExHID MMR. This protection does not remain after reset.

Protection can be set by writing into FEExPRO MMR. It takes effect only after a save protection command (0x0C) and a reset. The FEExPRO MMR is protected by a key to avoid direct access. The key is saved once and must be entered again to modify FEExPRO. A mass erase sets the key back to 0xFFFF but also erases all the user code.

The Flash/EE memory can be permanently protected by using the FEExPRO MMR and a particular value of the 0xDEADDEAD key. Entering the key again to modify the FEExPRO register is not allowed.

Sequence to Write the Key

1. Write the bit in FEExPRO corresponding to the page to be protected.
2. Enable key protection by setting Bit 6 of FEExMOD (Bit 5 must equal 0).
3. Write a 32-bit key in FEExADR, FEExDAT.
4. Run the write key command 0×0C in FEExCON; wait for the read to be successful by monitoring FEExSTA.
5. Reset the part.

To remove or modify the protection, the same sequence is used with a modified value of FEExPRO. If the key chosen is the value 0xDEADDEAD, then the memory protection cannot be removed. Only a mass erase unprotects the part; however, it also erases all user code.

The sequence to write the key is shown in the following example; this protects writing Page 4 to Page 7 of the Flash/EE memory:

```
FEE0PRO=0xFFFFFFFD;    //Protect pages 4 to 7
FEE0MOD=0x48;          //Write key enable
FEE0ADR=0x1234;        //16 bit key value
FEE0DAT=0x5678;        //16 bit key value
FEE0CON= 0x0C;         // Write key command
```

The same sequence should be followed to protect the part permanently with FEExADR = 0xDEAD and FEExDAT = 0xDEADDEAD.

FLASH/EE CONTROL INTERFACE

Table 38. FEE0DAT Register

Name	Address	Default Value	Access
FEE0DAT	0xFFFF0E0C	0xXXXX	R/W

FEE0DAT is a 16-bit data register.

Table 39. FEE0ADR Register

Name	Address	Default Value	Access
FEE0ADR	0xFFFF0E10	0x0000	R/W

FEE0ADR is a 16-bit address register.

Table 40. FEE0SGN Register

Name	Address	Default Value	Access
FEE0SGN	0xFFFF0E18	0xFFFFFF	R

FEE0SGN is a 24-bit code signature.

Table 41. FEE0PRO Register

Name	Address	Default Value	Access
FEE0PRO	0xFFFF0E1C	0x00000000	R/W

FEE0PRO provides protection following a subsequent reset. It requires a software key (see Table 57). As stated previously, each bit from 30 to 0 of the FEExPRO register protects a 2 kB block of memory; that is, setting Bit 0 low protects Page 0 to Page 3, and setting Bit 2 low protects Page 8 to Page 11.

Table 42. FEE0HID Register

Name	Address	Default Value	Access
FEE0HID	0xFFFF0E20	0xFFFFFFFF	R/W

FEE0HID provides immediate protection. It does not require any software keys (see Table 57).

Command Sequence for Executing a Mass Erase

```
FEE0DAT = 0x3CFF;
FEE0ADR = 0xFFC3;
FEE0MOD = FEE0MOD|0x8;          //Erase key enable
FEE0CON = 0x06;                 //Mass erase command
```

Table 43. FEE1DAT Register

Name	Address	Default Value	Access
FEE1DAT	0xFFFF0E8C	0xXXXX	R/W

FEE1DAT is a 16-bit data register.

Table 44. FEE1ADR Register

Name	Address	Default Value	Access
FEE1ADR	0xFFFF0E90	0x0000	R/W

FEE1ADR is a 16-bit address register.

Table 45. FEE1SGN Register

Name	Address	Default Value	Access
FEE1SGN	0xFFFF0E98	0xFFFFFF	R

FEE1SGN is a 24-bit code signature.

Table 46. FEE1PRO Register

Name	Address	Default Value	Access
FEE1PRO	0xFFFF0E9C	0x00000000	R/W

FEE1PRO provides protection following a subsequent reset. It requires a software key (see Table 58).

Table 47. FEE1HID Register

Name	Address	Default Value	Access
FEE1HID	0xFFFF0EA0	0xFFFFFFFF	R/W

FEE1HID provides immediate protection MMR. It does not require any software keys (see Table 58).

Table 48. FEE0STA Register

Name	Address	Default Value	Access
FEE0STA	0xFFFF0E00	0x0000	R

Table 49. FEE1STA Register

Name	Address	Default Value	Access
FEE1STA	0xFFFF0E80	0x0000	R

Table 50. FEE0MOD Register

Name	Address	Default Value	Access
FEE0MOD	0xFFFF0E04	0x80	R/W

Table 51. FEE1MOD Register

Name	Address	Default Value	Access
FEE1MOD	0xFFFF0E84	0x80	R/W

Table 52. FEE0CON Register

Name	Address	Default Value	Access
FEE0CON	0xFFFF0E08	0x0000	R/W

Table 53. FEE1CON Register

Name	Address	Default Value	Access
FEE1CON	0xFFFF0E88	0x0000	R/W

Table 54. FEExSTA MMR Bit Designations

Bit	Description
15:6	Reserved.
5	Reserved.
4	Reserved.
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 when reading the FEExSTA register.
2	Flash/EE controller busy. Set automatically when the controller is busy. Cleared automatically when the controller is not busy.
1	Command fail. Set automatically when a command completes unsuccessfully. Cleared automatically when reading the FEExSTA register.
0	Command complete. Set by ADuC7122 when a command is complete. Cleared automatically when reading the FEExSTA register.

Table 55. FEExMOD MMR Bit Designations

Bit	Description
7:5	Reserved. Always set these bits to 0 except when writing Flash memory control keys.
4	Flash/EE interrupt enable. Set by the user to enable the Flash/EE interrupt. The interrupt occurs when a command is complete. Cleared by user to disable the Flash/EE interrupt.
3	Erase/write command protection. Set by the user to enable the erase and write commands. Cleared to protect the Flash/EE memory against erase/write command.
2	Reserved. Should always be set to 0 by the user.
1:0	Flash/EE wait states. Both Flash/EE blocks must have the same wait state value for any change to take effect.

Table 56. Command Codes in FEExCON

Code	Command	Description
0x00[1]	Null	Idle state.
0x01[1]	Single read	Load FEExDAT with the 16-bit data indexed by FEExADR.
0x02[1]	Single write	Write FEExDAT at the address pointed by FEExADR. This operation takes 50 µs.
0x03[1]	Erase/write	Erase the page indexed by FEExADR and write FEExDAT at the location pointed by FEExADR. This operation takes 20 ms.
0x04[1]	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[1]	Single erase	Erase the page indexed by FEExADR.
0x06[1]	Mass erase	Erase user space. The 2 kB of kernel are protected in Block 0. This operation takes 2.48 sec. To prevent accidental execution, a command sequence is required to execute this instruction.
0x07	Reserved	Reserved.
0x08	Reserved	Reserved.
0x09	Reserved	Reserved.
0x0A	Reserved	Reserved.
0x0B	Signature	Gives a signature of the 64 kB of Flash/EE in the 24-bit FEExSIGN MMR. This operation takes 32,778 clock cycles.
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.
0x0E	Reserved	Reserved.
0x0F	Ping	No operation, interrupt generated.

[1] The FEExCON register always reads 0x07 immediately after execution of any of these commands.

Table 57. FEE0PRO and FEE0HID MMR Bit Designations

Bit	Description
31	Read protection. Cleared by the user to protect Block 0. Set by the user to allow reading of Block 0.
30:0	Write protection for Page 123 to Page 120, for Page 119 to Page 116, and for Page 3 to Page 0. Cleared by the user to protect the pages in writing. Set by the user to allow writing the pages.

Table 58. FEE1PRO and FEE1HID MMR Bit Designations

Bit	Description
31	Read protection. Cleared by the user to protect Block 1. Set by the user to allow reading of Block 1.
30	Write protection for Page 127 to Page 120. Cleared by the user to protect the pages in writing. Set by the user to allow writing the pages.
29:0	Write protection for Page 119 to Page 116 and for Page 3 to Page 0. Cleared by the user to protect the pages in writing. Set by the user to allow writing the pages.

EXECUTION TIME FROM SRAM AND FLASH/EE

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

Execution from SRAM

Fetching instructions from SRAM takes one clock cycle because the access time of the SRAM is 2 ns and a clock cycle is 22 ns 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 cycles if the data is in Flash/EE), one cycle to execute the instruction and two cycles to obtain the 32-bit data from Flash/EE. A control flow instruction, such as a branch instruction, takes one cycle to fetch, but it also takes two cycles to fill the pipeline with the new instructions.

Execution from Flash/EE

Because the Flash/EE width is 16 bits and access time for 16-bit words is 23 ns, execution from Flash/EE cannot be completed in one cycle (contrary to a SRAM fetch, which can be completed in a single cycle when CD bits = 0). Dependent on the instruction, some dead times may be required before accessing data for any value of CD bits.

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

Timing is identical in both modes when executing instructions that involve using 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 registers does not require any extra clock cycles, 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 obtain the 32-bit data from Flash/EE. An extra cycle must also be added before fetching another instruction. Data transfer instructions are more complex and are summarized in Table 59.

Table 59. Execution Cycles in ARM/Thumb Mode

Instructions	Fetch Cycles	Dead Time	Data Access	Dead Time
LD	2/1	1	2	1
LDH	2/1	1	1	1
LDM/PUSH	2/1	N	2 × N	N
STR	2/1	1	2 × 20 µs	1
STRH	2/1	1	20 µs	1
STRM/POP	2/1	N	2 × N × 20 µs	N

With 1 < N ≤ 16, N is the number of bytes of data to load or store in the multiple load/store instruction. The SWAP instruction combines an LD and STR instruction with only one fetch, giving a total of eight cycles plus 40 µs.

RESET AND REMAP

The ARM exception vectors are all situated at the bottom of the memory array, from Address 0x00000000 to Address 0x00000020, as shown in Figure 26.

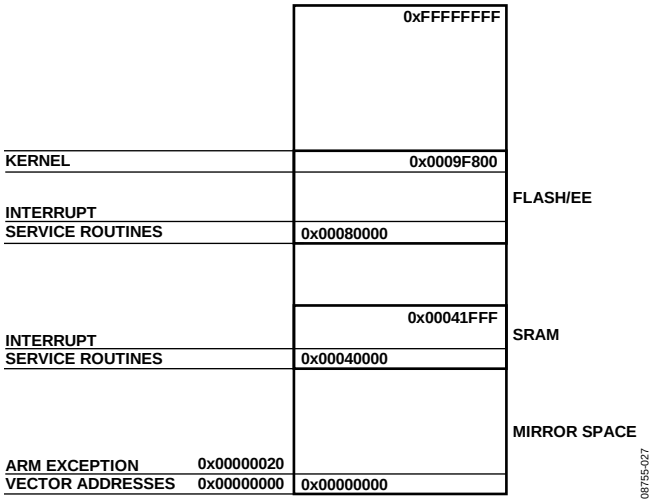


Figure 26. Remap for Exception Execution

By default and after any reset, the Flash/EE is mirrored at the bottom of the memory array. The remap function allows the programmer to mirror the SRAM at the bottom of the memory array, facilitating execution of exception routines from SRAM instead of from Flash/EE. This means exceptions are executed twice as fast, with the exception being executed in ARM mode (32 bits), and the SRAM being 32 bits wide instead of a 16-bit wide Flash/EE memory.

Remap Operation

When a reset occurs on the ADuC7122, execution starts automatically in factory-programmed internal configuration code. This kernel is hidden and cannot be accessed by user code. If the ADuC7122 is in normal mode (the $\overline{BM}$ pin is high), it executes the power-on configuration routine of the kernel and then jumps to the reset vector Address 0x00000000 to execute the reset exception routine of the user. Because the Flash/EE is mirrored at the bottom of the memory array at reset, the reset interrupt routine must always be written in Flash/EE.

The memory remap from Flash/EE is configured by setting Bit 0 of the REMAP register. Precautions must be taken to execute this command from Flash/EE, above Address 0x00080020, and not from the bottom of the array because this is replaced by the SRAM.

This operation is reversible: the Flash/EE can be remapped at Address 0x00000000 by clearing Bit 0 of the REMAP MMR. Precaution must again be taken to execute the remap function from outside the mirrored area. Any kind of reset remaps the Flash/EE memory at the bottom of the array.

Reset Operation

There are four types of reset: external reset, power-on reset, watchdog expiration, and software force reset. The RSTSTA register indicates the source of the last reset and RSTCLR clears the RSTSTA register. These registers can be used during a reset exception service routine to identify the source of the reset. If RSTSTA is null, the reset was external. Note that when clearing RSTSTA, all bits that are currently 1 must be cleared. Otherwise, a reset event occurs.

The RSTCFG register allows different peripherals to retain their state after a watchdog or software reset.

Table 60. REMAP MMR Bit Designations (Address = 0xFFFF0220, Default Value = 0x00)

Bit	Name	Description
0	Remap	Remap bit. Set by the user to remap the SRAM to Address 0x00000000. Cleared automatically after reset to remap the Flash/EE memory to Address 0x00000000.

Table 61. RSTSTA MMR Bit Designations (Address = 0xFFFF0230, Default Value = 0x0X)

Bit	Description
7:3	Reserved.
2	Software reset. Set by the user to force a software reset. Cleared by setting the corresponding bit in RSTCLR.
1	Watchdog timeout. Set 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.

RSTCFG Register

Name: RSTCFG

Address: 0xFFFF024C

Default value: 0x00

Access: Read/write

Table 62. RSTCFG MMR Bit Designations

Bit	Description
7 to 3	Reserved. Always set to 0.
2	This bit is set to 1 to configure the DAC outputs to retain their state after a watchdog or software reset. This bit is cleared for the DAC pins and registers to return to their default state.
1	Reserved. Always set to 0.
0	This bit is set to 1 to configure the GPIO pins to retain their state after a watchdog or software reset. This bit is cleared for the GPIO pins and registers to return to their default state.

RSTKEY1 Register

Name: RSTKEY1

Address: 0xFFFF0248

Default Value: N/A

Access Write

RSTKEY2 Register

Name: RSTKEY2

Address: 0xFFFF0250

Default Value: N/A

Access: Write

Table 63. RSTCFG Write Sequence

Name	Code
RSTKEY1	0x76
RSTCFG	User value
RSTKEY2	0xB1

OTHER ANALOG PERIPHERALS

DAC

The ADuC7122 incorporates 12 buffered, 12-bit voltage output string DACs on chip. Each DAC has a rail-to-rail voltage output buffer capable of driving 5 kΩ/100 pF.

Each DAC has two selectable ranges: 0 V to V_{REF} (internal band gap 2.5 V reference) and 0 V to AV_{DD}. The maximum signal range is 0 V to AV_{DD}.

MMRs Interface

Each DAC is independently configurable through a control register and a data register. These two registers are identical for the 12 DACs. DACxCON and DACxDAT (see Table 64 to Table 67) are described in detail in this section.

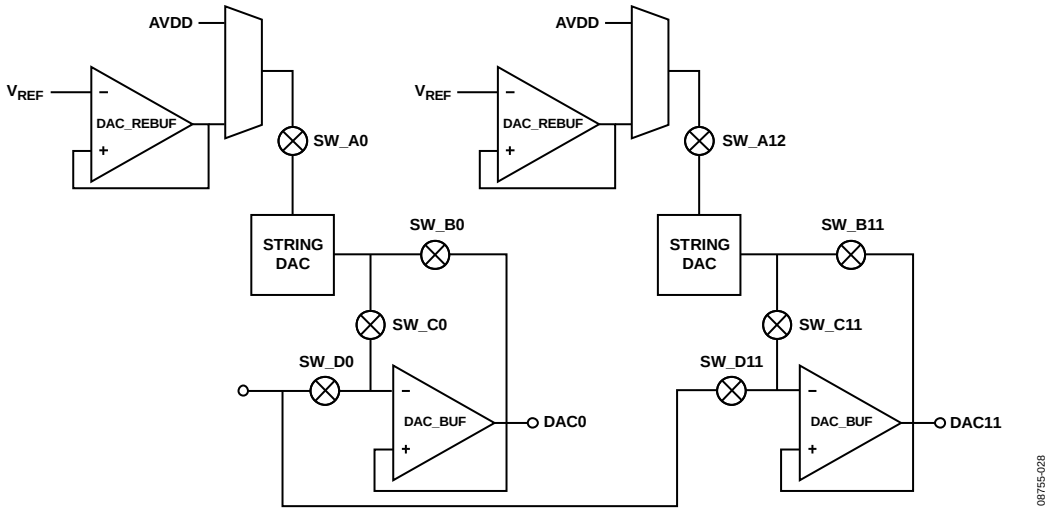


Figure 27. DAC Configuration

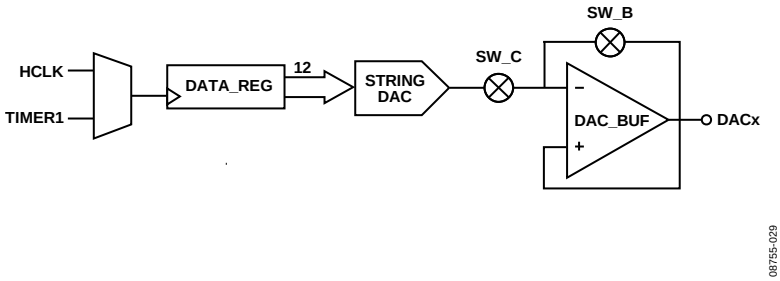


Figure 28. DAC User Functionality

Table 64. DACxCON Registers

Name	Address	Default Value	Access
DAC0CON	0xFFFF0580	0x100	R/W
DAC1CON	0xFFFF0588	0x100	R/W
DAC2CON	0xFFFF0590	0x100	R/W
DAC3CON	0xFFFF0598	0x100	R/W
DAC4CON	0xFFFF05A0	0x100	R/W
DAC5CON	0xFFFF05A8	0x100	R/W
DAC6CON	0xFFFF05B0	0x100	R/W
DAC7CON	0xFFFF05B8	0x100	R/W
DAC8CON	0xFFFF05C0	0x100	R/W
DAC9CON	0xFFFF05C8	0x100	R/W
DAC10CON	0xFFFF05D0	0x100	R/W
DAC11CON	0xFFFF05D8	0x100	R/W

Table 65. DACxCON MMR Bit Designations

Bit	Value	Name	Description
15:9	0		Reserved.
8	1	DACPD	DAC power-down. Set by user to set DACOUTx to tri-state mode.
7	0		Reserved.
6	0	BYP	DAC bypass bit. Set this bit to bypass the DAC buffer. Cleared to buffer the DAC output.
5	0	DACCLK	DAC update rate. Set by the user to update the DAC using Timer1. Cleared by the user to update the DAC using HCLK (core clock).
4	0	DACCLR	DAC clear bit. Set by the user to enable normal DAC operation. Cleared by the user to reset data register of the DAC to 0.
3	0		Reserved.
2	0		Reserved. Always clear to 0.
1:0		DACRNx	DAC range bits.
	00		V_{REF}/AGND.
	01		Reserved.
	10		Reserved.
	11		AV_{DD}/AGND.

Table 66. DACxDAT Registers

Name	Address	Default Value	Access
DAC0DAT	0xFFFF0584	0x00000000	R/W
DAC1DAT	0xFFFF058C	0x00000000	R/W
DAC2DAT	0xFFFF0594	0x00000000	R/W
DAC3DAT	0xFFFF059C	0x00000000	R/W
DAC4DAT	0xFFFF05A4	0x00000000	R/W
DAC5DAT	0xFFFF05AC	0x00000000	R/W
DAC6DAT	0xFFFF05B4	0x00000000	R/W
DAC7DAT	0xFFFF05BC	0x00000000	R/W
DAC8DAT	0xFFFF05C4	0x00000000	R/W
DAC9DAT	0xFFFF05CC	0x00000000	R/W
DAC10DAT	0xFFFF05D4	0x00000000	R/W
DAC11DAT	0xFFFF05DC	0x00000000	R/W

Table 67. DACxDAT MMR Bit Designations

Bit	Description
31:28	Reserved.
27:16	12-bit data for DACx.
15:12	Reserved.
11:0	Reserved.

Using the DACs

The on-chip DAC architecture consists of a resistor string DAC followed by an output buffer amplifier. The functional equivalent is shown in Figure 29.

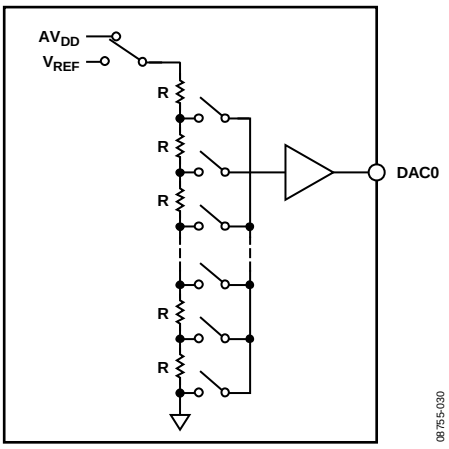


Figure 29. DAC Structure

As illustrated in Figure 29, the reference source for each DAC is user-selectable in software. It can be either AV_{DD} or V_{REF}. In 0 V-to-AV_{DD} mode, the DAC output transfer function spans from 0 V to the voltage at the AVDD pin. In 0 V-to-V_{REF} mode, the DAC output transfer function spans from 0 V to the internal 2.5 V reference, V_{REF}.

The DAC output buffer amplifier features a true, rail-to-rail output stage implementation. This means that when unloaded, each output is capable of swinging to within less than 5 mV of both AV_{DD} and ground. Moreover, the DAC linearity specification (when driving a 5 kΩ resistive load to ground) is guaranteed through the full transfer function except Code 0 to Code 100, and, in 0 V-to-AV_{DD} mode only, Code 3995 to Code 4095.

Linearity degradation near ground and AV_{DD} is caused by saturation of the output amplifier, and a general representation of its effects (neglecting offset and gain error) is illustrated in Figure 30. The dotted line in Figure 30 indicates the ideal transfer function, and the solid line represents what the transfer function may look like with endpoint nonlinearities due to saturation of the output amplifier. Note that Figure 30 represents a transfer function in 0-to-AV_{DD} mode only. In 0 V-to-V_{REF} mode (with V_{REF} < AV_{DD}), the lower nonlinearity is similar. However, the upper portion of the transfer function follows the ideal line right to the end (V_{REF} in this case, not AV_{DD}), showing no signs of endpoint linearity errors.

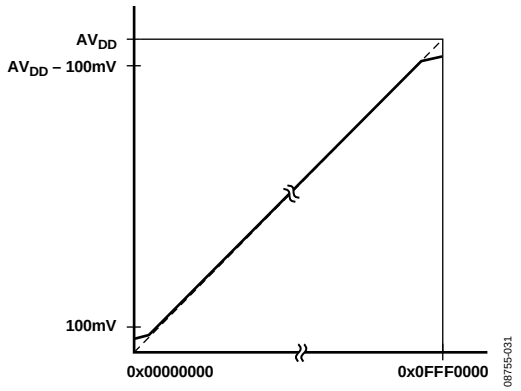


Figure 30. Endpoint Nonlinearities Due to Amplifier Saturation

The endpoint nonlinearities conceptually illustrated in Figure 30 become worse as a function of output loading. The ADuC7122 data sheet specifications assume a 5 kΩ resistive load to ground at the DAC output. As the output is forced to source or sink more current, the nonlinear regions at the top or bottom (respectively) of Figure 30 become larger. With larger current demands, this can significantly limit output voltage swing.

The DAC can be configured to retain its output voltage after a watchdog or software reset by writing to the RSTCFG register.

LDO (LOW DROPOUT REGULATOR)

The ADuC7122 contains an integrated LDO that generates the core supply voltage (LVDD) of approximately 2.6 V from the IOVDD supply. Because the LDO is driven from IOVDD, the IOVDD supply voltage needs to be greater than 2.7 V.

An external compensation capacitor (C_T) of 0.47 µF with low equivalent series resistance (ESR) must be placed very close to the LVDD pin. This capacitor also acts as a storage of charge and supplies an instantaneous charge required by the core, particularly at the positive edge of the core clock (HCLK).

The LVDD voltage generated by the LDO is solely for providing a supply for the ADuC7122. Therefore, users should not use the LVDD pin as the power supply pin for any other chip. Also, the IOVDD pin should have excellent power supply decoupling to help improve line regulation performance of the LDO.

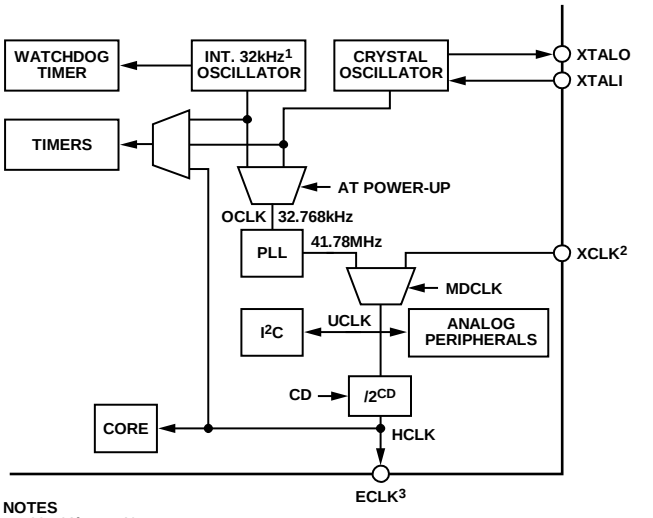
The LVDD pin has no reverse battery, current limit, or thermal shutdown protection; therefore, it is essential that users of the ADuC7122 do not short this pin to ground at anytime during normal operation or during board manufacture.

OSCILLATOR AND PLL—POWER CONTROL

The ADuC7122 integrates a 32.768 kHz oscillator, a clock divider, and a PLL. The PLL locks onto a multiple (1275) of the internal oscillator to provide a stable 41.78 MHz clock for the system. The core can operate at this frequency or at binary submultiples of it to allow power saving. The default core clock is the PLL clock divided by 8 (CD = 3) or 5.2 MHz. The core clock frequency can be output on the ECLK pin as described in Figure 31. Note that when the ECLK pin is used to output the core clock, the output signal is not buffered and is not suitable for use as a clock source to an external device without an external buffer.

A power-down mode is available on the ADuC7122.

The operating mode, clocking mode, and programmable clock divider are controlled via two MMRs, PLLCON (see Table 75) and POWCON (see Table 76). PLLCON controls the operating mode of the clock system, and POWCON controls the core clock frequency and the power-down mode.



NOTES
1. 32.768kHz ± 3%.
2. TO USE THE SECONDARY FUNCTION OF P1.4 AS XCLK, PLLCON BITS[1:0] MUST EQUAL 11.
3. WHEN THE SECONDARY FUNCTION FOR P1.4 IS SET TO 2 (THAT IS, GP1CON[17:16] = 10), THE ECLK FUNCTION IS SELECTED BY DEFAULT.

Figure 31. Clocking System

EXTERNAL CRYSTAL SELECTION

To switch to an external crystal, use the following procedure:

1. Enable the Timer2 interrupt and configure it for a timeout period of >120 μs.
2. Follow the write sequence to the PLLCON register, setting the MDCLK bits to 01 and clearing the OSEL bit.
3. Force the part into nap mode by following the correct write sequence to the POWCON register.
4. When the part is interrupted from nap mode by the Timer2 interrupt source, the clock source has switched to the external clock.

Example Source Code

```
  T2LD = 5;
  T2CON = 0x480;


  while ((T2VAL == t2val_old) || (T2VAL >
3)) //ensures timer value loaded
  IRQEN = 0x10;
//enable T2 interrupt
  PLLKEY1 = 0xAA;
  PLLCON = 0x01;
  PLLKEY2 = 0x55;

  POWKEY1 = 0x01;
  POWCON = 0x27;
// Set Core into Nap mode
  POWKEY2 = 0xF4;
```

In noisy environments, noise can couple to the external crystal pins, and PLL may lose lock momentarily. A PLL interrupt is provided in the interrupt controller. The core clock is immediately halted, and this interrupt is serviced only when the lock is restored.

In case of crystal loss, the watchdog timer should be used. During initialization, a test on the RSTSTA can determine if the reset came from the watchdog timer.

EXTERNAL CLOCK SELECTION

To switch to an external clock on P1.4, configure P1.4 in Mode 2. The external clock can be up to 41.78 MHz, providing the tolerance is 1%.

Example Source Code

```
  T2LD = 5;
  TCON = 0x480;
  while ((T2VAL == t2val_old) || (T2VAL >
3)) //ensures timer value loaded
  IRQEN = 0x10;
//enable T2 interrupt
  PLLKEY1 = 0xAA;
  PLLCON = 0x03; //Select external clock
  PLLKEY2 = 0x55;

  POWKEY1 = 0x01;
  POWCON = 0x27; // Set Core into Nap mode
  POWKEY2 = 0xF4;
```

POWER CONTROL SYSTEM

A choice of operating modes is available on the ADuC7122. Table 68 describes which blocks of the ADuC7122 are powered on in the different modes and indicates the power-up time. Table 69 gives some typical values of the total current consumption (analog and digital supply currents) in the different modes, depending on the clock divider bits when the ADC is turned off. Note that these values also include current consumption of the regulator and other parts on the test board on which these values were measured.

Table 68. Operating Modes

Mode	Core	Peripherals	PLL	XTAL/Timer2/Timer3	XIRQ	Start-Up/Power-On Time
Active	On	On	On	On	On	130 ms at CD = 0
Pause		On	On	On	On	24 ns at CD = 0; 3.06 µs at CD = 7
Nap			On	On	On	24 ns at CD = 0; 3.06 µs at CD = 7
Sleep				On	On	1.58 ms
Stop					On	1.7 ms

Table 69. Typical Current Consumption at 25°C

PC[2:0]	Mode	CD = 0	CD = 1	CD = 2	CD = 3	CD = 4	CD = 5	CD = 6	CD = 7
000	Active	30	21.2	13.8	11	8.1	7.2	6.7	6.45
001	Pause	22.7	13.3	8.5	6.1	4.9	4.3	4	3.85
010	Nap	3.8	3.8	3.8	3.8	3.8	3.8	3.8	3.8
011	Sleep	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
100	Stop	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25

MMRS AND KEYS

To prevent accidental programming, a certain sequence must be followed when writing in the PLLCON and POWCON registers (see Table 74).

Table 70. PLLKEYx Register

Name	Address	Default Value	Access
PLLKEY1	0xFFFF0410	0x0000	W
PLLKEY2	0xFFFF0418	0x0000	W

Table 71. PLLCON Register

Name	Address	Default Value	Access
PLLCON	0xFFFF0414	0x21	R/W

Table 72. POWKEYx Register

Name	Address	Default Value	Access
POWKEY1	0xFFFF0404	0x0000	W
POWKEY2	0xFFFF040C	0x0000	W

Table 73. POWCON Register

Name	Address	Default Value	Access
POWCON	0xFFFF0408	0x03	R/W

Table 74. PLLCON and POWCON Write Sequence

PLLCON	POWCON
PLLKEY1 = 0xAA	POWKEY1 = 0x01
PLLCON = 0x01	POWCON = user value
PLLKEY2 = 0x55	POWKEY2 = 0xF4

Table 75. PLLCON MMR Bit Designations

Bit	Value	Name	Description
7:6			Reserved.
5		OSEL	32 kHz PLL input selection.
	1		Set by the user to use the internal 32 kHz oscillator. Set by default.
	0		Cleared by the user to use the external 32 kHz crystal.
4:2			Reserved.
1:0		MDCLK	Clocking modes.
	00		Reserved.
	01		PLL. default configuration.
	10		Reserved.
	11		External clock on P1.4 pin.

Table 76. POWCON MMR Bit Designations

Bit	Value	Name	Description
7			Reserved.
6:4		PC	Operating modes.
	000		Active mode.
	001		Pause mode.
	010		Nap.
	011		Sleep mode. IRQ0 to IRQ3 and Timer2 can wake up the ADuC7122.
	100		Stop mode.
	Others		Reserved.
3		RSVD	Reserved.
2:0		CD	CPU clock divider bits.
	000		41.779200 MHz.
	001		20.889600 MHz.
	010		10.444800 MHz.
	011		5.222400 MHz.
	100		2.611200 MHz.
	101		1.305600 MHz.
	110		654.800 kHz.
	111		326.400 kHz.

DIGITAL PERIPHERALS

PWM GENERAL OVERVIEW

The ADuC7122 integrates a 6-channel PWM interface. The PWM outputs can be configured to drive an H-bridge or can be used as standard PWM outputs. On power-up, the PWM outputs default to H-bridge mode. This ensures that the H-bridge controlled motor is turned off by default. In standard PWM mode, the outputs are arranged as three pairs of PWM pins. Users have control over the period of each pair of outputs and over the duty cycle of each individual output.

Table 77. PWM MMRs

Name	Function
PWMCON1	PWM control
PWM1COM1	Compare Register 1 for PWM Output 1 and Output 2
PWM1COM2	Compare Register 2 for PWM Output 1 and Output 2
PWM1COM3	Compare Register 3 for PWM Output 1 and Output 2
PWM1LEN	Frequency control for PWM Output 1 and Output 2
PWM2COM1	Compare Register 1 for PWM Output 3 and Output 4
PWM2COM2	Compare Register 2 for PWM Output 3 and Output 4
PWM2COM3	Compare Register 3 for PWM Output 3 and Output 4
PWM2LEN	Frequency control for PWM Output 3 and Output 4
PWM3COM1	Compare Register 1 for PWM Output 5 and Output 6
PWM3COM2	Compare Register 2 for PWM Output 5 and Output 6
PWM3COM3	Compare Register 3 for PWM Output 5 and Output 6
PWM3LEN	Frequency control for PWM Output 5 and Output 6
PWMCON2	PWM convert start control
PWMICLR	PWM interrupt clear

In all modes, the PWMxCOMx MMRs control the point at which the PWM outputs change state. An example of the first pair of PWM outputs (PWM1 and PWM2) is shown in Figure 32.

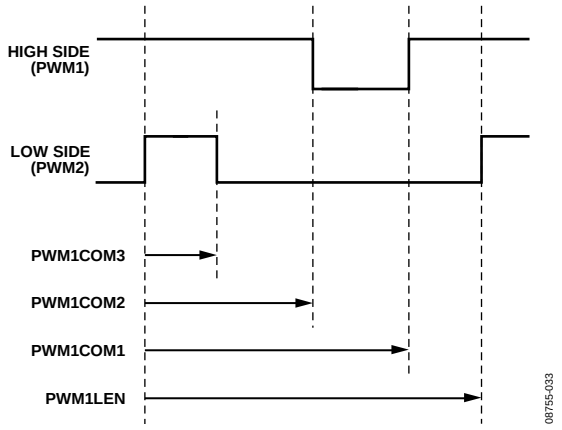


Figure 32. PWM Timing

The PWM clock is selectable via PWMCON1 with UCLK divided by one of the following values: 2, 4, 8, 16, 32, 64, 128, or 256. The length of a PWM period is defined by PWMxLEN.

The PWM waveforms are set by the count value of the 16-bit timer and the compare registers contents, as shown in the PWM1 and PWM2 waveforms in Figure 32.

The low-side waveform, PWM2, goes high when the timer count reaches PWM1LEN, and it goes low when the timer count reaches the value held in PWM1COM3 or when the high-side waveform PWM1 goes low.

The high-side waveform, PWM1, goes high when the timer count reaches the value held in PWM1COM1, and it goes low when the timer count reaches the value held in PWM1COM2.

In H-bridge mode, HMODE = 1 and Table 78 determine the PWM outputs.

Table 78. PWMCON1 MMR Bit Designations (Address = 0xFFFF0F80, Default Value = 0x0012)

Bit	Name	Description
15	Reserved	This bit is reserved.
14	SYNC	Enables PWM synchronization. Set to 1 by the user so that all PWM counters are reset on the next clock edge after the detection of a high-to-low transition on the SYNC pin. Cleared by the user to ignore transitions on the SYNC pin.
13	PWM6INV	Set to 1 by the user to invert PWM6. Cleared by the user to use PWM6 in normal mode.
12	PWM4NV	Set to 1 by the user to invert PWM4. Cleared by the user to use PWM4 in normal mode.
11	PWM2INV	Set to 1 by the user to invert PWM2. Cleared by the user to use PWM2 in normal mode.
10	PWMTRIP	Set to 1 by the user to enable PWM trip interrupt. When the PWMTRIP input is low, the PWMEN bit is cleared and an interrupt is generated. Cleared by the user to disable the PWMTRIP interrupt.
9	ENA	If HOFF = 0 and HMODE = 1. Set to 1 by the user to enable PWM outputs. Cleared by the user to disable PWM outputs. If HOFF = 1 and HMODE = 1, see Table 79. If not in H-Bridge mode, this bit has no effect.

Bit	Name	Description
8:6	PWMCP[2:0]	PWM clock prescaler bits. Sets the UCLK divider. 000 = UCLK/2. 001 = UCLK/4. 010 = UCLK/8. 011 = UCLK/16. 100 = UCLK/32. 101 = UCLK/64. 110 = UCLK/128. 111 = UCLK/256.
5	POINV	Set to 1 by the user to invert all PWM outputs. Cleared by the user to use PWM outputs as normal.
4	HOFF	High-side off. Set to 1 by the user to force PWM1 and PWM3 outputs high. This also forces PWM2 and PWM4 low. Cleared by the user to use the PWM outputs as normal.
3	LCOMP	Load compare registers. Set to 1 by the user to load the internal compare registers with the values in PWMxCOMx on the next transition of the PWM timer from 0x00 to 0x01. Cleared by the user to use the values previously stored in the internal compare registers.
2	DIR	Direction control. Set to 1 by the user to enable PWM1 and PWM2 as the output signals while PWM3 and PWM4 are held low. Cleared by the user to enable PWM3 and PWM4 as the output signals while PWM1 and PWM2 are held low.
1	HMODE	Enables H-bridge mode. Set to 1 by the user to enable H-Bridge mode and Bit 1 to Bit 5 of PWMCON1. Cleared by the user to operate the PWMs in standard mode.
0	PWMEN	Set to 1 by the user to enable all PWM outputs. Cleared by the user to disable all PWM outputs.

Table 79. PWM Output Selection[1]

PWMCON1 MMR				PWM Outputs			
ENA	HOFF	POINV	DIR	PWM1	PWM2	PWM3	PWM4
0	0	X	X	1	1	1	1
X	1	X	X	1	0	1	0
1	0	0	0	0	0	HS	LS
1	0	0	1	HS	LS	0	0
1	0	1	0	HS	LS	1	1
1	0	1	1	1	1	HS	LS

[1] X = don't care, HS = high side, LS = low side.

On power-up, PWMCON1 defaults to 0x12 (HOFF = 1 and HMODE = 1). All GPIO pins associated with the PWM are configured in PWM mode by default (see Table 80).

Table 80. Compare Registers

Name	Address	Default Value	Access
PWM1COM1	0xFFFF0F84	0x00	R/W
PWM1COM2	0xFFFF0F88	0x00	R/W
PWM1COM3	0xFFFF0F8C	0x00	R/W
PWM2COM1	0xFFFF0F94	0x00	R/W
PWM2COM2	0xFFFF0F98	0x00	R/W
PWM2COM3	0xFFFF0F9C	0x00	R/W
PWM3COM1	0xFFFF0FA4	0x00	R/W
PWM3COM2	0xFFFF0FA8	0x00	R/W
PWM3COM3	0xFFFF0FAC	0x00	R/W

The PWM trip interrupt can be cleared by writing any value to the PWMICLR MMR. Note that when using the PWM trip interrupt, users should make sure that the PWM interrupt has been cleared before exiting the ISR. This prevents generation of multiple interrupts.

PWM CONVERT START CONTROL

The PWM can be configured to generate an ADC convert start signal after the active low-side signal goes high. There is a programmable delay between when the low-side signal goes high and the convert start signal is generated.

This is controlled via the PWMCON2 MMR. If the delay selected is higher than the width of the PWM pulse, the interrupt remains low.

Table 81. PWMCON2 MMR Bit Designations (Address = 0xFFFF0FB4, Default Value = 0x00)

Bit	Value	Name	Description
7		CSEN	Convert start enable.
	1		Set to 1 by the user to enable the PWM to generate a convert start signal.
	0		Cleared by the user to disable the PWM convert start signal.
3:0		CSD3	Convert start delay. Delays the convert start signal by a number of clock pulses.
	0000		4 clock pulses.
	0001		8 clock pulses.
	0010		12 clock pulses.
	0011		16 clock pulses.
	0100		20 clock pulses.
	0101		24 clock pulses.
	0110		28 clock pulses.
	0111		32 clock pulses.
	1000		36 clock pulses.
	1001		40 clock pulses.
	1010		44 clock pulses.
	1011		48 clock pulses.
	1100		52 clock pulses.
	1101		56 clock pulses.
	1110		60 clock pulses.
	1111		64 clock pulses.

When calculating the time from the convert start delay to the start of an ADC conversion, the user needs to take account of internal delays. The example in Figure 33 shows the case for a delay of four clocks. One additional clock is required to pass the convert start signal to the ADC logic. When the ADC logic receives the convert start signal, an ADC conversion begins on the next ADC clock edge (see Figure 33).

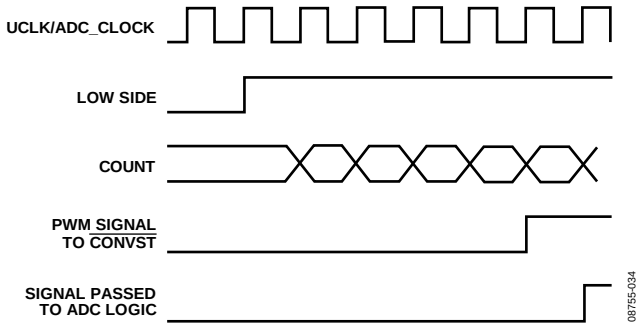


Figure 33. ADC Conversion

GENERAL-PURPOSE I/O

The ADuC7122 provides 32 general-purpose, bidirectional I/O (GPIO) pins. All I/O pins are 5 V tolerant, meaning that the GPIOs support an input voltage of 5 V. In general, many of the GPIO pins have multiple functions (see Table 84). By default, the GPIO pins are configured in GPIO mode.

All GPIO pins have an internal pull-up resistor (of about 100 kΩ), and their drive capability is 1.6 mA. Note that a maximum of 20 GPIOs can drive 1.6 mA at the same time. The 32 GPIOs are grouped in four ports: Port 0 to Port 3. Each port is controlled by four or five MMRs, with x representing the port number.

Table 82. GPxCON Register

Name	Address	Default Value	Access
GP0CON	0xFFFF0D00	0x00000000	R/W
GP1CON	0xFFFF0D04	0x00000000	R/W
GP2CON	0xFFFF0D08	0x00000000	R/W
GP3CON	0xFFFF0D0C	0x11111111	R/W

The input level of any GPIO can be read at any time in the GPxDAT MMR, even when the pin is configured in a mode other than GPIO. The PLA input is always active.

When the ADuC7122 parts enter a power-saving mode, the GPIO pins retain their state.

GPxCON is the Port x control register, and it selects the function of each pin of Port x, as described in Table 84.

Table 83. GPxCON MMR Bit Designations

Bit	Description
31:30	Reserved
29:28	Select function of Px.7 pin
27:26	Reserved
25:24	Select function of Px.6 pin
23:22	Reserved
21:20	Select function of Px.5 pin
19:18	Reserved
17:16	Select function of Px.4 pin
15:14	Reserved
13:12	Select function of Px.3 pin
11:10	Reserved
9:8	Select function of Px.2 pin
7:6	Reserved
5:4	Select function of Px.1 pin
3:2	Reserved
1:0	Select function of Px.0 pin

Table 84. GPIO Pin Function Designations[1]

		Configuration (see GPxCON)			
Port	Pin	00	01	10	11
0	P0.0	GPIO	SCL1	N/A	PLAI[5]
	P0.1	GPIO	SDA1	N/A	PLAI[4]
	P0.2	GPIO	SPICLK	ADC_{BUSY}	PLAO[13]
	P0.3	GPIO	SPIMISO	SYNC	PLAO[12]
	P0.4	GPIO	SPIMOSI	TRIP	PLAI[11]
	P0.5	GPIO	SPI$\overline{CS}$	$\overline{CONVST}$	PLAI[10]
	P0.6	GPIO	$\overline{MRST}$	N/A	PLAI[2]
	P0.7	GPIO	TRST	N/A	PLAI[3]
1[2]	P1.0	GPIO	SIN	SCL2	PLAI[7]
	P1.1	GPIO	SOUT	SDA2	PLAI[6]
	P1.4	GPIO	PWM1	ECLK/XCLK	PLAI[8]
	P1.5	GPIO	PWM2	N/A	PLAI[9]
	P1.6	GPIO	N/A	N/A	PLAO[5]
	P1.7	GPIO	N/A	N/A	PLAO[4]
2	P2.0	GPIO/IRQ0	N/A	N/A	PLAI[13]
	P2.1	GPIO/IRQ1	N/A	N/A	PLAI[12]
	P2.2	GPIO	N/A	N/A	PLAI[1]
	P2.3	GPIO/IRQ2	N/A	N/A	PLAI[14]
	P2.4	GPIO	PWM5	N/A	PLAO[7]
	P2.5	GPIO	PWM6	N/A	PLAO[6]
	P2.6	GPIO/IRQ3	N/A	N/A	PLAI[15]
	P2.7	GPIO	N/A	N/A	PLAI[0]
3	P3.0	GPIO	N/A	N/A	PLAO[0]
	P3.1	GPIO	N/A	N/A	PLAO[1]
	P3.2	GPIO/IRQ4	PWM3	N/A	PLAO[2]
	P3.3	GPIO/IRQ5	PWM4	N/A	PLAO[3]
	P3.4	GPIO	N/A	N/A	PLAO[8]
	P3.5	GPIO	N/A	N/A	PLAO[9]
	P3.6	GPIO	N/A	N/A	PLAO[10]
	P3.7/$\overline{BM}$	GPIO	N/A	N/A	PLAO[11]

[1] N/A means no secondary function exists.
[2] Never attempt a write to P1.2 or P1.3.

Table 85. GPxPAR Register

Name	Address	Default Value	Access
GP0PAR	0xFFFF0D2C	0x20000000	R/W
GP1PAR	0xFFFF0D3C	0x00000000	R/W
GP2PAR	0xFFFF0D4C	0x00000000	R/W
GP3PAR	0xFFFF0D5C	0x00222222	R/W

GPxPAR programs the parameters for Port 0, Port 1, Port 2, and Port 3. Note that the GPxDAT MMR must always be written after changing the GPxPAR MMR.

Table 86. GPxPAR MMR Bit Designations

Bit	Description
31:29	Reserved
28	Pull-up disable Px.7 pin
27:25	Reserved
24	Pull-up disable Px.6 pin
23:21	Reserved
20	Pull-up disable Px.5 pin
19:17	Reserved
16	Pull-up disable Px.4 pin
15:13	Reserved
12	Pull-up disable Px.3 pin
11:9	Reserved
8	Pull-up disable Px.2 pin
7:5	Reserved
4	Pull-up disable Px.1 pin
3:1	Reserved
0	Pull-up disable Px.0 pin

Table 87. GPxDAT Register

Name	Address	Default Value	Access
GP0DAT	0xFFFF0D20	0x000000XX	R/W
GP1DAT	0xFFFF0D30	0x000000XX	R/W
GP2DAT	0xFFFF0D40	0x000000XX	R/W
GP3DAT	0xFFFF0D50	0x000000XX	R/W

GPxDAT is a Port x configuration and data register. It configures the direction of the GPIO pins of Port x, sets the output value for the pins configured as outputs, and receives and stores the input value of the pins configured as inputs.

Table 88. GPxDAT MMR Bit Designations

Bit	Description
31:24	Direction of the data. Set to 1 by the user to configure the GPIO pin as an output. Cleared to 0 by user to configure the GPIO pin as an input.
23:16	Port x data output.
15:8	Reflect the state of Port x pins at reset (read only).
7:0	Port x data input (read only).

Table 89. GPxSET Register

Name	Address	Default Value	Access
GP0SET	0xFFFF0D24	0x000000XX	W
GP1SET	0xFFFF0D34	0x000000XX	W
GP2SET	0xFFFF0D44	0x000000XX	W
GP3SET	0xFFFF0D54	0x000000XX	W

Table 90. GPxSET MMR Bit Designations

Bit	Description
31: 24	Reserved.
23:16	Data Port x set bit. Set to 1 by the user to set bit on Port x; also sets the corresponding bit in the GPxDAT MMR. Cleared to 0 by the user; does not affect the data output.
15: 0	Reserved.

GPxSET is a data set Port x register.

Table 91. GPxCLR Register

Name	Address	Default Value	Access
GP0CLR	0xFFFF0D28	0x000000XX	W
GP1CLR	0xFFFF0D38	0x000000XX	W
GP2CLR	0xFFFF0D48	0x000000XX	W
GP3CLR	0xFFFF0D58	0x000000XX	W

GPxCLR is a data clear Port x register.

Table 92. GPxCLR MMR Bit Designations

Bit	Description
31:24	Reserved.
23:16	Data Port x clear bit. Set to 1 by the user to clear bit on Port x; also clears the corresponding bit in the GPxDAT MMR. Cleared to 0 by the user; does not affect the data output.
15:0	Reserved.

Open-collector functionality is available on the following GPIO pins: P1.7, P1.6, P2.x, and P3.x. Open-collector functionality can be configured using GP1OCE[7:6], GP2OCE[7:0], and GP3OCE[7:0].

Table 93. GPxOCE MMR Bit Designations

Bit	Description
31:8	Reserved.
7	GPIO Px.7 open-collector enable Set to 1 by the user to enable open-collector Set to 0 by the user to disable open collector
6	GPIO Px.6 open-collector enable Set to 1 by the user to enable open-collector Set to 0 by the user to disable open-collector
5	GPIO Px.5 open-collector enable Set to 1 by the user to enable open-collector Set to 0 by the user to disable open-collector
4	GPIO Px.4 open-collector enable Set to 1 by the user to enable open-collector Set to 0 by the user to disable open-collector
3	GPIO Px.3 open-collector enable Set to 1 by the user to enable open-collector Set to 0 by the user to disable open-collector
2	GPIO Px.2 open-collector enable Set to 1 by the user to enable open-collector Set to 0 by the user to disable open-collector
1	GPIO Px.1 open-collector enable Set to 1 by the user to enable open-collector Set to 0 by the user to disable open-collector
0	GPIO Px.0 open-collector enable Set to 1 by the user to enable open-collector Set to 0 by the user to disable open-collector

UART SERIAL INTERFACE

The ADuC7122 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 that facilitates high accuracy baud rate generation and a network addressable mode. The UART functionality is available on the P1.0 and P1.1 pins of the ADuC7122.

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

BAUD RATE GENERATION

The ADuC7122 features two methods of generating the UART baud rate: normal 450 UART baud rate generation and ADuC7122 fractional divider.

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

$$Baud\ rate = \frac{41.78\ \text{MHz}}{16 \times 2 \times DL} \quad (1)$$

Table 94 lists common baud rate values.

Table 94. Baud Rate Using the Standard Baud Rate Generator

Baud Rate	DL	Actual Baud Rate	% Error
9600	0x88	9600	0%
19,200	0x44	19,200	0%
115,200	0x0B	118,691	3%

ADuC7122 Fractional Divider

The fractional divider combined with the normal baud rate generator allows the generating of a wider range of more accurate baud rates.

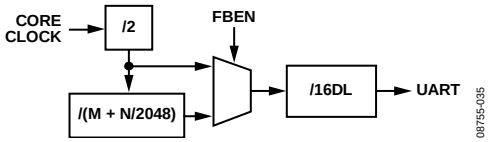


Figure 34. Baud Rate Generation Options

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

$$Baud\ Rate = \frac{41.78\ \text{MHz}}{16 \times DL \times 2 \times (M + \frac{N}{2048})} \quad (2)$$

$$M + \frac{N}{2048} = \frac{41.78\ \text{MHz}}{Baud\ Rate \times 16 \times DL \times 2}$$

For example, generation of 19,200 baud

$$M + \frac{N}{2048} = \frac{41.78\ \text{MHz}}{19{,}200 \times 16 \times 67 \times 2}$$

$$M + \frac{N}{2048} = 1.015$$

where:
$M = 1$.
$N = 0.015 \times 2048 = 30$.

$$Baud\ Rate = \frac{41.78\ \text{MHz}}{16 \times 67 \times 2 \times \left(1 + \frac{30}{2048}\right)}$$

where *Baud Rate* = 19,219 bps.

UART REGISTER DEFINITION

The UART interface consists of the following ten registers:

COMTX: 8-bit transmit register
COMRX: 8-bit receive register
COMDIV0: divisor latch (low byte)
COMDIV1: divisor latch (high byte)
COMCON0: line control register
COMCON1: line control register
COMSTA0: line status register
COMIEN0: interrupt enable register
COMIID0: interrupt identification register
COMDIV2: 16-bit fractional baud divide register

COMTX, COMRX, and COMDIV0 share the same address location. COMTX, COMRX, and COMIEN0 can be accessed when Bit 7 in the COMCON0 register is cleared. COMDIVx can be accessed when Bit 7 of COMCON0 is set

UART TX Register

Write to this 8-bit register to transmit data using the UART.

Name: COMTX

Address: 0xFFFF0800

Access: Write only

UART RX Register

This 8-bit register is read from to receive data transmitted using the UART.

Name: COMRX

Address: 0xFFFF0800

Default Value: 0x00

Access: Read only

UART Divisor Latch Register 0

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

Name: COMDIV0

Address: 0xFFFF0800

Default Value: 0x00

Access: Read/write

UART Divisor Latch Register 1

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

Name: COMDIV1

Address: 0xFFFF0804

Default Value: 0x00

Access: Read/write

UART Control Register 0

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

Name: COMCON0

Address: 0xFFFF080C

Default Value: 0x00

Access: Read/write

Table 95. COMCON0 MMR Bit Designations

Bit	Name	Description
7	DLAB	Divisor latch access. Set by the user to enable access to COMDIV0 and COMDIV1 registers. Cleared by the user to disable access to COMDIV0 and COMDIV1 and enable access to COMRX, COMTX, and COMIEN0.
6	BRK	Set break. Set by the user to force TxD to 0. Cleared to operate in normal mode.
5	SP	Stick parity. Set by the 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 the user to transmit and check the parity bit. Cleared by the user for no parity transmission or checking.
2	STOP	Stop bit. Set by the user to transmit 1.5 stop bits 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 the user to generate one stop bit in the transmitted data.
1 to 0	WLS	Word length select. 00 = 5 bits. 01 = 6 bits. 10 = 7 bits. 11 = 8 bits.

UART Control Register 1

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

Name: COMCON1

Address: 0xFFFF0810

Default Value: 0x00

Access: Read/write

Table 96. COMCON1 MMR Bit Designations

Bit	Name	Description
7:5		Reserved bits. Not used.
4	Loopback	Set by the user to enable loopback mode. In loopback mode, the TxD is forced high.
3:2		Reserved bits. Not used.
1	RTS	Request to send. Set by the user to force the RTS output to 0. Cleared by the user to force the RTS output to 1.
0	DTR	Data terminal ready. Set by the user to force the DTR output to 0. Cleared by the user to force the DTR output to 1.

UART Status Register 0

Name: COMSTA0

Address: 0xFFFF0814

Default Value: 0x60

Access: Read only

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

Table 97. COMSTA0 MMR Bit Designations

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, that is, no more data is 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 as soon as this bit is set, the previous data might not have been transmitted yet and can 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 the stop bit is invalid. Cleared automatically.
2	PE	Parity error. Set when a parity error occurs. Cleared automatically.
1	OE	Overrun error. Set automatically if data are overwritten before being 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: 0xFFFF0804

Default Value: 0x00

Access: Read/write

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

Table 98. COMIEN0 MMR Bit Designations

Bit	Name	Description
7:4		Reserved. Not used.
3	EDSSI	Modem status interrupt enable bit. Set by the user to enable generation of an interrupt if any of COMSTA0[3:1] are set. Cleared by the user.
2	ELSI	RxD status interrupt enable bit. Set by the user to enable generation of an interrupt if any of the COMSTA0[3:1] register bits are set. Cleared by the user.
1	ETBEI	Enable transmit buffer empty interrupt. Set by the user to enable an interrupt when the buffer is empty during a transmission, that is, when COMSTA[5] is set. Cleared by the user.
0	ERBFI	Enable receive buffer full interrupt. Set by the user to enable an interrupt when the buffer is full during a reception. Cleared by the user.

UART Interrupt Identification Register 0

Name: COMIID0

Address: 0xFFFF0808

Default Value: 0x01

Access: Read only

Function: This 8-bit register reflects the source of the UART interrupt.

Table 99. COMIID0 MMR Bit Designations

Bits[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

This 16-bit register controls the operation of the fractional divider for the ADuC7122.

Name: COMDIV2

Address: 0xFFFF082C

Default Value: 0x0000

Access: Read/write

Table 100. COMDIV2 MMR Bit Designations

Bit	Name	Description
15	FBEN	Fractional baud rate generator enable bit. Set by the user to enable the fractional baud rate generator. Cleared by the user to generate the baud rate using the standard 450 UART baud rate generator.
14:13		Reserved.
12:11	FBM[1:0]	M. If FBM = 0, M = 4. See Equation 2 for the calculation of the baud rate using a fractional divider and Table 94 for common baud rate values.
10:0	FBN[10:0]	N. See Equation 2 for the calculation of the baud rate using a fractional divider and Table 94 for common baud rate values.

I²C

The ADuC7122 incorporates two I²C peripherals that can be separately configured as a fully I²C-compatible I²C bus master device or as a fully I²C bus-compatible slave device. Because both peripherals are identical, only one is explained here.

The two pins used for data transfer, SDA and SCL, are configured in a wire-AND'ed format that allows arbitration in a multimaster system. These pins require external pull-up resistors. Typical pull-up values are between 4.7 kΩ and 10 kΩ.

The I²C bus peripheral address in the I²C bus system is programmed by the user. This ID can be modified any time a transfer is not in progress. The user can configure the interface to respond to four slave addresses.

The transfer sequence of an I²C system consists of a master device initiating a transfer by generating a start condition while the bus is idle. The master transmits the slave device address and the direction of the data transfer (read or write) during the initial address transfer. If the master does not lose arbitration and the slave acknowledges the last byte, the data transfer is initiated. This continues until the master issues a stop condition, and the bus becomes idle.

The I²C peripheral can only be configured as a master or slave at any given time. The same I²C channel cannot simultaneously support master and slave modes.

The I²C interface on the ADuC7122 includes the following features:

- Support for repeated start conditions. In master mode, the ADuC7122 can be programmed to generate a repeated start. In slave mode, the ADuC7122 recognizes repeated start conditions.
- In master and slave mode, the part recognizes both 7-bit and 10-bit bus addresses.
- In I²C master mode, the ADuC7122 supports continuous reads from a single slave up to 512 bytes in a single transfer sequence.
- Clock stretching is supported in both master and slave modes.
- In slave mode, the ADuC7122 can be programmed to return a NACK (no acknowledge). This allows the validiation of checksum bytes at the end of I²C transfers.
- Bus arbitration in master mode is supported.
- Internal and external loopback modes are supported for I²C hardware testing. In loopback mode.
- The transmit and receive circuits in both master and slave mode contain 2-byte FIFOs. Status bits are available to the user to control these FIFOs.

Configuring External Pins for I²C Functionality

The I²C pins of the ADuC7122 device are P0.0 and P0.1 for I2C0, and P1.0 and P1.1 for I2C1.

P0.0 and P1.0 are the I²C clock signals, and P0.1 and P1.1 are the I²C data signals. For instance, to configure the I2C0 pins (SCL1, SDA1), Bit 0 and Bit 4 of the GP0CON register must be set to 1 to enable I²C mode. Alternatively, to configure the I2C1 pins (SCL2, SDA2), Bit 1 and Bit 5 of the GP1CON register must be set to 1 to enable I²C mode.

SERIAL CLOCK GENERATION

The I²C master in the system generates the serial clock for a transfer. The master channel can be configured to operate in fast mode (400 kHz) or standard mode (100 kHz).

The bit rate is defined in the I2CxDIV MMR as follows:

$$f_{SERIAL\ CLOCK} = \frac{f_{UCLK}}{(2 + DIVH) + (2 + DIVL)}$$

where:
f_{UCLK} is the clock before the clock divider.
DIVH is the high period of the clock.
DIVL is the low period of the clock.

Thus, for 100 kHz operation,

DIVH = *DIVL* = 0xCF

and for 400 kHz,

DIVH = 0x28, DIVL = 0x3C

The I2CxDIV register corresponds to DIVH:DIVL.

I²C BUS ADDRESSES

Slave Mode

In slave mode, the I2CxID0, I2CxID1, I2CxID2, and I2xCID3 registers contain the device IDs. The device compares the four I2CxIDx registers to the address byte received from the bus Master. To be correctly addressed, the seven MSBs of either ID register must be identical to that of the seven MSBs of the first received address byte. The LSB of the ID registers (the transfer direction bit) is ignored in the process of address recognition.

The ADuC7122 also supports 10-bit addressing mode. When Bit 1 of I2CxSCTL (ADR10EN bit) is set to 1, then one 10-bit address is supported in slave mode and is stored in the I2CxID0 and I2CxID1 registers. The 10-bit address is derived as follows:

I2CxID0[0] is the read/write bit and is not part of the I²C address.

I2CxID0[7:1] = Address Bits[6:0].

I2CxID1[2:0] = Address Bits[9:7].

I2CxID1[7:3] must be set to 11110b.

Master Mode

In master mode, the I2CADR0 register is programmed with the I²C address of the device.

In 7-bit address mode, I2CADR0[7:1] are set to the device address. I2CADR0[0] is the read/write bit.

In 10-bit address mode, the 10-bit address is created as follows:

I2CADR0[7:3] must be set to 11110b.

I2CADR0[2:1] = Address Bits[9:8].

I2CADR1[7:0] = Address Bits[7:0].

I2CADR0[0] is the read/write bit.

I²C REGISTERS

The I²C peripheral interface consists of a number of MMRs. These are described in the following section.

I²C Master Registers

I²C Master Control Register

Name: I2C0MCTL, I2C1MCTL

Address: 0xFFFF0880, 0xFFFF0900

Default Value: 0x0000, 0x0000

Access: Read/write

Function: This 16-bit MMR configures I²C peripheral in master mode.

Table 101. I2CxMCTL MMR Bit Designations

Bit	Name	Description
15:9		Reserved. These bits are reserved and should not be written to.
8	I2CMCENI	I²C transmission complete interrupt enable bit. Set this bit to enable an interrupt on detecting a stop condition on the I²C bus. Clear this bit to disable the interrupt source.
7	I2CNACKENI	I²C NACK received interrupt enable bit. Set this bit to enable interrupts when the I²C master receives a NACK. Clear this bit to disable the interrupt source.
6	I2CALENI	I²C arbitration lost interrupt enable bit. Set this bit to enable interrupts when the I²C master has lost in trying to gain control of the I²C bus. Clear this bit to disable the interrupt source.
5	I2CMTENI	I²C transmit interrupt enable bit. Set this bit to enable interrupts when the I²C master has transmitted a byte. Clear this bit to disable the interrupt source.
4	I2CMRENI	I²C receive interrupt enable bit. Set this bit to enable interrupts when the I²C master receives data. Cleared by user to disable interrupts when the I²C master is receiving data.
3	I2CMSEN	I²C master SCL stretch enable bit. Set this bit to 1 to enable clock stretching. When SCL is low, setting this bit forces the device to hold SCL low until I2CMSEN is cleared. If SCL is high, setting this bit forces the device to hold SCL low after the next falling edge. Clear this bit to disable clock stretching.
2	I2CILEN	I²C internal loopback enable. Set this bit to enable loopback test mode. In this mode, the SCL and SDA signals are connected internally to their respective input signals. Cleared by user to disable loopback mode.
1	I2CBD	I²C master backoff disable bit. Set this bit to allow the device to compete for control of the bus even if another device is currently driving a start condition. Clear this bit to back off (wait) until the I²C bus becomes free.
0	I2CMEN	I²C master enable bit. Set by user to enable I²C master mode. Cleared disable I²C master mode.

I²C Master Status Register

Name: I2C0MSTA , I2C1MSTA

Address: 0xFFFF0884, 0xFFFF0904

Default Value: 0x0000, 0x0000

Access: Read

Function: This 16-bit MMR is I²C status register in master mode.

Table 102 I2CxMSTA MMR Bit Designations

Bit	Name	Description
15:11		Reserved. These bits are reserved.
10	I2CBBUSY	I²C bus busy status bit. This bit is set to 1 when a start condition is detected on the I²C bus. This bit is cleared when a stop condition is detected on the bus.
9	I2CMRxFO	Master Rx FIFO overflow. This bit is set to 1 when a byte is written to the Rx FIFO and it is already full. This bit is cleared in all other conditions.
8	I2CMTC	I²C transmission complete status bit. This bit is set to 1 when a transmission is complete between the master and the slave it was communicating with. If the I2CMCENI bit in I2CxMCTL is set, an interrupt is generated when this bit is set. Clear this bit to disable the interrupt source.
7	I2CMNA	I²C master NACK data bit. This bit is set to 1 when a NACK condition is received by the master in response to a data write transfer. If the I2CNACKENI bit in I2CxMCTL is set, an interrupt is generated when this bit is set. This bit is cleared in all other conditions.
6	I2CMBUSY	I²C master busy status bit Set to 1 when the master is busy processing a transaction. Cleared if the master is ready or if another Master device has control of the bus.
5	I2CAL	I²C arbitration lost status bit. This bit is set to 1 when the I²C master has lost in trying to gain control of the I²C bus. If the I2CALENI bit in I2CxMCTL is set, an interrupt is generated when this bit is set. This bit is cleared in all other conditions.
4	I2CMNA	I²C master NACK address bit. This bit is set to 1 when a NACK condition is received by the master from an I²C slave address. If the I2CNACKENI bit in I2CxMCTL is set, an interrupt is generated when this bit is set. This bit is cleared in all other conditions.
3	I2CMRXQ	I²C master receive request bit. This bit is set to 1 when data enters the Rx FIFO. If the I2CMRENI in I2CxMCTL is set, an interrupt is generated. This bit is cleared in all other conditions.
2	I2CMTXQ	I²C master transmit request bit. This bit goes high if the Tx FIFO is empty or only contains one byte and the master has transmitted an address and a write. If the I2CMTENI bit in I2CxMCTL is set, an interrupt is generated when this bit is set. This bit is cleared in all other conditions.
1:0	I2CMTFSTA	I²C master Tx FIFO status bits. 00 = I²C master Tx FIFO empty. 01 = one byte in master Tx FIFO. 10 = one byte in master Tx FIFO. 11 = I²C master Tx FIFO full.

I²C Master Receive Register

Name: I2C0MRX , I2C1MRX

Address: 0xFFFF0888, 0xFFFF0908

Default Value: 0x00, 0x00

Access: Read only

Function: This 8-bit MMR is the I²C master receive register.

I²C Master Transmit Register

Name: I2C0MTX, I2C1MTX

Address: 0xFFFF088C, 0xFFFF090C

Default Value: 0x00, 0x00

Access: Read/write

Function: This 8-bit MMR is the I²C master transmit register.

I²C Master Read Count Register

Name: I2C0MCNT0, I2C1MCNT0

Address: 0xFFFF0890, 0xFFFF0910

Default Value: 0x0000, 0x0000

Access: Read/write

Function: This 16-bit MMR holds the required number of bytes when the master begins a read sequence from a slave device.

I²C Master Current Read Count Register

Name: I2C0MCNT1, I2C1MCNT1

Address: 0xFFFF0894, 0xFFFF0914

Default Value: 0x00, 0x00

Access: Read

Function: This 8-bit MMR holds the number of bytes received so far during a read sequence with a slave device.

Table 103. I2CxMCNT0 MMR Bit Descriptions (Address = 0xFFFF0890, 0xFFFF0910, Default Value = 0x0000)

Bit	Name	Description
15:9		Reserved.
8	I2CRECNT	Set this bit if greater than 256 bytes are required from the slave. Clear this bit when reading 256 bytes or less.
7:0	I2CRCNT	These eight bits hold the number of bytes required during a slave read sequence, minus 1. If only a single byte is required, these bits should be set to 0.

I²C Address 0 Register

Name: I2C0ADR0, I2C1ADR0

Address: 0xFFFF0898, 0xFFFF0918

Default Value: 0x00, 0x00

Access: Read/write

Function: This 8-bit MMR holds the 7-bit slave address + the read/write bit when the master begins communicating with a slave.

I²C Address 1 Register

Name: I2C0ADR1, I2C1ADR1

Address: 0xFFFF089C, 0xFFFF091C

Default Value: 0x00, 0x00

Access: Read/write

Function: This 8-bit MMR is used in 10-bit addressing mode only. This register contains the least significant byte of the address.

I²C Master Clock Control Register

Name: I2C0DIV, I2C1DIV

Address: 0xFFFF08A4, 0xFFFF0924

Default Value: 0x1F1F, 0x1F1F

Access: Read/write

Function: This MMR controls the frequency of the I²C clock generated by the master on to the SCL pin. For further details, see the I²C section.

I²C Start Byte Register

Name: I2C0SBYTE, I2C1SBYTE

Address: 0xFFFF08A0, 0xFFFF0920

Default Value: 0x00, 0x00

Access: Read/write

Function: This MMR can be used to generate a start byte at the start of a transaction.

To generate a start byte followed by a normal address, first write to I2CxSBYTE, then write to the address register (I2CxADRx). This drives the byte written in I2CxSBYTE onto the bus followed by a repeated start. This register can be used to drive any byte onto the I²C bus followed by a repeated start (not a start byte only, for example, 00000001).

Table 104. I2CxADR0 MMR in 7-Bit Address Mode (Address = 0xFFFF0898, 0xFFFF0918, Default Value = 0x00)

Bit	Name	Description
7:1	I2CADR	These bits contain the 7-bit address of the required slave device.
0	R/W	Bit 0 is the read/write bit. When this bit = 1, a read sequence is requested. When this bit = 0, a write sequence is requested.

Table 105. I2CxADR0 MMR in 10-Bit Address Mode

Bit	Name	Description
7:3		These bits must be set to [11110b] in 10-bit address mode.
2:1	I2CMADR	These bits contain ADDR[9:8] in 10-bit addressing mode.
0	R/W	Read/write bit. When this bit = 1, a read sequence is requested. When this bit = 0, a write sequence is requested.

Table 106. I2CxADR1 MMR in 10-Bit Address Mode

Bit	Name	Description
7:0	I2CLADR	These bits contain ADDR[7:0] in 10-bit addressing mode.

Table 107. I2CxDIV MMR

Bit	Name	Description
15:8	DIVH	These bits control the duration of the high period of SCLx.
7:0	DIVL	These bits control the duration of the low period of SCLx.

I²C Slave Registers

I²C Slave Control Register

Name: I2C0SCTL, I2C1SCTL

Address: 0xFFFF08A8, 0xFFFF0928

Default Value: 0x0000, 0x000

Access: Read/write

Function: This 16-bit MMR configures the I²C peripheral in slave mode.

Table 108. I2CxSCTL MMR Bit Designations

Bit	Name	Description
15:11		Reserved bits.
10	I2CSTXENI	Slave transmit interrupt enable bit. Set this bit to enable an interrupt after a slave transmits a byte. Clear this interrupt source.
9	I2CSRXENI	Slave receive interrupt enable bit. Set this bit to enable an interrupt after the slave receives data. Clear this interrupt source.
8	I2CSSENI	I²C stop condition detected interrupt enable bit. Set this bit to enable an interrupt on detecting a stop condition on the I²C bus. Clear this interrupt source.
7	I2CNACKEN	I²C NACK enable bit. Set this bit to NACK the next byte in the transmission sequence. Clear this bit to let the hardware control the ACK/NACK sequence.
6	I2CSSEN	I²C slave SCLx stretch enable bit. Set this bit to 1 to enable clock stretching. When SCLx is low, setting this bit forces the device to hold SCLx low until I2CSSEN is cleared. If SCL is high, setting this bit forces the device to hold SCLx low after the next falling edge. Clear this bit to disable clock stretching.
5	I2CSETEN	I²C early transmit interrupt enable bit. Setting this bit enables a transmit request interrupt just after the positive edge of SCLx during the read bit transmission. Clear this bit to enable a transmit request interrupt just after the negative edge of SCLx during the read bit transmission.
4	I2CGCCLR	I²C general call status and ID clear bit. Writing a 1 to this bit clears the general call status (I2CGC) and ID (I2CGCID[1:0]) bits in the I2CxSSTA register. Clear this bit at all other times.
3	I2CHGCEN	I²C hardware general call enable. Hardware general call enable. When this bit and Bit 2 are set, and having received a general call (Address 0x00) and a data byte, the device checks the contents of I2CxALT against the receive register. If the contents match, the device has received a hardware general call. This method is used if a device needs urgent attention from a master device without knowing which master it needs to turn to. This is a broadcast message to all master devices on the bus. The ADuC7122 watches for these addresses. The device that requires attention embeds its own address into the message. All masters listen, and the one that can handle the device contacts its slave and acts appropriately. The LSB of the I2CxALT register should always be written to 1, as per the I²C January 2000 bus specification. Set this bit and I2CGCEN to enable hardware general call recognition in slave mode. Clear this bit to disable recognition of hardware general call commands.

Bit	Name	Description
2	I2CGCEN	I²C general call enable. Set this bit to enable the slave device to acknowledge an I²C general call, Address 0x00 (write). The device then recognizes a data bit. If it receives a 0x06 (reset and write programmable part of the slave address by hardware) as the data byte, the I²C interface resets as per the I²C January 2000 bus specification. This command can be used to reset an entire I²C system. If it receives a 0x04 (write programmable part of the slave address by hardware) as the data byte, the general call interrupt status bit sets on any general call. The user must take corrective action by reprogramming the device address. Set this bit to allow the slave ACK I²C general call commands. Clear to disable recognition of general call commands.
1	ADR10EN	I²C 10-bit address mode. Set to 1 to enable 10-bit address mode. Clear to 0 to enable normal address mode.
0	I2CSEN	I²C slave enable bit. Set by the user to enable I²C slave mode. Clear to disable I²C slave mode.

I²C Slave Status Registers

Name: I2C0SSTA, I2C1SSTA

Address: 0xFFFF08AC, 0xFFFF092C

Default Value: 0x0000, 0x0000

Access: Read only

Function: This 16-bit MMR is the I²C status register in slave mode.

Table 109. I2CxSSTA MMR Bit Designations

Bit	Name	Description
15		Reserved bit.
14	I2CSTA	This bit is set to 1 if a start condition followed by a matching address is detected. It is also set if a start byte (0x01) is received, or if general calls are enabled and a general call code of 0x00 is received. This bit is cleared upon receiving a stop condition.
13	I2CREPS	This bit is set to 1 if a repeated start condition is detected. This bit is cleared on receiving a stop condition.
12-11	I2CID[1:0]	I²C address matching register. These bits indicate which I2CxIDx register matches the received address. 00 = received address matches I2CxID0. 01 = received address matches I2CxID1. 10 = received address matches I2CxID2. 11 = received address matches I2CxID3.
10	I2CSS	I²C stop condition after start detected bit. This bit is set to 1 when a stop condition is detected after a previous start and matching address. When the I2CSSENI bit in I2CxSCTL is set, an interrupt is generated. This bit is cleared by reading this register.
9:8	I2CGCID[1:0]	I²C general call ID bits. 00 = no general call received. 01 = general call reset and program address. 10 = general program address. 11 = general call matching alternative ID. Note that these bits are not cleared by a general call reset command. Clear these bits by writing a 1 to the I2CGCCLR bit in I2CxSCTL.

Bit	Name	Description
7	I2CGC	I^2C general call status bit. This bit is set to 1 if the slave receives a general call command of any type. If the command received is a reset command, then all registers return to their default state. If the command received is a hardware general call, the Rx FIFO holds the second byte of the command and this can be compared with the I2CxALT register. Clear this bit by writing a 1 to the I2CGCCLR bit in I2CxSCTL.
6	I2CSBUSY	I^2C slave busy status bit. Set to 1 when the slave receives a start condition. Cleared by hardware if the received address does not match any of the I2CxIDx registers, if the slave device receives a stop condition, or if a repeated start address does not match any of the I2CxIDx registers.
5	I2CSNA	I^2C slave NACK data bit. This bit is set to 1 when the slave responds to a bus address with a NACK. This bit is asserted if NACK is returned because there is no data in the Tx FIFO or if the I2CNACKEN bit is set in the I2CxSCTL register. This bit is cleared in all other conditions.
4	I2CSRxFO	Slave Rx FIFO overflow. This bit is set to 1 when a byte is written to the Rx FIFO when it is already full. This bit is cleared in all other conditions.
3	I2CSRXQ	I2C slave receive request bit. This bit is set to 1 when the slave Rx FIFO is not empty. This bit causes an interrupt to occur if the I2CSRXENI bit in I2CxSCTL is set. The Rx FIFO must be read or flushed to clear this bit.
2	I2CSTXQ	I^2C slave transmit request bit. This bit is set to 1 when the slave receives a matching address followed by a read. If the I2CSETEN bit in I2CxSCTL = 0, this bit goes high just after the negative edge of SCL during the read bit transmission. If the I2CSETEN bit in I2CxSCTL = 1, this bit goes high just after the positive edge of SCL during the read bit transmission. This bit causes an interrupt to occur if the I2CSTXENI bit in I2CxSCTL is set. This bit is cleared in all other conditions.
1	I2CSTFE	I^2C slave FIFO underflow status bit. This bit goes high if the Tx FIFO is empty when a master requests data from the slave. This bit is asserted at the rising edge of SCL during the read bit. This bit is cleared in all other conditions.
0	I2CETSTA	I^2C slave early transmit FIFO status bit. If the I2CSETEN bit in I2CxSCTL = 0, this bit goes high if the slave Tx FIFO is empty. If the I2CSETEN bit in I2CxSCTL = 1, this bit goes high just after the positive edge of SCL during the write bit transmission. This bit asserts once only for a transfer. This bit is cleared after being read.

I²C Slave Receive Registers

Name: I2C0SRX, I2C1SRX

Address: 0xFFFF08B0, 0xFFFF0930

Default Value: 0x00

Access: Read

Function: This 8-bit MMR is the I²C slave receive register.

I²C Slave Transmit Registers

Name: I2C0STX, I2C1STX

Address: 0xFFFF08B4, 0xFFFF0934

Default Value: 0x00

Access: Read/write

Function: This 8-bit MMR is the I²C slave transmit register.

I²C Hardware General Call Recognition Registers

Name: I2C0ALT, I2C1ALT

Address: 0xFFFF08B8, 0xFFFF0938

Default Value: 0x00

Access: Read/write

Function: This 8-bit MMR is used with hardware general calls when I2CxSCTL Bit 3 is set to 1. This register is used in cases where a master is unable to generate an address for a slave, and instead, the slave must generate the address for the master.

I²C Slave Device ID Registers

Name: I2C0IDx, I2C1IDx

Addresses: 0xFFFF093C = I2C1ID0
0xFFFF08BC = I2C0ID0
0xFFFF0940 = I2C1ID1
0xFFFF08C0 = I2C0ID1
0xFFFF0944 = I2C1ID2
0xFFFF08C4 = I2C0ID2
0xFFFF0948 = I2C1ID3
0xFFFF08C8 = I2C0ID3

Default Value: 0x00

Access: Read/write

Function: These 8-bit MMRs are programmed with I²C bus IDs of the slave. See the I²C Bus Addresses section for further details.

I²C COMMON REGISTERS

I²C FIFO Status Register

Name: I2C0FSTA, I2C1FSTA

Address: 0xFFFF08CC, 0xFFFF094C

Default Value: 0x0000

Access: Read/write

Function: These 16-bit MMRs contain the status of the Rx/Tx FIFOs in both master and slave modes.

Table 110. I2CxFSTA MMR Bit Designations

Bit	Name	Description
15:10		Reserved bits.
9	I2CFMTX	Set this bit to 1 to flush the master Tx FIFO.
8	I2CFSTX	Set this bit to 1 to flush the slave Tx FIFO.
7:6	I2CMRXSTA	I²C master receive FIFO status bits. 00 = FIFO empty. 01 = byte written to FIFO. 10 = 1 byte in FIFO. 11 = FIFO full.
5:4	I2CMTXSTA	I²C master transmit FIFO status bits. 00 = FIFO empty. 01 = byte written to FIFO. 10 = 1 byte in FIFO. 11 = FIFO full.
3:2	I2CSRXSTA	I²C slave receive FIFO status bits. 00 = FIFO empty. 01 = byte written to FIFO. 10 = 1 byte in FIFO. 11 = FIFO full.
1:0	I2CSTXSTA	I²C slave transmit FIFO status bits. 00 = FIFO empty. 01 = byte written to FIFO. 10 = 1 byte in FIFO. 11 = FIFO full.

SERIAL PERIPHERAL INTERFACE

The ADuC7122 integrates a complete hardware serial peripheral interface (SPI) on chip. SPI is an industry standard, synchronous serial interface that allows eight bits of data to be synchronously transmitted and simultaneously received, that is, full duplex up to a maximum bit rate of 20 Mb.

The SPI port can be configured for master or slave operation and typically consists of four pins: SPIMISO, SPIMOSI, SPICLK, and SPI$\overline{CS}$.

SPIMISO (MASTER IN, SLAVE OUT) PIN

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

SPIMOSI (MASTER OUT, SLAVE IN) PIN

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

SPICLK (SERIAL CLOCK I/O) PIN

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

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

$$f_{SERIAL\ CLOCK} = \frac{f_{UCLK}}{2 \times (1 + SPIDIV)}$$

The maximum speed of the SPI clock is independent on the clock divider bits.

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 10 Mb.

In both master and slave modes, data is transmitted on one edge of the SPICLK 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.

SPI CHIP SELECT (SPI$\overline{CS}$ INPUT) PIN

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

In SPI master mode, SPI$\overline{CS}$ is an active low output signal. It asserts itself automatically at the beginning of a transfer and deasserts itself upon completion.

CONFIGURING EXTERNAL PINS FOR SPI FUNCTIONALITY

The SPI pins of the ADuC7122 device are P0.2 to P0.5.

P0.5 is the slave chip select pin. In slave mode, this pin is an input and must be driven low by the master. In master mode, this pin is an output and goes low at the beginning of a transfer and high at the end of a transfer.

P0.2 is the SPICLK pin.

P0.3 is the master in, slave out (SPIMISO) pin.

P0.4 is the master out, slave in (SPIMOSI) pin.

To configure P0.2 to P0.5 for SPI mode, see the General-Purpose I/O section.

SPI REGISTERS

The following MMR registers control the SPI interface: SPISTA, SPIRX, SPITX, SPIDIV, and SPICON.

SPI Status Register

Name: SPISTA

Address: 0xFFFF0A00

Default Value: 0x0000

Access: Read only

Function: This 16-bit MMR contains the status of the SPI interface in both master and slave modes.

Table 111. SPISTA MMR Bit Designations

Bit	Name	Description
15:12		Reserved bits.
11	SPIREX	SPI Rx FIFO excess bytes present. This bit is set when there are more bytes in the Rx FIFO than indicated in the SPIMDE bits in SPICON. This bit is cleared when the number of bytes in the FIFO is equal to or less than the number in SPIRXMDE.
10:8	SPIRXFSTA[2:0]	SPI Rx FIFO status bits. 000 = Rx FIFO is empty. 001 = 1 valid byte in the FIFO. 010 = 2 valid bytes in the FIFO. 011 = 3 valid bytes in the FIFO. 100 = 4 valid bytes in the FIFO. Clear this bit to disable clock stretching.
7	SPIFOF	SPI Rx FIFO overflow status bit. Set when the Rx FIFO was already full when new data was loaded to the FIFO. This bit generates an interrupt except when SPIRFLH is set in SPICON. Cleared when the SPISTA register is read.
6	SPIRXIRQ	SPI Rx IRQ status bit. Set when a receive interrupt occurs. This bit is set when SPITMDE in SPICON is cleared and the required number of bytes have been received. Cleared when the SPISTA register is read.
5	SPITXIRQ	SPI Tx IRQ status bit. Set when a transmit interrupt occurs. This bit is set when SPITMDE in SPICON is set and the required number of bytes have been transmitted. Cleared when the SPISTA register is read.
4	SPITXUF	SPI Tx FIFO underflow. This bit is set when a transmit is initiated without any valid data in the Tx FIFO. This bit generates an interrupt except when SPITFLH is set in SPICON. Cleared when the SPISTA register is read.
3:1	SPITXFSTA[2:0]	SPI Tx FIFO status bits. 000 = Tx FIFO is empty. 001 = 1 valid byte in the FIFO. 010 = 2 valid bytes in the FIFO. 011 = 3 valid bytes in the FIFO. 100 = 4 valid bytes in the FIFO. Clear this bit to enable 7-bit address mode.
0	SPIISTA	SPI interrupt status bit. Set to 1 when an SPI-based interrupt occurs. Cleared after reading SPISTA.

SPIRX Register

Name: SPIRX

Address: 0xFFFF0A04

Default Value: 0x00

Access: Read

Function: This 8-bit MMR is the SPI receive register.

SPITX Register

Name: SPITX

Address: 0xFFFF0A08

Default Value: 0x00

Access: Write

Function: This 8-bit MMR is the SPI transmit register.

SPIDIV Register

Name: SPIDIV

Address: 0xFFFF0A0C

Default Value: 0x1B

Access: Read/write

Function: This 8-bit MMR is the SPI baud rate selection register.

SPI Control Register

Name: SPICON

Address: 0xFFFF0A10

Default Value: 0x0000

Access: Read/write

Function: This 16-bit MMR configures the SPI peripheral in both master and slave modes.

Table 112. SPICON MMR Bit Designations

Bit	Name	Description
15:14	SPIMDE	SPI IRQ mode bits. These bits configure when the Tx/Rx interrupts occur in a transfer. 00 = Tx interrupt occurs when one byte has been transferred. Rx interrupt occurs when one or more bytes have been received by the FIFO. 01 = Tx interrupt occurs when two bytes have been transferred. Rx interrupt occurs when two or more bytes have been received by the FIFO. 10 = Tx interrupt occurs when three bytes have been transferred. Rx interrupt occurs when three or more bytes have been received by the FIFO. 11 = Tx interrupt occurs when four bytes have been transferred. Rx interrupt occurs when the Rx FIFO is full, or four bytes present.
13	SPITFLH	SPI Tx FIFO flush enable bit. Set this bit to flush the Tx FIFO. This bit does not clear itself and should be toggled if a single flush is required. If this bit is left high, then either the last transmitted value or 0x00 is transmitted depending on the SPIZEN bit. When the flush enable bit is set, the FIFO is cleared within a single microprocessor cycle. Any writes to the Tx FIFO are ignored while this bit is set. Clear this bit to disable Tx FIFO flushing.
12	SPIRFLH	SPI Rx FIFO flush enable bit. Set this bit to flush the Rx FIFO. This bit does not clear itself and should be toggled if a single flush is required. When the flush enable bit is set, the FIFO is cleared within a single microprocessor cycle. If this bit is set, all incoming data is ignored and no interrupts are generated. If set and SPITMDE = 0, a read of the Rx FIFO initiates a transfer. Clear this bit to disable Rx FIFO flushing.
11	SPICONT	Continuous transfer enable. Set by the user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the Tx register. SPI$\overline{CS}$ is asserted and remains asserted for the duration of each 8-bit serial transfer until Tx is empty. Cleared by the 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 of one serial clock cycle.
10	SPILP	Loop back enable bit. Set by the user to connect MISO to MOSI and test software. Cleared by the user to place in normal mode.
9	SPIOEN	Slave MISO output enable bit. Set this bit for SPIMISO to operate as normal. Clear this bit to disable the output driver on the SPIMISO pin. The SPIMISO pin is open-drain when this bit is clear.

Bit	Name	Description
8	SPIROW	SPIRX overflow overwrite enable. Set by the user, the valid data in the Rx register is overwritten by the new serial byte received. Cleared by the user, the new serial byte received is discarded.
7	SPIZEN	SPI transmit zeros when the Tx FIFO is empty. Set this bit to transmit 0x00 when there is no valid data in the Tx FIFO. Clear this bit to transmit the last transmitted value when there is no valid data in the Tx FIFO.
6	SPITMDE	SPI transfer and interrupt mode. Set by the user to initiate transfer with a write to the SPITX register. Interrupt only occurs when Tx is empty. Cleared by the user to initiate transfer with a read of the SPIRX register. Interrupt only occurs when Rx is full.
5	SPILF	LSB first transfer enable bit. Set by the user, the LSB is transmitted first Cleared by the user, the MSB is transmitted first.
4	SPIWOM	SPI wired or mode enable bit Set to 1 to enable open-drain data output enable. External pull-ups required on data out pins. Cleared for normal output levels.
3	SPICPO	Serial clock polarity mode bit. Set by the user, the serial clock idles high. Cleared by the user, the serial clock idles low.
2	SPICPH	Serial clock phase mode bit. Set by the user, the serial clock pulses at the beginning of each serial bit transfer. Cleared by the user, the serial clock pulses at the end of each serial bit transfer.
1	SPIMEN	Master mode enable bit. Set by the user to enable master mode. Cleared by the user to enable slave mode.
0	SPIEN	SPI enable bit. Set by the user to enable the SPI. Cleared by the user to disable the SPI.

PROGRAMMABLE LOGIC ARRAY (PLA)

The ADuC7122 integrates a fully programmable logic array (PLA) that consists of two, independent but interconnected PLA blocks. Each block consists of eight PLA elements, giving each part a total of 16 PLA elements.

Each PLA element contains a two-input look-up table that can be configured to generate any logic output function based on two inputs and a flip-flop. This is represented in Figure 35.

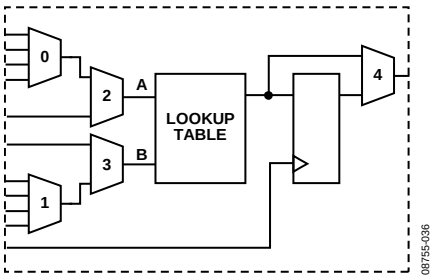


Figure 35. PLA Element

In total, 32 GPIO pins are available on each ADuC7122 for the PLA. These include 16 input pins and 16 output pins that need to be configured in the GPxCON register as PLA pins before using the PLA. Note that the comparator output is also included as one of the 16 input pins.

The PLA is configured via a set of user MMRs. The output(s) of the PLA can be routed to the internal interrupt system, to the $\overline{\text{CONVST}}$ signal of the ADC, to an MMR, or to any of the 16 PLA output pins.

The two blocks can be interconnected as follows:

- Output of Element 15 (Block 1) can be fed back to Input 0 of Mux 0 of Element 0 (Block 0)
- Output of Element 7 (Block 0) can be fed back to the Input 0 of Mux 0 of Element 8 (Block 1)

Table 113. Element Input/Output

PLA Block 0			PLA Block 1		
Element	**Input**	**Output**	**Element**	**Input**	**Output**
0	P2.7	P3.0	8	P1.4	P3.4
1	P2.2	P3.1	9	P1.5	P3.5
2	P0.6	P3.2	10	P0.5	P3.6
3	P0.7	P3.3	11	P0.4	P3.7
4	P0.1	P1.7	12	P2.1	P0.3
5	P0.0	P1.6	13	P2.0	P0.2
6	P1.1	P2.5	14	P2.3	P1.3
7	P1.0	P2.4	15	P2.6	P1.2

PLA MMRs Interface

The PLA peripheral interface consists of the 21 MMRs described in Table 114 to Table 128.

Table 114. PLAELMx Registers

Name	Address	Default Value	Access
PLAELM0	0xFFFF0B00	0x0000	R/W
PLAELM1	0xFFFF0B04	0x0000	R/W
PLAELM2	0xFFFF0B08	0x0000	R/W
PLAELM3	0xFFFF0B0C	0x0000	R/W
PLAELM4	0xFFFF0B10	0x0000	R/W
PLAELM5	0xFFFF0B14	0x0000	R/W
PLAELM6	0xFFFF0B18	0x0000	R/W
PLAELM7	0xFFFF0B1C	0x0000	R/W
PLAELM8	0xFFFF0B20	0x0000	R/W
PLAELM9	0xFFFF0B24	0x0000	R/W
PLAELM10	0xFFFF0B28	0x0000	R/W
PLAELM11	0xFFFF0B2C	0x0000	R/W
PLAELM12	0xFFFF0B30	0x0000	R/W
PLAELM13	0xFFFF0B34	0x0000	R/W
PLAELM14	0xFFFF0B38	0x0000	R/W
PLAELM15	0xFFFF0B3C	0x0000	R/W

PLAELMx are Element 0 to Element 15 control registers. They configure the input and output mux of each element, select the function in the look-up table, and bypass/use the flip-flop. See Table 115 and Table 118.

Table 115. PLAELMx MMR Bit Descriptions

Bit	Value	Description
31:11		Reserved.
10:9		Mux 0 control (see Table 118).
8:7		Mux 1 control (see Table 118).
6		Mux 2 control.
	1	Set by user to select the output of Mux 0.
	0	Cleared by user to select the bit value from the PLADIN register.
5		Mux 3 control.
	1	Set by user to select the input pin of the particular element.
	0	Cleared by user to select the output of Mux 1.
4:1		Look-up table control.
	0000	0.
	0001	NOR.
	0010	B AND NOT A.
	0011	NOT A.
	0100	A AND NOT B.
	0101	NOT B.
	0110	EXOR.
	0111	NAND.
	1000	AND.
	1001	EXNOR.
	1010	B.
	1011	NOT A OR B.
	1100	A.
	1101	A OR NOT B.
	1110	OR.
	1111	1.
0		Mux 4 control. Set by user to bypass the flip-flop. Cleared by user to select the flip-flop (cleared by default).

Table 116. PLACLK Register

Name	Address	Default Value	Access
PLACLK	0xFFFF0B40	0x00	R/W

PLACLK is the clock selection for the flip-flops of Block 0 and Block 1. Note that the maximum frequency when using the GPIO pins as the clock input for the PLA blocks is 41.78 MHz.

Table 117. PLACLK MMR Bit Descriptions

Bit	Value	Description
7		Reserved.
6:4		Block 1 clock source selection.
	000	GPIO clock on P0.5.
	001	GPIO clock on P0.0.
	010	GPIO clock on P0.7.
	011	HCLK.
	100	External crystal (OCLK) (32.768 kHz).
	101	Timer1 overflow.
	Other	Reserved.
3		Reserved.
2:0		Block 0 clock source selection.
	000	GPIO clock on P0.5.
	001	GPIO clock on P0.0.
	010	GPIO clock on P0.7.
	011	HCLK.
	100	External crystal (OCLK) (32.768 kHz).
	101	Timer1 overflow.
	Other	Reserved.

Table 118. Feedback Configuration

Bit	Value	PLAELM0	PLAELM1 to PLAELM7	PLAELM8	PLAELM9 to PLAELM15
10:9	00	Element 15	Element 0	Element 7	Element 8
	01	Element 2	Element 2	Element 10	Element 10
	10	Element 4	Element 4	Element 12	Element 12
	11	Element 6	Element 6	Element 14	Element 14
8:7	00	Element 1	Element 1	Element 9	Element 9
	01	Element 3	Element 3	Element 11	Element 11
	10	Element 5	Element 5	Element 13	Element 13
	11	Element 7	Element 7	Element 15	Element 15

Table 119. PLAIRQ Register

Name	Address	Default Value	Access
PLAIRQ	0xFFFF0B44	0x00000000	R/W

PLAIRQ enables IRQ0 and/or IRQ1 and selects the source of the IRQ.

Table 120. PLAIRQ MMR Bit Descriptions

Bit	Value	Description
15:13		Reserved.
12		PLA IRQ1 enable bit.
	1	Set by the user to enable the IRQ1 output from PLA.
	0	Cleared by the user to disable IRQ1 output from PLA.
11:8		PLA IRQ1 source.
	0000	PLA Element 0.
	0001	PLA Element 1.
	1111	PLA Element 15.
7:5		Reserved.
4		PLA IRQ0 enable bit. Set by the user to enable IRQ0 output from PLA. Cleared by the user to disable IRQ0 output from PLA.
3:0		PLA IRQ0 source.
	0000	PLA Element 0.
	0001	PLA Element 1.
	1111	PLA Element 15.

Table 121. PLAADC Register

Name	Address	Default Value	Access
PLAADC	0xFFFF0B48	0x00000000	R/W

PLAADC is the PLA source for the ADC start conversion signal.

Table 122. PLAADC MMR Bit Descriptions

Bit	Value	Description
31:5		Reserved.
4		ADC start conversion enable bit.
	1	Set by the user to enable ADC start conversion from PLA.
	0	Cleared by the user to disable ADC start conversion from PLA.
3:0		ADC start conversion source.
	0000	PLA Element 0.
	0001	PLA Element 1.
	1111	PLA Element 15.

Table 123. PLADIN Register

Name	Address	Default Value	Access
PLADIN	0xFFFF0B4C	0x00000000	R/W

Table 124. PLADIN MMR Bit Descriptions

Bit	Description
31:16	Reserved.
15:0	Input bit to Element 15 to Element 0.

PLADIN is a data input MMR for PLA.

Table 125. PLADOUT Register

Name	Address	Default Value	Access
PLADOUT	0xFFFF0B50	0x00000000	R

PLADOUT is a data output MMR for PLA. This register is always updated.

Table 126. PLADOUT MMR Bit Descriptions

Bit	Description
31:16	Reserved.
15:0	Output bit from Element 15 to Element 0.

Table 127. PLACLK Register

Name	Address	Default Value	Access
PLACLK	0xFFFF0B40	0x00	W

PLACLK is a PLA lock option. Bit 0 is written only once. When set, it does not allow modification of any of the PLA MMRs, except PLADIN. A PLA tool is provided in the development system to easily configure the PLA.

INTERRUPT SYSTEM

There are 27 interrupt sources on the ADuC7122 that are controlled by the interrupt controller. All interrupts are generated from the on-chip peripherals, except for the software interrupt (SWI), which is programmable by the user. The ARM7TDMI CPU core only recognizes 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 a number of interrupt-related registers. The bits in each IRQ and FIQ register represent the same interrupt source, as described in Table 128.

The ADuC7122 contains a vectored interrupt controller (VIC) that supports nested interrupts up to eight levels. The VIC also allows the programmer to assign priority levels to all interrupt sources. Interrupt nesting needs to be enabled by setting the ENIRQN bit in the IRQCONN register. A number of extra MMRs are used when the full vectored interrupt controller is enabled.

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

Table 128. IRQ/FIQ[1] MMRs Bit Designations

Bit	Description	Comments
0	All interrupts OR'ed (FIQ only)	This bit is set if any FIQ is active
1	Software interrupt	User programmable interrupt source
2	Timer0	General-Purpose Timer0
3	Timer1	General-Purpose Timer1
4	Timer2 or wake-up timer	General-Purpose Timer2 or wake-up timer
5	Timer3 or watchdog timer	General-Purpose Timer3 or watchdog timer
6	Timer4	General-Purpose Timer4
7	Reserved	Reserved
8	PSM	Power supply monitor
9	Undefined	This bit is not used
10	Flash Control 0	Flash controller for Block 0 interrupt
11	Flash Control 1	Flash controller for Block 1 interrupt
12	ADC	ADC interrupt source bit
13	UART	UART interrupt source bit
14	SPI	SPI interrupt source bit
15	I2C0 master IRQ	I^2C master interrupt source bit
16	I2C0 slave IRQ	I^2C slave interrupt source bit
17	I2C1 master IRQ	I^2C master interrupt source bit
18	I2C1 slave IRQ	I^2C slave interrupt source bit
19	XIRQ0 (GPIO IRQ0)	External Interrupt 0
20	XIRQ1 (GPIO IRQ1)	External Interrupt 1
21	XIRQ2 (GPIO IRQ2)	External Interrupt 2
22	XIRQ3 (GPIO IRQ3)	External Interrupt 3
23	PWM	PWM trip interrupt source bit
24	XIRQ4 (GPIO IRQ4)	External Interrupt 4
25	XIRQ5 (GPIO IRQ5)	External Interrupt 5
26	PLA IRQ0	PLA Block 0 IRQ bit
27	PLA IRQ1	PLA Block 1 IRQ bit

[1] Applies to IRQEN, FIQEN, IRQCLR, FIQCLR, IRQSTA, and FIQSTA registers.

IRQ

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

All 32 bits are logically OR'ed to create a single IRQ signal to the ARM7TDMI core. The four 32-bit registers dedicated to IRQ are IRQSIG, IRQEN, IRQCLR, and IRQSTA.

IRQSIG

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

IRQSIG Register

Name: IRQSIG

Address: 0xFFFF0004

Default Value: 0x00000000

Access: Read only

IRQEN

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

IRQEN Register

Name: IRQEN

Address: 0xFFFF0008

Default Value: 0x00000000

Access: Read/write

IRQCLR

IRQCLR is a write-only register that allows the IRQEN register to clear to mask an interrupt source. Each bit that is set to 1 clears 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.

IRQCLR Register

Name: IRQCLR

Address: 0xFFFF000C

Default Value: 0x00000000

Access: Write only

IRQSTA

IRQSTA is a read-only register that provides the current enabled IRQ source status (effectively a logic AND of the IRQSIG and IRQEN bits). When set to 1, that source generates an active IRQ request to the ARM7TDMI core. There is no priority encoder or interrupt vector generation. This function is implemented in software in a common interrupt handler routine.

IRQSTA Register

Name: IRQSTA

Address: 0xFFFF0000

Default Value: 0x00000000

Access: Read only

FAST INTERRUPT REQUEST (FIQ)

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

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

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

FIQSIG

FIQSIG reflects the status of the different FIQ sources. If a peripheral generates an FIQ signal, the corresponding bit in the FIQSIG is set; otherwise, it is cleared. The FIQSIG bits are cleared when the interrupt in the particular peripheral is cleared. All FIQ sources can be masked in the FIQEN MMR. FIQSIG is read only.

FIQSIG Register

Name: FIQSIG

Address: 0xFFFF0104

Default Value: 0x00000000

Access: Read only

FIQEN

FIQEN provides the value of the current enable mask. When a bit is set to 1, the corresponding source request is enabled to create an FIQ exception. When a bit is set to 0, the corresponding source request is disabled or masked, which does not create an FIQ exception. The FIQEN register cannot be used to disable an interrupt.

FIQEN Register

Name: FIQEN

Address: 0xFFFF0108

Default Value: 0x00000000

Access: Read/write

FIQCLR

FIQCLR is a write-only register that allows the FIQEN register to clear to mask an interrupt source. Each bit that is set to 1 clears the corresponding bit in the FIQEN register without affecting the remaining bits. The pair of registers, FIQEN and FIQCLR, allows independent manipulation of the enable mask without requiring an atomic read-modify-write.

FIQCLR Register

Name: FIQCLR

Address: 0xFFFF010C

Default Value: 0x00000000

Access: Write only

FIQSTA

FIQSTA is a read-only register that provides the current enabled FIQ source status (effectively a logic AND of the FIQSIG and FIQEN bits). When set to 1, that source generates an active FIQ 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.

FIQSTA Register

Name: FIQSTA

Address: 0xFFFF0100

Default Value: 0x00000000

Access: Read only

Programmed Interrupts

Because the programmed interrupts are not maskable, they are controlled by another register (SWICFG) that writes into both IRQSTA and IRQSIG registers and/or the FIQSTA and FIQSIG registers at the same time.

The 32-bit register dedicated to software interrupt is SWICFG, described in Table 129. This MMR allows the control of a programmed source interrupt.

Table 129. SWICFG MMR Bit Designations

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

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

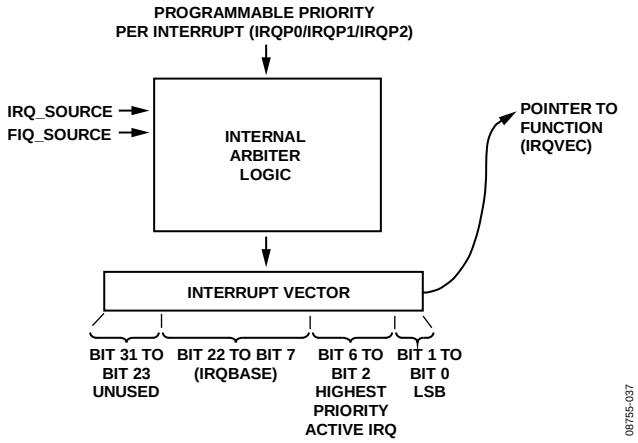


Figure 36. Interrupt Structure

Vectored Interrupt Controller (VIC)

The ADuC7122 incorporates an enhanced interrupt control system or vectored interrupt controller. The vectored interrupt controller for IRQ interrupt sources is enabled by setting Bit 0 of the IRQCONN register. Similarly, Bit 1 of IRQCONN enables the vectored interrupt controller for the FIQ interrupt sources. The vectored interrupt controller provides the following enhancements to the standard IRQ/FIQ interrupts:

- Vectored interrupts—allow a user to define separate interrupt service routine addresses for every interrupt source. This is achieved by using the IRQBASE and IRQVEC registers.
- IRQ/FIQ interrupts—can be nested up to eight levels depending on the priority settings. An FIQ still has a higher priority than an IRQ. Therefore, if the VIC is enabled for both the FIQ and IRQ and prioritization is maximized, then it is possible to have 16 separate interrupt levels.
- Programmable interrupt priorities—using the IRQP0 to IRQP2 registers, an interrupt source can be assigned an interrupt priority level value between 1 and 8.

VIC MMRs

IRQBASE Register

The vector base register, IRQBASE, is used to point to the start address of memory used to store 32 pointer addresses. These pointer addresses are the addresses of the individual interrupt service routines.

Name: IRQBASE

Address: 0xFFFF0014

Default Value: 0x00000000

Access: Read and write

Table 130. IRQBASE MMR Bit Designations

Bit	Type	Initial Value	Description
31:16	Read only	Reserved	Always read as 0
15:0	R/W	0	Vector base address

IRQVEC Register

The IRQ interrupt vector register, IRQVEC points to a memory address containing a pointer to the interrupt service routine of the currently active IRQ. This register should only be read when an IRQ occurs and IRQ interrupt nesting has been enabled by setting Bit 0 of the IRQCONN register.

Name: IRQVEC

Address: 0xFFFF001C

Default Value: 0x00000000

Access: Read only

Table 131. IRQVEC MMR Bit Designations

Bit	Type	Initial Value	Description
31:23	Read only	0	Always read as 0.
22:7	R/W	0	IRQBASE register value.
6:2	Read only	0	Highest priority IRQ source. This is a value between 0 to 27 representing the possible interrupt sources. For example, if the highest currently active IRQ is Timer1, then these bits are 00011.
1:0	Reserved	0	Reserved bits.

Priority Registers

IRQP0 Register

Name: IRQP0

Address: 0xFFFF0020

Default Value: 0x00000000

Access: Read and write

Table 132. IRQP0 MMR Bit Designations

Bit	Name	Description
31:27	Reserved	Reserved bit.
26:24	T4PI	A priority level of 0 to 7 can be set for Timer4.
23	Reserved	Reserved bit.
22:20	T3PI	A priority level of 0 to 7 can be set for Timer3.
19	Reserved	Reserved bit.
18:16	T2PI	A priority level of 0 to 7 can be set for Timer2.
15	Reserved	Reserved bit.
14:12	T1PI	A priority level of 0 to 7 can be set for Timer1.
11	Reserved	Reserved bit.
10:8	T0PI	A priority level of 0 to 7 can be set for Timer0.
7	Reserved	Reserved bit.
6:4	SWINTP	A priority level of 0 to 7 can be set for the software interrupt source.
3:0	Reserved	Interrupt 0 cannot be prioritized.

IRQP1 Register

Name: IRQP1

Address: 0xFFFF0024

Default Value: 0x00000000

Access: Read and write

Table 133. IRQP1 MMR Bit Designations

Bit	Name	Description
31	Reserved	Reserved bit.
30:28	I2C0MPI	A priority level of 0 to 7 can be set for the I2C0 master.
27	Reserved	Reserved bit.
26:24	SPIPI	A priority level of 0 to 7 can be set for the SPI.
23	Reserved	Reserved bit.
22:20	UARTPI	A priority level of 0 to 7 can be set for the UART.
19	Reserved	Reserved bit.
18:16	ADCPI	A priority level of 0 to 7 can be set for the ADC interrupt source.
15	Reserved	Reserved bit.
14:12	Flash1PI	A priority level of 0 to 7 can be set for the Flash Block 1 controller interrupt source.
11	Reserved	Reserved bit.
10:8	Flash0PI	A priority level of 0 to 7 can be set for the Flash Block 0 controller interrupt source.
7:3	Reserved	Reserved bits.
2:0	PSMPI	A priority level of 0 to 7 can be set for the Power supply monitor interrupt source.

IRQP2 Register

Name: IRQP2

Address: 0xFFFF0028

Default Value: 0x00000000

Access: Read and write

Table 134. IRQP2 MMR Bit Designations

Bit	Name	Description
31	Reserved	Reserved bit.
30:28	PWMPI	A priority level of 0 to 7 can be set for PWM.
27	Reserved	Reserved bit.
26:24	IRQ3PI	A priority level of 0 to 7 can be set for IRQ3.
23	Reserved	Reserved bit.
22:20	IRQ2PI	A priority level of 0 to 7 can be set for IRQ2.
19	Reserved	Reserved bit.
18:16	IRQ1PI	A priority level of 0 to 7 can be set for IRQ1.
15	Reserved	Reserved bit.
14:12	IRQ0PI	A priority level of 0 to 7 can be set for IRQ0.
11	Reserved	Reserved bit.
10:8	I2C1SPI	A priority level of 0 to 7 can be set for I2C1 slave.
7	Reserved	Reserved bit.
6:4	I2C1MPI	A priority level of 0 to 7 can be set for I2C1 master.
3	Reserved	Reserved bit.
2:0	I2C0SPI	A priority level of 0 to 7 can be set for I2C0 slave.

IRQP3 Register

Name: IRQP3

Address: 0xFFFF002C

Default Value: 0x00000000

Access: Read and write

Table 135. IRQP3 MMR Bit Designations

Bit	Name	Description
31:15	Reserved	Reserved bit.
14:12	PLA1PI	A priority level of 0 to 7 can be set for PLA0.
11	Reserved	Reserved bit.
10:8	PLA0PI	A priority level of 0 to 7 can be set for PLA0.
7	Reserved	Reserved bit.
6:4	IRQ5PI	A priority level of 0 to 7 can be set for IRQ5.
3	Reserved	Reserved bit.
2:0	IRQ4PI	A priority level of 0 to 7 can be set for IRQ4.

IRQCONN Register

The IRQCONN register is the IRQ and FIQ control register. It contains two active bits. The first enables nesting and prioritization of IRQ interrupts and the other enables nesting and prioritization of FIQ interrupts.

If these bits are cleared, then FIQs and IRQs can still be used, but it is not possible to nest IRQs or FIQs. Neither is it possible to set an interrupt source priority level. In this default state, an FIQ does have a higher priority than an IRQ.

Name: IRQCONN

Address: 0xFFFF0030

Default Value: 0x00000000

Access: Read and write

Table 136. IRQCONN MMR Bit Designations

Bit	Name	Description
31:2	Reserved	These bits are reserved and should not be written to.
1	ENFIQN	Setting this bit to 1 enables nesting of FIQ interrupts. Clearing this bit means no nesting or prioritization of FIQs is allowed.
0	ENIRQN	Setting this bit to 1 enables nesting of IRQ interrupts. Clearing this bit means no nesting or prioritization of IRQs is allowed.

IRQSTAN Register

If IRQCONN[0] is asserted and IRQVEC is read, then one of these bits is asserted. The bit that asserts depends on the priority of the IRQ. For example, if the IRQ is of Priority 0 then Bit 0 asserts; if it is Priority 1, then Bit 1 asserts. When a bit is set in this register, all interrupts of that priority and lower are blocked.

To clear a bit in this register, all bits of a higher priority must be cleared first. It is only possible to clear one bit at a time. For example, if this register is set to 0x09, then writing 0xFF changes the register to 0x08, and writing 0xFF a second time changes the register to 0x00.

Name: IRQSTAN

Address: 0xFFFF003C

Default Value: 0x00000000

Access: Read and write

Table 137. IRQSTAN MMR Bit Designations

Bit	Name	Description
31:8	Reserved	These bits are reserved and should not be written to.
7:0		Setting this bit to 1 enables nesting of FIQ interrupts. Clearing this bit means no nesting or prioritization of FIQs is allowed.

FIQVEC Register

The FIQ interrupt vector register, FIQVEC points to a memory address containing a pointer to the interrupt service routine of the currently active FIQ. This register should only be read when an FIQ occurs and FIQ interrupt nesting has been enabled by setting Bit 1 of the IRQCONN register.

Name: FIQVEC

Address: 0xFFFF011C

Default Value: 0x00000000

Access: Read only

Table 138. FIQVEC MMR Bit Designations

Bit	Type	Initial Value	Description
31:23	Read only	0	Always read as 0.
22:7	R/W	0	IRQBASE register value.
6:2		0	Highest priority FIQ source. This is a value between 0 to 27, which represents the possible interrupt sources. For example, if the highest currently active FIQ is Timer1, then these bits are 00011.
1:0	Reserved	0	Reserved bits.

FIQSTAN Register

If IRQCONN[1] is asserted and FIQVEC is read then one of these bits assert. The bit that asserts depends on the priority of the FIQ. For example, if the FIQ is of Priority 0, then Bit 0 asserts; if it is Priority 1, then Bit 1 asserts.

When a bit is set in this register, all interrupts of that priority and lower are blocked.

To clear a bit in this register, all bits of a higher priority must be cleared first. It is only possible to clear one bit as a time. For example, if this register is set to 0x09, then writing 0xFF changes the register to 0x08 and writing 0xFF a second time changes the register to 0x00.

Name: FIQSTAN

Address: 0xFFFF013C

Default Value: 0x00000000

Access: Read and write

Table 139. FIQSTAN MMR Bit Designations

Bit	Name	Description
31:8	Reserved	These bits are reserved and should not be written to.
7:0		Setting this bit to 1 enables nesting of FIQ interrupts. Clearing this bit means no nesting or prioritization of FIQs is allowed.

External Interrupts (IRQ0 to IRQ5)

The ADuC7122 provides up to six external interrupt sources. These external interrupts can be individually configured as level or rising/falling edge triggered.

To enable the external interrupt source, the appropriate bit must first be set in the FIQEN or IRQEN register. To select the required edge or level to trigger on, the IRQCONE register must be appropriately configured.

To properly clear an edge based external IRQ interrupt, set the appropriate bit in the IRQCLRE register.

IRQCONE Register

Name: IRQCONE

Address: 0xFFFF0034

Default Value: 0x00000000

Access: Read and write

Table 140. IRQCONE MMR Bit Designations

Bit	Value	Name	Description
31:12		Reserved	These bits are reserved and should not be written to.
11:10	11	IRQ5SRC[1:0]	External IRQ5 triggers on falling edge.
	10		External IRQ5 triggers on rising edge.
	01		External IRQ5 triggers on low level.
	00		External IRQ5 triggers on high level.
9:8	11	IRQ4SRC[1:0]	External IRQ4 triggers on falling edge.
	10		External IRQ4 triggers on rising edge.
	01		External IRQ4 triggers on low level.
	00		External IRQ4 triggers on high level.
7:6	11	IRQ3SRC[1:0]	External IRQ3 triggers on falling edge.
	10		External IRQ3 triggers on rising edge.
	01		External IRQ3 triggers on low level.
	00		External IRQ3 triggers on high level.
5:4	11	IRQ2SRC[1:0]	External IRQ2 triggers on falling edge.
	10		External IRQ2 triggers on rising edge.
	01		External IRQ2 triggers on low level.
	00		External IRQ2 triggers on high level.
3:2	11	IRQ1SRC[1:0]	External IRQ1 triggers on falling edge.
	10		External IRQ1 triggers on rising edge.
	01		External IRQ1 triggers on low level.
	00		External IRQ1 triggers on high level.
1:0	11	IRQ0SRC[1:0]	External IRQ0 triggers on falling edge.
	10		External IRQ0 triggers on rising edge.
	01		External IRQ0 triggers on low level.
	00		External IRQ0 triggers on high level.

IRQCLRE Register

Name: IRQCLRE

Address: 0xFFFF0038

Default Value: 0x00000000

Access: Write only

Table 141. IRQCLRE MMR Bit Designations

Bit	Name	Description
31:26	Reserved	These bits are reserved and should not be written to.
25	IRQ5CLRI	A 1 must be written to this bit in the IRQ5 interrupt service routine to clear an edge.
24	IRQ4CLRI	A 1 must be written to this bit in the IRQ4 interrupt service routine to clear an edge.
23	Reserved	This bit is reserved and should not be written to.
22	IRQ3CLRI	A 1 must be written to this bit in the IRQ3 interrupt service routine to clear an edge triggered IRQ3 interrupt.
21	IRQ2CLRI	A 1 must be written to this bit in the IRQ2 interrupt service routine to clear an edge triggered IRQ2 interrupt.
20	IRQ1CLRI	A 1 must be written to this bit in the IRQ1 interrupt service routine to clear an edge triggered IRQ1 interrupt.
19	IRQ0CLRI	A 1 must be written to this bit in the IRQO interrupt service routine to clear an edge triggered IRQ0 interrupt.
18:0	Reserved	These bits are reserved and should not be written to.

TIMERS

The ADuC7122 has five general-purpose timers/counters.

- Timer0
- Timer1
- Timer2 or wake-up timer
- Timer3 or watchdog timer
- Timer4

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

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

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 the clear register of the particular timer (TxCLRI).

The event selection feature allows flexible interrupt generation based on Timer0 and Timer1. T0CON and T1CON can be used to configure the interrupt sources, as shown in Table 142. When either Timer0 or Timer1 expires, an interrupt occurs based on the event selection in T0CON and T1CON MMRs.

Table 142. Event Selection Numbers

Event Selection (TxCON[16:12])	Interrupt Number	Name
00000	2	Timer0
00001	3	Timer1
00010	4	Wake-up timer (Timer2)
00011	5	Watchdog timer (Timer3)
00100	6	Timer4
00101	7	Reserved
00110	8	Power supply monitor
00111	9	Undefined
01000	10	Flash Block 0
01001	11	Flash Block 1
01010	12	ADC
01011	13	UART
01100	14	SPI
01101	15	I2C0 master
01110	16	I2C0 slave
01111	17	I2C1 master
10000	18	I2C1 slave
10001	19	External IRQ0

TIMER0—LIFETIME TIMER

Timer0 is a general-purpose 48-bit count up or a 16-bit count up/down timer with a programmable prescaler. Timer0 is clocked from the core clock, with a prescaler of 1, 16, 256, or 32,768. This gives a minimum resolution of 22 ns when the core is operating at 41.78 MHz and with a prescaler of 1. Timer0 can also be clocked from the undivided core clock, internal 32 kHz oscillator, or external 32 kHz crystal.

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

In 16-bit mode, Timer0 can count up or count down. A 16-bit value can be written to T0LD that is loaded into the counter. The current counter value can be read from T0VAL0. Timer0 has a capture register (T0CAP) that can be triggered by a selected IRQ's source initial assertion. When 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.

The Timer0 interface consists of six MMRs, shown in Table 143.

Table 143. Timer0 Interface MMRs

Name	Description
T0LD	16-bit register that holds the 16-bit value loaded into the counter. Available only in 16-bit mode.
T0CAP	16-bit register that holds the 16-bit value captured by an enabled IRQ event. Available only in 16-bit mode.
T0VAL0/T0VAL1	TOVAL0 is a 16-bit register that holds the 16 least significant bits (LSBs). T0VAL1 is a 32-bit register that holds the 32 most significant bits (MSBs).
T0CLRI	8-bit register. Writing any value to this register clears the interrupt. Available only in 16-bit mode.
T0CON	Configuration MMR.

Table 144. Timer0 Value Register

Name	Address	Default Value	Access
T0VAL0	0xFFFF0304	0x00,	R
T0VAL1	0xFFFF0308	0x00	R

T0VAL0 and T0VAL1 are 16-bit and 32-bit registers that hold the 16 least significant bits and 32 most significant bits, respectively. T0VAL0 and T0VAL1 are 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.

Table 145. Timer0 Capture Register

Name	Address	Default Value	Access
T0CAP	0xFFFF0314	0x00	R

This is a 16-bit register that holds the 16-bit value captured by an enabled IRQ event; it is only available in 16-bit mode.

Table 146. Timer0 Control Register

Name	Address	Default Value	Access
T0CON	0xFFFF030C	0x00	R/W

The 17-bit MMR configures the mode of operation of Timer0.

Table 147. T0CON MMR Bit Designations

Bit	Value	Description
31:18		Reserved.
17		Event select bit. Set by the user to enable time capture of an event. Cleared by the user to disable time capture of an event.
16:12		Event select (ES) range, 0 to 17. The events are as described in the Timers section.
11		Reserved.
10:9		Clock select.
	00	Internal 32 kHz oscillator.
	01	UCLK.
	10	External 32 kHz crystal.
	11	HCLK.
8		Count up. Available only in 16-bit mode. Set by the user for Timer0 to count up. Cleared by the user for Timer0 to count down (default).
7		Timer0 enable bit. Set by the user to enable Timer0. Cleared by the user to disable Timer0 (default).
6		Timer0 mode. Set by the user to operate in periodic mode. Cleared by the 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		Prescaler.
	0000	Source clock/1 (default).
	0100	Source clock/16.
	1000	Source clock/256.
	1111	Source clock/32,768.

Table 148. Timer0 Load Registers

Name	Address	Default Value	Access
T0LD	0xFFFF0300	0x00	R/W

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

Table 149. Timer0 Clear Register

Name	Address	Default Value	Access
T0CLRI	0xFFFF0310	0x00	W

This 8-bit, write-only MMR is written (with any value) by user code to refresh (reload) Timer0.

TIMER1—GENERAL-PURPOSE TIMER

Timer1 is a 32-bit general-purpose timer, count down or count up, with a programmable prescaler. The prescaler source can be from the 32 kHz internal oscillator, the 32 kHz external crystal, the core clock, or from the undivided PLL clock output. This source can be scaled by a factor of 1, 16, 256, or 32,768. This gives a minimum resolution of 42 ns when operating at CD = 0, the core is operating at 41.78 MHz, and with a prescaler of 1.

The counter can be formatted as a standard 32-bit value or as hours:minutes:seconds:hundredths.

Timer1 has a capture register (T1CAP) that can be triggered by a selected IRQ's source initial assertion. When 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.

The Timer1 interface consists of five MMRs, as shown in Table 150.

Table 150. Timer1 Interface Registers

Register	Description
T1LD	32-bit register. Holds 32-bit unsigned integers. This register is read only.
T1VAL	32-bit register. Holds 32-bit unsigned integers.
T1CAP	32-bit register. Holds 32-bit unsigned integers. This register is read only.
T1CLRI	8-bit register. Writing any value to this register clears the Timer1 interrupt.
T1CON	Configuration MMR.

Note that if the part is in a low power mode, and Timer1 is clocked from the GPIO or low power oscillator source, then Timer1 continues to operate.

Timer1 reloads the value from T1LD either when Timer1 overflows or immediately when T1ICLR is written.

Table 151. Timer1 Load Registers

Name	Address	Default Value	Access
T1LD	0xFFFF0320	0x00000	R/W

T1LD is a 32-bit register that holds the 32-bit value that is loaded into the counter.

Table 152. Timer1 Clear Register

Name	Address	Default Value	Access
T1CLRI	0xFFFF032C	0x00	W

This 8-bit, write-only MMR is written (with any value) by user code to refresh (reload) Timer1.

Table 153. Timer1 Value Register

Name	Address	Default Value	Access
T1VAL	0xFFFF0324	0x0000	R

T1VAL is a 32-bit register that holds the current value of Timer1.

Table 154. Timer1 Capture Register

Name	Address	Default Value	Access
T1CAP	0xFFFF0330	0x00	R

This is a 32-bit register that holds the 32-bit value captured by an enabled IRQ event.

Table 155. Timer1 Control Register.

Name	Address	Default Value	Access
T1CON	0xFFFF0328	0x0000	R/W

This 32-bit MMR configures the mode of operation of Timer1.

Table 156. T1CON MMR Bit Designations

Bit	Value	Description
31:24		8-bit postscaler.
23		Enable write to postscaler.
22:20		Reserved.
19		Postscaler compare flag.
18		T1 interrupt generation selection flag.
17		Event select bit.
	1	Set by the user to enable time capture of an event.
	0	Cleared by the user to disable time capture of an event.
16:12		Event select range, 0 to 17. The events are as described in the Timers section.
11:9		Clock select.
	000	Internal 32 kHz oscillator (default).
	001	Core clock.
	010	UCLK.
	011	P0.6.
8		Count up.
	1	Set by the user for Timer1 to count up.
	0	Cleared by the user for Timer1 to count down (default).
7		Timer1 enable bit.
	1	Set by the user to enable Timer1.
	0	Cleared by the user to disable Timer1 (default).
6		Timer1 mode.
	1	Set by the user to operate in periodic mode.
	0	Cleared by the 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 hours.
	11	Hr:min:sec:hundredths: 255 hours to 0 hours.
3:0		Prescaler.
	0000	Source clock/1 (default).
	0100	Source clock/16.
	1000	Source clock/256.
	1111	Source clock/32,768.

TIMER2—WAKE-UP TIMER

Timer2 is a 32-bit wake-up timer, count down or count up, with a programmable prescaler. The prescaler is clocked directly from 1 of 4 clock sources, including the core clock (default selection), the internal 32.768 kHz oscillator, the external 32.768 kHz watch crystal, or the PLL undivided clock. The selected clock source can be scaled by a factor of 1, 16, 256, or 32,768. The wake-up timer continues to run when the core clock is disabled. This gives a minimum resolution of 22 ns when the core is operating at 41.78 MHz and with a prescaler of 1. Capture of the current timer value is enabled if the Timer2 interrupt is enabled via IRQEN[4] (see Table 128).

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

Timer2 reloads the value from T2LD either when Timer2 overflows or immediately when T2ICLR is written.

The Timer2 interface consists of four MMRs, shown in Table 157.

Table 157. Timer2 Interface Registers

Register	Description
T2LD	32-bit register. Holds 32-bit unsigned integers.
T2VAL	32-bit register. Holds 32-bit unsigned integers. This register is read only.
T2CLRI	8-bit register. Writing any value to this register clears the Timer2 interrupt.
T2CON	Configuration MMR.

Table 158. Timer2 Load Registers

Name	Address	Default Value	Access
T2LD	0xFFFF0340	0x00000	R/W

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

Table 159. Timer2 Clear Register

Name	Address	Default Value	Access
T2CLRI	0xFFFF034C	0x00	W

This 8-bit write-only MMR is written (with any value) by user code to refresh (reload) Timer2.

Table 160. Timer2 Value Register

Name	Address	Default Value	Access
T2VAL	0xFFFF0344	0x0000	R

T2VAL is a 32-bit register that holds the current value of Timer2.

Table 161. Timer2 Control Register

Name	Address	Default Value	Access
T2CON	0xFFFF0348	0x0000	R/W

This 32-bit MMR configures the mode of operation for Timer2.

Table 162. T2CON MMR Bit Designations

Bit	Value	Description
31:11		Reserved.
10:9		Clock source select.
	00	Internal 32.768 kHz oscillator (default).
	01	Core clock.
	10	External 32.768 kHz watch crystal.
	11	UCLK.
8		Count up.
	1	Set by the user for Timer2 to count up.
	0	Cleared by the user for Timer2 to count down (default).
7		Timer2 enable bit.
	1	Set by the user to enable Timer2.
	0	Cleared by the user to disable Timer2 (default).
6		Timer2 mode.
	1	Set by the user to operate in periodic mode.
	0	Cleared by the 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 hours.
	11	Hr:min:sec:hundredths. 255 hours to 0 hours.
3:0		Prescaler.
	0000	Source clock/1 (default).
	0100	Source clock/16.
	1000	Source clock/256 (this setting should be used in conjunction with Timer2 Format 10 and Format 11).
	1111	Source clock/32,768.

TIMER3—WATCHDOG TIMER

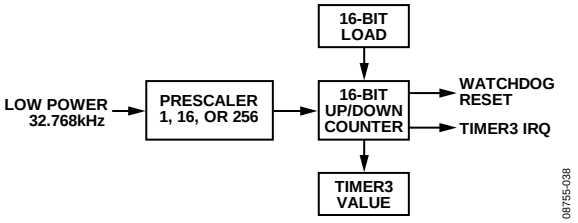


Figure 37. Timer3 Block Diagram

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 T3ICLR 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 32.768 kHz oscillator and can be scaled by a factor of 1, 16, or 256. Timer3 also features a capture facility that allows capture of the current timer value if the Timer2 interrupt is enabled via IRQEN[5].

Watchdog Mode

Watchdog mode is entered by setting T3CON[5]. Timer3 decrements from the timeout value present in the T3LD register until 0. The maximum timeout is 512 seconds, using the maximum prescalar/256 and full scale in T3LD.

User software should only configure a minimum timeout period of 30 milliseconds. This is to avoid any conflict with Flash/EE memory page erase cycles, requiring 20 ms 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 T3ICLR 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 cannot 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.

Timer3 is automatically halted during JTAG debug access and only recommences counting after JTAG has relinquished control of the ARM7 core. By default, Timer3 continues to count during power-down. This can be disabled by setting Bit 0 in T3CON. It is recommended that the default value is used, that is, the watchdog timer continues to count during power-down.

Timer3 Interface

Timer3 interface consists of four MMRS as shown in the table below.

Table 163. Timer3 Interface Registers

Register	Description
T3CON	The configuration MMR.
T3LD	6-bit registers (Bit 0 to Bit15); holds 16-bit unsigned integers.
T3VAL	6-bit registers (Bit 0 to Bit 15); holds 16-bit unsigned integers. This register is read only.
T3ICLR	8-bit register. Writing any value to this register clears the Timer3 interrupt in normal mode or resets a new timeout period in watchdog mode.

Table 164. Timer3 Load Register

Name	Address	Default Value	Access
T3LD	0xFFFF0360	0x03D7	R/W

This 16-bit MMR holds the Timer3 reload value.

Table 165. Timer3 Value Register

Name	Address	Default Value	Access
T3VAL	0xFFFF0364	0x03D7	R

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

Table 166. Timer3 Clear Register

Name	Address	Default Value	Access
T3CLRI	0xFFFF036C	0x00	W

This 8-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.

Table 167. Timer3 Control Register

Name	Address	Default Value	Access
T3CON	0xFFFF0368	0x00	R/W once only

The 16-bit MMR configures the mode of operation of Timer3 as is described in detail in Table 168.

Table 168. T3CON MMR Bit Designations

Bit	Value	Description
16:9		These bits are reserved and should be written as 0s by user code.
8		Count up/down enable.
	1	Set by user code to configure Timer3 to count up.
	0	Cleared by user code to configure Timer3 to count down.
7		Timer3 enable.
	1	Set by user code to enable Timer3.
	0	Cleared by user code to disable Timer3.
6		Timer3 operating mode.
	1	Set by user code to configure Timer3 to operate in periodic mode.
	0	Cleared by user to configure Timer3 to operate in free-running mode.
5		Watchdog timer mode enable.
	1	Set by user code to enable watchdog mode.
	0	Cleared by user code to disable watchdog mode.
4		Secure clear bit.
	1	Set by the user to use the secure clear option.
	0	Cleared by the user to disable the secure clear option by default.
3:2		Timer3 clock (32.768 kHz) prescaler.
	00	Source clock/1 (default).
	01	Reserved.
	10	Reserved.
	11	Reserved.
1		Watchdog timer IRQ enable.
	1	Set by the user code to produce an IRQ instead of a reset when the watchdog reaches 0.
	0	Cleared by the user code to disable the IRQ option.
0		PD_OFF.
	1	Set by user code to stop Timer3 when the peripherals are powered down via Bits[6:4] in the POWCON MMR.
	0	Cleared by user code to enable Timer3 when the peripherals are powered down via Bits[6:4] in the POWCON MMR.

Secure Clear Bit (Watchdog Mode Only)

The secure clear bit(T3CPM[4]) is provided for a higher level of protection. When set, a specific sequential value must be written to T3CLRI to avoid a watchdog reset. The value is a sequence generated by the 8-bit linear feedback shift register (LFSR) polynomial = X8 + X6 + X5 + X + 1 (see Figure 38).

The initial value or seed is written to T3ICLR before entering watchdog mode. After entering watchdog mode, a write to T3CLRI must match this expected value. If it matches, the LFSR is advanced to the next state when the counter reload happens. If it fails to match the expected state, a reset is immediately generated, even if the count has not yet expired.

The value 0x00 should not be used as an initial seed due to the properties of the polynomial. The value 0x00 is always guaranteed to force an immediate reset. The value of the LFSR cannot be read; it must be tracked/generated in software.

Example of a sequence:

1. Enter initial seed, 0xAA, in T3CLRI before starting Timer3 in watchdog mode.
2. Enter 0xAA in T3CLRI; Timer3 is reloaded.
3. Enter 0x37 in T3CLRI; Timer3 is reloaded.
4. Enter 0x6E in T3CLRI; Timer3 is reloaded.
5. Enter 0x66. 0xDC was expected; the watchdog resets the chip.

TIMER4—GENERAL-PURPOSE TIMER

Timer4 is a 32-bit general-purpose timer, count down or count up, with a programmable prescalar. The prescalar source can be the 32 kHz oscillator, the core clock, or the PLL undivided output. This source can be scaled by a factor of 1, 16, 256, or 32,768. This gives a minimum resolution of 42 ns when operating at CD = 0, the core is operating at 41.78 MHz, and with a prescalar of 1 (ignoring external GPIO).

The counter can be formatted as a standard 32-bit value or as hours:minutes:seconds:hundredths.

Timer4 has a capture register (T4CAP) that can be triggered by a selected IRQ's source initial assertion. When 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.

The Timer4 interface consists of five MMRS.

- T4LD, T4VAL, and T4CAP are 32-bit registers and hold 32-bit unsigned integers. T4VAL and T4CAP are read only.
- T4ICLR is an 8-bit register. Writing any value to this register clears the Timer1 interrupt.
- T4CON is the configuration MMR.

Note that if the part is in a low power mode, and Timer4 is clocked from the GPIO or oscillator source, then Timer4 continues to operate.

Timer4 reloads the value from T4LD either when Timer 4 overflows or immediately when T4CLRI is written.

Table 169. Timer4 Load Registers

Name	Address	Default Value	Access
T4LD	0xFFFF0380	0x00000	R/W

T4LD is a 32-bit register, which holds the 32-bit value that is loaded into the counter.

Table 170. Timer4 Clear Register

Name	Address	Default Value	Access
T4CLRI	0xFFFF038C	0x00	W

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

Table 171. Timer4Value Register

Name	Address	Default Value	Access
T4VAL	0xFFFF0384	0x0000	R

T4VAL is a 32-bit register that holds the current value of Timer4.

Table 172. Timer4 Capture Register

Name	Address	Default Value	Access
T4CAP	0xFFFF0390	0x00	R

This is a 32-bit register that holds the 32-bit value captured by an enabled IRQ event.

Table 173. Timer4 Control Register

Name	Address	Default Value	Access
T4CON	0xFFFF0388	0x0000	R/W

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

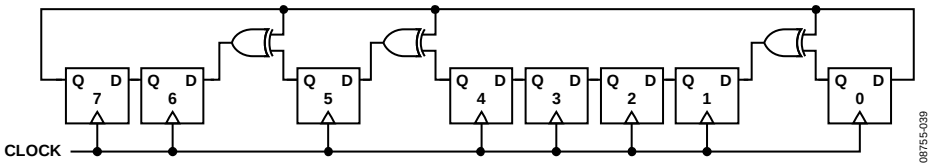
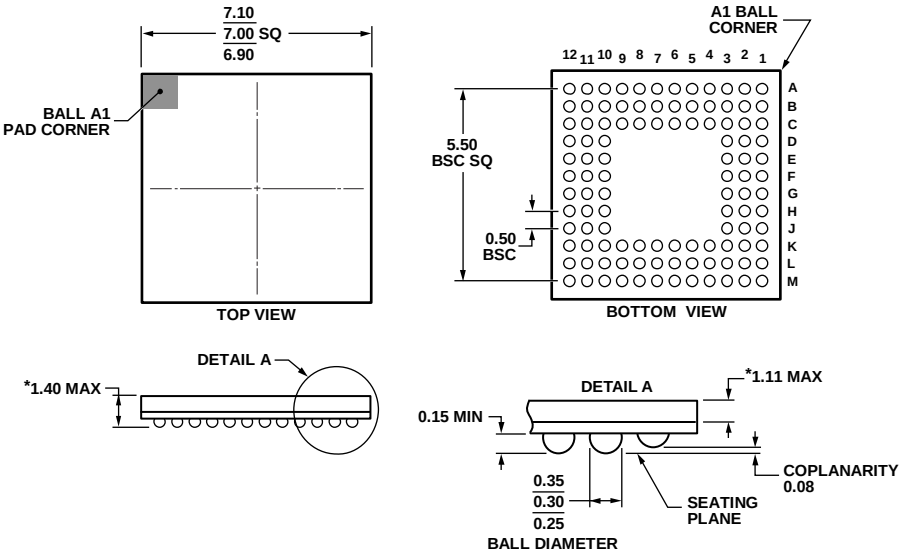


Figure 38. 8-Bit LFSR

Table 174. T4CON MMR Bit Designations

Bit	Value	Description
31:18		Reserved. Set by user to 0.
17		Event select bit.
	1	Set by the user to enable time capture of an event.
	0	Cleared by the user to disable time capture of an event.
16:12		Event select range, 0 to 31. The events are as described in the Timers section.
11:9		Clock select.
	000	32.768 kHz oscillator.
	001	Core clock.
	010	UCLK.
	011	UCLK.
8		Count up.
	1	Set by the user for Timer4 to count up.
	0	Cleared by the user for Timer4 to count down (default).
7		Timer4 enable bit.
	1	Set by the user to enable Timer4.
	0	Cleared by the user to disable Timer4 (default).
6		Timer4 mode.
	1	Set by the user to operate in periodic mode.
	0	Cleared by the 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 hours.
	11	Hr:min:sec:hundredths: 255 hours to 0 hours.
3:0		Prescaler.
	0000	Source clock/1 (default).
	0100	Source clock/16.
	1000	Source clock/256.
	1111	Source clock/32,768.

OUTLINE DIMENSIONS



*COMPLIANT WITH JEDEC STANDARDS MO-195-BD WITH EXCEPTION TO PACKAGE HEIGHT AND THICKNESS.

Figure 39. 108-Ball Chip Scale Package Ball Grid Array [CSP_BGA]
(BC-108-4)
Dimensions shown in millimeters

ORDERING GUIDE

Model[1]	Temperature Range	Package Description	Package Option
ADuC7122BBCZ	−10°C to +95°C	108-Ball Chip Scale Package Ball Grid Array [CSP_BGA]	BC-108-4
ADuC7122BBCZ-RL	−10°C to +95°C	108-Ball Chip Scale Package Ball Grid Array [CSP_BGA], 13″Tape and Reel	BC-108-4

[1] Z = RoHS Compliant Part.

NOTES

NOTES

NOTES

I²C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).

D08755-0-4/10(0)