WINSOURCE
ELECTRONICS

# THE DATASHEET OF

# ADUC7121BBCZ

**ANALOG DEVICES**

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

Data Sheet

# ADuC7121

## FEATURES

**Analog input/output**
- 9-channel, 12-bit, 1 MSPS ADC
  - 2 differential pairs with input PGA
  - 7 general-purpose inputs (differential or single-ended)
- Fully differential and single-ended modes
- 0 V to $V_{REF}$ analog input range
- 5 low noise current digital-to-analog converters (IDACs)
  - 250 mA, 200 mA, 45 mA, 80 mA, 20 mA
- 4 × 12-bit voltage output DACs
- On-chip voltage reference
- On-chip temperature sensor

**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/O
- 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 CSP_BGA
- Fully specified for −10°C to +95°C operation

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

## APPLICATIONS

**Optical modules—tunable laser**

## FUNCTIONAL BLOCK DIAGRAM

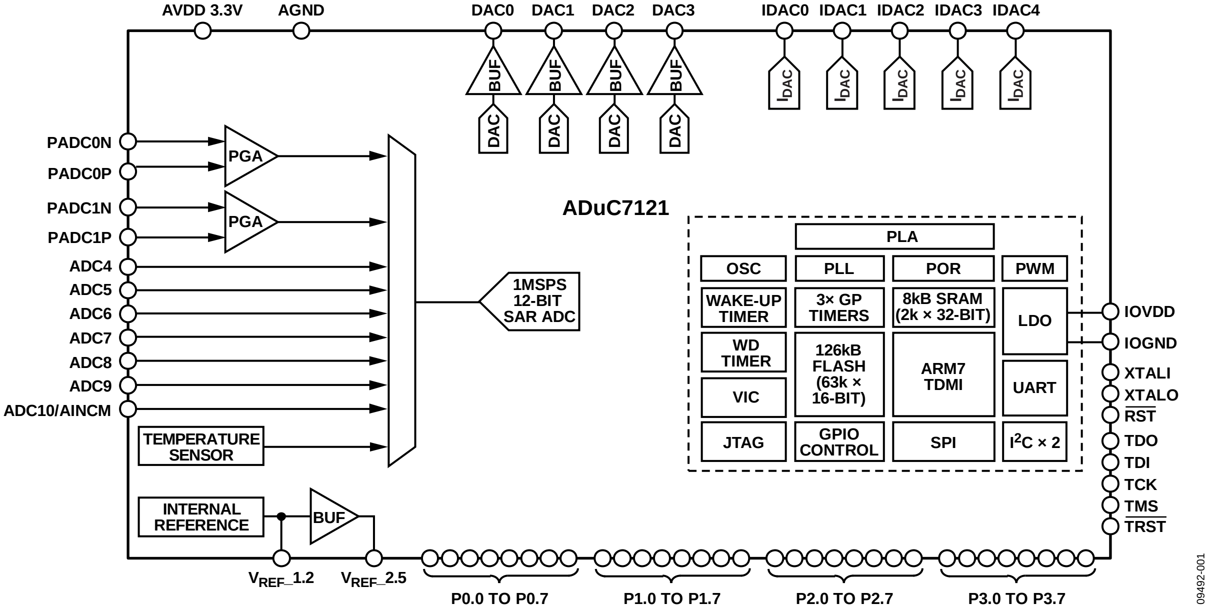

*Figure 1.*

**Rev. C** [Document Feedback]

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700

Technical Support
www.analog.com

# TABLE OF CONTENTS

## REVISION HISTORY

**11/14—Rev. B to Rev. C**

Changes to Flash/EE Memory Section.........................................23
Changes to PADC0x/PADC1x Pins Section...............................28
Changes to Flash/EE Memory Section and Serial Downloading (In-Circuit Programming) Section...............................................35
Changes to Flash/EE Memory Security Section .........................36
Changes to Table 41, Table 42, and FEE0STA Default Value ....37
Changes to I²C Peripherals Section ..............................................62
Changes to Table 89 .......................................................................63
Changes to Table 96 .......................................................................67
Changes to Table 97 .......................................................................68
Added Hardware Design Consideration Section........................95

**3/13—Rev. A to Rev. B**

Changes to Table 9 .........................................................................19
Changes to Table 11 .......................................................................24
Changes to Reset Operation Section ............................................40
Added RSTCFGKEY0 Register and RSTCFGKEY1 Register Sections............................................................................................40
Added Table 49 and Table 50; Renumbered Sequentially .........40
Changed $I_{REF}$ = 370.37 µA to $I_{REF}$ = 380 µA.................................44
Changes to Figure 32 .....................................................................44
Changes to Table 57 .......................................................................45
Changes to Table 64 .......................................................................46

**3/12—Rev. 0 to Rev. A**

Changed IDAC2 Full-Scale Output to 45 mA...........Throughout
Changed IDAC3 Full-Scale Output to 80 mA...........Throughout
Added BUFFER $V_{REF}$ Out Parameter to Table 1, Specifications Section ......................................................................6
Changes to IDAC3 and IDAC2, Full-Scale Output Parameter, Table 1, Specifications Section........................................................6
Changes to Flash/EE Memory Section..........................................35
Changes to PADC0x/PADC1x Pins Section...............................28
Added New Figure 15, Renumbered Sequentially......................28
Changes to Temperature Sensor Section ....................................29
Changes to Current Output DACs (IDAC) Section...................43
Changes to Table 58 .......................................................................45
Change to External Interrupts (IRQ0 to IRQ5) Section Heading ...........................................................................................84
Added Hour:Minute:Second:1/128 Format Section...................86
Added New Table 122, Renumbered Sequentially......................86
Changes to Timer3 Control Register Section.............................91
Changes to Table 130 .....................................................................92

**1/11—Revision 0: Initial Version**

# GENERAL DESCRIPTION

The ADuC7121 is a fully integrated, 1 MSPS, 12-bit data acquisition system incorporating a high performance multichannel ADC, 16-bit/32-bit MCU, and Flash®/EE memory on a single chip.

The ADC consists of up to seven single-ended inputs and two extra differential input pairs. The two differential pair inputs can be routed through a programmable gain amplifier (PGA). 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 voltage comparator complete the ADC peripheral set.

The ADuC7121 provides five current output digital-to-analog converters (DACs). The current sources (five current DACs) feature low noise and low drift high-side current output at 11-bit resolution. The five IDACs are as follows: IDAC0 with 250 mA full-scale (FS) output, IDAC1 with 200 mA FS output, IDAC2 with 45 mA FS output, IDAC3 with 80 mA FS output, and IDAC4 with 20 mA FS output.

The ADuC7121 also contains four voltage output digital-to-analog converters (DACs). The DAC output range is programmable to one of three voltage ranges.

The devices operate from an on-chip oscillator and a PLL generating an internal high frequency clock of 41.78 MHz (UCLK). 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, which offers up to 41 MIPS peak performance. Eight kB of SRAM and 126 kB of nonvolatile Flash/EE memory are provided on chip. The ARM7TDMI core views all memory and registers as a single linear array.

On-chip factory firmware supports in-circuit serial download via the $I^2C$ serial interface port; 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 device operates from 3.0 V to 3.6 V, and it is specified over an industrial temperature range of −10°C to +95°C. The IDACs are powered from a separate 2 V input power supply. When operating at 41.78 MHz, the power dissipation is typically 120 mW. The ADuC7121 is available in a 108-ball chip scale package ball grid array [CSP_BGA].

# SPECIFICATIONS

$AV_{DD} = IOV_{DD} = 3.0$ V to 3.6 V, $PV_{DD} = 2.0$ V ± 5%, $V_{REF} = 2.5$ 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 and PADC1 channels |
| Differential Nonlinearity[3, 4] | | ±0.5 | +1.4/−0.99 | LSB | 2.5 V internal reference, guaranteed monotonic |
| DC Code Distribution | | 1 | | LSB | ADC input is a dc voltage |
| ENDPOINT ERRORS[5] | | | | | Internally unbuffered channels |
| Offset Error | | | | | |
| All Channels Except IDACx Channels | | ±2 | ±5 | LSB | |
| IDACx Channels Only | | 1 | | % of full scale | |
| 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 38 |
| Single-Ended Mode | | | 0 to $V_{REF}$ | V | Buffer bypassed |
| | 0.15 | | $AV_{DD} - 1.5$ | V | Buffer enabled |
| Leakage Current | | ±0.2 | ±1 | µA | |
| Input Capacitance | | 20 | | pF | During ADC acquisition buffer bypassed |
| | | 20 | | pF | During ADC acquisition buffer enabled |
| PADC0x INPUT | | | | | 28.3 kΩ resistor, PGA gain = 3, acquisition time = 3.2 µs, pseudo differential mode |
| Full-Scale Input Range | 20 | | 1000 | µA | |
| Input Leakage at PADC0x[4] | | 0.15 | 2 | nA | |
| Resolution | 11 | | | Bits | 0.1% accuracy, 5 ppm external resistor for current to voltage |
| 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 | |
| PADC0x Compliant Range | 0.1 | | $AV_{DD} - 1.2$ | V | |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| PADC1x INPUT | | | | | 53.5 kΩ resistor, PGA gain = 3 |
| Full-Scale Input Range | 10.6 | | 700 | µA | |
| Input Leakage at PADC1x[4] | | 0.15 | 2 | nA | |
| Resolution | 11 | | | Bits | 0.1% accuracy, 5 ppm external resistor for current to voltage |
| 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 | |
| PADC1x-Compliant Range | 0.1 | | $AV_{DD}$ − 1.2 | V | |
| 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 | |
| BUFFER $V_{REF}$ OUT | | | | | |
| BUF_VREFx | | ±80 | | mV | |
| EXTERNAL REFERENCE INPUT | | | | | |
| Input Voltage Range | 1.2 | | $AV_{DD}$ | V | |
| IDAC CHANNEL SPECIFICATIONS | | | | | |
| Voltage Compliance Range | 0.4 | | 1.6 | V | Output voltage compliance |
| Voltage Compliance Range, IDAC0 | −0.2 | | +1.6[8] | V | For IDAC0 channel only, linearity not guaranteed below 0 V |
| REFERENCE CURRENT GENERATOR | | | | | |
| Reference Current | | 0.38 | | mA | Using internal reference, 0.1% 5 ppm 3.16 kΩ external resistor |
| Temperature Coefficient | | 25 | | ppm/°C | Using internal reference |
| Short-Circuit Detection | | 1 | | mA | |
| Overheat Shutdown | | 135 | | °C | Junction temperature |
| RESOLUTION | | | | | |
| | | 11 | | Bits | Guaranteed monotonic |
| FULL-SCALE OUTPUT | | | | | |
| IDAC4 | | 20 | | mA | |
| IDAC3 | | 80 | | mA | |
| IDAC2 | | 45 | | mA | |
| IDAC1 | | 200 | | mA | |
| IDAC0 | | 250 | | mA | |
| Integral Nonlinearity | | ±2 | | LSB | 11-bit mode |
| Noise Current | | 20 | | µA | RMS value, bandwidth 20 Hz to 10 MHz |
| Full-Scale Error | | ±3 | | % | $V_{OUT}$ = 1.6 V |
| Full-Scale Error Drift | | 50 | | ppm/°C | Internal $V_{REF}$, 5 ppm external resistor |
| Zero-Scale Error | | | | | Pull-down switch off, $V_{OUT}$ = 0 V |
| IDAC4 Channel | | | ±30 | µA | |
| IDAC3 Channel | | | +42/−70 | µA | |
| IDAC2 Channel | | | +70/−110 | µA | |
| IDAC1 Channel | | | ±240 | µA | |
| IDAC0 Channel | | | ±250 | µA | Output range 0.4 V to 1.6 V |
| | | | +580/−430 | µA | Output range −0.2 V to +1.6 V |
| Settling Time | | 1 | | ms | To 0.1% |
| Signal Bandwidth | | 20 | | kHz | |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| LINE REGULATION | | | | | Measured with full-scale current load on current DACs |
| IDAC4 | | 10 | | μA/V | |
| IDAC3 | | 22.5 | | μA/V | |
| IDAC2 | | 40 | | μA/V | |
| IDAC1 | | 100 | | μA/V | |
| IDAC0 | | 750 | | μA/V | |
| LOAD REGULATION | | | | | Measured with full-scale current load on current DACs |
| IDAC4 | | 10 | | μA/V | |
| IDAC3 | | 22.5 | | μA/V | |
| IDAC2 | | 40 | | μA/V | |
| IDAC1 | | 100 | | μA/V | |
| IDAC0 | | 750 | | μA/V | |
| ACPSRR[4] | 0.75% | | | % of full-scale/V | 10 kHz, percentage of each current DAC full-scale current per volt |
| | 6% | | | % of full-scale/V | 2.25 MHz, percentage of each current DAC full-scale current per volt |
| PULL-DOWN | | | | | |
| NMOS | | 100 | | mV | Drain 40 mA |
| Speed[4] | | | 10 | µs | Triggered by PLA, draw the pin voltage to 10% of its original value |
| Voltage DAC (VDAC) CHANNEL | | | | | $R_L$ = 5 kΩ, $C_L$ = 100 pF |
| DC Accuracy[9] | | | | | 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[10] | | ±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 | |
| DRIFT | | | | | |
| 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 | | 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) |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| TEMPERATURE SENSOR[11] | | | | | After user calibration |
| Voltage Output at 25°C | | 707 | | mV | |
| Voltage Temperature Coefficient | | −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 | |
| FLASH/EE MEMORY | | | | | |
| Endurance[12] | 10,000 | | | Cycles | |
| Data Retention[13] | 20 | | | Years | $T_J$ = 85°C |
| DIGITAL INPUTS | | | | | All digital inputs excluding XTALI and XTALO |
| Logic 1 Input Current | | ±0.2 | ±1 | µA | $V_{IH} = V_{DD}$ |
| 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[14] | | | 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 | Clock divider (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 (HCLK) = 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[15, 16] | | | | | |
| 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 |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| 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 | $T_A$ = 25°C |
| Additional Power Supply Currents | | | | | |
| ADC | | 2.7 | | mA | @1 MSPS |
| IDAC | | 21 | | mA | All current DACs (IDACs) on |
| DAC | | 250 | | µA | per VDAC |
| ESD TESTS | | | | | 2.5 V reference, $T_A$ = 25°C |
| HBM Passed Up to | | | 4 | kV | |
| FICDM Passed Up to | | | 0.5 | kV | |

[1] All ADC channel specifications are guaranteed during normal MicroConverter core operation.

[2] Apply to all ADC input channels.

[3] Measured using the factory set default values in the ADC offset register (ADCOF) and gain coefficient register (ADCGN).

[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 24. 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 section).

[6] The input signal can be centered on any dc common-mode voltage ($V_{CM}$) provided that this value is within the ADC voltage input range specified.

[7] $V_{REF}$ calibration and trimming are performed under the following conditions: the core is operating in normal mode CD = 0, the ADC is on, the current DACs are on, and all VDACs are on. $V_{REF}$ accuracy may vary under other operating conditions.

[8] The PVDD_IDAC0 pad voltage must be at least 300 mV greater than the IDAC0 pad voltage. These voltages are measured via the PVDD0 and IDAC0 channels of the ADC. This allows the IDAC0 pin to be pulled up to 1.7 V provided that this 300 mV differential voltage is maintained between the pads. This may require the PVDD_IDAC0 being supplied with a voltage greater than 2.0 V. The 2.1 V maximum PVDD_IDACx rating must not be exceeded.

[9] DAC linearity is calculated using a reduced code range of 100 to 3995.

[10] DAC gain error is calculated using a reduced code range of 100 to internal 2.5 V $V_{REF}$.

[11] Die temperature.

[12] Endurance is qualified as per JEDEC Standard 22 Method A117 and measured at −40°C, +25°C, +85°C, and +125°C.

[13] Retention lifetime equivalent at junction temperature ($T_J$) = 85°C as per JEDEC Standard 22 Method A117. Retention lifetime derates with junction temperature.

[14] Test carried out with a maximum of eight I/Os set to a low output level.

[15] Power supply current consumption is measured in normal, pause, and sleep modes under the following conditions: normal mode using a 3.6 V supply, pause mode using a 3.6 V supply, and sleep mode using 3.6 V supply.

[16] $IOV_{DD}$ power supply current increases typically by 2 mA during a Flash/EE erase cycle.

## TIMING SPECIFICATIONS

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

| Parameter | Description | Slave Min | Slave Max | Master Typ | Unit |
|---|---|---|---|---|---|
| $t_L$ | SCL low pulse width | 200 | | 1360 | ns |
| $t_H$ | SCL high pulse width | 100 | | 1140 | ns |
| $t_{SHD}$ | Start condition hold time | 300 | | 251,350 | 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 | | 12.51350 | ns |
| $t_{PSU}$ | Stop condition setup time | 100 | | 400 | ns |
| $t_{BUF}$ | Bus-free time between a stop condition and a start condition | 1.3 | | | µs |
| $t_R$ | Rise time for both SCL and SDA | | 300 | 200 | ns |
| $t_F$ | Fall time for both SCL and SDA | | 300 | | ns |
| $t_{SUP}$ | Pulse width of spike suppressed | | 50 | | ns |

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

| Parameter | Description | Slave Min | Slave 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 |

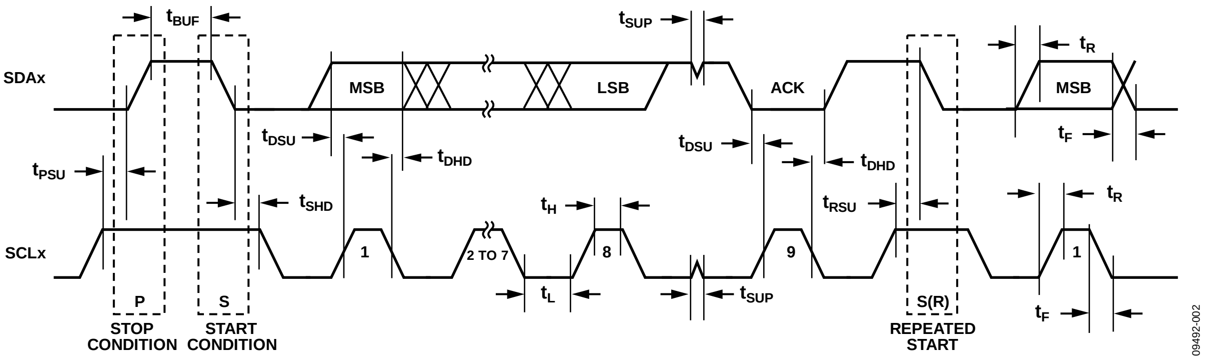

*Figure 2. I²C-Compatible Interface Timing*

### Table 4. SPI Master Mode Timing (Phase Mode = 1)

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

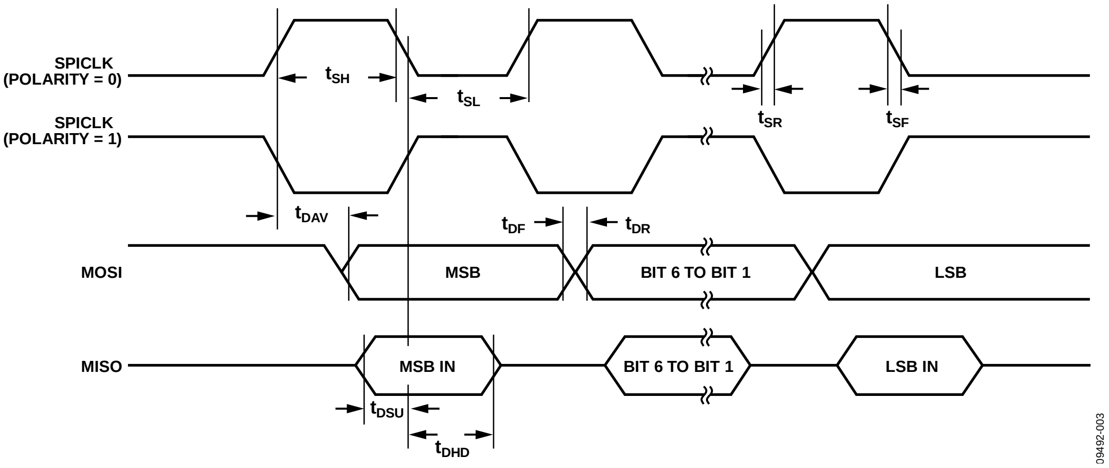

*Figure 3. SPI Master Mode Timing (Phase Mode = 1)*

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

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

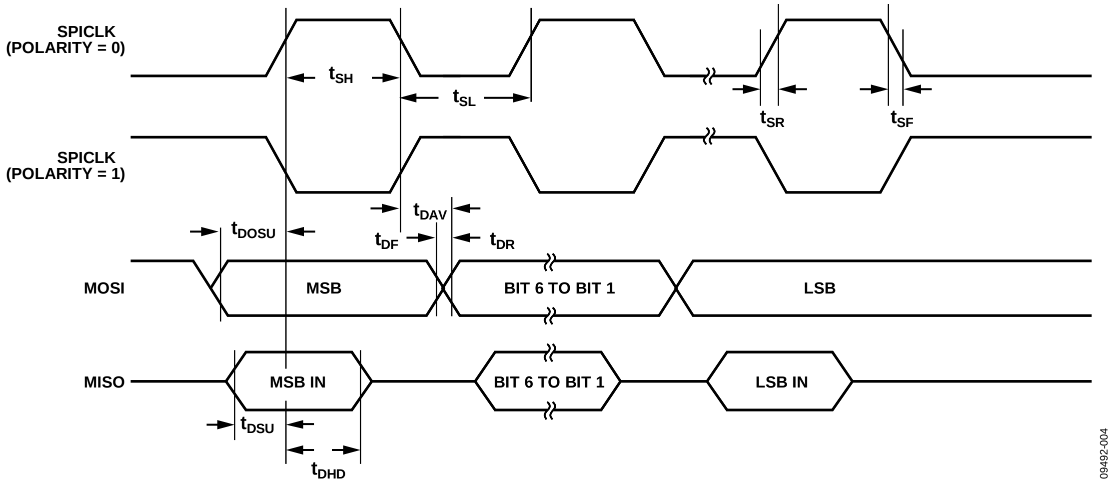

Figure 4. SPI Master Mode Timing (Phase Mode = 0)

### Table 6. SPI Slave Mode Timing (Phase Mode = 1)

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

[1] $\overline{CS}$ is the $\overline{CS}$ (SPI slave select input) function of the multifunction Pin F3.

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

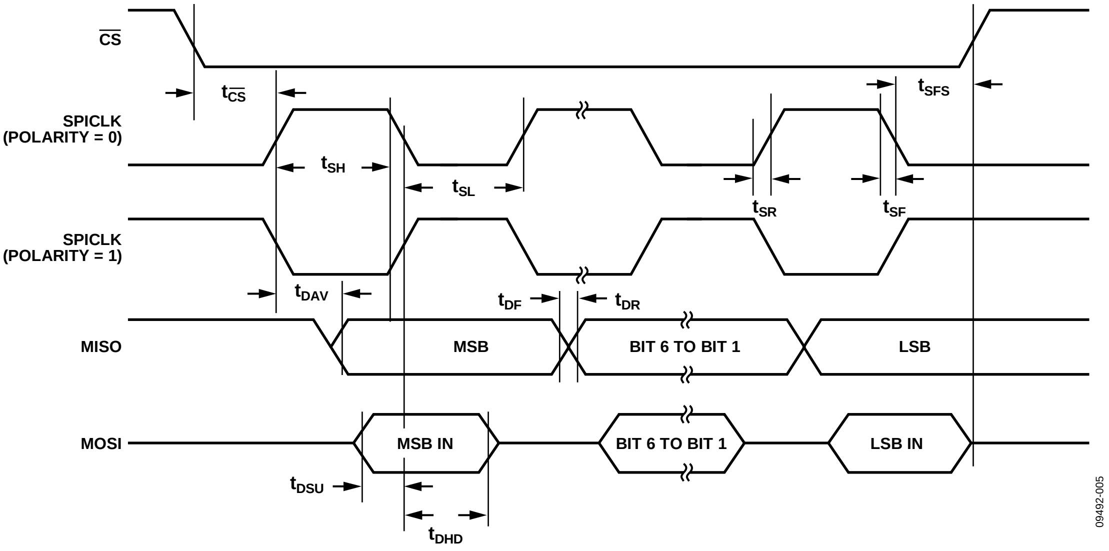

Figure 5. SPI Slave Mode Timing (Phase Mode = 1)

**Table 7. SPI Slave Mode Timing (Phase Mode = 0)**

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

[1] $\overline{CS}$ is the $\overline{CS}$ (SPI slave select input) function of the multifunction Pin F3.

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

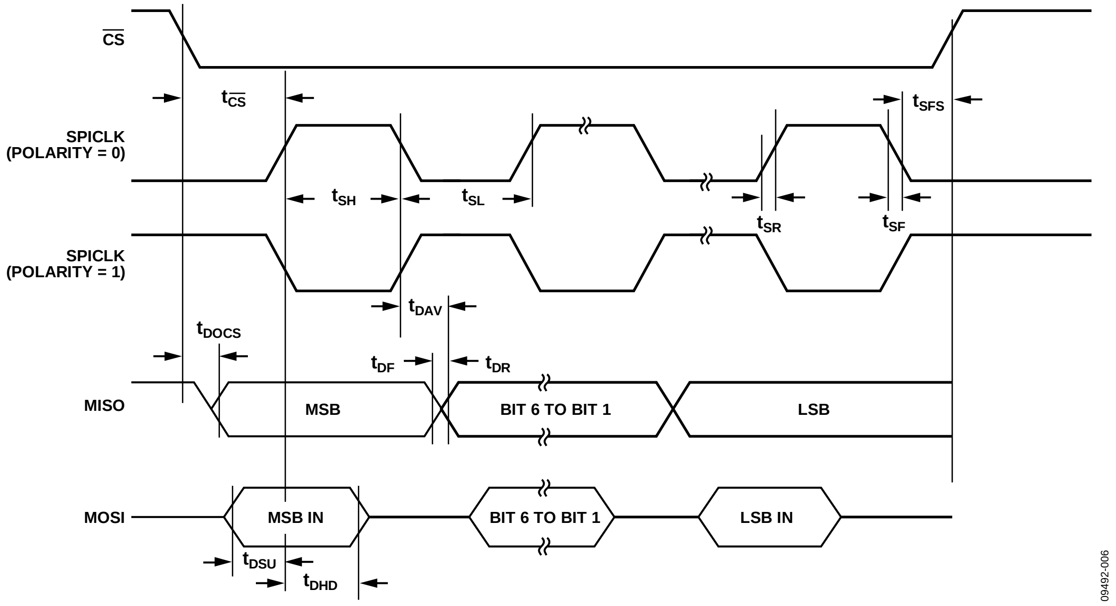

*Figure 6. SPI Slave Mode Timing (Phase Mode = 0)*

# ABSOLUTE MAXIMUM RATINGS

AGND = 0 V, $T_A$ = 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, Industrial | −10°C to +95°C |
| Storage Temperature Range | −65°C to +150°C |
| Junction Temperature | 150°C |
| $\theta_{JA}$ 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

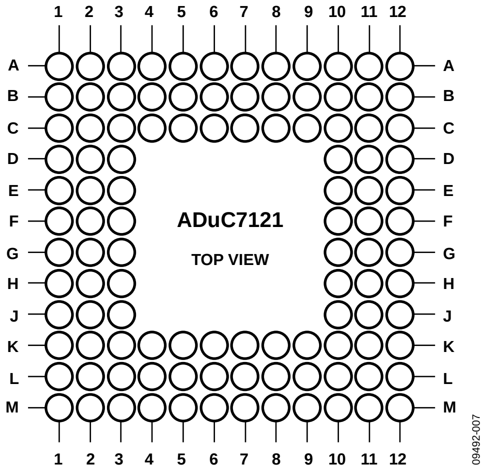

*Figure 7. Pin Configuration*

**Table 9. Pin Function Descriptions**

| Pin No. | Mnemonic | Type[1] | Description |
|---|---|---|---|
| C12 | $\overline{\text{RST}}$ | I | Reset Input (Active Low). |
| D11 | P0.0/SCL0/PLAI[5] | I/O | General-Purpose Input and Output Port 0.0 (P0.0).<br>$I^2C$ Interface Serial Clock for $I^2C0$ (SCL0).<br>Programmable Logic Array for Input Element 5 (PLAI[5]). |
| E11 | P0.1/SDA0/PLAI[4] | I/O | General-Purpose Input and Output Port 0.1 (P0.1).<br>$I^2C$ Interface Serial Data for $I^2C0$ (SDA0).<br>Programmable Logic Array for Input Element 4 (PLAI[4]). |
| C3 | P0.2/SPICLK/$\text{ADC}_{\text{BUSY}}$/PLAO[13] | I/O | General-Purpose Input and Output Port 0.2 (P0.2).<br>SPI Clock (SPICLK).<br>Status of the ADC ($\text{ADC}_{\text{BUSY}}$).<br>Programmable Logic Array for Output Element 13 (PLAO[13]). |
| D3 | P0.3/MISO/PLAO[12]/SYNC | I/O | General-Purpose Input and Output Port 0.3 (P0.3).<br>SPI Master In Slave Out (MISO).<br>Programmable Logic Array for Output Element 12 (PLAO[12]).<br>Synchronous Reset (SYNC). Input to reset synchronously the PWM counters using an external source. |
| E3 | P0.4/MOSI/PLAI[11]/TRIP | I/O | General-Purpose Input and Output Port 0.4 (P0.4).<br>SPI Master Out Slave In (MOSI).<br>Programmable Logic Array for Input Element 11 (PLAI[11]).<br>PWM Trip Interrupt (TRIP). The TRIP function of Pin E3 is the input that allows the PWM trip interrupt to be triggered. |
| F3 | P0.5/$\overline{\text{CS}}$/PLAI[10]/$\text{ADC}_{\overline{\text{CONVST}}}$ | I/O | General-Purpose Input and Output Port 0.5 (P0.5).<br>SPI Slave Select Input ($\overline{\text{CS}}$).<br>Programmable Logic Array for Input Element 10 (PLAI[10]).<br>ADC Conversions ($\text{ADC}_{\overline{\text{CONVST}}}$). The $\text{ADC}_{\overline{\text{CONVST}}}$ function of Pin F3 initiates the ADC conversions using the PLA or the timer output. |
| G3 | P0.6/$\overline{\text{MRST}}$/PLAI[2] | I/O | General-Purpose Input and Output Port 0.6 (P0.6).<br>Power On Reset Output ($\overline{\text{MRST}}$).<br>Programmable Logic Array for Input Element 2 (PLAI[2]). |
| G10 | P0.7/$\overline{\text{TRST}}$/PLAI[3] | I/O | General-Purpose Input and Output Port 0.7 (P0.7).<br>JTAG Test Port Input, Test Reset ($\overline{\text{TRST}}$). Debug and download access.<br>Programmable Logic Array for Input Element 3 (PLAI[3]). |

| Pin No. | Mnemonic | Type[1] | Description |
|---|---|---|---|
| C2 | P1.0/SIN/SCL1/PLAI[7] | I/O | General-Purpose Input and Output Port 1.0 (P1.0).<br>Serial Input, Receive Data, UART (SIN).<br>$I^2C$ Interface Serial Clock for $I^2C1$ (SCL1).<br>Programmable Logic Array for Input Element 7 (PLAI[7]). |
| D2 | P1.1/SOUT/SDA1/PLAI[6] | I/O | General-Purpose Input and Output Port 1.1 (P1.1).<br>Serial Output, Transmit Data, UART (SOUT).<br>$I^2C$ Interface Serial Data for $I^2C1$ (SDA1).<br>Programmable Logic Array for Input Element 6 (PLAI[6]). |
| C10 | P1.2/TDI/PLAO[15] | DI | General-Purpose Input and Output Port 1.2 (P1.2).<br>JTAG Test Port Input, Test Data In (TDI). The TDI function of Pin C10 is for debug and download access.<br>Programmable Logic Array for Output Element 15 (PLAO[15]). |
| D10 | P1.3/TDO/PLAO[14] | DO | General-Purpose Input and Output Port 1.3 (P1.3).<br>JTAG Test Port Output, Test Data Out (TDO). The TDO function of Pin D10 is for debug and download access.<br>Programmable Logic Array for Output Element 14 (PLAO[14]). |
| H3 | P1.4/PWM1/ECLK/XCLK/PLAI[8] | I/O | General-Purpose Input and Output Port 1.4 (P1.4).<br>Pulse-Width Modulator 1 Output (PWM1).<br>Base System Clock Output (ECLK).<br>Base System Clock Input (XCLK).<br>Programmable Logic Array for Input Element 8 (PLAI[8]). |
| J3 | P1.5/PWM2/PLAI[9] | I/O | General-Purpose Input and Output Port 1.5 (P1.5).<br>Pulse-Width Modulator 2 Output (PWM2).<br>Programmable Logic Array for Input Element 9 (PLAI[9]). |
| B3 | P1.6/PLAO[5] | I/O | General-Purpose Input and Output Port 1.6 (P1.6).<br>Programmable Logic Array for Output Element 5 (PLAO[5]). |
| B2 | P1.7/PLAO[4] | I/O | General-Purpose Input and Output Port 1.7 (P1.7).<br>Programmable Logic Array for Output 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, Active High.<br>Programmable Logic Array for Input Element 13 (PLAI[13]). |
| G11 | P2.1/IRQ1/PLAI[12] | I/O | General-Purpose Input and Output Port 2.1 (P2.1)<br>External Interrupt Request 1, Active High (IRQ1).<br>Programmable Logic Array for Input Element 12 (PLAI[12]). |
| H11 | P2.2/PLAI[1] | I/O | General-Purpose Input and Output Port 2.2 (P2.2).<br>Programmable Logic Array for Input Element 1 (PLAI[1]). |
| J11 | P2.3/IRQ2/PLAI[14] | I/O | General-Purpose Input and Output Port 2.3 (P2.3).<br>External Interrupt Request 2, Active High (IRQ2).<br>Programmable Logic Array for Input Element 14 (PLAI[14]). |
| H10 | P2.4/PWM5/PLAO[7] | I/O | General-Purpose Input and Output Port 2.4 (P2.4).<br>Pulse-Width Modulator 5 Output (PWM5).<br>Programmable Logic Array for Output Element 7 (PLAO[7]). |
| J10 | P2.5/PWM6/PLAO[6] | I/O | General-Purpose Input and Output Port 2.5 (P2.5).<br>Pulse-Width Modulator 6 Output (PWM6).<br>Programmable Logic Array for Output Element 6 (PLAO[6]). |
| C1 | P2.6/IRQ3/PLAI[15] | I/O | General-Purpose Input and Output Port 2.6 (P2.6).<br>External Interrupt Request 3, Active High (IRQ3).<br>Programmable Logic Array for Input Element 15 (PLAI[15]). |
| C9 | P2.7/PLAI[0] | I/O | General-Purpose Input and Output Port 2.7 (P2.7). |
| C4 | P3.0/PLAO[0] | I/O | General-Purpose Input and Output Port 3.0 (P3.0).<br>Programmable Logic Array for Output Element 0 (PLAO[0]). |
| C11 | P3.1/PLAO[1] | I/O | General-Purpose Input and Output Port 3.1 (P3.1).<br>Programmable Logic Array for Output Element 1 (PLAO[1]). |

| Pin No. | Mnemonic | Type[1] | Description |
|---|---|---|---|
| D1 | P3.2/IRQ4/PWM3/PLAO[2] | I/O | General-Purpose Input and Output Port 3.2 (P3.2).<br>External Interrupt Request 4, Active High (IRQ4).<br>Pulse-Width Modulator 3 Output (PWM3).<br>Programmable Logic Array for Output Element 2 (PLAO[2]). |
| E1 | P3.3/IRQ5/PWM4/PLAO[3] | I/O | General-Purpose Input and Output Port 3.3 (P3.3).<br>External Interrupt Request 5, Active High (IRQ5).<br>Pulse-Width Modulator 4 Output (PWM4).<br>Programmable Logic Array for Output Element 3 (PLAO[3]). |
| E2 | P3.4/PLAO[8] | I/O | General-Purpose Input and Output Port 3.4 (P3.4).<br>Programmable Logic Array for Output Element 8 (PLAO[8]). |
| F2 | P3.5/PLAO[9] | I/O | General-Purpose Input and Output Port 3.5 (P3.5).<br>Programmable Logic Array for Output Element 9 (PLAO[9]). |
| D12 | P3.6/PLAO[10] | I/O | General-Purpose Input and Output Port 3.6 (P3.6).<br>Programmable Logic Array for Output Element 10 (PLAO[10]). |
| E12 | P3.7/$\overline{\text{BM}}$/PLAO[11] | I/O | General-Purpose Input and Output Port 3.7 (P3.7).<br>Programmable Logic Array for Output Element 11 (PLAO[11]). |
| L8 | $V_{REF}$_2.5 | AI/O | 2.5 V Reference Output and External 2.5 V Reference Input. |
| L5 | $V_{REF}$_1.2 | AI/O | 1.2 V Reference Output and External 1.2 V Reference Input. Cannot be used to source current externally. |
| B8 | $I_{REF}$ | AI/O | Generates Reference Current for IDACs. Set by the external resistor, $R_{EXT}$. |
| K6 | BUF_VREF1 | AO | Buffered 2.5 V. The maximum load for BUF_VREF1 is 1.2 mA. |
| K7 | BUF_VREF2 | AO | Buffered 2.5 V. The maximum load for BUF_VREF2 is 1.2 mA. |
| L6 | PADC0P | AI | PGA Channel 0+. |
| M5 | PADC0N | AI | PGA Channel 0−. |
| L7 | PADC1P | AI | PGA Channel 1+. |
| M8 | PADC1N | AI | PGA Channel 1−. |
| K5 | NC | NC | No Connect. Do not connect to this pin. |
| K4 | NC | NC | No Connect. Do not connect to this pin. |
| M4 | NC | NC | No Connect. Do not connect to this pin. |
| L4 | NC | NC | No Connect. Do not connect to this pin. |
| 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 (ADC10).<br>Common Mode (AINCM). The common-mode function of this pin is for pseudo differential input. |
| K1 | DAC0 | AO | 12-Bit DAC0 Output. |
| K2 | DAC1 | AO | 12-Bit DAC1 Output. |
| J2 | NC | NC | No Connect. Do not connect to this pin. |
| L2 | NC | NC | No Connect. Do not connect to this pin. |
| M2 | NC | NC | No Connect. Do not connect to this pin. |
| L3 | NC | NC | No Connect. Do not connect to this pin. |
| M11 | DAC2 | AO | 12-Bit DAC2 Output. |
| L11 | NC | NC | No Connect. Do not connect to this pin. |
| L10 | NC | NC | No Connect. Do not connect to this pin. |
| K10 | NC | NC | No Connect. Do not connect to this pin. |
| K11 | NC | NC | No Connect. Do not connect to this pin. |
| K12 | DAC3 | AO | 12-Bit DAC3 Output. |
| B5 | IDAC4 | AO | IDAC4 Output. The output for this pin is 20 mA. |
| C6 | PVDD_IDAC4 | S | 2.0 V Power for IDAC4. |
| A6 | $C_{DAMP}$_IDAC4 | AI | Damping Capacitor Pin for IDAC4. |

| Pin No. | Mnemonic | Type[1] | Description |
|---|---|---|---|
| A8 | IDAC2 | AO | IDAC2 Output. The output for this pin is 45 mA. |
| A7 | PVDD_IDAC2 | S | 2.0 V Power for IDAC2. |
| C8 | $C_{DAMP}$_IDAC2 | AI | Damping Capacitor Pin for IDAC2. |
| A5 | IDAC3 | AO | IDAC3 Output. The output for this pin is 80 mA. |
| C5 | PVDD_IDAC3 | S | 2.0 V Power for the IDAC3. |
| B4 | $C_{DAMP}$_IDAC3 | AI | Damping Capacitor for IDAC3. |
| A4 | IDAC1 | AO | IDAC1 Output. The output for this pin is 200 mA. |
| A1 | IDAC1 | AO | IDAC1 Output. The output for this pin is 200 mA. |
| A3 | PVDD_IDAC1 | S | Power for IDAC1. |
| A2 | PVDD_IDAC1 | S | Power for IDAC1. |
| B1 | $C_{DAMP}$_IDAC1 | AI | Damping Capacitor for IDAC1. |
| A12 | IDAC0 | AO | IDAC0 Output. The output for this pin is 250 mA. |
| A9 | IDAC0 | AO | IDAC0 Output. The output for this pin is 250 mA. |
| A11 | PVDD_IDAC0 | S | Power for IDAC0. |
| A10 | PVDD_IDAC0 | S | Power for IDAC0. |
| B12 | $C_{DAMP}$_IDAC0 | AI | Damping Capacitor Pin for IDAC0. |
| B11 | IDAC_TST | AI/O | IDAC Test Purposes. |
| B10 | PGND | S | Power Ground. |
| B9 | PGND | S | Power 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 | AVDD | S | Analog Supply (3.3 V). |
| B7 | AVDD_IDAC | S | Output of 2.5 V LDO regulator for internal IDACs. A 470 nF capacitor to AGND must be connected to this pin. |
| G1 | DVDD | S | Output of 2.6 V On-Chip LDO Regulator. A 470 nF capacitor to DGND must be connected to this pin. |
| G12 | DVDD | 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 | Crystal Oscillator Inverter Output. If an external crystal is not being used, this pin can remain unconnected. |
| H2 | XTALI | DI | Crystal Oscillator Inverter Input and Internal Clock Generator Circuits Input. If an external crystal is not being used, connect this pin to the DGND system ground. |
| 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] A is analog, D is digital, I is input, O is output, and S is supply, NC is no connect.

# TERMINOLOGY

## ADC SPECIFICATIONS

### Integral Nonlinearity

Integral nonlinearity (INL) is 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

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

### Offset Error

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

### Gain Error

Gain error is 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

Signal to (noise + distortion) ratio, or SINAD, is 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 on the number of quantization levels in the digitization process: the more levels there are, the smaller the quantization noise becomes.

The theoretical SINAD ratio for an ideal N-bit converter with a sine wave input is given by

$$\textit{Signal to } (\textit{Noise} + \textit{Distortion}) = (6.02\,N + 1.76)\text{ dB}$$

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

### Total Harmonic Distortion

Total harmonic distortion (THD) is 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

This is the amount of time it takes the output to settle to within a one 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 8 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, as follows:

- T support for the thumb (16-bit) instruction set
- D support for debug
- 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, as follows:

- 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 debug state. When 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 interrupt as IRQ, but for higher priority interrupt, that is, faster response time, the programmer can define interrupt as FIQ.

## ARM REGISTERS

ARM7TDMI has a total of 37 registers: 31 general-purpose registers and 6 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 used for system level programming and exception handling only.

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 that the interrupt processing can begin without the need to save or restore these registers, thus saving critical time in the interrupt handling process.

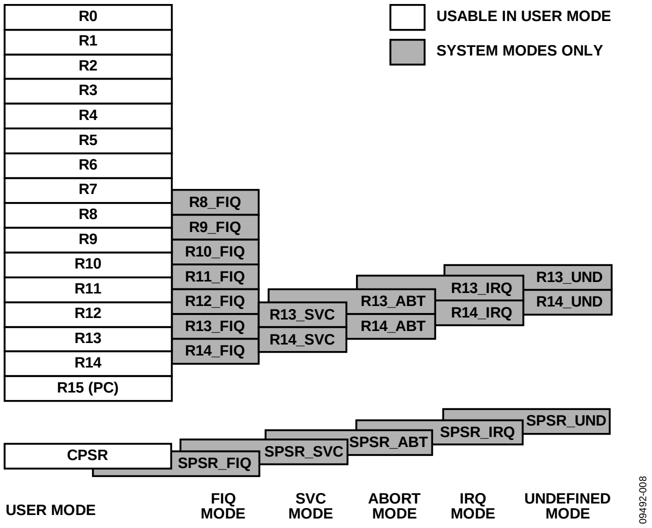

Figure 8. Register Organization

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

- ARM DDI 0029G, *ARM7TDMI Technical Reference Manual*
- ARM 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 may 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 wherein 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 modes, for example, when executing interrupt service routines.

# MEMORY ORGANIZATION

The ADuC7121 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.

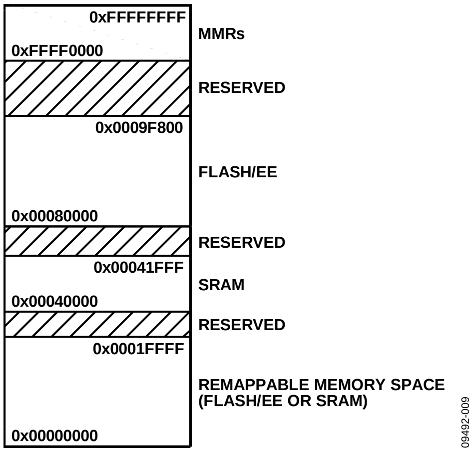

Figure 9. Physical Memory Map

## 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 ADuC7121 memory organization is configured in little endian format: the least significant byte is located in the lowest byte address and the most significant byte is located in the highest byte address.

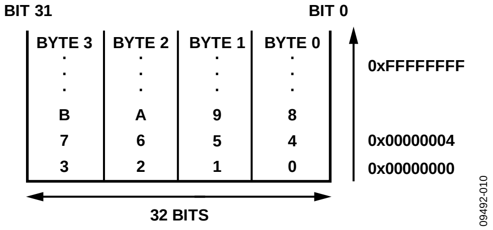

Figure 10. Little Endian Format

## FLASH/EE MEMORY

The 128 kB of Flash/EE are organized as two banks of 32k × 16 bits. Block 0 starts at Address 0x90000 and finishes at Address 0x9F700. In this block, 31k × 16 bits is 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.

Block 1 starts at Address 0x80000 and finishes at Address 0x90000. In this block 64 kB block is arranged in 32k × 16 bits, all of which 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 through Table 27 provide the complete MMR memory maps.

The access time reading or writing an 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 a lower performance peripheral. Access to the AHB is one cycle, and access to the APB is two cycles. All peripherals on the ADuC7121 are on the APB except the Flash/EE memory and the GPIOs.

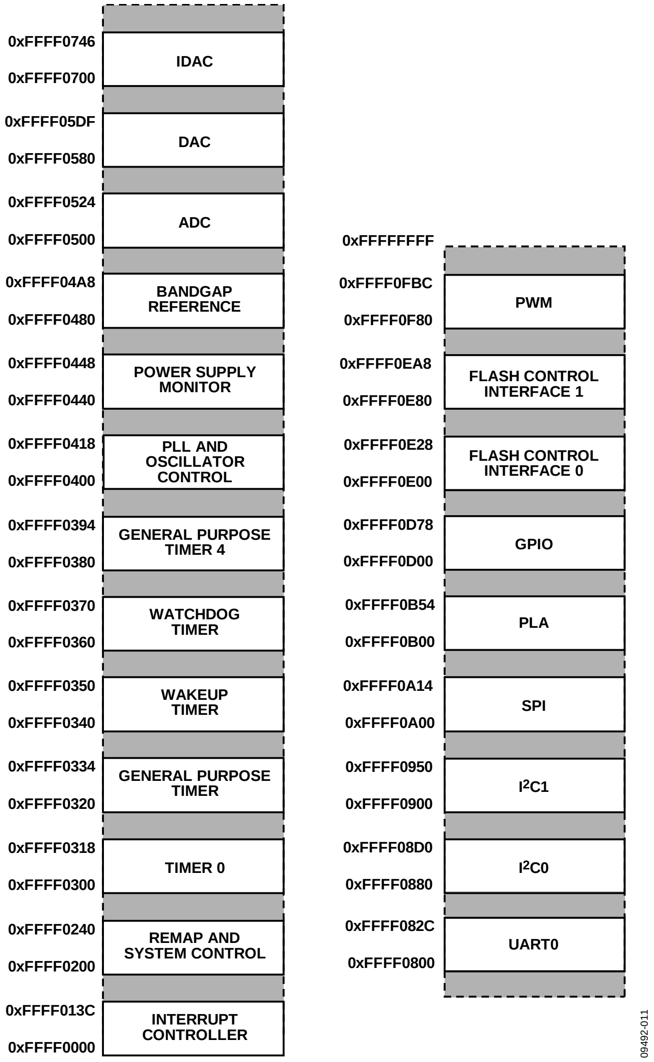

*Figure 11. Memory Mapped Registers*

## COMPLETE MMR LISTING

Note that the Access Type column corresponds to the access time reading or writing an MMR, where R is read, W is write, and R/W is read/write. It depends on the AMBA bus that is used to access the peripheral. The processor has two AMBA buses: the advanced high performance bus (AHB ) used for system modules and the advanced peripheral bus (APB) 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 | R/W | 1 |
| 0x0010 | SWICFG | 4 | W | 1 |
| 0x0014 | IRQBASE | 4 | R/W | 1 |
| 0x001C | IRQVEC | 4 | R/W | 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 | 4 | R/W | 1 |
| 0x0038 | IRQCLRE | 1 | W | 1 |
| 0x003C | IRQSTAN | 1 | R/W | 1 |
| 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 | RSTCFGKEY0 | 1 | W | 1 |
| 0x024C | RSTCFG | 1 | R/W | 1 |
| 0x0250 | RSTCFGKEY1 | 1 | W | 1 |

**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 |
| 0x0518 | ADCGN | 2 | R/W | 2 |
| 0x051C | ADCOF | 2 | R/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 |
| 0x05B0 | DAC2CON | 2 | R/W | 2 |
| 0x05B4 | DAC2DAT | 4 | R/W | 2 |
| 0x05D8 | DAC3CON | 2 | R/W | 2 |
| 0x05DC | DAC3DAT | 4 | R/W | 2 |

### Table 18. IDAC Base Address = 0xFFFF0700

| Address | Name | Byte | Access Type | Cycle |
|---|---|---|---|---|
| 0x0700 | IDAC0CON | 2 | R/W | 2 |
| 0x0704 | IDAC0DAT | 4 | R/W | 2 |
| 0x0708 | IDAC0BW | 1 | R/W | 2 |
| 0x070C | IDAC1CON | 2 | R/W | 2 |
| 0x0710 | IDAC1DAT | 4 | R/W | 2 |
| 0x0714 | IDAC1BW | 1 | R/W | 2 |
| 0x0718 | IDAC2CON | 2 | R/W | 2 |
| 0x071C | IDAC2DAT | 4 | R/W | 2 |
| 0x0720 | IDAC2BW | 1 | R/W | 2 |
| 0x0724 | IDAC3CON | 2 | R/W | 2 |
| 0x0728 | IDAC3DAT | 4 | R/W | 2 |
| 0x072C | IDAC3BW | 1 | R/W | 2 |
| 0x0730 | IDAC4CON | 2 | R/W | 2 |
| 0x0734 | IDAC4DAT | 4 | R/W | 2 |
| 0x0738 | IDAC4BW | 1 | R/W | 2 |
| 0x073C | TSDCON | 1 | R/W | 2 |
| 0x0740 | IDACSTA | 1 | R/W | 2 |
| 0x0744 | IDAC0PULLDOWN | 1 | R/W | 2 |

### Table 19. UART0 Base Address = 0xFFFF0800

| Address | Name | Byte | Access Type | Cycle |
|---|---|---|---|---|
| 0x0800 | COMTX | 1 | W | 2 |
|  | COMRX | 1 | R | 2 |
|  | COMDIV0 | 1 | R/W | 2 |
| 0x0804 | COMIEN0 | 1 | R/W | 2 |
|  | 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 |
| 0X082C | COMDIV2 | 2 | R/W | 2 |

### Table 20. I$^2$C0 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 | 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 | 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 21. I$^2$C1 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 | 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 | W | 2 |
| 0x0938 | I2C1ALT | 1 | R/W | 2 |
| 0x093C | I2C1ID0 | 1 | R/W | 2 |

| Address | Name | Byte | Access Type | Cycle |
|---|---|---|---|---|
| 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 22. SPI Base Address = 0xFFFF0A00

| Address | Name | Byte | Access Type | Cycle |
|---|---|---|---|---|
| 0x0A00 | SPISTA | 2 | 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 23. 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 | 2 | R/W | 2 |
| 0x0B48 | PLAADC | 4 | R/W | 2 |
| 0x0B4C | PLADIN | 4 | R/W | 2 |
| 0x0B50 | PLAOUT | 4 | R | 2 |
| 0x0B54 | PLALCK | 1 | W | 2 |

Table 24. 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 |

Table 25. 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 26. 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 27. 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 a 3.0 V to 3.6 V supply 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, a differential track-and-hold, an on-chip reference, and an 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 V 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 of 0 V to $AV_{DD}$ and with a maximum amplitude of 2 $V_{REF}$ (see Figure 12).

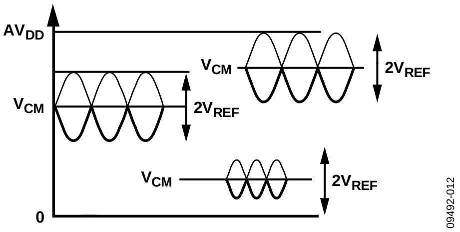

*Figure 12. Examples of Balanced Signals for Fully Differential Mode*

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 the software. An external $ADC_{\overline{CONVST}}$ pin, an output generated from the on-chip PLA, a Timer0, or a 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 creating an additional ADC channel input. This facilitates an internal temperature sensor channel, measuring die temperature to an accuracy of ±3°C.

The ADuC7121 is modified in a way that differentiates its ADC structure from other devices in the ADuC702x family.

The PADC0x and PADC1x inputs connect to a PGA and allow for a gain from 1 to 5 with 32 steps. The remaining channels can be configured as single ended or differential. A buffer is provided before the ADC for measuring internal channels.

### ADC TRANSFER FUNCTION

#### Pseudo Differential and Single-Ended Modes

For both pseudo differential and single-ended modes, the input range is 0 to $V_{REF}$. In addition, the output coding is straight binary in both pseudo differential and single-ended modes with

$$1\text{ LSB} = FS/4096\text{, or } 2.5\text{ V}/4096 = 0.61\text{ mV, or } 610\ \mu\text{V when } V_{REF} = 2.5\text{ V}$$

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.

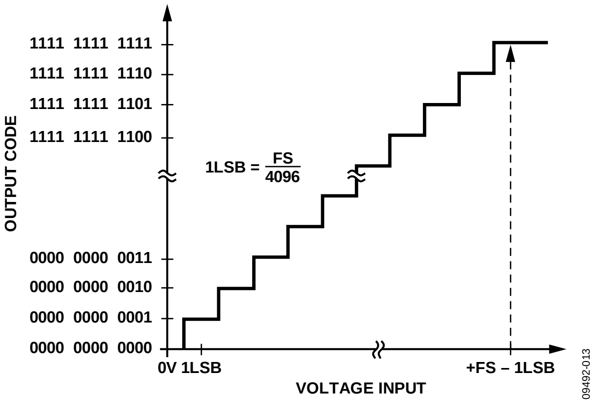

*Figure 13. ADC Transfer Function in Pseudo Differential Mode or Single-Ended Mode*

#### 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, $V_{IN+} - V_{IN-}$). Therefore, the maximum amplitude of the differential signal is $-V_{REF}$ to $+V_{REF}$ p-p (2 × $V_{REF}$). 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, which 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

$$1\text{ LSB} = 2\ V_{REF}/4096 \text{ or } 2 \times 2.5\text{ V}/4096 = 1.22\text{ mV when } V_{REF} = 2.5\text{ V}$$

The output result is ±11 bits, but this is shifted by one bit to the right. This 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.

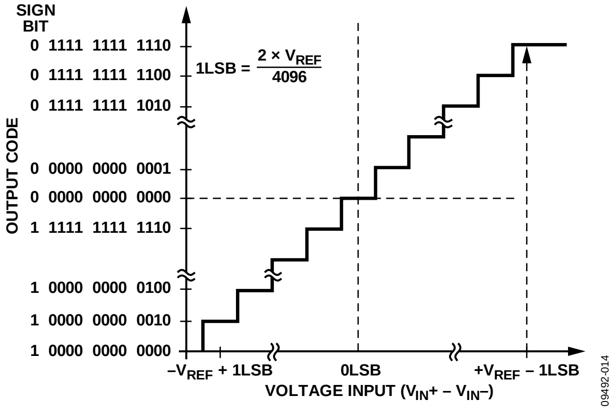

*Figure 14. ADC Transfer Function in Differential Mode*

### PADC0x/PADC1x Pins

The PADC0x and PADC1x pins are differential input channels to the ADC that each have a programmable gain amplifier (PGA ) on their front ends.

An external precision resistor converts the current to voltage and the PGA then amplifies this voltage signal with gain up to 5 by 32 steps. The intention is to compensate the variation of the detector diode responsivity and normalize optical power read by the ADC. The external resistor is assumed 0.1% accuracy, 5 ppm. A 1 nF capacitor is shunted with the resistor to suppress wideband noise. Select the resistor value such that the full-scale voltage developed on the resistor is less than $AV_{DD}$ – 1.2 V, or typically 1.8 V.

The PGA is designed to handle 10 mV minimum input. To minimize noise, bypass the ADC input buffer.

PADC0N is driven by a buffer to 0.15 V to keep the PGA from saturation when the input current drops to zero (see Figure 15).

Another buffer on the output of the MUX can be disabled by setting the ADCCON[14] so that the PADCxN pin can be connected to the ground plane as well. This is the same for the PADCxP pin using ADCCON[15]. The buffer is longside the switch labeled in Figure 15.

The ADC needs to be placed in pseudo differential mode and assumes that the negative input is close to ground.

All of the controls are independently set through register bits for giving maximum flexibility to the user. Typically, users need to take the following steps:

1. Select PADCxP and PADCxN in the ADCCP and ADCCN registers.
2. Optionally, bypass the ADC input buffers in ADCCON[15:14].
3. Set the proper gain value for the PGA in PGA_GN.
4. Set the ADC to pseudo differential mode in ADCCON[4:3].
5. Start the conversion.

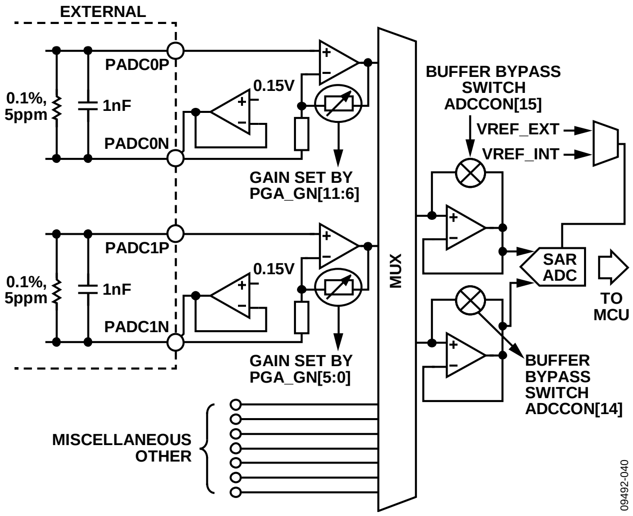

*Figure 15. PADC0x/PADC1x ADC Input*

### Other Input Channels

ADuC7121 contains seven extra ADC input pins. These pins can also be configured as differential input pairs or single-ended inputs, or pseudo differential inputs. The buffer and ADC are configured independently from the 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 ADuC7121 provides two pins for each thermistor input. The negative input removes the error of the ground difference. When selecting the thermistor input, always bypass the negative side buffer to ensure that the amplifier is not saturated. Configure the ADC to work in positive pseudo differential mode.

Besides these external inputs, the ADC can also select internal inputs to monitor three power supplies: IOVDD, PVDD_IDAC0, and PVDD_IDAC1. The voltage of the five IDAC outputs can also be monitored by the ADC by selecting the required channel in Register ADCCP. These internal signals are single-ended and can select AGND/PGND/IOGND as the negative input of the ADC via the ADCCN register.

Note that when monitoring IDAC outputs or PVDD_IDAC0, PVDD_IDAC1, or IOVDD_MON, the buffer must be enabled to isolate interference from ADC sampling.

An on-chip diode can also be selected to provide chip temperature monitoring. The ADC can also select $V_{REF}$ and AGND as inputs for calibration purposes.

### PGA and Input Buffer

The PGA is a one stage, positive gain amplifier that is able to accept input from 0.1 V to $AV_{DD}$ – 1.2 V, and the output swing should be at least 2.5 V. The gain of the PGA is from 1 to 5 with 32 linear steps. The PGA cannot be bypassed for the PADC0x and PADC1x channels.

The input level for PGA is limited to a maximum value of AVDD – 1.2 V and minimum value of 0.1 V to ensure that the amplifiers are not saturated. The input buffer is a rail-to-rail buffer. It can accept signals from 0.15 to AVDD − 0.15 V. Both the positive and negative input buffers can be bypassed independently by setting ADCCON Bits[15:14].

### 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 ADC data register.

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

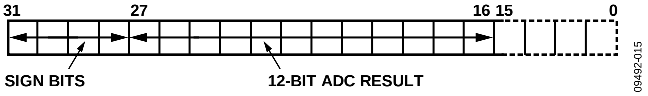

Figure 16. ADC Result Format

### *Timing*

Figure 17 provides details of the ADC timing. Users control the ADC clock speed and the number of acquisition clocks in the ADCCON MMR. By default, the acquisition time is eight clocks and the clock divider is two. The number of additional 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.

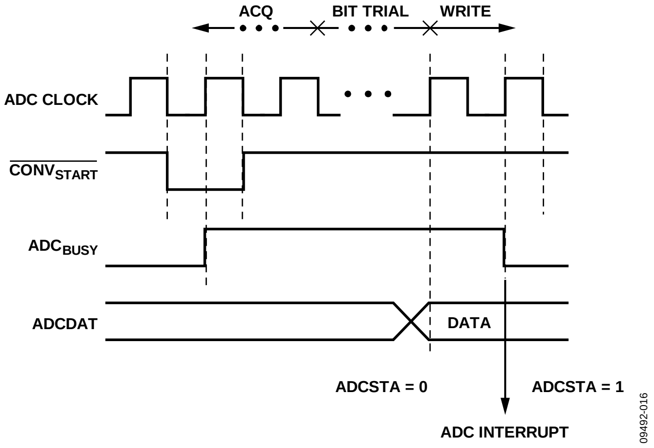

Figure 17. ADC Timing

## TEMPERATURE SENSOR

The ADuC7121 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 ADuC7121 die. The typical temperature coefficient is −1.25 mV/°C.

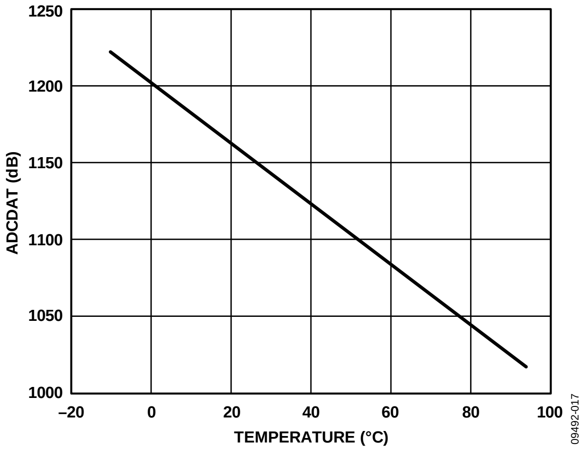

Figure 18. ADC Output vs. Temperature

### *ADC MMR Interface*

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

**Table 28. ADC MMRs**

| Name | Description |
|---|---|
| ADCCON | ADC control register. ADCCON 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 29). |
| ADCCP | ADC positive channel selection register. |
| ADCCN | ADC negative channel selection register. |
| ADCSTA | ADC status register. ADCSTA 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$_{Busy}$ function of Pin C3. This pin is high during a conversion. When the conversion is finished, ADC$_{Busy}$ returns to low. This information can be available on P0.2 (see the General-Purpose Input/Output section) if enabled in the GP0CON register. |
| ADCDAT | ADC data result register. ADCDAT holds the 12-bit ADC result, as shown in Figure 16. |
| ADCRST | ADC reset register. ADCRST resets all of the ADC registers to their default values. |
| ADCGN | ADC gain calibration register for non-PGA channels. |
| ADCOF | ADC offset calibration register for all ADC channels. |
| PGA_GN | Gain of PGA_PADC0 and PGA_PADC1. |

**Table 29. 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 | $f_{ADC}$ divide-by-1. This divider is provided to obtain a 1 MSPS ADC with an external clock of <41.78 MHz. |
| | 001 | $f_{ADC}$ divide-by-2 (default value). |
| | 010 | $f_{ADC}$ divide-by-4. |
| | 011 | $f_{ADC}$ divide-by-8. |
| | 100 | $f_{ADC}$ divide-by-16. |
| | 101 | $f_{ADC}$ divide-by-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. |
| | | Set by the user to 1 to enable conversion mode. |
| | | Cleared by the user to 0 to disable conversion mode. |
| 6 | | Reserved. The user sets this bit to 0. |
| 5 | | ADC power control. |
| | 1 | Set by the 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 the 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 the $ADC_{\overline{CONVST}}$ function on Pin F3 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 | PWM conversion. |
| | Other | Reserved. |

### Table 30. ADCCP[1] MMR Bit Designations

| Bit | Value | Description |
|---|---|---|
| 7:5 | | Reserved |
| 4:0 | | Positive channel selection bits |
| | 00000 | PADC0P |
| | 00001 | PADC1P |
| | 00010 | Reserved |
| | 00011 | Reserved |
| | 00100 | Reserved |
| | 00101 | Reserved |
| | 00110 | ADC4 |
| | 00111 | ADC5 |
| | 01000 | ADC6 |
| | 01001 | ADC7 |
| | 01010 | ADC8 |
| | 01011 | ADC9 |
| | 01100 | ADC10/AINCM |
| | 01101 | Temperature sensor |
| | 01110 | DVDD_IDAC0 |
| | 01111 | DVDD_IDAC1 |
| | 10000 | DVDD_IDAC2 |
| | 10001 | DVDD_IDAC3 |
| | 10010 | DVDD_IDAC4 |
| | 10011 | IOVDD_MON |
| | 10100 | Reserved |
| | 10101 | Reserved |
| | 10110 | $V_{REF}$ |
| | 10111 | AGND |
| | Others | Reserved |

[1] ADC channel availability depends on part model.

### Table 31. ADCCN[1] MMR Bit Designations

| Bit | Value | Description |
|---|---|---|
| 7:5 | | Reserved |
| 4:0 | | Negative channel selection bits |
| | 00000 | PADC0N |
| | 00001 | PADC1N |
| | 00010 | Reserved |
| | 00011 | Reserved |
| | 00100 | Reserved |
| | 00101 | Reserved |
| | 00110 | ADC4 |
| | 00111 | ADC5 |
| | 01000 | ADC6 |
| | 01001 | ADC7 |
| | 01010 | ADC8 |
| | 01011 | ADC9 |
| | 01100 | ADC10/AINCM |
| | 01101 | $V_{REF}$ |
| | 01110 | AGND |
| | 01111 | PGND |
| | 10000 | IOGND |
| | Others | Reserved |

[1] ADC channel availability depends on part model.

### Table 32. ADCSTA MMR Bit Designations

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

### Table 33. ADCDAT MMR Bit Designations

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

### Table 34. ADCRST MMR Bit Designations

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

### Table 35. PGA_GN MMR Bit Designations[1]

| Bit | Value[2] | Description |
|---|---|---|
| 11:6 | N/A | Gain of PGA for PADC0 = 1 + 4 × (PGA_PADC0_GN/32). |
| 5:0 | N/A | Gain of PGA for PADC1 = 1 + 4 × (PGA_PADC1_GN/32). |

[1] PGA_PADC0_GN and PGA_PADC1_GN must be ≤ 32.
[2] N/A means not applicable.

### Table 36. ADCGN MMR Bit Designations

| Bit | Value[1] | Description |
|---|---|---|
| 11:6 | N/A | These bits are reserved. |
| 9:0 | N/A | 10-bit ADC gain calibration value for non-PGA channels. |

[1] N/A means not applicable.

### Table 37. ADCOF MMR Bit Designations

| Bit | Value[1] | Description |
|---|---|---|
| 15:10 | N/A | These bits are reserved. |
| 9:0 | N/A | 10-bit ADC offset calibration value. |

[1] N/A means not applicable.

## 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, pseudo differential, and single-ended.

### *Differential Mode*

The ADuC7121 contains a successive approximation ADC based on two capacitive DACs. Figure 19 and Figure 20 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 19 (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.

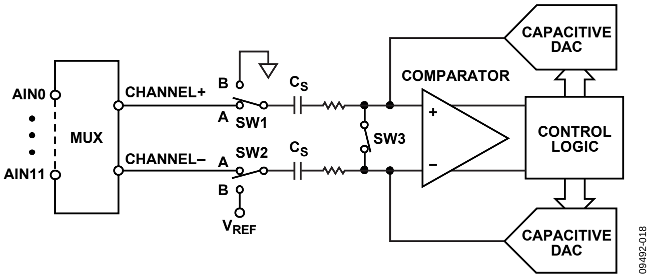

Figure 19. ADC Acquisition Phase

When the ADC starts a conversion (see Figure 20), SW3 opens, and SW1 and SW2 move to Position B, causing the comparator to become unbalanced. Both inputs are disconnected as soon as 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 return the comparator to 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 $V_{IN+}$ input and the $V_{IN-}$ input must be matched; otherwise, the two inputs have different settling times, resulting in errors.

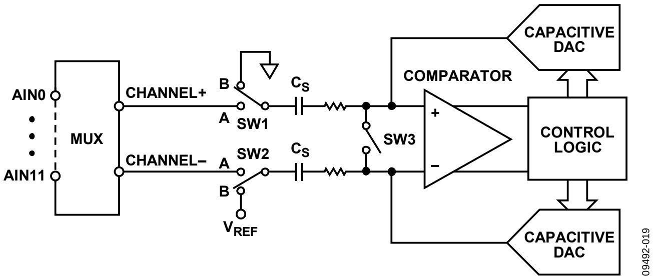

Figure 20. ADC Conversion Phase

### Pseudo Differential Mode

In pseudo differential mode, Channel– is linked to the $V_{IN-}$ input of the ADuC7121, 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 chosen so that $V_{REF}$ + $V_{IN-}$ does not exceed $AV_{DD}$.

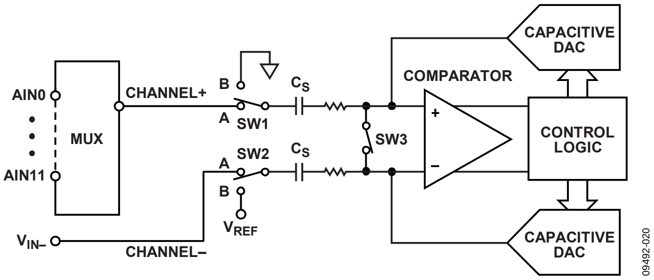

Figure 21. ADC in Pseudo Differential Mode

### Single-Ended Mode

In single-ended mode, SW2 is always connected internally to ground. The $V_{IN-}$ input pin can be floating. The input signal range on $V_{IN+}$ is 0 V to $V_{REF}$.

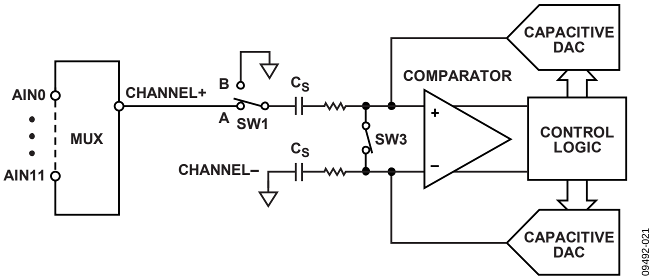

Figure 22. ADC in Single-Ended Mode

### Analog Input Structure

Figure 23 shows the equivalent circuit of the analog input structure of the ADC. The four diodes provide ESD protection for the analog inputs. Take care to ensure that the analog input signals never exceed the supply rails by more than 300 mV. Voltage in excess of 300 mV causes these diodes to become forward biased and to 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 23 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.

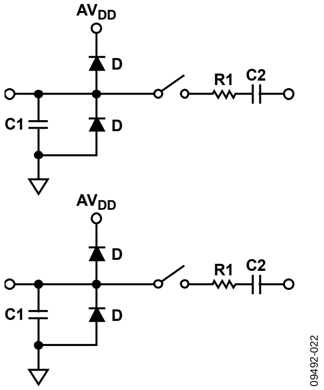

Figure 23. Equivalent Analog Input Circuit Conversion Phase: Switches Open, Track Phase: Switches Closed

For ac applications, removing high frequency components from the analog input signal is recommended with an RC low-pass filter on the relevant analog input pins. In applications where harmonic distortion and signal-to-noise ratio are critical, drive the analog input 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 24 and Figure 25 give an example of an ADC front end.

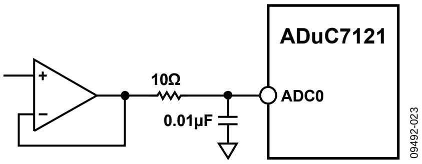

Figure 24. Buffering Single-Ended/Pseudo Differential Input

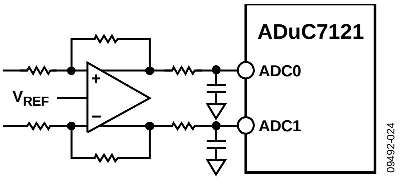

*Figure 25. Buffering Differential Inputs*

When no amplifier is used to drive the analog input, limit the source impedance 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

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

Table 38 gives some calculated $V_{CM}$ minimum and $V_{CM}$ maximum values.

**Table 38. $V_{CM}$ Ranges**

| $AV_{DD}$ | $V_{REF}$ | $V_{CM}$ Min | $V_{CM}$ Max | Signal Peak-to-Peak |
|---|---|---|---|---|
| 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 ADuC7121 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 $V_{REF}$ pins ($V_{REF}$_2.5 and $V_{REF}$_1.2). When using the internal reference, a capacitor of 0.47 μF must be connected between each external $V_{REF}$ pin and AGND to ensure stability and fast response during ADC conversions. This reference can also be connected to the external pin, BUF_VREF2, and used as a reference for other circuits in the system.

The band gap reference also connects through buffers to the BUF_VREF1 and the BUF_VREF2 pins. To damp the noise, connect a minimum of 0.1 μF capacitor to these pins. The band gap reference interface consists of an 8-bit REFCON MMR, described in Table 39.

**Table 39. REFCON MMR Bit Designations (Address = 0xFFFF0480, Default Value = 0x01)**

| Bit | Description |
|---|---|
| 7:1 | Reserved. |
| 2 | BUF_VREF1/BUF_VREF2 is driven from the internal 2.5 V reference when set to 1. |
| 1 | Internal 2.5 V reference output enable.<br>Set by the user to connect the internal 2.5 V reference to the $V_{REF}$_2.5 pin.<br>Cleared by the user to disconnect the reference from the $V_{REF}$_2.5 pin. The $V_{REF}$_2.5 pin should also be cleared to connect an external reference source to it. |
| 0 | Internal 1.2 V reference output enable.<br>Set by the user to connect the internal 1.2 V reference to the $V_{REF}$_1.2 pin.<br>Cleared by the user to disconnect the reference from the $V_{REF}$_1.2 pin. |

## POWER SUPPLY MONITOR

The power supply monitor on the ADuC7121 indicates when the IOVDD 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 after CMP goes high. Note that if the interrupt generated is exited before CMP goes high (IOVDD supply voltage is above the trip point), no further interrupts are generated until CMP returns high. The user needs to ensure that the 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 low supply or brownout conditions. It also ensures that normal code execution does not resume until a safe supply level has been established.

The PSM does not operate correctly when using JTAG debug; therefore, disable PSM while in JTAG debug mode.

**Table 40. PSMCON MMR Bit Designations (Address = 0xFFFF0440, Default Value = 0x0008)**

| Bit | Name | Description |
|---|---|---|
| 15:4 | Reserved | These bits are reserved. |
| 3 | CMP | Comparator bit. This is a read-only bit that directly reflects the state of the comparator.<br>Read 1 indicates that the IOVDD supply is above its selected trip point or the PSM is in power-down mode.<br>Read 0 indicates the IOVDD supply is below its selected trip point. Set this bit before leaving the interrupt service routine. |
| 2 | TP | Trip point selection bit.<br>0 = 2.79 V.<br>1 = 3.07 V. |
| 1 | PSMEN | Power supply monitor enable bit.<br>Set to 1 by the user to enable the power supply monitor circuit.<br>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 MicroConverter 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 after CMP goes high. |

# NONVOLATILE FLASH/EE MEMORY

## FLASH/EE MEMORY OVERVIEW

The ADuC7121 incorporates Flash/EE memory technology on chip to provide the user with nonvolatile, in circuit reprogrammable memory space.

Similar to 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 ADuC7121, 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 ADuC7121 contains two 64 kB arrays of Flash/EE memory. In Flash Block 0, the lower 62 kB is available to the user and the upper 2 kB of this Flash/EE memory array program 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 forth). 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 Flash Block 1, 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.

### *Flash/EE Memory Reliability*

The Flash/EE memory arrays on the ADuC7121 are fully qualified for two key Flash/EE memory characteristics: Flash/EE memory cycling endurance and Flash/EE memory data retention.

Endurance quantifies the ability of the Flash/EE memory to be cycled through many program, read, and erase cycles. A single endurance cycle is composed of four independent, sequential events, defined as follows:

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

In reliability qualification, every half word (16-bit wide) location of the three pages (top, middle, and bottom) in the Flash/EE memory is cycled 10,000 times from 0x0000 to 0xFFFF.

As indicated in the Specifications 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 supply temperature of 10,000 cycles.

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. Again, the parts 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 26.

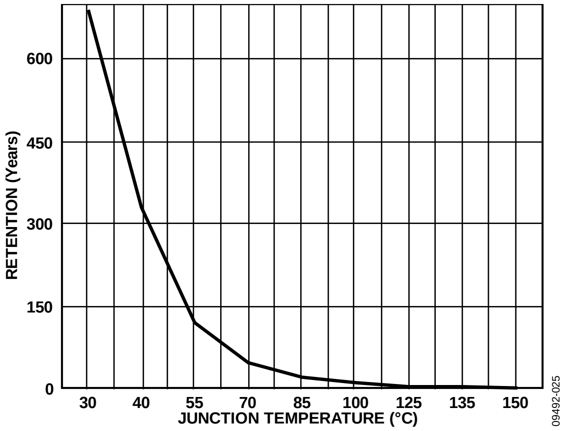

*Figure 26. Flash/EE Memory Data Retention*

### *Serial Downloading (In-Circuit Programming)*

The ADuC7121 facilitates code download via the I$^2$C serial port. The ADuC7121 enters serial download mode after a reset or power cycle if the $\overline{\text{BM}}$ function of the P3.7/$\overline{\text{BM}}$/PLAO[11] pin is pulled low through an external 1 kΩ resistor. This is combined with the state of Address 0x00014 in the flash. If this address is 0xFFFFFFFF and $\overline{\text{BM}}$ 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 serial download executable and hardware dongle are provided as part of the development system for serial downloads via the I$^2$C port. The I$^2$C maximum allowed baud rate is 100 kHz for the I$^2$C downloader.

### *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 62 kB of Block 0 from being read through JTAG and also in I$^2$C programming mode. 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 Flash Block 1. Bit 31 of the FEE1PRO/FEE1HID MMR protects the 64 kB of Block 1 from being read through JTAG. Bit 30 protects writing to the top 8 pages of Block 1. The other 30 bits of this register protect writing to the Flash/EE memory; each bit protects four pages, that is, 2 kB

### *Three Levels of Protection*

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

Protection can be set by writing into the 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 one time only and must be reentered 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 FEEPRO 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 to Protection Registers*

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 0x0C 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 0xDEAD, then the memory protection cannot be removed. Only a mass erase unprotects the part, but 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 Page 4 to Page 7
FEE0MOD=0x48;        //Write key enable
FEE0ADR=0x1234;      //16-bit key value
FEE0DAT=0x5678;      //16-bit key value
FEE0CON= 0x0C;       //Write key command
```

Follow the same sequence to permanently protect the part with FEExADR = 0xDEAD and FEExDAT = 0xDEAD.

## FLASH/EE CONTROL INTERFACE

### *FEE0DAT Register*

FEE0DAT is a 16-bit data register.

| | |
|---|---|
| Name: | FEE0DAT |
| Address: | 0xFFFF0E0C |
| Default value: | 0xXXXX |
| Access: | Read and write |

### *FEE0ADR Register*

FEE0ADR is a 16-bit address register.

| | |
|---|---|
| Name: | FEE0ADR |
| Address: | 0xFFFF0E10 |
| Default value: | 0x0000 |
| Access: | Read and write |

### *FEE0SGN Register*

FEE0SGN is a 24-bit code signature.

| | |
|---|---|
| Name: | FEE0SGN |
| Address: | 0xFFFF0E18 |
| Default value: | 0xFFFFFF |
| Access: | Read only |

### *FEE0PRO Register*

FEE0PRO provides protection following subsequent reset MMR. It requires a software key (see Table 41).

| | |
|---|---|
| Name: | FEE0PRO |
| Address: | 0xFFFF0E1C |
| Default value: | 0x00000000 |
| Access: | Read and write |

### *FEE0HID Register*

FEE0HID provides immediate protection MMR. It does not require any software keys (see Table 41).

| | |
|---|---|
| Name: | FEE0HID |
| Address: | 0xFFFF0E20 |
| Default value: | 0xFFFFFFFF |
| Access: | Read and write. |

Table 41. FEE0PRO and FEE0HID MMR Bit Designations

| Bit | Description |
|---|---|
| 31 | Read protection.<br>Cleared by the user to protect Block 0.<br>Set by the user to allow reading Block 0. |
| 30:0 | Write protection for Page 123 to Page 0. Each bit protects a group of 4 pages.<br>Cleared by the user to protect the pages when writing to flash. Thus preventing an accidental write to specific pages in flash.<br>Set by the user to allow writing the pages. |

### *Command Sequence for Executing a Mass Erase*

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

### *FEE1DAT Register*

FEE1DAT is a 16-bit data register.

| | |
|---|---|
| Name: | FEE1DAT |
| Address: | 0xFFFF0E8C |
| Default value: | 0xXXXX |
| Access: | Read and write |

### *FEE1ADR Register*

FEE1ADR is a 16-bit address register.

| | |
|---|---|
| Name: | FEE1ADR |
| Address: | 0xFFFF0E90 |
| Default value: | 0x0000 |
| Access: | Read and write |

### *FEE1SGN Register*

FEE1SGN is a 24-bit code signature.

| | |
|---|---|
| Name: | FEE1SGN |
| Address: | 0xFFFF0E98 |
| Default value: | 0xFFFFFF |
| Access: | Read only |

### *FEE1PRO Register*

FEE1PRO provides protection following subsequent reset MMR. It requires a software key (see Table 42).

| | |
|---|---|
| Name: | FEE1PRO |
| Address: | 0xFFFF0E9C |
| Default value: | 0x00000000 |
| Access: | Read and write |

### *FEE1HID Register*

FEE1HID provides immediate protection MMR. It does not require any software keys (see Table 42).

| | |
|---|---|
| Name: | FEEHID |
| Address: | 0xFFFF0EA0 |
| Default value: | 0xFFFFFFFF |
| Access: | Read and write |

Table 42. FEE1PRO and FEE1HID MMR Bit Designations

| Bit | Description |
|---|---|
| 31 | Read protection.<br>Cleared by the user to protect Block 1.<br>Set by the user to allow reading Block 1. |
| 30 | Write protection for Page 127 to Page 120.<br>Cleared by the user to protect the pages when writing to flash. Thus preventing an accidental write to specific pages in flash.<br>Set by the user to allow writing the pages. |
| 29:0 | Write protection for Page 119 to Page 0. Each bit protects a group of 4 pages.<br>Cleared by the user to protect the pages when writing to flash. Thus preventing an accidental write to specific pages in flash.<br>Set by the user to allow writing the pages. |

### *FEE0STA Register*

| | |
|---|---|
| Name: | FEE0STA |
| Address: | 0xFFFF0E00 |
| Default value: | 0x0001 |
| Access: | Read and write |

### *FEE1STA Register*

| | |
|---|---|
| Name: | FEE1STA |
| Address: | 0xFFFF0E80 |
| Default value: | 0x0000 |
| Access: | Read and write |

Table 43. FEExSTA MMR Bit Designations

| Bit | Description |
|---|---|
| 15:6 | Reserved. |
| 5 | Reserved. |
| 4 | Reserved. |
| 3 | Flash/EE interrupt status bit.<br>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.<br>Cleared when reading FEExSTA register. |
| 2 | Flash/EE controller busy.<br>Set automatically when the controller is busy.<br>Cleared automatically when the controller is not busy. |

| Bit | Description |
|---|---|
| 1 | Command fail.<br>Set automatically when a command completes unsuccessfully.<br>Cleared automatically when reading FEExSTA register. |
| 0 | Command complete.<br>Set by MicroConverter when a command is complete.<br>Cleared automatically when reading FEExSTA register. |

#### *FEE0MOD Register*

Name: FEE0MOD

Address: 0xFFFF0E04

Default value: 0x80

Access: Read and write

#### *FEE1MOD Register*

Name: FEE1MOD

Address: 0xFFFF0E84

Default value: 0x80

Access: Read and write

**Table 44. FEExMOD MMR Bit Designations**

| Bit | Description |
|---|---|
| 7:5 | Reserved. These bits are always set to 0 except when writing keys. See the Sequence to Write the Key to Protection Registers section for details. |
| 4 | Flash/EE interrupt enable.<br>Set by the user to enable the Flash/EE interrupt. The interrupt occurs when a command is complete.<br>Cleared by the user to disable the Flash/EE interrupt. |
| 3 | Erase/write command protection.<br>Set by the user to enable the erase and write commands.<br>Cleared to protect the Flash/EE memory against erase/write command. |
| 2 | Reserved. The user must set this bit to 0. |
| 1:0 | Flash/EE wait states. Both Flash/EE blocks must have the same wait state value for any change to take effect. |

#### *FEE0CON Register*

Name: FEE0CON

Address: 0xFFFF0E08

Default value: 0x00

Access: Read and write

#### *FEE1CON Register*

Name: FEE1CON

Address: 0xFFFF0E88

Default value: 0x00

Access: Read and write

**Table 45. 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.

## EXECUTION TIME FROM SRAM AND FLASH/EE

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

### Execution from SRAM

Fetching instructions from SRAM takes one clock cycle 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 retrieve 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 accomplished in one cycle (as can be done from SRAM when the CD bit = 0). In addition, some dead times are needed 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 the Flash/EE for data memory. If the instruction to be executed is a control flow instruction, an extra cycle is needed to decode the new address of the program counter and then four cycles are needed to fill the pipeline. A data processing instruction involving only core registers does not require any extra clock cycles, but if it involves data in Flash/EE, one additional clock cycle is needed to decode the address of the data and two additional cycles are needed 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 46.

**Table 46. 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 situated at the bottom of the memory array, from Address 0x00000000 to Address 0x00000020, as shown in Figure 27.

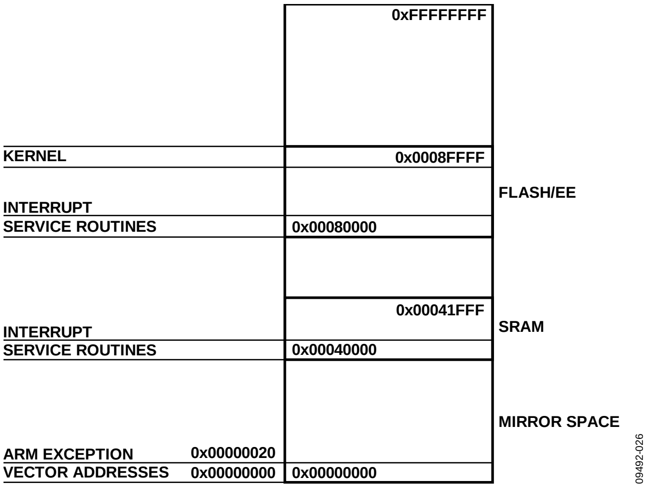

*Figure 27. Remap for Exception Execution*

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 being 16-bit wide Flash/EE memory.

### Remap Operation

When a reset occurs on the ADuC7121, execution starts automatically in factory programmed internal configuration code. This kernel is hidden and cannot be accessed by user code.

If the ADuC7121 is in normal mode (the P3.7/$\overline{\text{BM}}$/PLAO[11] 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 user's reset exception routine. 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 remap is performed from Flash/EE 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, the defined memory space, 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. 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 set to 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 47. Remap MMR Bit Designations (Address = 0xFFFF0220, Default Value = 0x00)**

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

**Table 48. RSTSTA MMR Bit Designations (Address = 0xFFFF0230, Default Value = 0x0X)**

| Bit | Description |
|---|---|
| 7:3 | Reserved. |
| 2 | Software reset.<br>Set by the user to force a software reset.<br>Cleared by setting the corresponding bit in RSTCLR. |
| 1 | Watchdog timeout.<br>Set automatically when a watchdog timeout occurs.<br>Cleared by setting the corresponding bit in RSTCLR. |
| 0 | Power-on reset.<br>Set automatically when a power-on reset occurs.<br>Cleared by setting the corresponding bit in RSTCLR. |

### RSTCFGKEY0 Register

| | |
|---|---|
| Name: | RSTCFGKEY0 |
| Address: | 0xFFFF0248 |
| Default value: | 0xXX |
| Access | Write |

### RSTCFGKEY1 Register

| | |
|---|---|
| Name: | RSTCFGKEY1 |
| Address: | 0xFFFF0250 |
| Default value: | 0xXX |
| Access: | Write |

**Table 49. RSTCFG Write Sequence**

| Name | Code |
|---|---|
| RSTCFGKEY0 | 0x76 |
| RSTCFG | User value |
| RSTCFGKEY1 | 0xB1 |

**Table 50. RSTCFG MMR Bit Designations (Address = 0xFFFF024C, Default Value = 0x00)**

| Bit | Description |
|---|---|
| 7:4 | Reserved. Always set to 0. |
| 3 | This bit is set to 1 to configure the IDAC outputs to retain their state after a watchdog or software reset.<br>This bit is cleared for the IDAC output pins and registers to return to their default state. |
| 2 | This bit is set to 1 to configure the DAC outputs to retain their state after a watchdog or software reset.<br>This bit is cleared for the DAC output 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.<br>This bit is cleared for the GPIO pins and registers to return to their default state. |

# OTHER ANALOG PERIPHERALS

## DIGITAL-TO-ANALOG CONVERTERS

The ADuC7121 incorporates four buffered 12-bit voltage output string digital-to-analog converters (DACs) on chip. Each DAC has a rail-to-rail voltage output buffer capable of driving 5 kΩ/100 pF.

Each DAC has three selectable ranges: 0 V to $V_{REF}$ (internal band gap 2.5 V reference), 0 V to $AV_{DD}$, and 0 V to EXT_REF (see Figure 28). The signal range is 0 V to $AV_{DD}$. Note that the DAC can also operate in interpolation mode.

### *MMR Interfaces*

Each DAC is independently configurable through a control register and a data register. These two registers are identical for the 12 DACs. Only DAC0CON and DAC0DAT are described in detail in this section.

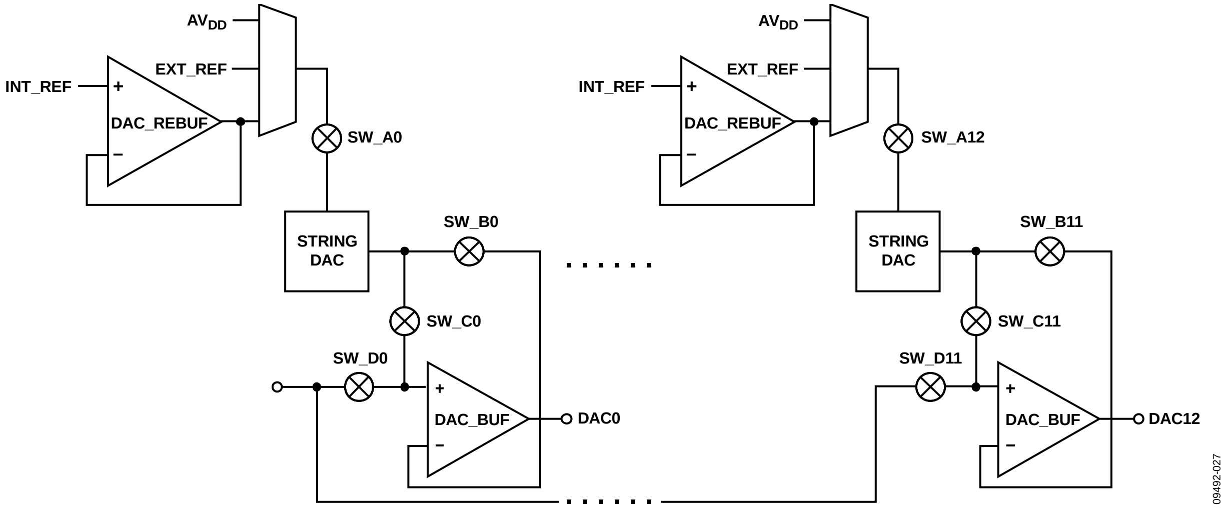

*Figure 28. DAC Configuration*

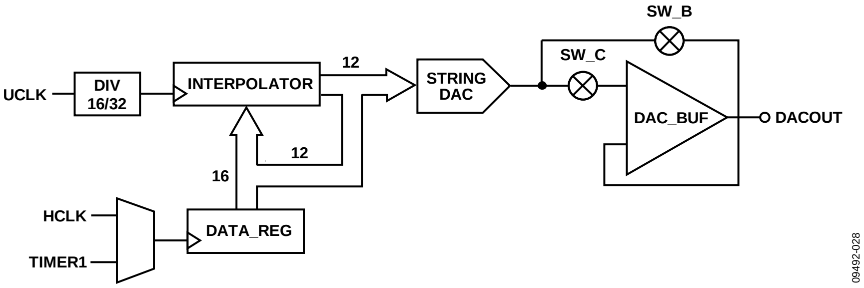

*Figure 29. DAC User Functionality*

Table 51. DACxCON Registers (Default Value = 0x100, Read/Write Access)

| Name | Address |
|---|---|
| DAC0CON | 0xFFFF0580 |
| DAC1CON | 0xFFFF0588 |
| DAC2CON | 0xFFFF05B0 |
| DAC3CON | 0xFFFF05D8 |

Table 52. DAC0CON MMR Bit Designations

| Bit | Value | Name | Description |
|---|---|---|---|
| 15:9 | 0 | | Reserved. |
| 8 | 1 | DACPD | DAC power-down. Set by the user to set DACOUTx to tristate mode. |
| 7 | 0 | DACBUF_LP | DAC buffer low power mode. Set by the user to place DAC_BUFF into a low power mode. |
| 6 | 0 | BYP | DAC bypass bit.<br>Set this bit to bypass the DAC buffer.<br>Cleared to buffer the DAC output. |
| 5 | 0 | DACCLK | DAC update rate.<br>Set by the user to update the DAC using Timer1.<br>Cleared by the user to update the DAC using HCLK (core clock). |
| 4 | 0 | DACCLR | DAC clear bit.<br>Set by the user to enable normal DAC operation.<br>Cleared by the user to reset data register of the DAC to 0. |
| 3 | 0 | Mode | Mode bit.<br>Set by the user to operate on DAC normal mode and turn off the interpolator clock source. Cleared by the user to enable the interpolation mode. |
| 2 | 0 | Rate | Rate bit. Set by the user to enable the interpolation clock to HCLK/16. Cleared by the user to HCLK/32. |
| 1:0 | | DACRNx | DAC range bits. |
| | 00 | | DAC range is from AGND to the internal reference. |
| | 01 | | EXT_REF DAC range is from AGND to the external reference. See the REFCON MMR in Table 39 for details. |
| | 10 | | EXT_REF DAC range is from AGND to the external reference. See the REFCON MMR in Table 39 for more details. |
| | 11 | | AVDD and AGND. |

Table 53. DACxDAT Registers (Default Value = 0x00000000, Read/Write Access)

| Name | Address |
|---|---|
| DAC0DAT | 0xFFFF0584 |
| DAC1DAT | 0xFFFF058C |
| DAC2DAT | 0xFFFF05B4 |
| DAC3DAT | 0xFFFF05DC |

Table 54. DACxDAT MMR Bit Designations

| Bit | Description |
|---|---|
| 31:28 | Reserved. |
| 27:16 | 12-bit data for DACx. |
| 15:12 | Extra bits for interpolation mode. |
| 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 30.

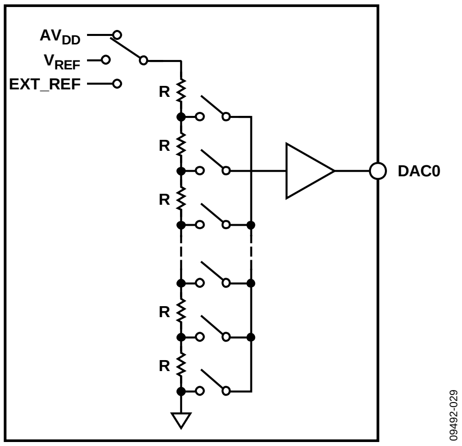

*Figure 30. DAC Structure*

As shown in Figure 30, the reference source for each DAC is user-selectable in software. It can be either $AV_{DD}$, $V_{REF}$, or EXT_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-EXT_REF mode, the DAC output transfer function spans from 0 V to the voltage at the $V_{REF}$_2.5 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 linearity specification of the DAC (when driving a 5 kΩ resistive load to ground) is guaranteed through the full transfer function except for 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 shown in Figure 31. The dotted line in Figure 31 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 31 represents a transfer function in 0 V-to-$AV_{DD}$ mode only. In 0 V-to-$V_{REF}$ or 0 V-to-EXT_REF modes (with $V_{REF} < AV_{DD}$ or EXT_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.

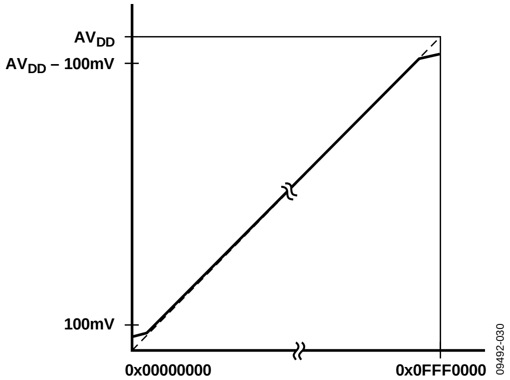

*Figure 31. Endpoint Nonlinearities Due to Amplifier Saturation*

The endpoint nonlinearities conceptually illustrated in Figure 31 worsen as a function of output loading. The ADuC7121 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 31 become larger. With larger current demands, this can significantly limit output voltage swing.

## LDO (LOW DROPOUT REGULATOR)

The ADuC7121 contains an integrated LDO, which generates the core supply voltage (DVDD) of approximately 2.6 V from the IOVDD supply. As the LDO is driven from IOVDD, the IOVDD supply voltage needs to be greater than 2.7 V.

An external compensation capacitor (CT) of 0.47 µF with low ESR must be placed very close to each of the DVDD pins. This capacitor also acts as a storage tank of charge, and supplies an instantaneous charge required by the core, particularly at the positive edge of the core clock (HCLK).

The DVDD voltage generated by the LDO is solely for providing a supply for the ADuC7121. Therefore, users should not use a DVDD pin as the power supply pin for any other chip. In addition, it is recommended that the IOVDD has excellent power supply decoupling to help improve line regulation performance of the LDO.

The DVDD pin has no reverse battery, current limit, or thermal shutdown protection; therefore, it is essential that users of the ADuC7121 do not short this pin to ground at any time during normal operation or during board manufacture.

## CURRENT OUTPUT DACs (IDAC)

The ADuC7121 provides five current output digital-to-analog converters (DACs). The current sources (five current DACs) feature low noise and low drift high-side current output with 11-bit resolution. The five IDACs are as follows: IDAC0 with 250 mA full-scale (FS) output, IDAC1 with 200 mA FS output, IDAC2 with 45 mA FS output, IDAC3 with 80 mA FS output, and IDAC4 with 20 mA FS output.

The reference current of each IDAC is generated by a precision internal band gap voltage reference and an external precision resistor, and as such, the gain error of each IDAC is impacted by the accuracy of the external resistor. Connect the resistor to the $I_{REF}$ pin. The noise of each IDAC is limited by its damping capacitor, $C_{DAMP}$, which is selected to band limit noise as well as to meet the signal bandwidth. Connect $C_{DAMP}$_IDACx to PVDD.

An NMOS switch is provided to shut down the IDAC0 diode. Note that the output current switches off while this switch is on. When the switch is on, the IDAC0 pin is able to withstand −0.5 V. At power-up or reset, IDAC0 is powered down by default and its output is high impedance. When enabled, the IDAC0 output current does not overshoot.

To reduce the heat dissipation on chip, a separate power supply can be used. An internal LDO provides a stable 2.5 V supply for all low current internal IDACs.

### *Precision Current Generation and Fault Protection*

The reference current is generated either from an on-chip precision band gap voltage source or from an external voltage reference by default, which is applied to an external precision resistor. This resistor is connected to the $I_{REF}$ pin. The band gap is factory trimmed to obtain a precise initial value and low temperature drift. The external resistor is an assumed 0.1% accuracy with 5 ppm drift, and a 0.1 µF external capacitor is required to bypass high frequency noise.

A fault detection block is included to stop problems from occurring if too small a reference resistor is detected. By sending the developed reference current into an on-board resistor of half the expected size of the external resistor, a fault signal is generated if the resistor is less than half the expected value (to an accuracy of about 20%). The external resistor value is calculated by $R_{EXT} = V_{REF}/I_{REF}$, where $I_{REF} = 380$ µA and $V_{REF}$ is the selected reference voltage for the voltage-to-current circuit.

### *IDAC and Output Stage Fault Protection*

All five IDACs use the same architecture to generate high-side current whereby only the section that generates the reference current is shared. A low current is generated first using a current-mode DAC, which is then mirrored up to give the large output current that is desired. A thermal shutdown circuit protects the chip from overheating. The IDACs are guaranteed monotonic to within 11 bits of resolution.

The bandwidth limit is provided by a programmable internal resistor and an external capacitor. This is to filter high frequency noise. It is also used to generate a triangle wave from a square wave input for the IDAC4 only.

The thermal shutdown circuit automatically shuts down all of the output stages when the chip temperature exceeds a certain threshold. The intention of the thermal shutdown is only for protection in the case of a short on an IDAC output. The overheating of the chip from other causes also triggers a thermal shutdown but only the IDAC output stage is automatically shut down. It triggers an interrupt and sets the TSHUT bit in the IDACSTA register to indicate the overheating of the chip.

In case the digital core malfunctions at a temperature lower than the thermal shutdown trigger point, the circuit can still shut down the IDAC, but a watchdog reset must be used to reset the chip. The TSHUT bit retains its value after a software reset or a watchdog reset. This bit can only be cleared by a power-on reset, a hardware reset, or when 0 is written to the IDACSTA register.

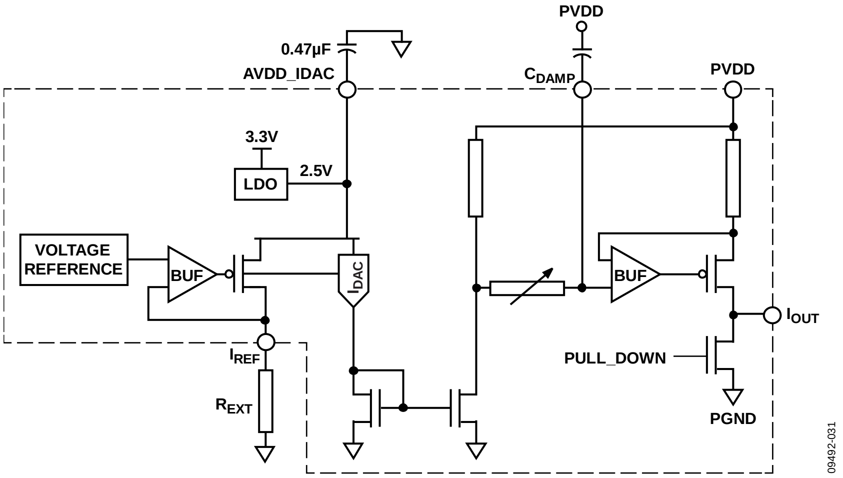

*Figure 32.*

## IDAC MMRs

Table 55. IDAC Control Registers (Read and Write Access)

| Name | Address (Hex) | Default Value |
|---|---|---|
| IDAC0CON | 0xFFFF0700 | 0x0010 |
| IDAC1CON | 0xFFFF070C | 0x0010 |
| IDAC2CON | 0xFFFF0718 | 0x0010 |
| IDAC3CON | 0xFFFF0724 | 0x0010 |
| IDAC4CON | 0xFFFF0730 | 0x0010 |
| TDSCON | 0xFFFF073C | 0x00 |
| IDAC0PULLDOWN | 0xFFFF0744 | 0x00 |

Table 56. IDACxCON MMR Bit Designations

| Bit | Name | Value | Description |
|---|---|---|---|
| 15:9 | | | These bits are reserved. |
| 8:7 | SFHMODE | | Bit shuffling is a method of increasing the ac precision of an IDAC. Do not use in applications where dc performance is important. |
| | | 00 | Shuffle one increment at a time. |
| | | 01 | Shuffle based on an internal counter. |
| | | 10 | Shuffle based on the input data. |
| | | 11 | Reserved. |
| 6 | MSBSHFEN | 0 | MSB shuffle enable.<br>Set by the user to 1 to enable MSB shuffling.<br>Set by the user to 0 to disable MSB shuffling. |
| 5 | LSBSHFEN | 0 | LSB shuffle enable.<br>Set by the user to 1 to enable LSB shuffling.<br>Set by the user to 0 to disable LSB shuffling. |
| 4 | IDACPD | 1 | IDAC power-down bit.<br>Set by the user to 1 to power down the IDAC. IDAC output is high impedance.<br>Set by the user to 0 to power up the IDAC. |
| 3 | IDACCLK | 0 | IDAC update rate.<br>Set by the user to update the IDAC using Timer1.<br>Cleared by the user to update the IDAC using HCLK (core clock). |
| 2 | IDACCLR | 0 | IDAC clear bit.<br>Set by the user to enable normal IDAC operation.<br>Cleared by the user to reset data register of the IDAC to 0. |
| 1 | Mode | 0 | Mode bit. This bit must always be cleared. |
| 0 | Reserved | 0 | Set this bit to 0. |

Table 57. TDSCON MMR Bit Designations

| Bit | Value | Name | Description |
|---|---|---|---|
| 7:3 | | Reserved | The user sets these bits to 0. |
| 2 | | Reserved | The user must set this to 1. |
| 1 | 0 | DISINT | Disable thermal trigger interrupt.<br>Set by the user to 0 to generate an interrupt if the temperature passes the thermal shutdown point. |
| 0 | 0 | DISSD | Set by the user to 0 to disable the output current DACs when the temperature passes a trip point. |

Table 58. IDAC0PULLDOWN MMR Bit Designations

| Bit | Value | Name | Description |
|---|---|---|---|
| 7:6 | | Reserved | These bits are set to 0 by the user. |
| 5 | 0 | Pulldown | IDAC0 pull-down.<br>Set to 1 by the user to pull down the IDAC0 pin as well as power down the IDAC0.<br>Set to 0 by the user to disable the pull-down. |
| 4 | 0 | PLA_PD_EN | PLA output trigger enable.<br>Set to 1 by the user to enable the PLA output to trigger the IDAC0 pull-down.<br>Set to 0 by the user to disable this feature. |
| 3:0 | | PLA Source | PLA output source for PLA output trigger enable.<br>Can select the output of any element, 0 to 15, by programming these bits with the corresponding binary value. |

Table 59. IDAC Data Registers (Default Value = 0x00000000, Read and Write Access)

| Name | Address (Hex) |
|---|---|
| IDAC0DAT | 0xFFFF0704 |
| IDAC1DAT | 0xFFFF0710 |
| IDAC2DAT | 0xFFFF071C |
| IDAC3DAT | 0xFFFF0728 |
| IDAC4DAT | 0xFFFF0734 |

Table 60. IDACxDAT MMR Bit Designations

| Bit | Name | Value | Description |
|---|---|---|---|
| 31:28 | Reserved | | These bits are reserved. |
| 27:17 | Data | | Data from IDACx. |
| 16:0 | Reserved | 000 | These bits are reserved. |

**Table 61. IDAC Bandwidth Registers (Default Value = 0x00, Read and Write Access)**

| Name | Address |
|---|---|
| IDAC0BW | 0xFFFF0708 |
| IDAC1BW | 0xFFFF0714 |
| IDAC2BW | 0xFFFF0720 |
| IDAC3BW | 0xFFFF072C |
| IDAC4BW | 0xFFFF0738 |

**Table 62. IDACxBW MMR Bit Designations**

| Bit | Name | Value | Description |
|---|---|---|---|
| 7:4 | Reserved | | The user sets these bits to 0. |
| 3:0 | BW | | Bandwidth control bits. Defines the 3 dB bandwidth of the RC low-pass filter, assuming a 0.01 µF capacitor on the $C_{DAMP}$_IDACx pins of the IDACx. |
| | | 000 | 100 kHz. |
| | | 001 | 28.7 kHz. |
| | | 010 | 15 kHz. |
| | | 011 | 7.8 kHz. |
| | | 100 | 4 kHz. |
| | | 101 | 2.2 kHz. |
| | | 110 | 1.2 kHz. |
| | | Others | Not defined. |

**Table 63. IDAC Status Register (Default Value = 0x00, Read and Write Access)**

| Name | Address (Hex) |
|---|---|
| IDACSTA | 0xFFFF0740 |

**Table 64. IDACSTA MMR Bit Designations**

| Bit | Value | Name | Description |
|---|---|---|---|
| 7:2 | | Reserved | These bits are set to 0 by the user. |
| 1 | 0 | TSHUT | Thermal shutdown error status bit.<br>Set to 1 by the core indicating a thermal shutdown event.<br>Set to 0 by the core indicating the IDACs are within operating temperature. |
| 0 | | Reserved | Reserved. |

## OSCILLATOR AND PLL—POWER CONTROL

The ADuC7121 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 for 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 XCLK pin as described in Figure 33. Note that when the XCLK 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 ADuC7121.

The operating mode, clocking mode, and programmable clock divider are controlled via two MMRs, PLLCON (see Table 67) and POWCON (see Table 68). PLLCON controls the operating mode of the clock system, and POWCON controls the core clock frequency and the power-down mode.

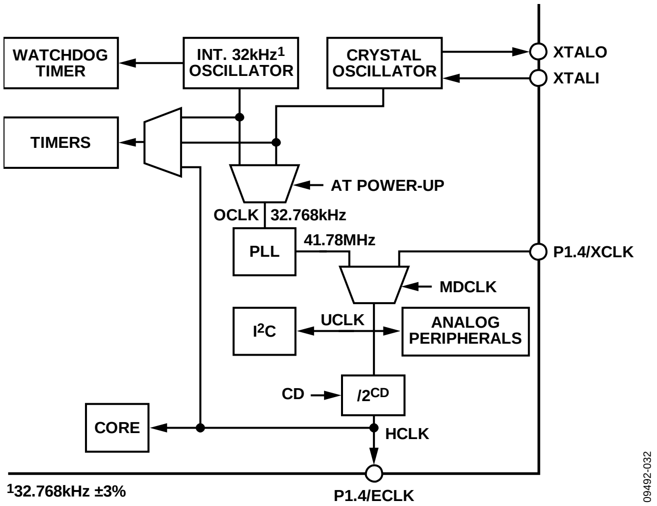

*Figure 33. Clocking System*

### *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 writing 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.

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 register can determine if the reset came from the watchdog timer.

### 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 = 0x01;
  PLLKEY2 = 0x55;

  POWKEY1 = 0x01;
  POWCON =
0x27;                              // set
core into nap mode
  POWKEY2 = 0xF4;
```

### *External Clock Selection*

To switch to an external clock on P1.4 (of the P1.4/PWM1/ECLK/XCLK/PLAI[8] pin), configure P1.4 in Mode 2. The external clock can be up to 41.78 MHz.

### 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 ADuC7121.

Table 65 describes what part of the ADuC7121 is powered on in the different modes and indicates the power-up time. Table 66 gives some typical values of the total current consumption (analog + digital supply currents) in the different modes, depending on the clock divider bits. 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 65. Operating Modes

| Mode | Core | Peripherals | PLL | XTAL/Timer2/Timer3 | External IRQ | Start-Up/Power-On Time |
|---|---|---|---|---|---|---|
| Active | On | On | On | On | On | 66 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 66. 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 | 33.1 | 21.2 | 13.8 | 10 | 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 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |
| 100 | Stop | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 | 0.4 |

### MMRs and Keys

To prevent accidental programming, a certain sequence must be followed when writing in the PLLCON and POWCON registers (see Table 69).

### PLLKEYx Registers

| | |
|---|---|
| Name: | PLLKEY1 |
| Address: | 0xFFFF0410 |
| Default value: | 0x0000 |
| Access: | Write only |

| | |
|---|---|
| Name: | PLLKEY2 |
| Address: | 0xFFFF0418 |
| Default value: | 0x0000 |
| Access: | Write only |

### PLLCON Register

| | |
|---|---|
| Name: | PLLCON |
| Address: | 0xFFFF0414 |
| Default value: | 0x21 |
| Access: | Read and write |

Table 67. PLLCON MMR Bit Designations

| Bit | Value | Name | Description |
|---|---|---|---|
| 7:6 | | | Reserved. |
| 5 | | OSEL | 32 kHz PLL input selection. |
| | | | Set by the user to use the internal 32 kHz oscillator. |
| | | | Set by default. |
| | | | 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 the P1.4/PWM1/ECLK/XCLK/PLAI[8] pin. |

### POWKEYx Registers

| | |
|---|---|
| Name: | POWKEY1 |
| Address: | 0xFFFF0404 |
| Default value: | 0x0000 |
| Access: | Write only |

| | |
|---|---|
| Name: | POWKEY2 |
| Address: | 0xFFFF040C |
| Default value: | 0x0000 |
| Access: | Write only |

### POWCON Register

| | |
|---|---|
| Name: | POWCON |
| Address: | 0xFFFF0408 |
| Default value: | 0x0003 |
| Access: | Read and write |

Table 68. POWCON MMR Bit Designations

| Bit | Value | Name | Description |
|---|---|---|---|
| 7 | | | Reserved. |
| 6:4 | | PC | Operating modes. |
| | 000 | | Active mode. |
| | 001 | | Pause mode. |
| | 010 | | Nap mode. |
| | 011 | | Sleep mode. IRQ0 to IRQ3 and Timer2 can wake up the ADuC7121. |
| | 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. |

Table 69. PLLCON and POWCON Write Sequence

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

# DIGITAL PERIPHERALS

## PWM GENERAL OVERVIEW

The ADuC7121 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 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.

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 34.

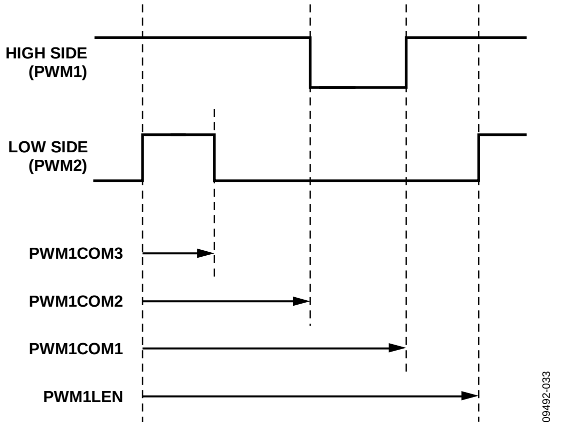

*Figure 34. PWM Timing*

The PWM clock is selectable via PWMCON1 with one of the following values: UCLK divide-by-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 with the PWM1 and PWM2 waveforms above.

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.

**Table 70. PWM MMRs**

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

**Table 71. PWMCON1 MMR Bit Designations (Address = 0xFFFF0F80, Default Value = 0x0012)**

| Bit | Name | Description |
|---|---|---|
| 15 | Reserved | This bit is reserved. |
| 14 | SYNC | Enables PWM synchronization.<br>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 SYNC of the P0.3/MISO/PLAO[12]/SYNC pin.<br>Cleared by the user to ignore transitions on SYNC of the P0.3/MISO/PLAO[12]/SYNC pin. |
| 13 | PWM6INV | Set to 1 by the user to invert PWM6.<br>Cleared by the user to use PWM6 in normal mode. |
| 12 | PWM4NV | Set to 1 by the user to invert PWM4.<br>Cleared by the user to use PWM4 in normal mode. |
| 11 | PWM2INV | Set to 1 by the user to invert PWM2.<br>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.<br>Cleared by the user to disable the PWMTRIP interrupt. |
| 9 | ENA | If HOFF = 0 and HMODE = 1. If HOFF = 1 and HMODE = 1, see Table 72. If not in H-Bridge mode, this bit has no effect.<br>Set to 1 by the user to enable PWM outputs.<br>Cleared by the user to disable PWM outputs. |
| 8:6 | PWMCP[2:0] | PWM clock prescaler bits. Sets the UCLK divider.<br>000 = UCLK divide-by-2.<br>001 = UCLK divide-by-4.<br>010 = UCLK divide-by-8.<br>011 = UCLK divide-by-16.<br>100 = UCLK divide-by-32.<br>101 = UCLK divide-by-64.<br>110 = UCLK divide-by-128.<br>111 = UCLK divide-by-256. |
| 5 | POINV | Set to 1 by the user to invert all PWM outputs.<br>Cleared by the user to use PWM outputs as normal. |
| 4 | HOFF | High-side off.<br>Set to 1 by the user to force PWM1 and PWM3 outputs high. This also forces PWM2 and PWM4 low.<br>Cleared by the user to use the PWM outputs as normal. |
| 3 | LCOMP | Load compare registers.<br>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.<br>Cleared by the user to use the values previously stored in the internal compare registers. |
| 2 | DIR | Direction control.<br>Set to 1 by the user to enable PWM1 and PWM2 as the output signals while PWM3 and PWM4 are held low.<br>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.<br>Set to 1 by the user to enable H-Bridge mode and Bits[5:2] of PWMCON1.<br>Cleared by the user to operate the PWMs in standard mode. |
| 0 | PWMEN | Set to 1 by the user to enable all PWM outputs.<br>Cleared by the user to disable all PWM outputs. |

In H-bridge mode, HMODE = 1 and Table 71 determine the PWM outputs, as listed in Table 72.

**Table 72. PWM Output Selection**

| PWMCOM1 MMR | | | | PWM Outputs | | | |
|---|---|---|---|---|---|---|---|
| **ENA** | **HOFF** | **POINV** | **DIR** | **PWM1** | **PWM2** | **PWMR3** | **PWM4** |
| 0 | 0 | X[1] | X[1] | 1 | 1 | 1 | 1 |
| X[1] | 1 | X[1] | X[1] | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | HS[1] | LS[1] |
| 1 | 0 | 0 | 1 | HS[1] | LS[1] | 0 | 0 |
| 1 | 0 | 1 | 0 | HS[1] | LS[1] | 1 | 1 |
| 1 | 0 | 1 | 1 | 1 | 1 | HS[1] | LS[1] |

[1] HS is high side, LS is low side, X is a don't care bit.

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 73).

**Table 73. Compare Register (Default Value = 0x0000, Access is Read/Write)**

| Name | Address | Default Value | Access |
|---|---|---|---|
| PWM1COM1 | 0xFFFF0F84 | 0x0000 | R/W |
| PWM1COM2 | 0xFFFF0F88 | 0x0000 | R/W |
| PWM1COM3 | 0xFFFF0F8C | 0x0000 | R/W |
| PWM2COM1 | 0xFFFF0F94 | 0x0000 | R/W |
| PWM2COM2 | 0xFFFF0F98 | 0x0000 | R/W |
| PWM2COM3 | 0xFFFF0F9C | 0x0000 | R/W |
| PWM3COM1 | 0xFFFF0FA4 | 0x0000 | R/W |
| PWM3COM2 | 0xFFFF0FA8 | 0x0000 | R/W |
| PWM3COM3 | 0xFFFF0FAC | 0x0000 | 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 74. PWMCON2 MMR Bit Designations (Address = 0xFFFF0FB4, Default Value = 0x00)**

| Bit | Name | Value | Description |
|---|---|---|---|
| 7 | CSEN | | Set to 1 by the user to enable the PWM to generate a convert start signal.<br>Cleared by the user to disable the PWM convert start signal. |
| 3:0 | CSD3 to CSD0 | | 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 following example 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 35).

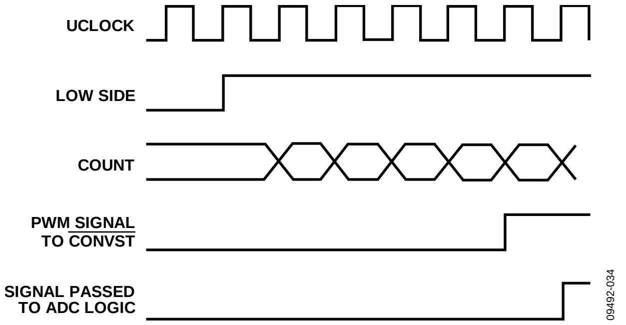

*Figure 35. ADC Conversion*

# GENERAL-PURPOSE INPUT/OUTPUT

The ADuC7121 provides 32 general-purpose, bidirectional input/output (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 75). 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 into four ports: Port 0 to Port 3. Each port is controlled by four or five MMRs, with x representing the port number.

### GPxCON Registers

| | |
|---|---|
| Name: | GP0CON |
| Address: | 0xFFFF0D00 |
| Default value: | 0x11000000 |
| Access: | Read and write |

| | |
|---|---|
| Name: | GP1CON |
| Address: | 0xFFFF0D04 |
| Default value: | 0x00000000 |
| Access: | Read and write |

| | |
|---|---|
| Name: | GP2CON |
| Address: | 0xFFFF0D08 |
| Default value: | 0x00000000 |
| Access: | Read and write |

| | |
|---|---|
| Name: | GP3CON |
| Address: | 0xFFFF0D0C |
| Default value: | 0x00000000 |
| Access: | Read and write |

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 ADuC7121 device enters 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 75.

**Table 75. GPIO Pin Function Designations**

| Port | Pin | Configuration (See GPxCON Table 76) | | | |
|---|---|---|---|---|---|
| | | **00** | **01** | **10** | **11** |
| 0 | P0.0 | GPIO | SCL0 | | PLAI[5] |
| | P0.1 | GPIO | SDA0 | JTAG disabled | PLAI[4] |
| | P0.2 | GPIO | SPICLK | JTAG disabled $\text{ADC}_{\text{BUSY}}$ | PLAO[13] |
| | P0.3 | GPIO | MISO | SYNC (PWM) | PLAO[12] |
| | P0.4 | GPIO | MOSI | TRIP (PWM) | PLAI[11] |
| | P0.5 | GPIO | $\overline{\text{CS}}$ | $\text{ADC}_{\overline{\text{CONVST}}}$ | PLAI[10] |
| | P0.6 | GPIO | $\overline{\text{MRST}}$ | | PLAI[2] |
| | P0.7 | GPIO | $\overline{\text{TRST}}$ | | PLAI[3] |
| 1 | P1.0 | GPIO | SIN | SCL1 | PLAI[7] |
| | P1.1 | GPIO | SOUT | SDA1 | PLAI[6] |
| | P1.2[1] | TDI (JTAG) | | | PLAO[15] |
| | P1.3[1] | TDO (JTAG) | | | PLAO[14] |
| | P1.4 | GPIO | PWM1 | ECLK/XCLK | PLAI[8] |
| | P1.5 | GPIO | PWM2 | | PLAI[9] |
| | P1.6 | GPIO | | | PLAO[5] |
| | P1.7 | GPIO | | | PLAO[4] |
| 2 | P2.0 | GPIO/IRQ0 | | | PLAI[13] |
| | P2.1 | GPIO/IRQ1 | | | PLAI[12] |
| | P2.2 | GPIO | | | PLAI[1] |
| | P2.3 | GPIO/IRQ2 | | | PLAI[14] |
| | P2.4 | GPIO | PWM5 | | PLAO[7] |
| | P2.5 | GPIO | PWM6 | | PLAO[6] |
| | P2.6 | GPIO/IRQ3 | | | PLAI[15] |
| | P2.7 | GPIO | | | PLAI[0] |
| 3 | P3.0 | GPIO | | | PLAO[0] |
| | P3.1 | GPIO | | | PLAO[1] |
| | P3.2 | GPIO/IRQ4 | PWM3 | | PLAO[2] |
| | P3.3 | GPIO/IRQ5 | PWM4 | | PLAO[3] |
| | P3.4 | GPIO | | | PLAO[8] |
| | P3.5 | GPIO | | | PLAO[9] |
| | P3.6 | GPIO | | | PLAO[10] |
| | P3.7 | GPIO/$\overline{\text{BM}}$ | | | PLAO[11] |

[1] Reconfiguring these pins disables JTAG mode. Erase part to reenable JTAG access after changing default value.

Table 76. GPxCON MMR Bit Designations

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

### *GPxPAR Registers*

The GPxPAR registers program 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.

| | |
|---|---|
| Name: | GP0PAR |
| Address: | 0xFFFF0D2C |
| Default value: | 0x20000000 |
| Access: | Read and write |

| | |
|---|---|
| Name: | GP1PAR |
| Address: | 0xFFFF0D3C |
| Default value: | 0x00000000 |
| Access: | Read and write |

| | |
|---|---|
| Name: | GP2PAR |
| Address: | 0xFFFF0D4C |
| Default value: | 0x00000000 |
| Access: | Read and write |

| | |
|---|---|
| Name: | GP3PAR |
| Address: | 0xFFFF0D5C |
| Default value: | 0x00222222 |
| Access: | Read and write |

Table 77. GPxPAR MMR Bit Designations

| Bit | Description |
|---|---|
| 31:29 | Reserved |
| 28 | Pull-up disable Px.7 pin<br>Set to 1 to enable the pull-up<br>Clear to 0 to disable the pull-up |
| 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 |

### *GPxDAT Register*

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 output, and receives and stores the input value of the pins configured as inputs.

| | |
|---|---|
| Name: | GP0DAT |
| Address: | 0xFFFF0D20 |
| Default value: | 0x000000XX |
| Access: | Read and write |

| | |
|---|---|
| Name: | GP1DAT |
| Address: | 0xFFFF0D30 |
| Default value: | 0x000000XX |
| Access: | Read and write |

| | |
|---|---|
| Name: | GP2DAT |
| Address: | 0xFFFF0D40 |
| Default value: | 0x000000XX |
| Access: | Read and write |

| | |
|---|---|
| Name: | GP3DAT |
| Address: | 0xFFFF0D50 |
| Default value: | 0x000000XX |
| Access: | Read and write |

**Table 78. GPxDAT MMR Bit Designations**

| Bit | Description |
|---|---|
| 31:24 | Direction of the data.<br>Set to 1 by the user to configure the GPIO pin as an output.<br>Cleared to 0 by the 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). |

### *GPxSET Registers*

The GPxSET registers provide a data set for the Port x registers.

| | |
|---|---|
| Name: | GP0SET |
| Address: | 0xFFFF0D24 |
| Default value: | 0x000000XX |
| Access: | Write only |

| | |
|---|---|
| Name: | GP1SET |
| Address: | 0xFFFF0D34 |
| Default value: | 0x000000XX |
| Access: | Write only |

| | |
|---|---|
| Name: | GP2SET |
| Address: | 0xFFFF0D44 |
| Default value: | 0x000000XX |
| Access: | Write only |

| | |
|---|---|
| Name: | GP3SET |
| Address: | 0xFFFF0D54 |
| Default value: | 0x000000XX |
| Access: | Write only |

**Table 79. GPxSET MMR Bit Designations**

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

### *GPxCLR Registers*

The GPxCLR registers are data clear for Port x registers.

| | |
|---|---|
| Name: | GP0CLR |
| Address: | 0xFFFF0D28 |
| Default value: | 0x000000XX |
| Access: | Write only |

| | |
|---|---|
| Name: | GP1CLR |
| Address: | 0xFFFF0D38 |
| Default value: | 0x000000XX |
| Access: | Write only |

| | |
|---|---|
| Name: | GP2CLR |
| Address: | 0xFFFF0D48 |
| Default value: | 0x000000XX |
| Access: | Write only |

| | |
|---|---|
| Name: | GP3CLR |
| Address: | 0xFFFF0D58 |
| Default value: | 0x000000XX |
| Access: | Write only |

**Table 80. GPxCLR MMR Bit Designations**

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

### GPxOCE Registers

Open-collector functionality is available on the following GPIO pins: P1.7, P1.6, Port 2, and Port 3.

**Table 81. GPxOCE MMR Bit Designations**

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

# UART SERIAL INTERFACE

The ADuC7121 features a 16,450-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. The UART functionality is available on the P1.0/SIN/SCL1/PLAI[7] and P1.1/SOUT/SDA1/PLAI[6] pins of the ADuC7121.

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 ADuC7121 features two methods of generating the UART baud rate: normal 450 UART baud rate generation and ADuC7121 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

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

Table 82 lists common baud rate values.

**Table 82. 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% |

### *Fractional Divider*

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

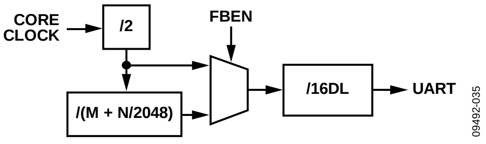

*Figure 36. Baud Rate Generation Options*

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

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

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

For example, generation of 19,200 baud

$$M + \frac{N}{2048} = \frac{41.78\,\text{MHz}}{19200 \times 16 \times 67 \times 2}$$

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

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 10 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 and COMRX can be accessed when Bit 7 in the COMCON0 register is cleared. COMDIV0 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 and 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 and 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 and write

Table 83. COMCON0 MMR Bit Designations

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

Table 84. COMCON1 MMR Bit Designations

| Bit | Name | Description |
|---|---|---|
| 7:5 | | Reserved bits. Not used. |
| 4 | LOOPBACK | Loopback. Set by the user to enable loopback mode. In loopback mode, SOUT is forced high. |
| 3:2 | | Reserved bits. Not used. |
| 1 | RTS | Request to send.<br>Set by the user to force the RTS output to 0.<br>Cleared by the user to force the RTS output to 1. |
| 0 | DTR | Data terminal ready.<br>Set by the user to force the DTR output to 0.<br>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 85. COMSTA0 MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 7 | | Reserved. |
| 6 | TEMT | COMTX and shift register empty status bit.<br>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.<br>Cleared automatically when writing to COMTX. |
| 5 | THRE | COMTX empty status bit.<br>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.<br>Cleared automatically when writing to COMTX. |
| 4 | BI | Break indicator.<br>Set when SIN of the P1.0/SIN/SCL1/PLAI[7] pin is held low for more than the maximum word length.<br>Cleared automatically. |
| 3 | FE | Framing error.<br>Set when the stop bit is invalid.<br>Cleared automatically. |
| 2 | PE | Parity error.<br>Set when a parity error occurs.<br>Cleared automatically. |
| 1 | OE | Overrun error.<br>Set automatically if data are overwritten before being read.<br>Cleared automatically. |
| 0 | DR | Data ready.<br>Set automatically when COMRX is full.<br>Cleared by reading COMRX. |

### UART Interrupt Enable Register 0

This 8-bit register enables and disables the individual UART interrupt sources.

Name: COMIEN0

Address: 0xFFFF0804

Default value: 0x00

Access: Read and write

Table 86. COMIEN0 MMR Bit Designations

| Bit | Name | Description |
|---|---|---|
| 7 to 3 | | Reserved. Not used.<br>Cleared by the user. |
| 2 | ELSI | Receive pin (SIN) status interrupt enable bit.<br>Set by the user to enable generation of an interrupt if any of the COMSTA0[3:1] register bits are set.<br>Cleared by the user. |
| 1 | ETBEI | Enable transmit buffer empty interrupt.<br>Set by the user to enable an interrupt when the buffer is empty during a transmission, that is, when COMSTA[5] is set.<br>Cleared by the user. |
| 0 | ERBFI | Enable receive buffer full interrupt.<br>Set by the user to enable an interrupt when the buffer is full during a reception.<br>Cleared by the user. |

### UART Interrupt Identification Register 0

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

Name: COMIID0

Address: 0xFFFF0808

Default value: 0x01

Access: Read only

Table 87. 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 ADuC7121.

Name: COMDIV2

Address: 0xFFFF082C

Default value: 0x0000

Access: Read and write

Table 88. COMDIV2 MMR Bit Designations

| Bit | Name | Description |
|---|---|---|
| 15 | FBEN | Fractional baud rate generator enable bit.<br>Set by the user to enable the fractional baud rate generator.<br>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 82 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 82 for common baud rate values. |

# I²C PERIPHERALS

The ADuC7121 incorporates two I²C peripherals that may be configured as a fully I²C-compatible bus master device or as a fully I²C-compatible bus slave device. Both peripherals are identical.

The two pins used for data transfer, SDA and SCL, are configured in a wired-AND 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 address of the I²C bus peripheral 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 (R/$\overline{\text{W}}$) during the initial address transfer. If the master does not lose arbitration and the slave acknowledges, 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 ADuC7121 includes the following features:

- Support for repeated start conditions. In master mode, the ADuC7121 can be programmed to generate a repeated start. In slave mode, the ADuC7121 recognizes repeated start conditions.
- In master and slave modes, the device recognizes both 7-bit and 10-bit bus addresses.
- In I²C Master mode, the ADuC7121 supports continuous reads from a single slave up to 512 bytes in a single transfer sequence.
- Clock stretching can be enabled by other devices on the bus without causing any issues with the ADuC7121. However, the ADuC7121 cannot enable clock stretching.
- In slave mode, the ADuC7121 can be programmed to return a no acknowledge. This allows the validation 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 ADuC7121 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 (SCL0, SDA0), Bit 0 and Bit 4 of the GP0CON register must be set to 1 to enable I²C mode. To configure the I²C1 pins (SCL1, SDA1), Bit 1 and Bit 5 of the GP1CON register must be set to 1 to enable I²C mode, as shown in the General-Purpose Input/Output section.

## 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 I2CDIV MMR as follows:

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

where:

$f_{UCLK}$ = clock before the clock divider.
$DIVH$ = the high period of the clock.
$DIVL$ = the low period of the clock.

Thus, for 100 kHz operation

$$DIVH = DIVL = \text{0xCF}$$

and for 400 kHz

$$DIVH = \text{0x28},\ DIVL = \text{0x3C}$$

The I2CDIV register corresponds to DIVH:DIVL.

## I²C BUS ADDRESSES

### Slave Mode

In slave mode, the registers I2CxID0, I2CxID1, I2CxID2, and I2CxID3 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 7 MSBs of either ID register must be identical to that of the 7 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 ADuC7121 also supports 10-bit addressing mode. When Bit 1 of I2CxSCON (ADR10EN bit) is set to 1, then one 10-bit address is supported in slave mode and is stored in registers I2CxID0 and I2CxID1. The 10-bit address is derived as follows:

I2CxID0[0] is the read/$\overline{\text{write}}$ 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 I2CxADR0 register is programmed with the I²C address of the device.

In 7-bit address mode, I2CxADR0[7:1] are set to the device address. I2CxADR0[0] is the read/$\overline{\text{write}}$ bit.

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

- I2CxADR0[7:3] must be set to 11110b.
- I2CxADR0[2:1] = Address Bits[9:8].
- I2CxADR1[7:0] = Address Bits[7:0].
- I2CxADR0[0] is the read/$\overline{\text{write}}$ bit.

## I$^2$C REGISTERS

The I$^2$C peripheral interfaces consists of a number of MMRs. These are described in the following section.

### *I$^2$C Master Registers*

#### I$^2$C Master Control Register

This 16-bit MMR configures I$^2$C peripheral in master mode.

| | |
|---|---|
| Name: | I2C0MCTL, I2C1MCTL |
| Address: | 0xFFFF0880, 0xFFFF0900 |
| Default value: | 0x0000, 0x0000 |
| Access: | Read/write |

**Table 89. I2CxMCTL MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 15:9 | | Reserved. These bits are reserved; do not write to these bits. |
| 8 | I2CMCENI | I$^2$C transmission complete interrupt enable bit.<br>Set this bit to enable an interrupt on detecting a stop condition on the I$^2$C bus.<br>Clear this interrupt source. |
| 7 | I2CNACKENI | I$^2$C no acknowledge received interrupt enable bit.<br>Set this bit to enable interrupts when the I$^2$C master receives a no acknowledge.<br>Clear this interrupt source. |
| 6 | I2CALENI | I$^2$C arbitration lost interrupt enable bit.<br>Set this bit to enable interrupts when the I$^2$C master has been unsuccessful in gaining control of the I$^2$C bus.<br>Clear this interrupt source. |
| 5 | I2CMTENI | I$^2$C transmit interrupt enable bit.<br>Set this bit to enable interrupts when the I$^2$C master has transmitted a byte.<br>Clear this interrupt source. |
| 4 | I2CMRENI | I$^2$C receive interrupt enable bit.<br>Set this bit to enable interrupts when the I$^2$C master receives data.<br>Cleared by the user to disable interrupts when the I$^2$C master is receiving data. |
| 3 | | Reserved. A value of 0 should be written to this bit. |
| 2 | I2CILEN | I$^2$C internal loopback enable.<br>Set this bit to enable loopback test mode. In this mode, the SCL and SDA signals are connected internally to their respective input signals.<br>Cleared by the user to disable loopback mode. |
| 1 | I2CBD | I$^2$C master back off disable bit.<br>Set this bit to allow the device to compete for control of the bus even if another device is currently driving a start condition.<br>Clear this bit to back off until the I$^2$C bus becomes free. |
| 0 | I2CMEN | I$^2$C master enable bit.<br>Set by the user to enable I$^2$C master mode.<br>Clear this bit to disable I$^2$C master mode. |

### I$^2$C Master Status Register

This 16-bit MMR is I$^2$C status register in master mode.

| | |
|---|---|
| Name: | I2C0MSTA, I2C1MSTA |
| Address: | 0xFFFF0884, 0xFFFF0904 |
| Default value: | 0x0000, 0x0000 |
| Access: | Read only |

**Table 90 I2CxMSTA MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 15:11 | | Reserved. These bits are reserved. |
| 10 | I2CBBUSY | I$^2$C bus busy status bit.<br>This bit is set to 1 when a start condition is detected on the I$^2$C bus.<br>This bit is cleared when a stop condition is detected on the bus. |
| 9 | I2CMRxFO | Master receiver (Rx) FIFO overflow.<br>This bit is set to 1 when a byte is written to the Rx FIFO when it is already full.<br>This bit is cleared in all other conditions. |
| 8 | I2CMTC | I$^2$C transmission complete status bit.<br>This bit is set to 1 when a transmission is complete between the master and the slave with which it was communicating. If the I2CMCENI bit in I2CxMCTL is set, an interrupt is generated when the I2CMTC bit is set.<br>Clear this interrupt source. |
| 7 | I2CMNA | I$^2$C master no acknowledge data bit.<br>This bit is set to 1 when a no acknowledge 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 the I2CMNA bit is set.<br>This bit is cleared in all other conditions. |
| 6 | I2CMBUSY | I$^2$C master busy status bit.<br>Set to 1 when the master is busy processing a transaction.<br>Cleared if the master is ready or if another master device has control of the bus. |
| 5 | I2CAL | I$^2$C arbitration lost status bit.<br>This bit is set to 1 when the I$^2$C master is unsuccessful in gaining control of the I$^2$C bus. If the I2CALENI bit in I2CxMCTL is set, an interrupt is generated when the I2CAL bit is set.<br>This bit is cleared in all other conditions. |
| 4 | I2CMNA | I$^2$C master no acknowledge address bit.<br>This bit is set to 1 when a no acknowledge condition is received by the master in response to an address. If the I2CNACKENI bit in I2CxMCTL is set, an interrupt is generated when the I2CMNA bit is set.<br>This bit is cleared in all other conditions. |
| 3 | I2CMRXQ | I$^2$C master receive request bit.<br>This bit is set to 1 when data enters the Rx FIFO. If the I2CMRENI in I2CxMCTL is set, an interrupt is generated.<br>This bit is cleared in all other conditions. |
| 2 | I2CMTXQ | I$^2$C master transmit request bit.<br>This bit goes high if the transmitter (Tx) FIFO is empty or only contains one byte and the master has transmitted an address + write. If the I2CMTENI bit in I2CxMCTL is set, an interrupt is generated when the I2CMTXQ bit is set.<br>This bit is cleared in all other conditions. |
| 1:0 | I2CMTFSTA | I$^2$C master Tx FIFO status bits.<br>00 = I$^2$C master Tx FIFO empty.<br>01 = one byte in master Tx FIFO.<br>10 = one byte in master Tx FIFO.<br>11 = I$^2$C master Tx FIFO full. |

### I$^2$C Master Receive Registers

This 8-bit MMR is the I$^2$C master receive register.

Name: I2C0MRX, I2C1MRX

Address: 0xFFFF0888, 0xFFFF0908

Default value: 0x00

Access: Read only

### I$^2$C Master Transmit Registers

This 8-bit MMR is the I$^2$C master transmit register.

Name: I2C0MTX, I2C1MTX

Address: 0xFFFF088C, 0xFFFF090C

Default value: 0x00

Access: Write only

### I$^2$C Master Read Count Registers

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

Name: I2C0MCNT0, I2C1MCNT0

Address: 0xFFFF0890, 0xFFFF0910

Default value: 0x0000

Access: Read and write

**Table 91. I2CxMCNT0 MMR Bit Descriptions**

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

### I$^2$C Master Current Read Count Registers

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

Name: I2C0MCNT1, I2C1MCNT1

Address: 0xFFFF0894, 0xFFFF0914

Default value: 0x00

Access: Read only

### I$^2$C Address 0 Registers

This 8-bit MMR holds the 7-bit slave address + the read/$\overline{\text{write}}$ bit when the master begins communicating with a slave.

Name: I2C0ADR0, I2C1ADR0

Address: 0xFFFF0898, 0xFFFF0918

Default value: 0x00

Access: Read and write

**Table 92. I2CxADR0 MMR in 7-Bit Address Mode**

| Bit | Name | Description |
|---|---|---|
| 7:1 | I2CADR | These bits contain the 7-bit address of the required slave device. |
| 0 | R/$\overline{\text{W}}$ | Bit 0 is the read/$\overline{\text{write}}$ bit.<br>When this bit = 1, a read sequence is requested.<br>When this bit = 0, a write sequence is requested. |

**Table 93. 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/$\overline{\text{W}}$ | Read/$\overline{\text{write}}$ bit.<br>When this bit = 1, a read sequence is requested.<br>When this bit = 0, a write sequence is requested. |

### I$^2$C Address 1 Register

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

Name: I2C0ADR1, I2C1ADR1

Address: 0xFFFF089C, 0xFFFF091C

Default value: 0x00

Access: Read and write

**Table 94. I2CxADR1 MMR in 10-Bit Address Mode**

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

### I$^2$C Master Clock Control Register

This MMR controls the frequency of the I$^2$C clock generated by the master on to the SCL pin.

| | |
|---|---|
| Name: | I2C0DIV, I2C1DIV |
| Address: | 0xFFFF08A4, 0xFFFF0924 |
| Default value: | 0x1F1F |
| Access: | Read and write |

**Table 95. I2CxDIV MMR**

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

## *I$^2$C Slave Registers*

### I$^2$C Slave Control Register

This 16-bit MMR configures the I$^2$C peripheral in slave mode.

| | |
|---|---|
| Name: | I2C0SCTL, I2C1SCTL |
| Address: | 0xFFFF08A8, 0xFFFF0928 |
| Default value: | 0x0000 |
| Access: | Read and write |

**Table 96. I2CxSCTL MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 15:11 | | Reserved bits. |
| 10 | I2CSTXENI | Slave transmit interrupt enable bit.<br>Set this bit to enable an interrupt after a slave transmits a byte.<br>Clear this interrupt source. |
| 9 | I2CSRXENI | Slave receive interrupt enable bit.<br>Set this bit to enable an interrupt after the slave receives data.<br>Clear this interrupt source. |
| 8 | I2CSSENI | I$^2$C stop condition detected interrupt enable bit.<br>Set this bit to enable an interrupt on detecting a stop condition on the I$^2$C bus.<br>Clear this interrupt source. |
| 7 | I2CNACKEN | I$^2$C no acknowledge enable bit.<br>Set this bit to no acknowledge the next byte in the transmission sequence.<br>Clear this bit to let the hardware control the acknowledge/no acknowledge sequence. |
| 6 | | Reserved. A value of 0 should be written to this bit. |
| 5 | I2CSETEN | I$^2$C early transmit interrupt enable bit.<br>Setting this bit enables a transmit request interrupt just after the positive edge of SCL during the read bit transmission.<br>Clear this bit to enable a transmit request interrupt just after the negative edge of SCL during the read bit transmission. |
| 4 | I2CGCCLR | I$^2$C general call status and ID clear bit.<br>Writing a 1 to this bit clears the general call status and ID bits in the I2CxSSTA register.<br>Clear this bit at all other times. |
| 3 | I2CHGCEN | I$^2$C 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 the I2CALT against the receive register. If the contents match, the device has received a hardware general call. This is used if a device needs urgent attention from a master device without knowing which master it needs to turn to. This is a "to whom it may concern" call. The ADuC7121 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 I$^2$C January 2000 bus specification.<br>Set this bit and I2CGCEN to enable hardware general call recognition in slave mode.<br>Clear to disable recognition of hardware general call commands. |
| 2 | I2CGCEN | I$^2$C general call enable. Set this bit to enable the slave device to acknowledge an I$^2$C 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 I$^2$C interface resets as per the I$^2$C January 2000 bus specification. This command can be used to reset an entire I$^2$C 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.<br>Set this bit to allow the slave acknowledge I$^2$C general call commands.<br>Clear to disable recognition of general call commands. |
| 1 | Reserved | Always set this bit to 0. |
| 0 | I2CSEN | I$^2$C slave enable bit.<br>Set by the user to enable I$^2$C slave mode.<br>Clear to disable I$^2$C slave mode. |

### I$^2$C Slave Status Registers

These 16-bit MMRs are the I$^2$C status registers in slave mode.

| | |
|---|---|
| Name: | I2C0SSTA, I2C1SSTA |
| Address: | 0xFFFF08AC, 0xFFFF092C |
| Default value: | 0x0000, 0x0000 |
| Access: | Read and write |

**Table 97. I2CxSSTA MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 15 | | Reserved bit. |
| 14 | I2CSTA | This bit is set to 1 if:<br>A start condition followed by a matching address is detected.<br>A start byte (0x01) is received.<br>General calls are enabled and a general call code of (0x00) is received.<br>This bit is cleared on receiving a stop condition. |
| 13 | I2CREPS | This bit is set to 1 if a repeated start condition is detected.<br>This bit is cleared on receiving a stop condition. A read of the I2CxSSTA register also clears this bit. |
| 12:11 | I2CID[1:0] | I$^2$C address matching register. These bits indicate which I2CxIDx register matches the received address.<br>[00] = Received address matches I2CxID0.<br>[01] = Received address matches I2CxID1.<br>[10] = Received address matches I2CxID2.<br>[11] = Received address matches I2CxID3. |
| 10 | I2CSS | I$^2$C stop condition after start detected bit.<br>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.<br>This bit is cleared by reading this register. |
| 9:8 | I2CGCID[1:0] | I$^2$C general call ID bits.<br>[00] = no general call received.<br>[01] = general call reset and program address.<br>[10] = general program address.<br>[11] = general call matching alternative ID.<br>Clear these bits by writing a 1 to the I2CGCCLR bit in I2CxSCTL. Note that these bits are not cleared by a general call reset command. |
| 7 | I2CGC | I$^2$C general call status bit.<br>This bit is set to 1 if the slave receives a general call command of any type. If the command received was a reset command, all registers return to their default state. If the command received was a hardware general call, the Rx FIFO holds the second byte of the command and this can be compared with the I2CxALT register.<br>Clear this bit by writing a 1 to the I2CGCCLR bit in I2CxSCTL. |
| 6 | I2CSBUSY | I$^2$C slave busy status bit.<br>Set to 1 when the slave receives a start condition.<br>Cleared by hardware under the following conditions:<br>The received address does not match any of the I2CxIDx registers.<br>The slave device receives a stop condition.<br>A repeated start address does not match any of the I2CxIDx registers. |
| 5 | I2CSNA | I$^2$C slave no acknowledge data bit.<br>This bit sets to 1 when the slave responds to a bus address with a no acknowledge. This bit is asserted under the following conditions:<br>If no acknowledge was returned because there was no data in the Tx FIFO.<br>If the I2CNACKEN bit was set in the I2CxSCTL register.<br>This bit is cleared in all other conditions. |

| Bit | Name | Description |
|---|---|---|
| 4 | I2CSRxFO | Slave Rx FIFO overflow.<br>This bit is set to 1 when a byte is written to the Rx FIFO when it is already full.<br>This bit is cleared in all other conditions. |
| 3 | I2CSRXQ | I$^2$C slave receive request bit.<br>This bit is set to 1 when the Rx FIFO of the slave is not empty. This bit causes an interrupt to occur if the I2CSRXENI bit in I2CxSCTL is set.<br>The Rx FIFO must be read or flushed to clear this bit. |
| 2 | I2CSTXQ | I$^2$C slave transmit request bit.<br>This bit is set to 1 when the slave receives a matching address followed by a read.<br>If the I2CSETEN bit in I2CxSCTL is = 0, this bit goes high just after the negative edge of SCL during the read bit transmission.<br>If the I2CSETEN bit in I2CxSCTL is = 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.<br>This bit is cleared in all other conditions. |
| 1 | I2CSTFE | I$^2$C slave FIFO underflow status bit.<br>This bit is high if the Tx FIFO is empty when a master requests data from the slave. This bit asserts at the rising edge of SCL during the read bit.<br>This bit clears in all other conditions. |
| 0 | I2CETSTA | I$^2$C slave early transmit FIFO status bit.<br>If the I2CSETEN bit in I2CxSCTL is = 0, this bit goes high if the slave Tx FIFO is empty.<br>If the I2CSETEN bit in I2CxSCTL is = 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.<br>This bit is cleared after being read. |

### I$^2$C Slave Receive Registers

This 8-bit MMR is the I$^2$C slave receive register.

Name: I2C0SRX, I2C1SRX
Address: 0xFFFF08B0, 0xFFFF0930
Default value: 0x00
Access: Read only

### I$^2$C Slave Transmit Registers

This 8-bit MMR is the I$^2$C slave transmit register.

Name: I2C0STX, I2C1STX
Address: 0xFFFF08B4, 0xFFFF0934
Default value: 0x00
Access: Write only

### I$^2$C Hardware General Call Recognition Registers

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.

Name: I2C0ALT, I2C1ALT
Address: 0xFFFF08B8, 0xFFFF0938
Default value: 0x00
Access: Read and write

### *I$^2$C Slave Device ID Registers*

### I2C0IDx Registers

These eight I2C0IDx 8-bit MMRs are programmed with I$^2$C bus IDs of the slave. See the section, I$^2$C Bus Addresses, for further details.

Name: I2C0ID0
Address: 0xFFFF08BC
Default value: 0x00
Access: Read and write

Name: I2C0ID1
Address: 0xFFFF08C0
Default value: 0x00
Access: Read and write

Name: I2C0ID2
Address: 0xFFFF08C4
Default value: 0x00
Access: Read and write

Name: I2C0ID3
Address: 0xFFFF08C8
Default value: 0x00
Access: Read and write

Name: I2C1ID0
Address: 0xFFFF093C
Default value: 0x00
Access: Read and write

Name: I2C1ID1
Address: 0xFFFF0940
Default value: 0x00
Access: Read and write

Name: I2C1ID2
Address: 0xFFFF0944
Default value: 0x00
Access: Read and write

Name: I2C1ID3
Address: 0xFFFF0948
Default value: 0x00
Access: Read and write

## I$^2$C COMMON REGISTERS

### *I$^2$C FIFO Status Registers*

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

| | |
|---|---|
| Name: | I2C0FSTA |
| Address: | 0xFFFF08CC |
| Default value: | 0x0000 |
| Access: | Read and write |

| | |
|---|---|
| Name: | I2C1FSTA |
| Address: | 0xFFFF094C |
| Default value: | 0x0000 |
| Access: | Read and write |

**Table 98. 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 | I$^2$C master receive FIFO status bits.<br>[00] = FIFO empty.<br>[01] = byte written to FIFO.<br>[10] = one byte in FIFO.<br>[11] = FIFO full. |
| 5:4 | I2CMTXSTA | I$^2$C master transmit FIFO status bits.<br>[00] = FIFO empty.<br>[01] = byte written to FIFO.<br>[10] = one byte in FIFO.<br>[11] = FIFO full. |
| 3:2 | I2CSRXSTA | I$^2$C slave receive FIFO status bits.<br>[00] = FIFO empty.<br>[01] = byte written to FIFO.<br>[10] = one byte in FIFO.<br>[11] = FIFO full. |
| 1:0 | I2CSTXSTA | I$^2$C slave transmit FIFO status bits.<br>[00] = FIFO empty.<br>[01] = byte written to FIFO.<br>[10] = one byte in FIFO.<br>[11] = FIFO full. |

# SERIAL PERIPHERAL INTERFACE

The ADuC7121 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 Mbps.

The SPI port can be configured for master or slave operation and typically consists of four pins: P0.3/MISO/PLAO[12]/SYNC, P0.4/MOSI/PLAI[11]/TRIP, P0.2/SPICLK/ADC$_{\text{BUSY}}$/PLAO[13], and P0.5/$\overline{\text{CS}}$/PLAI[10]/ ADC$_{\overline{\text{CONVST}}}$.

## SPI MISO (MASTER IN, SLAVE OUT) PIN

MISO on the P0.3/MISO/PLAO[12]/SYNC pin is configured as an input line in master mode and an output line in slave mode. Connect the MISO line on the master (data in) to the MISO line in the slave device (data out). The data is transferred as byte wide (8-bit) serial data, MSB first.

## SPI MOSI (MASTER OUT, SLAVE IN) PIN

MOSI on the P0.4/MOSI/PLAI[11]/TRIP pin is configured as an output line in master mode and an input line in slave mode. The MOSI line on the master (data out) should be connected to the MOSI line in the slave device (data in). The data is transferred as byte wide (8-bit) serial data, MSB first.

## 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 P0.2/SPICLK/ADC$_{\text{BUSY}}$/PLAO[13] pin is configured as an output in master mode and as an input in slave mode.

In master mode, the 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 Mbps.

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 INPUT PIN

In SPI slave mode, a transfer is initiated by the assertion of $\overline{\text{CS}}$ on the P0.5/$\overline{\text{CS}}$/PLAI[10]/ADC$_{\overline{\text{CONVST}}}$ pin. $\overline{\text{CS}}$ is an active low input signal. The SPI port then transmits and receives 8-bit data until the transfer is concluded by deassertion of $\overline{\text{CS}}$. In slave mode, $\overline{\text{CS}}$ is always an input.

In SPI master mode, $\overline{\text{CS}}$ 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 ADuC7121 device are P0.2 to P0.5.

- P0.5/$\overline{\text{CS}}$/PLAI[10]/ADC$_{\overline{\text{CONVST}}}$ 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/SPICLK/ADC$_{\text{BUSY}}$/PLAO[13] is the SPICLK pin.
- P0.3/MISO/PLAO[12]/SYNC is the master in, slave out pin.
- P0.4/MOSI/PLAI[11]/TRIP is the master out, slave in pin.

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

## SPI REGISTERS

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

### ***SPI Status Register***

This 32-bit MMR contains the status of the SPI interface in both master and slave modes.

| | |
|---|---|
| Name: | SPISTA |
| Address: | 0xFFFF0A00 |
| Default value: | 0x0000 |
| Access: | Read only |

**Table 99. 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.<br>This bit is cleared when the number of bytes in the FIFO is equal or less than the number in SPIMDE. |
| 10:8 | SPIRXFSTA[2:0] | SPI Rx FIFO status bits.<br>[000] = Rx FIFO is empty.<br>[001] = one valid byte in the FIFO.<br>[010] = two valid bytes in the FIFO.<br>[011] = three valid bytes in the FIFO.<br>[100] = four valid bytes in the FIFO. |
| 7 | SPIFOF | SPI Rx FIFO overflow status bit.<br>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.<br>Cleared when the SPISTA register is read. |
| 6 | SPIRXIRQ | SPI Rx IRQ status bit.<br>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.<br>Cleared when the SPISTA register is read. |
| 5 | SPITXIRQ | SPI Tx IRQ status bit.<br>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.<br>Cleared when the SPISTA register is read. |
| 4 | SPITXUF | SPI Tx FIFO underflow.<br>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.<br>Cleared when the SPISTA register is read. |
| 3:1 | SPITXFSTA[2:0] | SPI Tx FIFO status bits.<br>[000] = Tx FIFO is empty.<br>[001] = one valid byte in the FIFO.<br>[010] = two valid bytes in the FIFO.<br>[011] = three valid bytes in the FIFO.<br>[100] = four valid bytes in the FIFO. |
| 0 | SPIISTA | SPI interrupt status bit.<br>Set to 1 when an SPI based interrupt occurs.<br>Cleared after reading SPISTA. |

### *SPIRX Register*

This 8-bit MMR is the SPI receive register.

Name: SPIRX

Address: 0xFFFF0A04

Default value: 0x00

Access: Read only

### *SPITX Register*

This 8-bit MMR is the SPI transmit register.

Name: SPITX

Address: 0xFFFF0A08

Default value: 0x00

Access: Write only

### *SPIDIV Register*

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

Name: SPIDIV

Address: 0xFFFF0A0C

Default value: 0x00

Access: Read and write

### *SPI Control Register*

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

Name: SPICON

Address: 0xFFFF0A10

Default value: 0x0000

Access: Read and write

**Table 100. 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.<br>[00] = Tx interrupt occurs when one byte has been transferred. Rx interrupt occurs when one or more bytes have been received into the FIFO.<br>[01] = Tx interrupt occurs when two bytes has been transferred. Rx interrupt occurs when two or more bytes have been received into the FIFO.<br>[10] = Tx interrupt occurs when three bytes has been transferred. Rx interrupt occurs when three or more bytes have been received into the FIFO.<br>[11] = Tx interrupt occurs when four bytes has been transferred. Rx interrupt occurs when the Rx FIFO is full, or four bytes present. |
| 13 | SPITFLH | SPI Tx FIFO flush enable bit.<br>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. Any writes to the Tx FIFO are ignored while this bit is set.<br>Clear this bit to disable Tx FIFO flushing. |
| 12 | SPIRFLH | SPI Rx FIFO flush enable bit.<br>Set this bit to flush the Rx FIFO. This bit does not clear itself and should be toggled if a single flush is required. If this bit is set, all incoming data is ignored and no interrupts are generated. If this bit is set and SPITMDE = 0, a read of the Rx FIFO initiates a transfer.<br>Clear this bit to disable Rx FIFO flushing. |
| 11 | SPICONT | Continuous transfer enable.<br>Set by the user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the Tx register. The P0.5/$\overline{\text{CS}}$/PLAI[10]/ADC$_{\overline{\text{CONVST}}}$ pin is asserted and remains asserted for the duration of each 8-bit serial transfer until Tx is empty.<br>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 | Loopback enable bit.<br>Set by the user to connect MISO to MOSI and test software.<br>Cleared by the user to be in normal mode. |
| 9 | SPIOEN | Slave MISO output enable bit.<br>Set this bit for normal operation of MISO.<br>Clear this bit to disable the output driver on the MISO pin. The MISO pin is open drain when this bit is clear. |
| 8 | SPIROW | SPIRX overflow overwrite enable.<br>Set by the user, the valid data in the Rx register is overwritten by the new serial byte that is received.<br>Cleared by the user, the new serial byte that is received is discarded. |
| 7 | SPIZEN | SPI transmits zeros when Tx FIFO is empty.<br>Set this bit to transmit 0x00 when there is no valid data in the Tx FIFO.<br>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.<br>Set by the user to initiate a transfer with a write to the SPITX register. Interrupt occurs only when Tx is empty.<br>Cleared by the user to initiate a transfer with a read of the SPIRX register. Interrupt occurs only when Rx is full. |
| 5 | SPILF | LSB first transfer enable bit.<br>Set by the user, the LSB is transmitted first.<br>Cleared by the user, the MSB is transmitted first. |
| 4 | SPIWOM | SPI wired or mode enable bit.<br>Set to 1 to enable open-drain data output enable. External pull-ups are required on data output pins.<br>Clear for normal output levels. |
| 3 | SPICPO | Serial clock polarity mode bit.<br>Set by the user, the serial clock idles high.<br>Cleared by the user, the serial clock idles low. |
| 2 | SPICPH | Serial clock phase mode bit.<br>Set by the user, the serial clock pulses at the beginning of each serial bit transfer.<br>Cleared by the user, the serial clock pulses at the end of each serial bit transfer. |
| 1 | SPIMEN | Master mode enable bit.<br>Set by the user to enable master mode.<br>Cleared by the user to enable slave mode. |
| 0 | SPIEN | SPI enable bit.<br>Set by the user to enable the SPI.<br>Cleared by the user to disable the SPI. |

# PROGRAMMABLE LOGIC ARRAY (PLA)

The ADuC7121 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 dual input lookup table that can be configured to generate any logic output function based on two inputs and a flip-flop. This is represented in Figure 37.

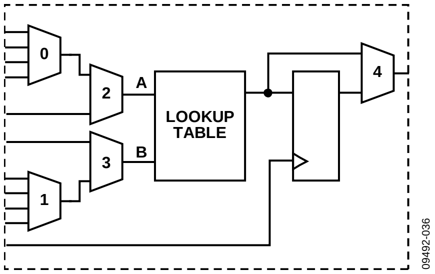

*Figure 37. PLA Element*

In total, 32 GPIO pins are available on each ADuC7121 for the PLA. These include 16 input pins and 16 output pins, which 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, and that the JTAG TDI and TDO pins are included as PLA outputs. If you want to use JTAG programming or debugging, then you cannot use the JTAG TDI and TDO pins as PLA outputs.

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 $\text{ADC}_{\overline{\text{CONVST}}}$ 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 to Input 0 of Mux 0 of Element 0 (Block 0).
- Output of Element 7 (Block 0) can be fed to the Input 0 of Mux 0 of Element 8 (Block 1).

**Table 101. 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 the following sections.

### *PLAELMx Registers*

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 103 and Table 106.

**Table 102. PLAELMx MMR Addresses (Default Value = 0x0000, Access is Read/Write)**

| Name | Address |
|---|---|
| PLAELM0 | 0xFFFF0B00 |
| PLAELM1 | 0xFFFF0B04 |
| PLAELM2 | 0xFFFF0B08 |
| PLAELM3 | 0xFFFF0B0C |
| PLAELM4 | 0xFFFF0B10 |
| PLAELM5 | 0xFFFF0B14 |
| PLAELM6 | 0xFFFF0B18 |
| PLAELM7 | 0xFFFF0B1C |
| PLAELM8 | 0xFFFF0B20 |
| PLAELM9 | 0xFFFF0B24 |
| PLAELM10 | 0xFFFF0B28 |
| PLAELM11 | 0xFFFF0B2C |
| PLAELM12 | 0xFFFF0B30 |
| PLAELM13 | 0xFFFF0B34 |
| PLAELM14 | 0xFFFF0B38 |
| PLAELM15 | 0xFFFF0B3C |

**Table 103. PLAELMx MMR Bit Descriptions**

| Bit | Value | Description |
|---|---|---|
| 31:11 | | Reserved. |
| 10:9 | | Mux 0 control (see Table 106). |
| 8:7 | | Mux 1 control (see Table 106). |
| 6 | | Mux 2 control.<br>Set by the user to select the output of Mux 0.<br>Cleared by the user to select the bit value from PLADIN. |
| 5 | | Mux 3 control.<br>Set by the user to select the input pin of the particular element.<br>Cleared by the 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.<br>Set by the user to bypass the flip-flop.<br>Cleared by the user to select the flip-flop (cleared by default). |

**Table 104. 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 |

### PLACLK Register

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

| | |
|---|---|
| Name: | PLACLK |
| Address: | 0xFFFF0B40 |
| Default value: | 0x00 |
| Access: | Read and write |

**Table 105. PLACLK MMR Bit Descriptions**

| Bit | Value | Description |
|---|---|---|
| 7 | | Reserved. |
| 6:4 | | Block 1 clock source selection. |
| | 000 | GPIO clock on P0.5 of the P0.5/$\overline{\text{CS}}$/PLAI[10]/ ADC$_{\overline{\text{CONVST}}}$ pin. |
| | 001 | GPIO clock on P0.0 of the P0.0/SCL0/PLAI[5] pin. |
| | 010 | GPIO clock on the P0.7 of the P0.7/$\overline{\text{TRST}}$/PLAI[3] pin. |
| | 011 | HCLK (core clock). |
| | 100 | OCLK (32.768 kHz external crystal). |
| | 101 | Timer1 overflow. |
| | Other | Reserved. |
| 3 | | Reserved. |
| 2:0 | | Block 0 clock source selection. |
| | 000 | GPIO clock on P0.5. on P0.5 of the P0.5/$\overline{\text{CS}}$/ PLAI[10]/ADC$_{\overline{\text{CONVST}}}$ pin. |
| | 001 | GPIO clock on P0.0 of the P0.0/SCL0/PLAI[5] pin. |
| | 010 | GPIO clock on P0.7 of the P0.7/$\overline{\text{TRST}}$/PLAI[3] pin. |
| | 011 | HCLK (core clock). |
| | 100 | OCLK (32.768 kHz external crystal). |
| | 101 | Timer1 overflow. |
| | Other | Reserved. |

### PLAIRQ Register

PLAIRQ enables IRQ0 and/or IRQ1 and selects the source of the normal interrupt request IRQ (IRQ).

| | |
|---|---|
| Name: | PLAIRQ |
| Address: | 0xFFFF0B44 |
| Default value: | 0x0000 |
| Access: | Read and write |

**Table 106. PLAIRQ MMR Bit Descriptions**

| Bit | Value | Description |
|---|---|---|
| 15:13 | | Reserved. |
| 12 | | PLA IRQ1 enable bit.<br>Set by the user to enable IRQ1 output from the PLA.<br>Cleared by the user to disable IRQ1 output from the 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.<br>Set by the user to enable IRQ0 output from the PLA.<br>Cleared by the user to disable IRQ0 output from the PLA. |
| 3:0 | | PLA IRQ0 source. |
| | 0000 | PLA Element 0. |
| | 0001 | PLA Element 1. |
| | 1111 | PLA Element 15. |

### PLAADC Register

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

| | |
|---|---|
| Name: | PLAADC |
| Address: | 0xFFFF0B48 |
| Default value: | 0x00000000 |
| Access: | Read and write |

**Table 107. PLAADC MMR Bit Descriptions**

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

### PLADIN Register

PLADIN is a data input MMR for PLA.

| | |
|---|---|
| Name: | PLADIN |
| Address: | 0xFFFF0B4C |
| Default value: | 0x00000000 |
| Access: | Read and write |

**Table 108. PLADIN MMR Bit Descriptions**

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

### PLADOUT Register

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

| | |
|---|---|
| Name: | PLADOUT |
| Address: | 0xFFFF0B50 |
| Default value: | 0x00000000 |
| Access: | Read only |

**Table 109. PLADOUT MMR Bit Descriptions**

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

### PLALCK Register

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

| | |
|---|---|
| Name: | PLALCK |
| Address: | 0xFFFF0B54 |
| Default value: | 0x00 |
| Access: | Write only |

# INTERRUPT SYSTEM

**Table 110. IRQ/FIQ 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 | IDAC fault | IDAC fault IRQ |
| 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 | $I^2$C0 Master IRQ | $I^2$C master interrupt source bit |
| 16 | $I^2$C0 Slave IRQ | $I^2$C slave interrupt source bit |
| 17 | $I^2$C1 Master IRQ | $I^2$C master interrupt source bit |
| 18 | $I^2$C1 Slave IRQ | $I^2$C 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 |

There are 27 interrupt sources on the ADuC7121 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 recognizes interrupts as one of two types only: 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 110.

The ADuC7121 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.

Upon entering the interrupt service routine (ISR), immediately save IRQSTA/FIQSTA to ensure that all valid interrupt sources are serviced.

## NORMAL INTERRUPT REQUEST (IRQ)

The normal interrupt request (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 follow.

### *IRQSIG Register*

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 a read-only register. Do not use this register in an interrupt service routine for determining the source of an IRQ exception; use only IRQSTA for this purpose.

Name: IRQSIG

Address: 0xFFFF0004

Default value: 0x00000000

Access: Read only

### *IRQEN Register*

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.

Name: IRQEN

Address: 0xFFFF0008

Default value: 0x00000000

Access: Read and write

### *IRQCLR Register*

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.

Use this register to disable an interrupt source only when:

- The device is in the interrupt sources interrupt service routine.
- The peripheral is temporarily disabled by its own control register.

Do not use the IRQCLR to disable an IRQ source if that IRQ source has an interrupt pending or could have an interrupt pending.

Name: IRQCLR

Address: 0xFFFF000C

Default value: 0x00000000

Access: Write only

### *IRQSTA Register*

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.

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 Register*

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.

Name: FIQSIG

Address: 0xFFFF0104

Default value: 0x00000000

Access: Read only

### *FIQEN Register*

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 and 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.

Use this register to disable an interrupt source only when:

- The device is in the interrupt sources interrupt service routine.
- The peripheral is temporarily disabled by its own control register.

Do not use this register to disable an FIQ source if that FIQ source has an interrupt pending or could have an interrupt pending.

#### 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 111. This MMR allows the control of a programmed source interrupt.

**Table 111. SWICFG MMR Bit Designations**

| Bit | Description |
|---|---|
| 31:3 | Reserved. |
| 2 | Programmed Interrupt FIQ. Setting/clearing this bit corresponds to setting/clearing Bit 1 of FIQSTA and FIQSIG. |
| 1 | Programmed Interrupt IRQ1. Setting or clearing this bit corresponds to setting or 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 to be detected by the user in the IRQSTA/FIQSTA register.

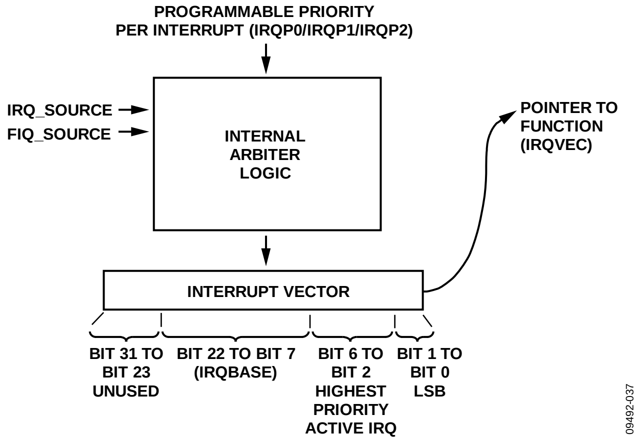

*Figure 38. Interrupt Structure*

### *Vectored Interrupt Controller (VIC)*

The ADuC7121 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—allows 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, it is possible to have 16 separate interrupt levels.
- Programmable interrupt priorities—using the IRQP0 to IRQP3 registers, an interrupt source can be assigned an interrupt priority level value between 0 and 7.

### *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 112. IRQBASE MMR Bit Designations**

| Bit | Type | Initial Value | Description |
|---|---|---|---|
| 31:16 | Read only | Reserved | Always read as 0 |
| 15:0 | Read and write | 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. Read this register only 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 and write |

**Table 113. IRQVEC MMR Bit Designations**

| Bit | Type | Initial Value | Description |
|---|---|---|---|
| 31:23 | Read only | 0 | Always read as 0. |
| 22:7 | Read and write | 0 | IRQBASE register value. |
| 6:2 | Read only | 0 | Highest priority source. This is a value between 0 and 27 representing the possible interrupt sources. For example, if the highest currently active IRQ is Timer2, these bits are [00100]. |
| 1:0 | Reserved | 0 | Reserved bits. |

### *Priority Registers*

#### IRQP0 Register

| | |
|---|---|
| Name: | IRQP0 |
| Address: | 0xFFFF0020 |
| Default value: | 0x00000000 |
| Access: | Read and write |

**Table 114. IRQP0 MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 31 | Reserved | Reserved bit. |
| 30:28 | IDAC_Fault | A priority level of 0 to 7 can be set for an IDAC fault interrupt. |
| 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 | Reserved bit. |

#### IRQP1 Register

| | |
|---|---|
| Name: | IRQP1 |
| Address: | 0xFFFF0024 |
| Default value: | 0x00000000 |
| Access: | Read and write |

**Table 115. 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 $I^2C$ 0 master. |
| 27 | Reserved | Reserved bit. |
| 26:24 | SPIPI | A priority level of 0 to 7 can be set for SPI. |
| 23 | Reserved | Reserved bit. |
| 22:20 | UARTPI | A priority level of 0 to 7 can be set for 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 116. 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 $I^2C1$ slave. |
| 7 | Reserved | Reserved bit. |
| 6:4 | I2C1MPI | A priority level of 0 to 7 can be set for $I^2C1$ master. |
| 3 | Reserved | Reserved bit. |
| 2:0 | I2C0SPI | A priority level of 0 to 7 can be set for $I^2C0$ slave. |

### IRQP3 Register

Name: IRQP3

Address: 0xFFFF002C

Default value: 0x00000000

Access: Read and write

**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 to enable nesting and prioritization of IRQ interrupts the other to enable 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, nor 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 117. 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. If the IRQ is of Priority 0 then Bit 0 asserts, Priority 1 then Bit 1 asserts, and so forth. 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 118. 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.<br>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. Read this register only 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 119. FIQVEC MMR Bit Designations**

| Bit | Type | Initial Value | Description |
|---|---|---|---|
| 31:23 | Read only | 0 | Always read as 0. |
| 22:7 | Read and write | 0 | IRQBASE register value. |
| 6:2 | | 0 | Highest priority source. This is a value between 0 and 27 representing the possible interrupt sources. For example, if the highest currently active FIQ is Timer2, then these bits are [00100]. |
| 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. If the FIQ is of Priority 0, then Bit 0 asserts; if Priority 1, then Bit 1 asserts, and so forth.

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: | FIQSTAN |
| Address: | 0xFFFF013C |
| Default value: | 0x00000000 |
| Access: | Read and write |

**Table 120. 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 ADuC7121 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, first, the appropriate bit must 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 121. IRQCONEMMR 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. |

| Bit | Value | Name | Description |
|---|---|---|---|
| 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: | Read and write |

Table 122. IRQCLRE MMR Bit Designations

| Bit | Name | Description |
|---|---|---|
| 31:25 | Reserved | These bits are reserved and should not be written to. |
| 24 | IRQ5CLRI | A 1 must be written to this bit in the IRQ5 interrupt service routine to clear an edge triggered IRQ5 interrupt. |
| 24 | IRQ4CLRI | A 1 must be written to this bit in the IRQ4 interrupt service routine to clear an edge triggered IRQ4 interrupt. |
| 23 | Reserved | This bit is reserved. |
| 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 ADuC7121 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).

**Table 123. Event Selection (ES) Numbers**

| ES | Interrupt No. | Name |
|---|---|---|
| 00000 | 2 | RTOS timer (Timer0) |
| 00001 | 3 | GP Timer0 (Timer1) |
| 00010 | 4 | Wake-up timer (Timer2) |
| 00011 | 5 | Watchdog timer (Timer3) |
| 00100 | 6 | GP Timer4 (Timer4) |
| 00101 | 7 | IDAC Fault IRQ |
| 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 | I$^2$C0 master |
| 01110 | 16 | I$^2$C0 slave |
| 01111 | 17 | I$^2$C1 master |
| 10000 | 18 | I$^2$C1 slave |
| 10001 | 19 | External IRQ0 |

## HOUR:MINUTE:SECOND:1/128 FORMAT

To use the timer in hour:minute:second:hundredths format, select the 32,768 kHz clock and prescaler of 256. The hundredths field does not represent milliseconds but 1/128 of a second (256/32,768). The bits representing the hour, minute, and second are not consecutive in the register. This arrangement applies to TxLD and TxVAL when using the hour:minute:second:hundredths format as set in TxCON[5:4]. See Table 124 for additional details.

**Table 124. Hour:Minnute:Second:Hundredths Format**

| Bit | Value | Description |
|---|---|---|
| 31:24 | 0 to 23 or 0 to 255 | Hours |
| 23:22 | 0 | Reserved |
| 21:16 | 0 to 59 | Minutes |
| 15:14 | 0 | Reserved |
| 13.8 | 0 to 59 | Seconds |
| 7 | 0 | Reserved |
| 6:0 | 0 to 127 | 1/128 second |

## 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 one. 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 T0ICLR is written.

Timer0 interface consists of six MMRs as listed in Table 125.

**Table 125. 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 | T0VAL0 is a 16-bit register that holds the 16 least significant bits (LSBs).<br>T0VAL1 is a 32-bit register that holds the 32 most significant bits (MSBs).<br>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. |
| T0ICLR | 8-bit register. Writing any value to this register clears the interrupt. Available only in 16-bit mode. |
| T0CON | Configuration MMR. |

### *Timer0 Value Registers*

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 registers. In 16-bit mode, 16-bit T0VAL0 is used. In 48-bit mode, both 16-bit T0VAL0 and 32-bit T0VAL1 are used.

Name: T0VAL0

Address: 0xFFFF0304

Default value: 0x0000

Access: Read only

Name: T0VAL1

Address: 0xFFFF0308

Default value: 0x00000000

Access: Read only

### *Timer0 Capture Register*

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

Name: T0CAP

Address: 0xFFFF0314

Default value: 0x0000

Access: Read only

### *Timer0 Control Register*

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

Name: T0CON

Address: 0xFFFF030C

Default value: 0x00000000

Access: Read and write

**Table 126. T0CON MMR Bit Designations**

| Bit | Value | Description |
|---|---|---|
| 31:18 | | Reserved. |
| 17 | | Event select bit.<br>Set by the user to enable time capture of an event.<br>Cleared by the user to disable time capture of an event. |
| 16:12 | | Event select range, 0 to 16. The events are described in the introduction to 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 in 16-bit mode only.<br>Set by the user for Timer0 to count up.<br>Cleared by the user for Timer0 to count down (default). |
| 7 | | Timer0 enable bit.<br>Set by the user to enable Timer0.<br>Cleared by the user to disable Timer0 (default). |
| 6 | | Timer0 mode.<br>Set by the user to operate in periodic mode.<br>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 divide-by-1 (default). |
| | 0100 | Source clock divide-by-16. |
| | 1000 | Source clock divide-by-256. |
| | 1111 | Source clock divide-by-32,768. |

### *Timer0 Load Registers*

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

| | |
|---|---|
| Name: | T0LD |
| Address: | 0xFFFF0300 |
| Default value: | 0x00 |
| Access: | Read and write |

### *Timer0 Clear Register*

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

| | |
|---|---|
| Name: | T0CLRI |
| Address: | 0xFFFF0310 |
| Default value: | 0x00 |
| Access: | Write only |

## 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 22 ns when operating at CD zero, the core is operating at 41.78 MHz, and with a prescaler of one.

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 source initial assertion of a selected IRQ. 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. Timer1 interface consists of five MMRs as shown in Table 127.

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 127. 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. |

### *Timer1 Load Registers*

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

| | |
|---|---|
| Name: | T1LD |
| Address: | 0xFFFF0320 |
| Default value: | 0x00000000 |
| Access: | Read and write |

### *Timer1 Clear Register*

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

| | |
|---|---|
| Name: | T1CLRI |
| Address: | 0xFFFF032C |
| Default value: | 0x00 |
| Access: | Write only |

### *Timer1 Value Register*

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

| | |
|---|---|
| Name: | T1VAL |
| Address: | 0xFFFF0324 |
| Default value: | 0x00000000 |
| Access: | Read only |

### *Timer1 Capture Register*

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

| | |
|---|---|
| Name: | T1CAP |
| Address: | 0xFFFF0330 |
| Default value: | 0x0000 |
| Access: | Read only |

### *Timer1 Control Register*

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

| | |
|---|---|
| Name: | T1CON |
| Address: | 0xFFFF0328 |
| Default value: | 0x00000000 |
| Access: | Read and write |

Table 128. 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.<br>Set by the user to enable time capture of an event.<br>Cleared by the user to disable time capture of an event. |
| 16:12 | | Event select range, 0 to 16. The events are as described in the introduction to the Timers section. |
| 11:9 | | Clock select. |
| | 000 | Internal 32 kHz oscillator (default). |
| | 001 | Core clock. |
| | 010 | UCLK. |
| | 011 | P0.6. of the P0.6/$\overline{\text{MRST}}$/PLAI[2] pin. |
| 8 | | Count up.<br>Set by the user for Timer1 to count up.<br>Cleared by the user for Timer1 to count down (default). |
| 7 | | Timer1 enable bit.<br>Set by the user to enable Timer1.<br>Cleared by the user to disable Timer1 (default). |
| 6 | | Timer1 mode.<br>Set by the user to operate in periodic mode.<br>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 hour. |
| | 11 | Hr:Min:Sec:Hundredths: 255 hours to 0 hour. |
| 3:0 | | Prescaler. |
| | 0000 | Source clock divide-by-1 (default). |
| | 0100 | Source clock divide-by-16. |
| | 1000 | Source clock divide-by-256. |
| | 1111 | Source clock divide-by-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 one of four clock sources, namely, 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].

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

Timer2 reloads the value from T2LD either when Timer2 overflows or immediately when T2CLRI is written. The Timer2 interface consists of four MMRs, as shown in Table 129.

**Table 129. 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. |

### *Timer2 Load Registers*

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

| | |
|---|---|
| Name: | T2LD |
| Address: | 0xFFFF0340 |
| Default value: | 0x00000000 |
| Access: | Read and write |

### *Timer2 Clear Register*

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

| | |
|---|---|
| Name: | T2CLRI |
| Address: | 0xFFFF034C |
| Default value: | 0x00 |
| Access: | Write only |

### *Timer2 Value Register*

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

| | |
|---|---|
| Name: | T2VAL |
| Address: | 0xFFFF0344 |
| Default value: | 0x00000000 |
| Access: | Read only |

### *Timer2 Control Register*

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

| | |
|---|---|
| Name: | T2CON |
| Address: | 0xFFFF0348 |
| Default value: | 0x00000000 |
| Access: | Read and write |

**Table 130. 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 32.768kHz watch crystal. |
| | 11 | UCLK. |
| 8 | | Count up.<br>Set by the user for Timer2 to count up.<br>Cleared by the user for Timer2 to count down (default). |
| 7 | | Timer2 enable bit.<br>Set by the user to enable Timer2.<br>Cleared by the user to disable Timer2 (default). |
| 6 | | Timer2 mode.<br>Set by the user to operate in periodic mode.<br>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 hour. |
| | 11 | Hr:Min:Sec:Hundredths: 255 hours to 0 hour. |
| 3:0 | | Prescaler. |
| | 0000 | Source clock divide-by-1 (default). |
| | 0100 | Source clock divide-by-16. |
| | 1000 | Source clock divide-by-256. (Use this setting in conjunction with Timer2 Format 1,0 and Format 1,1.) |
| | 1111 | Source clock divide-by-32,768. |

## TIMER3—WATCHDOG TIMER

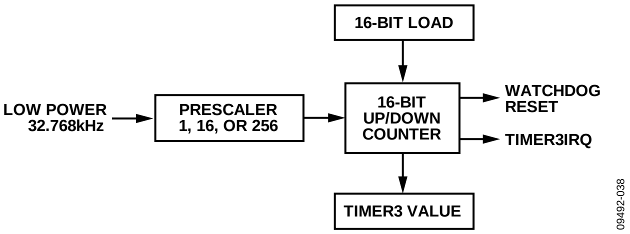

*Figure 39. Timer3 Block Diagram*

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

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

### *Normal Mode*

The Timer3 in normal mode is identical to Timer0 in 16-bit mode of operation, except for the clock source. The clock source is the 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 prescaler divide-by-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 once 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 Table 131.

**Table 131. Timer3 Interface Registers**

| Register | Description |
|---|---|
| T3CON | The configuration MMR. |
| T3LD | 6-bit register (Bit 0 to Bit15); holds 16-bit unsigned integers. |
| T3VAL | 6-bit register (Bit 0 to Bit 15); holds 16-bit unsigned integers. This register is read only. |
| T3CLRI | 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. |

### *Timer3 Load Register*

This 16-bit MMR holds the Timer3 reload value.

| | |
|---|---|
| Name: | T3LD |
| Address: | 0xFFFF0360 |
| Default value: | 0x3BF8 |
| Access: | Read and write |

### *Timer3 Value Register*

This 16-bit, read-only MMR holds the current Timer3 count value.

| | |
|---|---|
| Name: | T3VAL |
| Address: | 0xFFFF0364 |
| Default value: | 0x3BF8 |
| Access: | Read only |

### *Timer3 Clear Register*

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.

| | |
|---|---|
| Name: | T3CLRI |
| Address: | 0xFFFF036C |
| Default value: | 0x0000 |
| Access: | Write only |

### *Timer3 Control Register*

The 16-bit MMR configures the mode of operation of Timer3 and is described in detail in Table 132.

| | |
|---|---|
| Name: | T3CON |
| Address: | 0xFFFF0368 |
| Default value: | 0x0000 |
| Access: | Read and write one time only |

Table 132. T3CON MMR Bit Designations

| Bit | Value | Description |
|---|---|---|
| 15:9 | | These bits are reserved and should be written as 0s by user code. |
| 8 | | Count up/down enable.<br>Set by user code to configure Timer3 to count up.<br>Cleared by user code to configure Timer3 to count down. |
| 7 | | Timer3 enable.<br>Set by user code to enable Timer3.<br>Cleared by user code to disable Timer3. |
| 6 | | Timer3 operating mode.<br>Set by user code to configure Timer3 to operate in periodic mode.<br>Cleared by user to configure Timer3 to operate in free-running mode. |
| 5 | | Watchdog timer mode enable.<br>Set by user code to enable watchdog mode.<br>Cleared by user code to disable watchdog mode. |
| 4 | | Secure clear bit.<br>Set by the user to use the secure clear option.<br>Cleared by the user to disable the secure clear option by default. |
| 3:2 | | Timer3 Clock(32.768 kHz) prescaler. |
| | 00 | Source clock divide-by-1 (default). |
| | 01 | Reserved. |
| | 10 | Reserved. |
| | 11 | Reserved. |
| 1 | | Watchdog timer IRQ enable.<br>Set by user code to produce an IRQ instead of a reset when the watchdog reaches 0.<br>Cleared by user code to disable the IRQ option. |
| 0 | | PD_OFF.<br>Set by user code to stop Timer3 when the peripherals are powered down via Bit 4 in the POWCON MMR.<br>Cleared by user code to enable Timer3 when the peripherals are powered down via Bit 4 in the POWCON MMR. |

### *Secure Clear Bit (Watchdog Mode Only)*

The secure clear bit 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.

The initial value or seed is written to T3CLRI 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, reset is immediately generated, even if the count has not yet expired.

Because of the properties of the polynomial, do not use the value, 0x00, as an initial seed. 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.

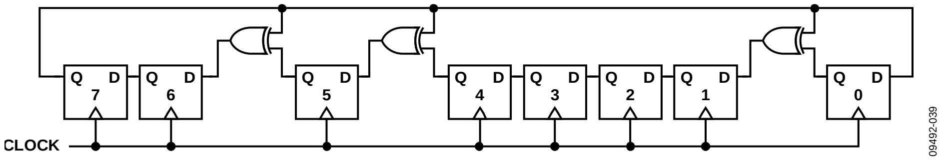

Figure 40. 8-Bit LFSR

## TIMER4—GENERAL-PURPOSE TIMER

Timer4 is a 32-bit general-purpose timer, count down or count up, with a programmable prescaler. The prescaler source can be the 32 kHz oscillator, the core clock, or 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 zero, the core is operating at 41.78 MHz, and with a prescaler 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), which can be triggered by a selected IRQ's source initial assertion. Once triggered, the current timer value is copied to T4CAP, and the timer keeps running. This feature can be used to determine the assertion of an event with increased accuracy.

Timer4 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 Timer4 overflows, or immediately when T4ICLR is written.

### *Timer4 Load Registers*

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

Name: T4LD

Address: 0xFFFF0380

Default value: 0x00000000

Access: Read and write

### *Timer4 Clear Register*

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

Name: T4CLRI

Address: 0xFFFF038C

Default value: 0x00

Access: Write only

### *Timer4 Value Register*

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

Name: T4VAL

Address: 0xFFFF0384

Default value: 0x00000000

Access: Read only

### *Timer4 Capture Register*

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

Name: T4CAP

Address: 0xFFFF0390

Default value: 0x00000000

Access: Read only

### *Timer4 Control Register*

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

Name: T4CON

Address: 0xFFFF0388

Default value: 0x0000

Access: Read and write

Table 133. T4CON MMR Bit Designations

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

# HARDWARE DESIGN CONSIDERATIONS

## POWER SUPPLIES

The ADuC7121 operational power supply voltage range is 3.0 V to 3.6 V. Separate analog and digital power supply pins (AVDD and IOVDD, respectively) allow AVDD to be kept relatively free of noisy digital signals often present on the system IOVDD line. In this mode, the part can also operate with split supplies, that is, using different voltage levels for each supply. For example, the system can be designed to operate with an IOVDD voltage level of 3.3 V while the AVDD level can be at 3 V, or vice versa. A typical split supply configuration is shown in Figure 41.

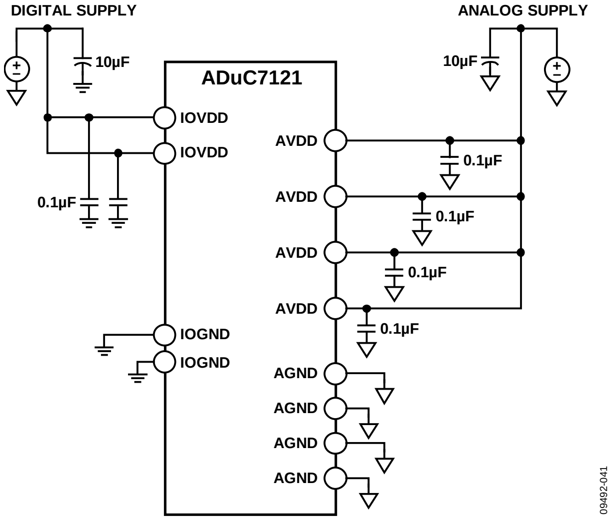

*Figure 41. External Dual Supply Connections*

As an alternative to providing two separate power supplies, the user can reduce noise on AVDD by placing a small series resistor and/or ferrite bead between AVDD and IOVDD, and then decouple AVDD separately to ground. An example of this configuration is shown in Figure 42. With this configuration, other analog circuitry (such as op amps, voltage reference, and others) can be powered from the AVDD supply line as well.

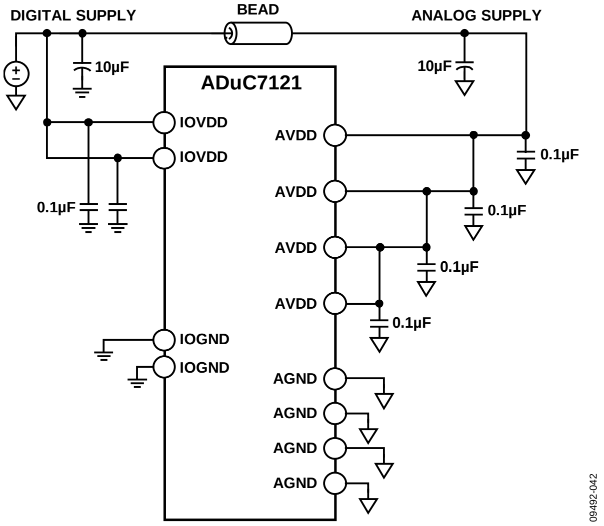

*Figure 42. External Single Supply Connections*

Notice that in both Figure 41 and Figure 42, a large value (10 µF) reservoir capacitor sits on IOVDD, and a separate 10 µF capacitor sits on AVDD. In addition, local small-value (0.1 µF) capacitors are located at each AVDD and IOVDD pin of the chip. As per standard design practice, be sure to include all of these capacitors and ensure the smaller capacitors are close to each AVDD pin with trace lengths as short as possible. Connect the ground terminal of each of these capacitors directly to the underlying ground plane. Finally, note that the analog and digital ground pins on the ADuC7121 must be referenced to the same system ground reference point at all times.

### *IOVDD Supply Sensitivity*

The IOVDD supply is sensitive to high frequency noise because it is the supply source for the internal oscillator and PLL circuits. When the internal PLL loses lock, the clock source is removed by a gating circuit from the CPU, and the ARM7TDMI core stops executing code until the PLL regains lock. This feature ensures that no flash interface timings or ARM7TDMI timings are violated.

Typically, frequency noise greater than 50 kHz and 50 mV p-p on top of the supply causes the core to stop working.

If decoupling values recommended in the Power Supplies section do not sufficiently dampen all noise sources below 50 mV on IOVDD, a filter such as the one shown in Figure 43 is recommended.

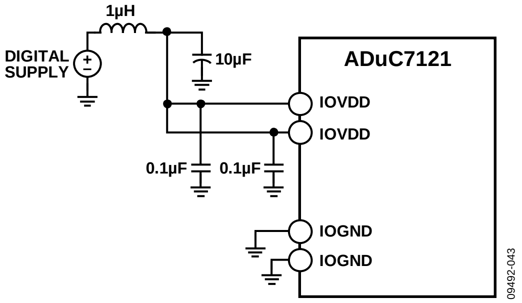

*Figure 43. Recommended IOVDD Supply Filter*

### Linear Voltage Regulator

Each ADuC7121 requires a single 3.3 V supply, but the core logic requires a 2.6 V supply. An on-chip linear regulator generates the 2.6 V from IOVDD for the core logic. The DVDD pins are the 2.6 V supply for the core logic. An external compensation capacitor of 0.47 µF must be connected between each DVDD and DGND (as close as possible to these pins) to act as a tank of charge as shown in Figure 44. The internal IDACs require a 2.5 V supply. An internal LDO provides a stable 2.5 V supply. The AVDD_IDAC pin is the 2.5 V supply for the IDACs. An external compensation capacitor of 0.47 µF must be connected between AVDD_IDAC and AGND (as close as possible to these pins) to act as a tank of charge as shown in Figure 44.

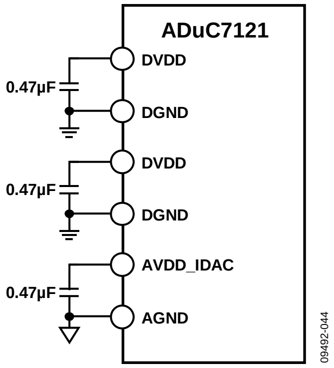

*Figure 44. Voltage Regulator Connections*

The DVDD pins should not be used for any other chip. It is also recommended to use excellent power supply decoupling on IOVDD to help improve line regulation performance of the on-chip voltage regulator.

## GROUNDING AND BOARD LAYOUT RECOMMENDATIONS

As with all high resolution data converters, special attention must be paid to grounding and PC board layout of ADuC7121-based designs to achieve optimum performance from the ADCs and DAC.

Although the part has separate has separate pins for analog and digital ground (AGND and IOGND), the user must not tie these to two separate ground planes unless the two ground planes are connected very close to the part. This is illustrated in the simplified example shown in Figure 45a. In systems where digital and analog ground planes are connected together somewhere else (at the system's power supply, for example), the planes can not be reconnected near the part, because a ground loop would result. In these cases, tie all the AGND and IOGND pins of the ADuC7121 to the analog ground plane, as illustrated in Figure 45b. In systems with only one ground plane, ensure that the digital and analog components are physically separated onto separate halves of the board so that digital return currents do not flow near analog circuitry and vice versa. The ADuC7121 can then be placed between the digital and analog sections, as illustrated in Figure 45c.

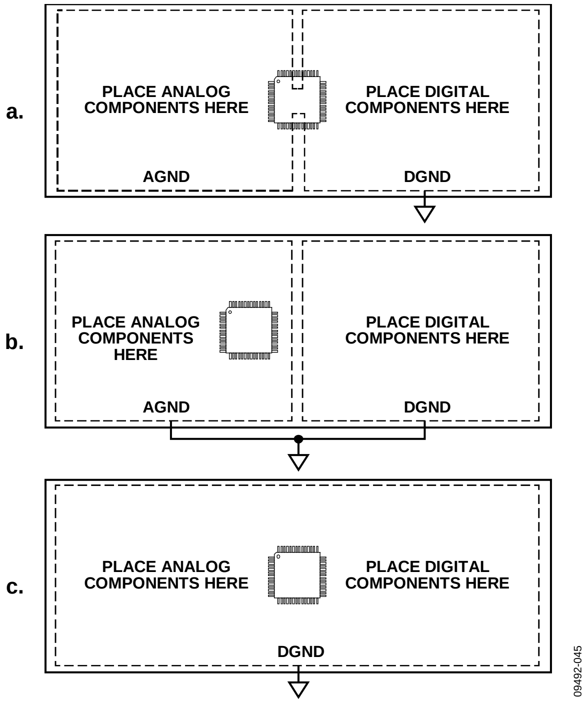

*Figure 45. System Grounding Schemes*

In all of these scenarios, and in more complicated real-life applications, pay particular attention to the flow of current from the supplies and back to ground. Make sure the return paths for all currents are as close as possible to the paths the currents took to reach their destinations. For example, do not power components on the analog side, as seen in Figure 45b, with IOVDD because that would force return currents from IOVDD to flow through AGND. Also, avoid digital currents flowing under analog circuitry, which could occur if a noisy digital chip is placed on the left half of the board shown in Figure 45c. If possible, avoid large discontinuities in the ground plane(s) (such as those formed by a long trace on the same layer), because they force return signals to travel a longer path. In addition, make all connections to the ground plane directly, with little or no trace separating the pin from its via to ground.

When connecting fast logic signals (rise/fall time < 5 ns) to any of the ADuC7121 digital inputs, add a series resistor to each relevant line to keep rise and fall times longer than 5 ns at the input pins of the part. A value of 100 Ω or 200 Ω is usually sufficient enough to prevent high speed signals from coupling capacitively into the part and affecting the accuracy of ADC conversions.

## CLOCK OSCILLATOR

The clock source for the ADuC7121 can be generated by the internal PLL or by an external clock input. To use the internal PLL, connect a 32.768 kHz parallel resonant crystal between XTALI and XTALO, and connect a capacitor from each pin to ground as shown Figure 46. This crystal allows the PLL to lock correctly to give a frequency of 41.78 MHz. If no external crystal is present, the internal oscillator is used to give a frequency of 41.78 MHz ± 3% typically.

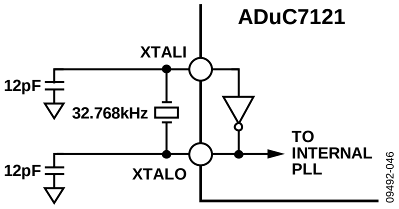

*Figure 46. External Parallel Resonant Crystal Connections*

To use an external source clock input instead of the PLL (see Figure 47), Bit 1 and Bit 0 of PLLCON must be modified. The external clock uses P1.4 and XCLK.

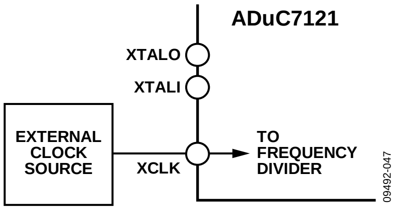

*Figure 47. Connecting an External Clock Source*

Using an external clock source, the ADuC7121 specified operational clock speed range is 50 kHz to 41.78 MHz ± 1% to ensure correct operation of the analog peripherals and Flash/EE.

# OUTLINE DIMENSIONS

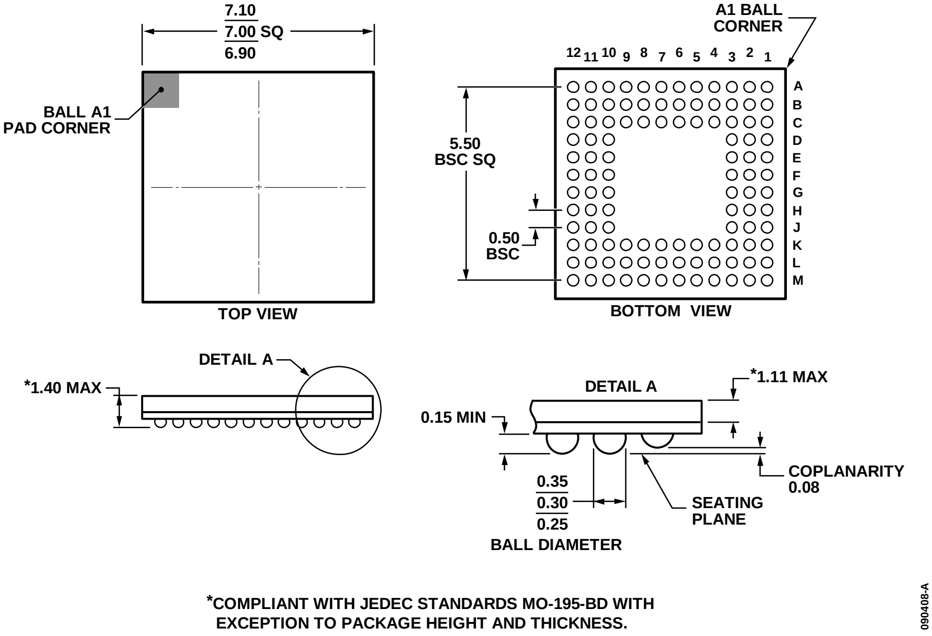

*Figure 48. 108-Ball Chip Scale Package Ball Grid Array [CSP_BGA]*
*(BC-108-4)*
*Dimensions shown in millimeters*

## ORDERING GUIDE

| Model[1] | Temperature Range | Package Description | Package Option |
|---|---|---|---|
| ADuC7121BBCZ | −10°C to +95°C | 108-Ball CSP_BGA | BC-108-4 |
| ADuC7121BBCZ-RL | −10°C to +95°C | 108-Ball CSP_BGA, 13" Tape and Reel | BC-108-4 |
| EVAL-ADuC7121QSPZ | | ADuC7121 QuickStart Development System | |

[1] Z = RoHS Compliant Part.

# NOTES

## NOTES

I$^2$C refers to a communications protocol originally developed by Philips Semiconductors (now NXP Semiconductors).