Precision Analog Microcontroller 12-bit Analog I/O, ARM7TDMI® MCU

ADuC7019/20/21/22/24/25/26/27

FEATURES

Analog I/O
- Multichannel, 12-bit, 1 MSPS ADC
  - Up to 16 ADC channels[1]
- Fully differential and single-ended modes
- 0 to V_{REF} analog input range
- 12-bit voltage output DACs
  - Up to 4 DAC outputs available[1]
- On-chip voltage reference
- On-chip temperature sensor (±3°C)
- Voltage comparator

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 44 MHz
- 41.78 MHz PLL with programmable divider

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

On-chip peripherals
- UART, 2 × I²C® and SPI® serial I/O
- Up to 40-pin GPIO port[1]
- 4 × general-purpose timers
- Wake-up and watchdog timers (WDT)
- Power supply monitor
- Three-phase, 16-bit PWM generator[1]
- Programmable logic array (PLA)
- External memory interface, up to 512 kB[1]

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

Packages and temperature range
- From 40-lead 6 mm × 6 mm LFCSP to 80-lead LQFP[1]
- Fully specified for –40°C to +125°C operation

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

APPLICATIONS

Industrial control and automation systems
Smart sensors, precision instrumentation
Base station systems, optical networking

FUNCTIONAL BLOCK DIAGRAM

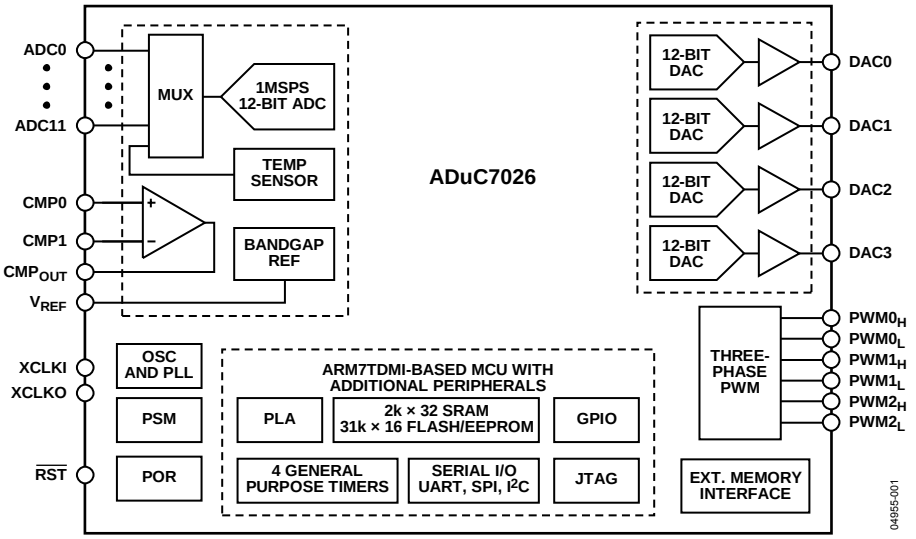


Figure 1.

[1] Depending on part model. See Ordering Guide for more information.

Rev. A

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

TABLE OF CONTENTS

REVISION HISTORY

1/06—Rev. 0 to Rev. A

Changes to Table 1 6
Added the Flash/EE Memory Reliability Section 43
Changes to Table 30 52
Changes to Serial Peripheral Interface 66
Changes to Ordering Guide 90

10/05—Revision 0: Initial Version

GENERAL DESCRIPTION

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 are fully integrated, 1 MSPS, 12-bit data acquisition systems incorporating high performance multichannel ADCs, 16-bit/32-bit MCUs and Flash/EE memory on a single chip.

The ADC consists of up to 12 single-ended inputs. An additional four inputs are available but are multiplexed with the four DAC output pins. The four DAC outputs are only available on certain models (ADuC7020, and ADuC7026). However, in many cases where the DAC outputs are not present, these pins can still be used as additional ADC inputs, giving a maximum of 16 ADC input channels. The ADC can operate in single-ended or differential input modes. The ADC input voltage is 0 to V_{REF}. Low-drift bandgap reference, temperature sensor, and voltage comparator complete the ADC peripheral set.

Depending on the part model, up to four buffered voltage output DACs are available on-chip. 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. 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 kilobytes of SRAM and 62 kilobytes 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 UART or I^2C serial interface ports, while 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 parts operate from 2.7 V to 3.6 V and are specified over an industrial temperature range of −40°C to +125°C. When operating at 41.78 MHz, the power dissipation is typically 120 mW. The ADuC7019/7020/7021/7022/7024/7025/7026/7027 are available in a variety of memory models and packages.

DETAILED BLOCK DIAGRAM

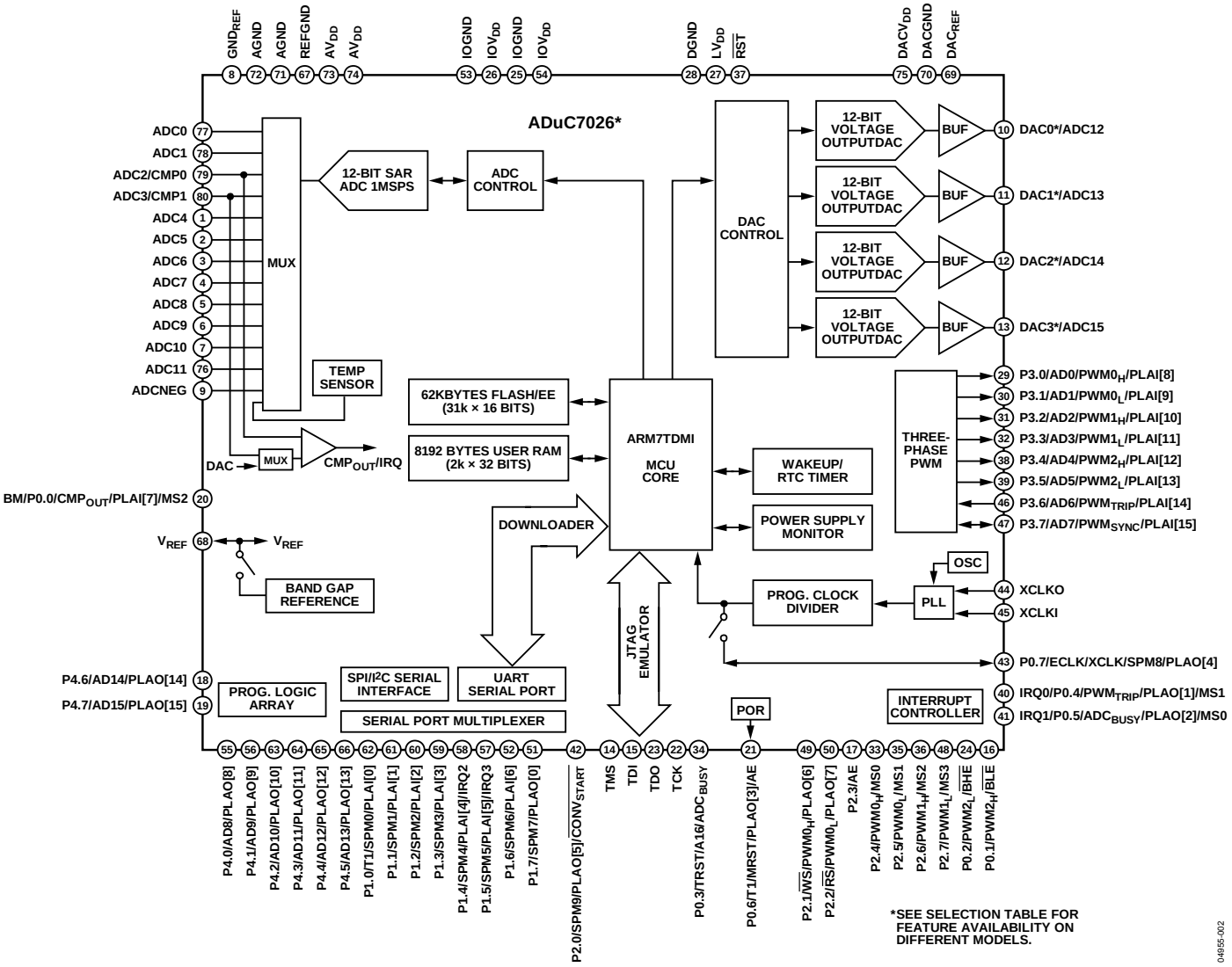


Figure 2.

SPECIFICATIONS

AVDD = IOV_{DD} = 2.7 V to 3.6 V, V_{REF} = 2.5 V internal reference, f_{CORE} = 41.78 MHz, T_A = 40°C to 125°C, unless otherwise noted.

Table 1.

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
ADC CHANNEL SPECIFICATIONS					Eight acquisition clocks and fADC/2
ADC Power-Up Time		5		µs	
DC Accuracy[1, 2]					
Resolution	12			Bits	
Integral Nonlinearity		±0.6	±1.5	LSB	2.5 V internal reference
		±1.0		LSB	1.0 V external reference
Differential Nonlinearity[3, 4]		±0.5	+1/−0.9	LSB	2.5 V internal reference
		+0.7/−0.6		LSB	1.0 V external reference
DC Code Distribution		1		LSB	ADC input is a dc voltage
ENDPOINT ERRORS[5]					
Offset Error		±1	±2	LSB	
Offset Error Match		±1		LSB	
Gain Error		±2	±5	LSB	
Gain Error Match		±1		LSB	
DYNAMIC PERFORMANCE					f_{IN} = 10 kHz sine wave, f_{SAMPLE} = 1 MSPS
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	
Single-Ended Mode			0 to V_{REF}	V	
Leakage Current		±1	±6	µA	
Input Capacitance		20		pF	During ADC acquisition
ON-CHIP VOLTAGE REFERENCE					0.47 µF from V_{REF} to AGND
Output Voltage		2.5		V	
Accuracy			±5	mV	T_A = 25°C
Reference Temperature Coefficient		±40		ppm/°C	
Power Supply Rejection Ratio		75		dB	
Output Impedance		70		Ω	T_A = 25°C
Internal V_{REF} Power-On Time		1		ms	
EXTERNAL REFERENCE INPUT[7]					
Input Voltage Range	0.625		AV_{DD}	V	
Input Impedance		65		kΩ	
DAC CHANNEL SPECIFICATIONS					R_L = 5 kΩ, C_L = 100 pF
DC ACCURACY[8]					
Resolution		12		Bits	
Relative Accuracy		±2		LSB	
Differential Nonlinearity			±1	LSB	Guaranteed monotonic
Offset Error			±15	mV	2.5 V internal reference
Gain Error[9]			±1	%	
Gain Error Mismatch		0.1		%	% of full scale on DAC0

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
ANALOG OUTPUTS					
Output Voltage Range_0		0 to DAC_{REF}		V	DAC_{REF} range: DACGND to $DACV_{DD}$
Output Voltage Range_1		0 to 2.5		V	
Output Voltage Range_2		0 to $DACV_{DD}$		V	
Output Impedance		2		Ω	
DAC AC CHARACTERISTICS					
Voltage Output Settling Time		10		µs	
Digital to Analog Glitch Energy		±20		nV-sec	1 LSB change at major carry
COMPARATOR					
Input Offset Voltage		±15		mV	
Input Bias Current		1		µA	
Input Voltage Range	AGND		AV_{DD} – 1.2	V	
Input Capacitance		7		pF	
Hysteresis[4, 6]	2		15	mV	Hysteresis can be turned on or off via the CMPHYST bit in the CMPCON register
Response Time		3		µs	100 mV overdrive and configured with CMPRES = 11
TEMPERATURE SENSOR					
Voltage Output at 25°C		780		mV	
Voltage TC		−1.3		mV/°C	
Accuracy		±3		°C	
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	
GLITCH IMMUNITY ON RESET PIN[3]		50		µs	
WATCHDOG TIMER (WDT)					
Timeout Period	0		512	sec	
FLASH/EE MEMORY					
Endurance[10]	10,000			cycles	
Data Retention[11]	20			years	T_J = 85°C
DIGITAL INPUTS					All digital inputs excluding XCLKI and XCLKO
Logic 1 Input Current		±0.2	±1	µA	V_{IH} = VDD or V_{IH} = 5 V
Logic 0 Input Current		−40	−60	µA	V_{IL} = 0 V; except TDI on ADuC7019/20/21/22/24/25
		−80	−120	µA	V_{IL} = 0 V; TDI, on ADuC7019/20/21/22/24/25
Input Capacitance		10		pF	
LOGIC INPUTS[3]					All logic inputs excluding XCLKI and XCLKO
V_{INL}, Input Low Voltage			0.8	V	
V_{INH}, Input High Voltage	2.0			V	
LOGIC OUTPUTS					All digital outputs excluding XCLKI and XCLKO
V_{OH}, Output High Voltage	2.4			V	I_{SOURCE} = 1.6 mA
V_{OL}, Output Low Voltage[12]			0.4	V	I_{SINK} = 1.6 mA
CRYSTAL INPUTS XCLKI and XCLKO					
Logic Inputs, XCLKI Only					
V_{INL}, Input Low Voltage		1.1		V	
V_{INH}, Input High Voltage		1.7		V	
XCLKI Input Capacitance		20		pF	
XCLKO Output Capacitance		20		pF	

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
INTERNAL OSCILLATOR		32.768		kHz	
			±3	%	
MCU CLOCK RATE					
From 32 kHz Internal Oscillator		326		kHz	CD = 7
From 32 kHz External Crystal		41.78		MHz	CD = 0
Using an External Clock	0.05		44	MHz	T_A = 85°C
	0.05		41.78	MHz	T_A = 125°C
START-UP TIME					Core clock = 41.78 MHz
At Power-On		130		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 [13, 14]					
Power Supply Voltage Range					
AV_{DD} − AGND and IOV_{DD} − IOGND	2.7		3.6	V	
Analog Power Supply Currents					
AV_{DD} Current		200		µA	ADC in idle mode; all parts except ADuC7019
		400		µA	ADC in idle mode; ADuC7019 only
$DACV_{DD}$ Current[15]		3	25	µA	
Digital Power Supply Current					
IOV_{DD} Current in Normal Mode					Code executing from Flash/EE
		7	10	mA	CD = 7
		11	15	mA	CD = 3
		40	45	mA	CD = 0 (41.78 MHz clock)
IOV_{DD} Current in Pause Mode		25	30	mA	CD = 0 (41.78 MHz clock)
IOV_{DD} Current in Sleep Mode		250	400	µA	T_A = 85°C
		600	1000	µA	T_A = 125°C
Additional Power Supply Currents					
ADC		2		mA	@ 1 MSPS
		0.7		mA	@ 62.5 kSPS
DAC		700		µA	per DAC

[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 ADCOF and 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 using an external AD845 op amp as an input buffer stage as shown in Figure 47. 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 Calibration section).
[6] The input signal can be centered on any dc common-mode voltage (V_{CM}) as long as this value is within the ADC voltage input range specified.
[7] When using an external reference input pin, the internal reference must be disabled by setting the LSB in the REFCON memory mapped register to 0.
[8] DAC linearity is calculated using a reduced code range of 100 to 3995.
[9] DAC gain error is calculated using a reduced code range of 100 to internal 2.5 V V_{REF}.
[10] Endurance is qualified as per JEDEC Standard 22 method A117 and measured at −40°C, +25°C, +85°C, and +125°C.
[11] Retention lifetime equivalent at junction temperature (T_J) = 85°C as per JEDEC Standard 22 method A117. Retention lifetime derates with junction temperature.
[12] Test carried out with a maximum of eight I/O set to a low output level.
[13] Power supply current consumption is measured in normal, pause, and sleep modes under the following conditions: Normal Mode: 3.6 V supply, Pause Mode: 3.6 V supply, Sleep Mode: 3.6 V supply.
[14] IOV_{DD} power supply current decreases typically by 2 mA during a flash/EE erase cycle.
[15] On the ADuC7019/20/21/22, this current must be added to AV_{DD} current.

TIMING SPECIFICATIONS

Table 2. External Memory Write Cycle

Parameter	Min	Typ	Max	Unit
CLK		UCLK		
$T_{MS\_AFTER\_CLKH}$	0		4	ns
$T_{ADDR\_AFTER\_CLKH}$	4		8	ns
$T_{AE\_H\_AFTER\_MS}$		½ CLK		
T_{AE}		(XMxPAR[14:12] + 1) x CLK		
$T_{HOLD\_ADDR\_AFTER\_AE\_L}$		½ CLK + (!XMxPAR[10]) x CLK		
$T_{HOLD\_ADDR\_BEFORE\_WR\_L}$		(!XMxPAR[8]) x CLK		
$T_{WR\_L\_AFTER\_AE\_L}$		½ CLK + (!XMxPAR[10] + !XMxPAR[8]) x CLK		
$T_{DATA\_AFTER\_WR\_L}$	8		12	ns
T_{WR}		(XMxPAR[7:4] + 1) x CLK		
$T_{WR\_H\_AFTER\_CLKH}$	0		4	ns
$T_{HOLD\_DATA\_AFTER\_WR\_H}$		(!XMxPAR[8]) x CLK		
$T_{BEN\_AFTER\_AE\_L}$		½ CLK		
$T_{RELEASE\_MS\_AFTER\_WR\_H}$		(!XMxPAR[8] + 1) x CLK		

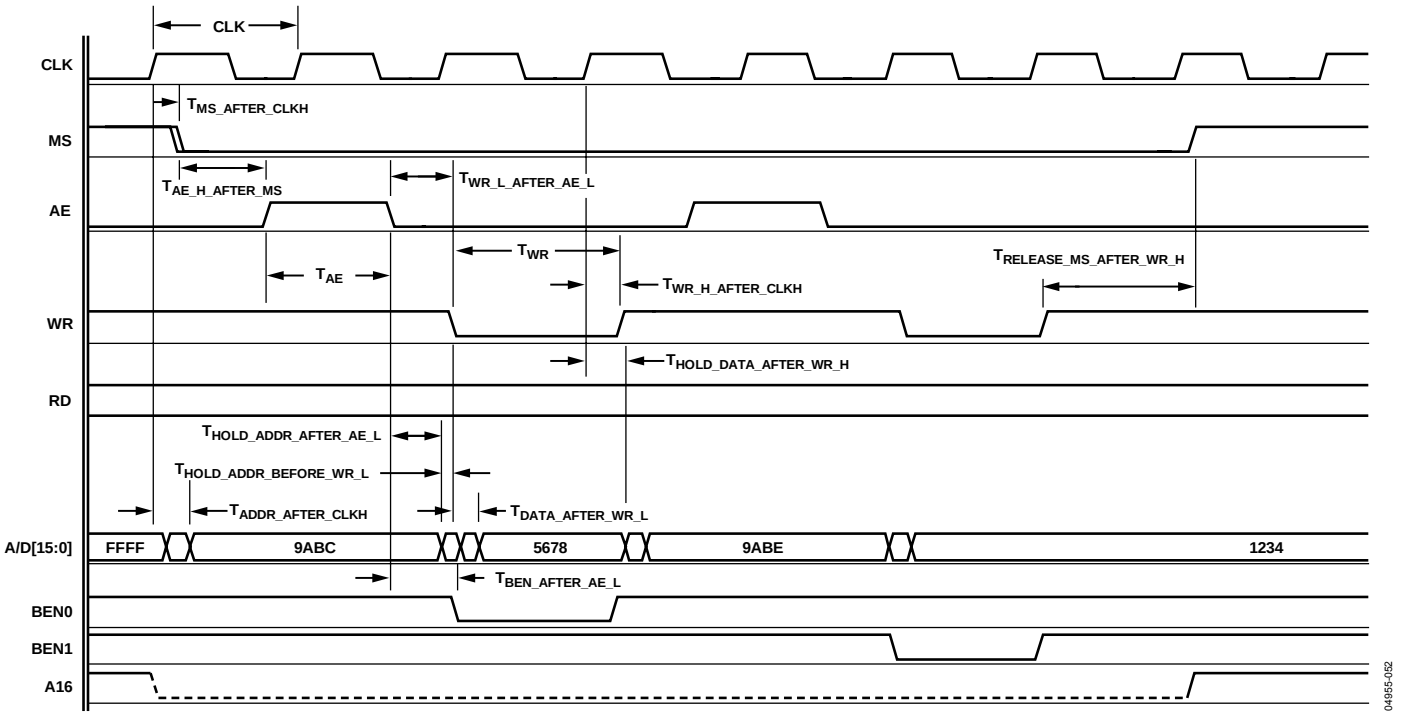


Figure 3. External Memory Write Cycle

Table 3. External Memory Read Cycle

Parameter	Min	Typ	Max	Unit
CLK		UCLK		
$T_{MS\_AFTER\_CLKH}$	4		8	ns
$T_{ADDR\_AFTER\_CLKH}$	4		16	ns
$T_{AE\_H\_AFTER\_MS}$		½ CLK		
T_{AE}		(XMxPAR[14:12] + 1) x CLK		
$T_{HOLD\_ADDR\_AFTER\_AE\_L}$		½ CLK + (!XMxPAR[10]) x CLK		
$T_{RD\_L\_AFTER\_AE\_L}$		½ CLK + (!XMxPAR[10] + !XMxPAR[9]) x CLK		
$T_{DATA\_AFTER\_RD\_L}$	8		12	ns
T_{RD}		(XMxPAR[3:0] + 1) x CLK		
$T_{RD\_H\_AFTER\_CLKH}$	0		4	ns
$T_{HOLD\_DATA\_AFTER\_RD\_H}$		(!XMxPAR[9]) x CLK		
$T_{RELEASE\_MS\_AFTER\_RD\_H}$		CLK		

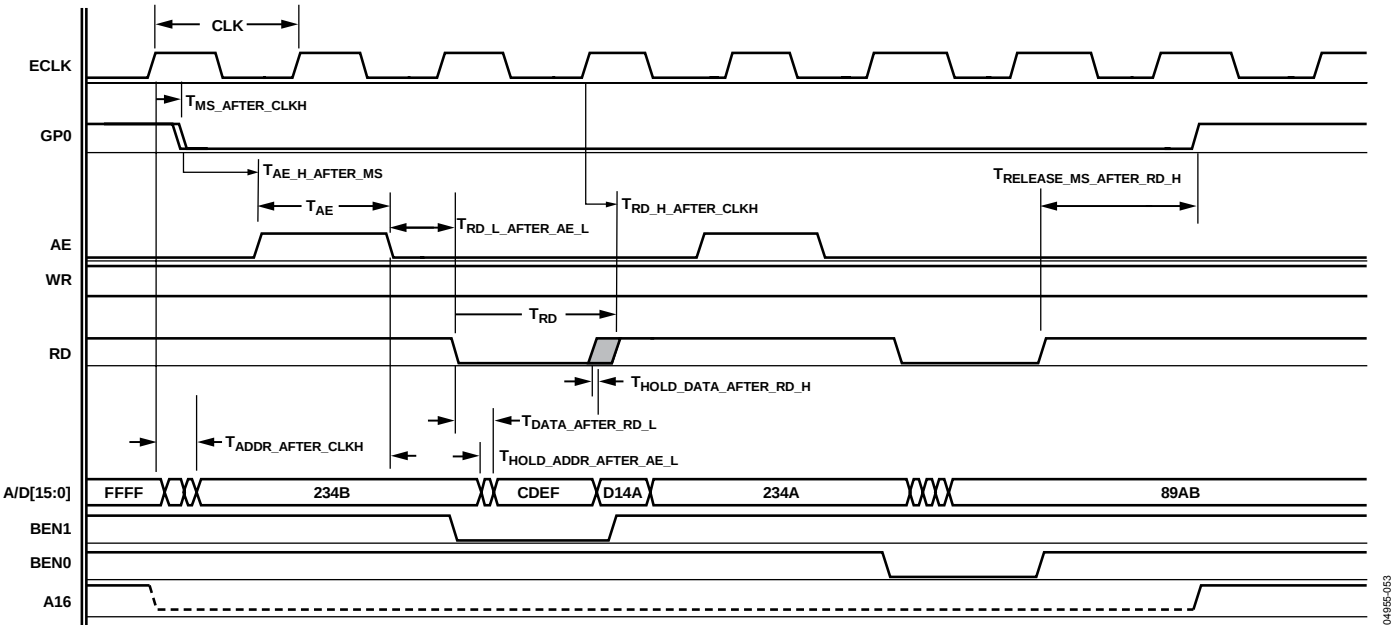


Figure 4. External Memory Read Cycle

Table 4. I²C Timing in Fast Mode (400 kHz)

Parameter	Description	Slave		Master Typ	Unit
		Min	Max		
t_L	SCLOCK low pulse width[1]	200		1360	ns
t_H	SCLOCK high pulse width[1]	100		1140	ns
t_{SHD}	Start condition hold time	300		251350	ns
t_{DSU}	Data setup time	100		740	ns
t_{DHD}	Data hold time	50		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 CLOCK and SDATA	100	300	200	ns
t_F	Fall time for both CLOCK and SDATA	60	100	20	ns
t_{SUP}	Pulse width of spike suppressed		50		ns

[1] t_{HCLK} depends on the clock divider or CD bits in PLLCON MMR. $t_{HCLK} = t_{UCLK}/2^{CD}$.

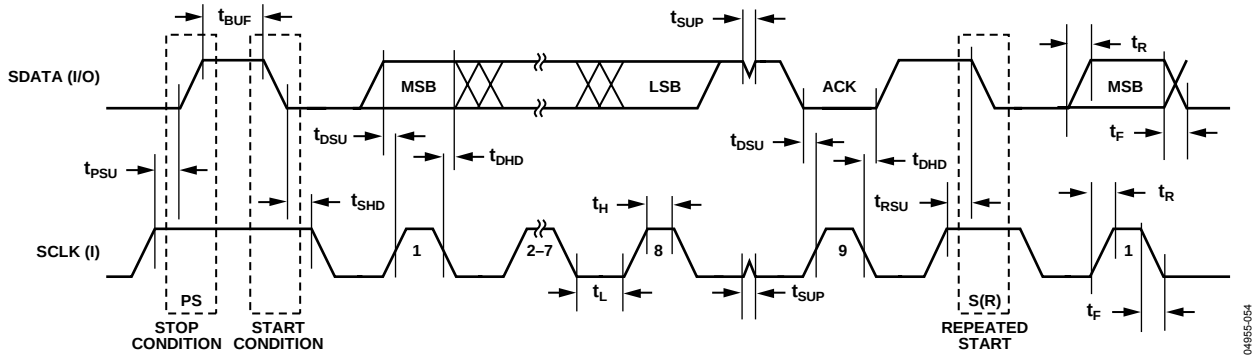


Figure 5. I²C Compatible Interface Timing

Table 5. SPI Master Mode Timing (PHASE Mode = 1)

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

[1] t_{HCLK} depends on the clock divider or CD bits in PLLCON MMR. $t_{HCLK} = t_{UCLK}/2^{CD}$.
[2] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider.

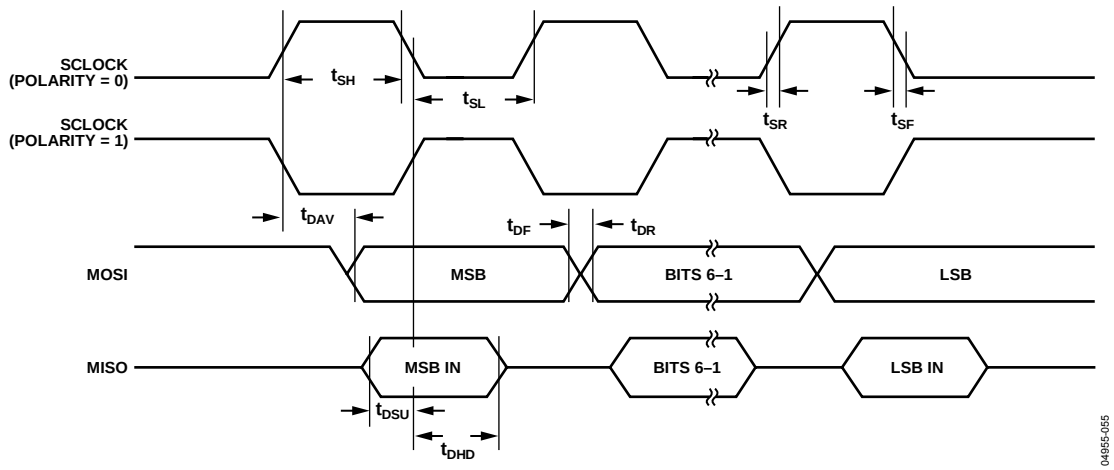


Figure 6. SPI Master Mode Timing (PHASE Mode = 1)

Table 6. SPI Master Mode Timing (PHASE Mode = 0)

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

[1] t_{HCLK} depends on the clock divider or CD bits in PLLCON MMR. $t_{HCLK} = t_{UCLK}/2^{CD}$.
[2] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider.

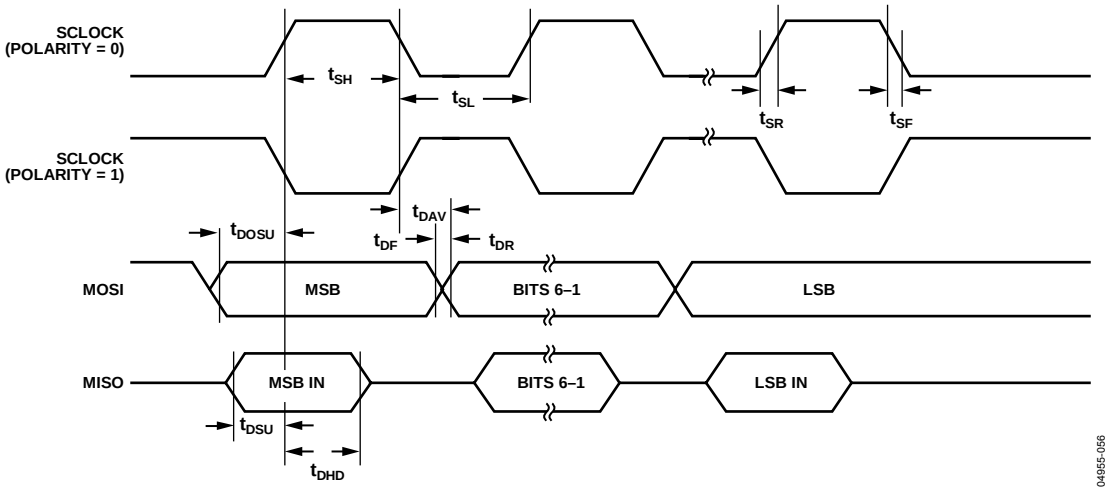


Figure 7. SPI Master Mode Timing (PHASE Mode = 0)

Table 7. SPI Slave Mode Timing (PHASE Mode = 1)

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

[1] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider.
[2] t_{HCLK} depends on the clock divider or CD bits in PLLCON MMR. $t_{HCLK} = t_{UCLK}/2^{CD}$.

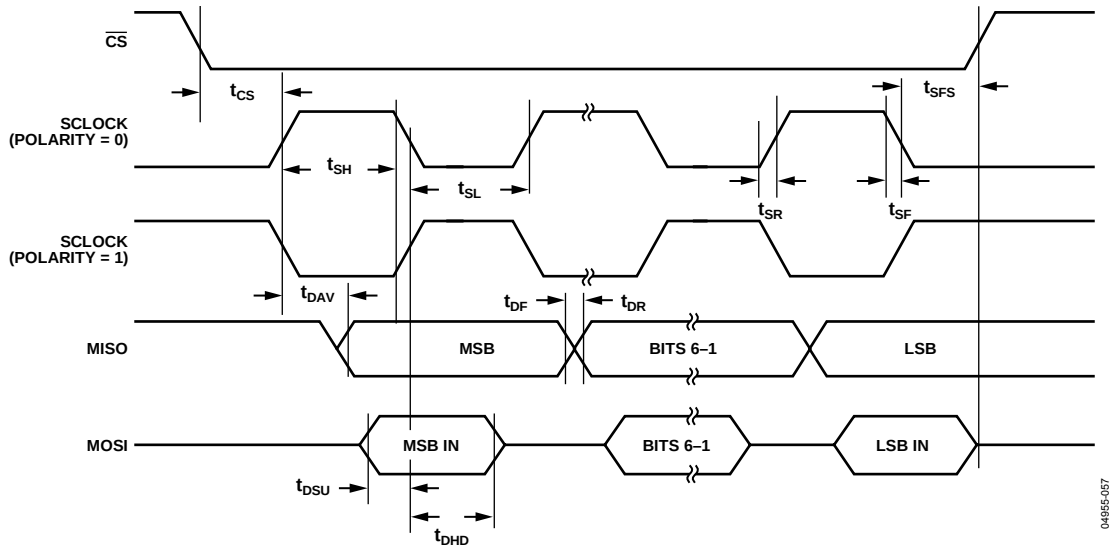


Figure 8. SPI Slave Mode Timing (PHASE Mode = 1)

Table 8. SPI Slave Mode Timing (PHASE Mode = 0)

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

[1] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider.
[2] t_{HCLK} depends on the clock divider or CD bits in PLLCON MMR. $t_{HCLK} = t_{UCLK}/2^{CD}$.

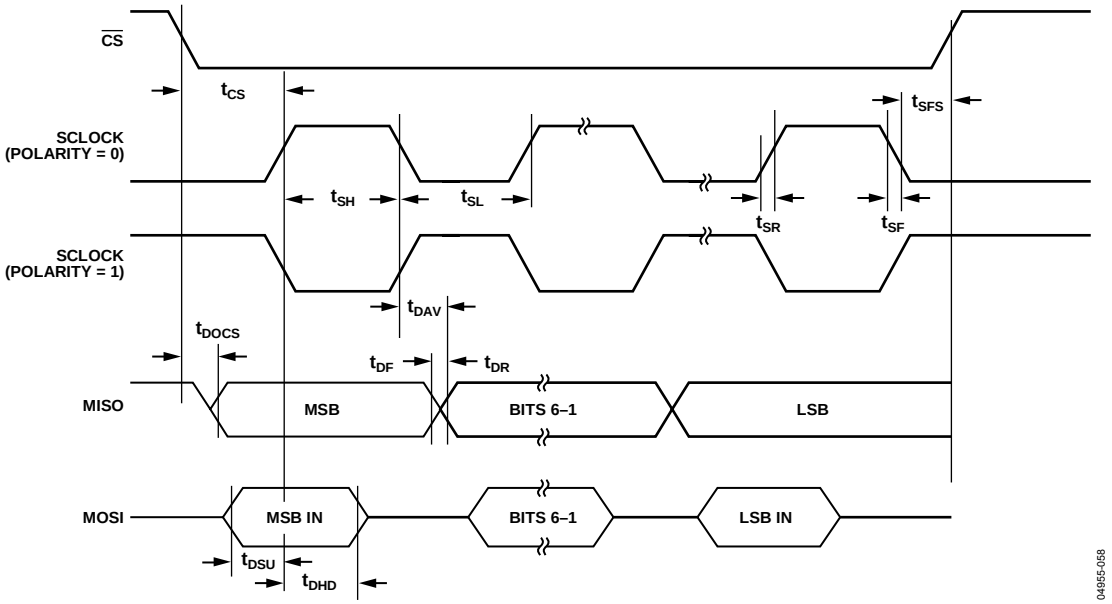


Figure 9. SPI Slave Mode Timing (PHASE Mode = 0)

ABSOLUTE MAXIMUM RATINGS

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

Table 9.

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} 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	−40°C to +125°C
Storage Temperature Range	−65°C to +150°C
Junction Temperature	150°C
θ_{JA} Thermal Impedance (40-pin CSP)	26°C/W
θ_{JA} Thermal Impedance (64-pin CSP)	24°C/W
θ_{JA} Thermal Impedance (64-pin LQFP)	47°C/W
θ_{JA} Thermal Impedance (80-pin LQFP)	38°C/W
Peak Solder Reflow Temperature	
SnPb Assemblies (10 sec to 30 sec)	240°C
Pb-Free 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. Electrostatic charges as high as 4000 V readily accumulate on the human body and test equipment and can discharge without detection. Although this product features proprietary ESD protection circuitry, permanent damage may occur on devices subjected to high energy electrostatic discharges. Therefore, proper ESD precautions are recommended to avoid performance degradation or loss of functionality.

PIN CONFIGURATIONS AND FUNCTION DESCRIPTIONS

ADuC7019/ADuC7020/ADuC7021/ADuC7022

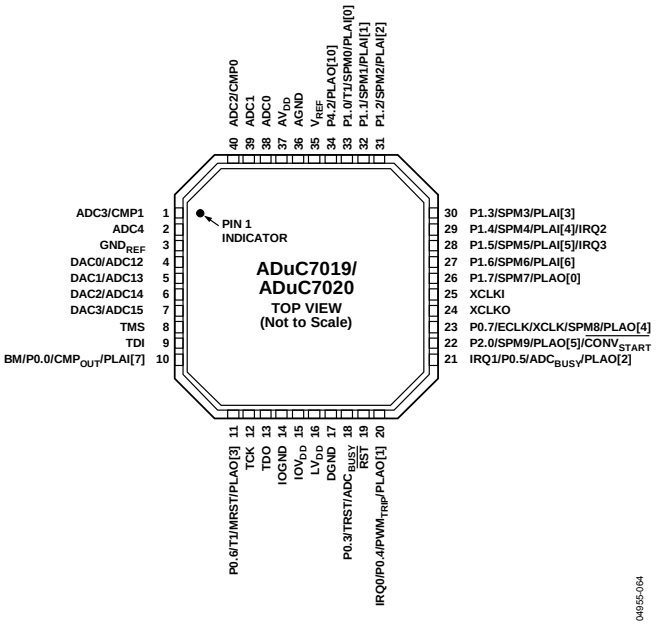


Figure 10. ADuC7019/ADuC7020 40-Lead LFCSP_VQ Pin Configuration

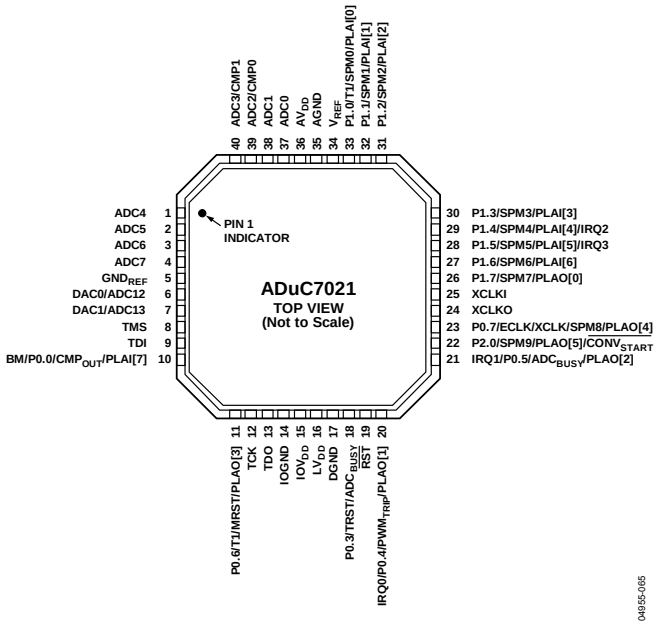


Figure 11. ADuC702140-Lead LFCSP_VQ Pin Configuration

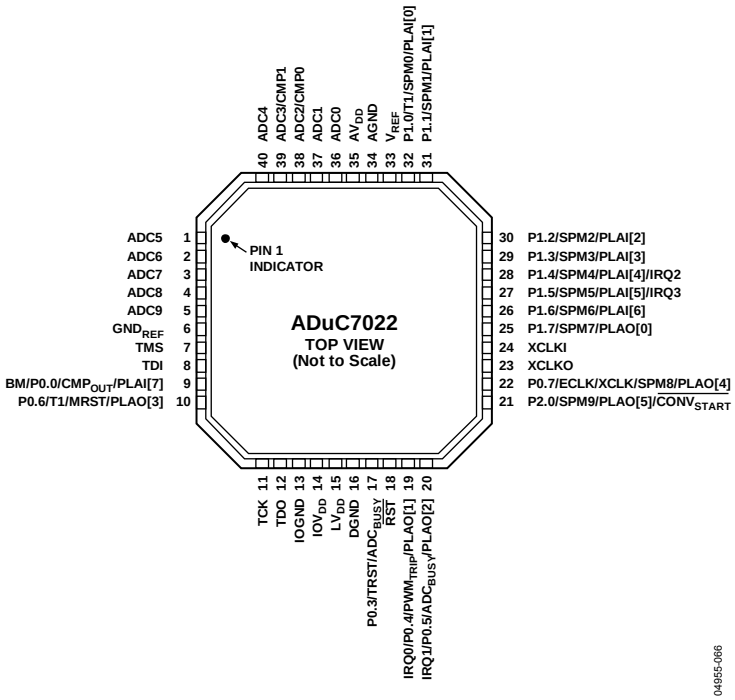


Figure 12. ADuC7022 40-Lead LFCSP_VQ Pin Configuration

Table 10. Pin Function Descriptions (ADuC7019/ADuC7020/ADuC7021/ADuC7022)

Pin No.				
7019/7020	**7021**	**7022**	**Mnemonic**	**Description**
38	37	36	ADC0	Single-Ended or Differential Analog Input 0.
39	38	37	ADC1	Single-Ended or Differential Analog Input 1.
40	39	38	ADC2/CMP0	Single-Ended or Differential Analog Input 2/Comparator Positive Input.
1	40	39	ADC3/CMP1	Single-Ended or Differential Analog Input 3 (Buffered Input on ADuC7019)/Comparator Negative Input.
2	1	40	ADC4	Single-Ended or Differential Analog Input 4.
–	2	1	ADC5	Single-Ended or Differential Analog Input 5.
–	3	2	ADC6	Single-Ended or Differential Analog Input 6.
–	4	3	ADC7	Single-Ended or Differential Analog Input 7.
–	–	4	ADC8	Single-Ended or Differential Analog Input 8.
–	–	5	ADC9	Single-Ended or Differential Analog Input 9.
3	5	6	GND_{REF}	Ground Voltage Reference for the ADC. For optimal performance, the analog power supply should be separated from IOGND and DGND.
4	6	–	DAC0/ADC12	DAC0 Voltage Output/Single-Ended or Differential Analog Input 12.
5	7	–	DAC1/ADC13	DAC1 Voltage Output/Single-Ended or Differential Analog Input 13.
6	–	–	DAC2/ADC14	DAC2 Voltage Output/Single-Ended or Differential Analog Input 14.
7	–	–	DAC3/ADC15	DAC3 Voltage Output on ADuC7020. On the ADuC7019, a 10 nF capacitor needs to be connected between this pin and AGND/Single-Ended or Differential Analog Input 15.
8	8	7	TMS	Test Mode Select, JTAG Test Port Input. Debug and download access.
9	9	8	TDI	Test Data In, JTAG Test Port Input. Debug and download access.
10	10	9	BM/P0.0/CMP_{OUT}/PLAI[7]	Multifunction I/O Pin. Boot Mode (BM). The ADuC7019/20/21/22 enter serial download mode if BM is low at reset and execute code if BM is pulled high at reset through a 1 kΩ resistor. General-Purpose Input and Output Port 0.0/Voltage Comparator Output/Programmable Logic Array Input Element 7.
11	11	10	P0.6/T1/MRST/PLAO[3]	Multifunction Pin, Driven Low After Reset. General-Purpose Output Port 0.6/Timer1 Input/Power-On Reset Output/Programmable Logic Array Output Element 3.
12	12	11	TCK	Test Clock, JTAG Test Port Input. Debug and download access.
13	13	12	TDO	Test Data Out, JTAG Test Port Output. Debug and download access.
14	14	13	IOGND	Ground for GPIO. Typically connected to DGND.
15	15	14	IOV_{DD}	3.3 V Supply for GPIO and Input of the On-Chip Voltage Regulator.
16	16	15	LV_{DD}	2.6 V Output of the On-Chip Voltage Regulator. This output must be connected to a 0.47 µf capacitor to DGND only.
17	17	16	DGND	Ground for Core Logic.
18	18	17	P0.3/TRST/ADC_{BUSY}	General-Purpose Input and Output Port 0.3/Test Reset, JTAG Test Port Input/ ADC_{BUSY} Signal Output.
19	19	18	$\overline{RST}$	Reset Input, Active Low.
20	20	19	IRQ0/P0.4/PWM_{TRIP}/PLAO[1]	Multifunction I/O Pin. External Interrupt Request 0, Active High/General-Purpose Input and Output Port 0.4/PWM Trip External Input/Programmable Logic Array Output Element 1.
21	21	20	IRQ1/P0.5/ADC_{BUSY}/PLAO[2]	Multifunction I/O Pin. External Interrupt Request 1, Active High/General-Purpose Input andOutput Port 0.5/ADC_{BUSY} Signa l Output/Programmable Logic Array Output Element 2.
22	22	21	P2.0/SPM9/PLAO[5]/$\overline{CONV_{START}}$	Serial Port Multiplexed. General-Purpose Input and Output Port 2.0/UART/ Programmable Logic Array Output Element 5/Start Conversion Input Signal for ADC.
23	23	22	P0.7/ECLK/XCLK/SPM8/PLAO[4]	Serial Port Multiplexed. General-Purpose Input and Output Port 0.7/Output for External Clock Signal/Input to the Internal Clock Generator Circuits/UART/ Programmable Logic Array Output Element 4.
24	24	23	XCLKO	Output from the Crystal Oscillator Inverter.

Pin No.				
7019/7020	7021	7022	Mnemonic	Description
25	25	24	XCLKI	Input to the Crystal Oscillator Inverter and Input to the Internal Clock Generator Circuits.
26	26	25	P1.7/SPM7/PLAO[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.7/UART, SPI/Programmable Logic Array Output Element 0.
27	27	26	P1.6/SPM6/PLAI[6]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.6/UART, SPI/Programmable Logic Array Input Element 6.
28	28	27	P1.5/SPM5/PLAI[5]/IRQ3	Serial Port Multiplexed. General-Purpose Input and Output Port 1.5/UART, SPI/Programmable Logic Array Input Element 5/External Interrupt Request 3, Active High.
29	29	28	P1.4/SPM4/PLAI[4]/IRQ2	Serial Port Multiplexed. General-Purpose Input and Output Port 1.4/UART, SPI/Programmable Logic Array Input Element 4/External Interrupt Request 2, Active High.
30	30	29	P1.3/SPM3/PLAI[3]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.3/UART, I^2C1/Programmable Logic Array Input Element 3.
31	31	30	P1.2/SPM2/PLAI[2]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.2/UART, I^2C1/Programmable Logic Array Input Element 2.
32	32	31	P1.1/SPM1/PLAI[1]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.1/UART, I^2C0/Programmable Logic Array Input Element 1.
33	33	32	P1.0/T1/SPM0/PLAI[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.0/ Timer1 Input/UART, I^2C0/Programmable Logic Array Input Element 0.
34	–	–	P4.2/PLAO[10]	General-Purpose Input and Output Port 4.2/Programmable Logic Array Output Element 10.
35	34	33	V_{REF}	2.5 V Internal Voltage Reference. Must be connected to a 0.47 µF capacitor when using the internal reference.
36	35	34	AGND	Analog Ground. Ground reference point for the analog circuitry.
37	36	35	AV_{DD}	3.3 V Analog Power.

ADuC7024/ADuC7025

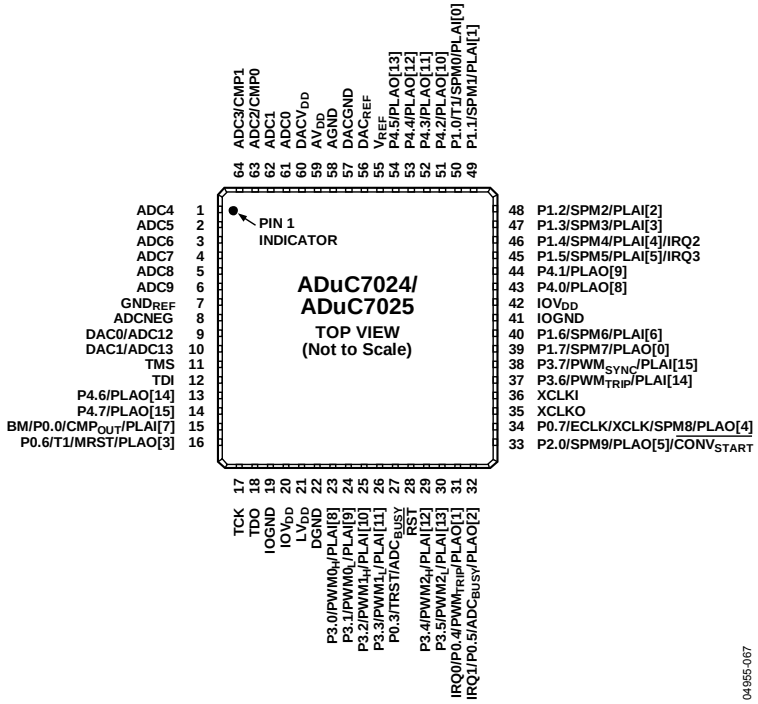


Figure 13. ADuC7024/ADuC7025 64-Lead LFCSP_VQ Pin Configuration

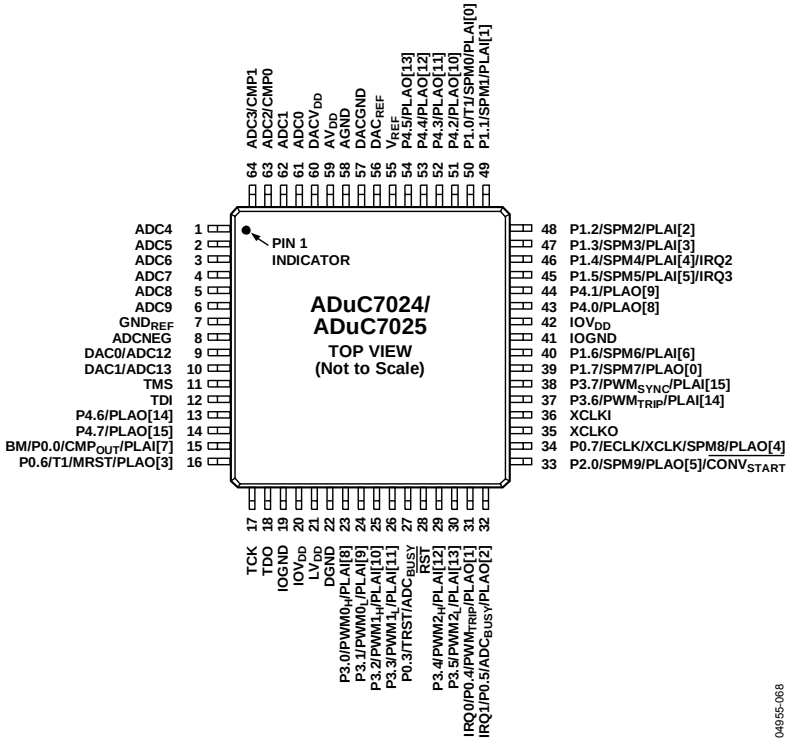


Figure 14. ADuC7024/ADuC7025 64-Lead LQFP Pin Configuration

Table 11. Pin Function Descriptions (ADuC7024/ADuC7025 64-Lead CSP and ADuC7024/ADuC7025 64-Lead LQFP)

Pin No.	Mnemonic	Description
1	ADC4	Single-Ended or Differential Analog Input 4.
2	ADC5	Single-Ended or Differential Analog Input 5.
3	ADC6	Single-Ended or Differential Analog Input 6.
4	ADC7	Single-Ended or Differential Analog Input 7.
5	ADC8	Single-Ended or Differential Analog Input 8.
6	ADC9	Single-Ended or Differential Analog Input 9.
7	GND_{REF}	Ground Voltage Reference for the ADC. For optimal performance, the analog power supply should be separated from IOGND and DGND.
8	ADCNEG	Bias Point or Negative Analog Input of the ADC in Pseudo Differential Mode. Must be connected to the ground of the signal to convert. This bias point must be between 0 V and 1 V.
9	DAC0/ADC12	DAC0 Voltage Output/Single-Ended or Differential Analog Input 12. DAC outputs are not present on the ADuC7025.
10	DAC1/ADC13	DAC1 Voltage Output/Single-Ended or Differential Analog Input 13. DAC outputs are not present on the ADuC7025.
11	TMS	JTAG Test Port Input, Test Mode Select. Debug and download access.
12	TDI	JTAG Test Port Input, Test Data In. Debug and download access
13	P4.6/PLAO[14]	General-Purpose Input and Output Port 4.6/Programmable Logic Array Output Element 14.
14	P4.7/PLAO[15]	General-Purpose Input and Output Port 4.7/Programmable Logic Array Output Element 15.
15	BM/P0.0/CMP_{OUT}/PLAI[7]	Multifunction I/O Pin. Boot mode. The ADuC7024/ADuC7025 enter download mode if BM is low at reset and executes code if BM is pulled high at reset through a 1 kΩ resistor/General-Purpose Input and Output Port 0.0/Voltage Comparator Output/Programmable Logic Array Input Element 7.
16	P0.6/T1/MRST/PLAO[3]	Multifunction Pin, Driven Low After Reset. General-Purpose Output Port 0.6/Timer1 Input/Power-On Reset Output/Programmable Logic Array Output Element 3.
17	TCK	JTAG Test Port Input, Test Clock. Debug and download access.
18	TDO	JTAG Test Port Output, Test Data Out. Debug and download access.
19	IOGND	Ground for GPIO. Typically connected to DGND.
20	IOV_{DD}	3.3 V Supply for GPIO and Input of the On-Chip Voltage Regulator.
21	LV_{DD}	2.6 V Output of the On-Chip Voltage Regulator. This output must be connected to a 0.47 µF capacitor to DGND only.
22	DGND	Ground for Core Logic.
23	P3.0/$PWM0_H$/PLAI[8]	General-Purpose Input and Output Port 3.0/PWM Phase 0 High-Side Output/Programmable Logic Array Input Element 8.
24	P3.1/$PWM0_L$/PLAI[9]	General-Purpose Input and Output Port 3.1/PWM Phase 0 Low-Side Output/Programmable Logic Array Input Element 9.
25	P3.2/$PWM1_H$/PLAI[10]	General-Purpose Input and Output Port 3.2/PWM Phase 1 High-Side Output/Programmable Logic Array Input Element 10.
26	P3.3/$PWM1_L$/PLAI[11]	General-Purpose Input and Output Port 3.3/PWM Phase 1 Low-Side Output/Programmable Logic Array Input Element 11.
27	P0.3/TRST/ADC_{BUSY}	General-Purpose Input and Output Port 0.3/JTAG Test Port Input, Test Reset/ADC_{BUSY} Signal Output.
28	$\overline{RST}$	Reset Input, Active Low.
29	P3.4/$PWM2_H$/PLAI[12]	General-Purpose Input and Output Port 3.4/PWM Phase 2 High-Side Output/Programmable Logic Array Input 12.
30	P3.5/$PWM2_L$/PLAI[13]	General-Purpose Input and Output Port 3.5/PWM Phase 2 Low-Side Output/Programmable Logic Array Input Element 13.
31	IRQ0/P0.4/PWM_{TRIP}/PLAO[1]	Multifunction I/O Pin. External Interrupt Request 0, Active High/General-Purpose Input and Output Port 0.4/PWM Trip External Input/Programmable Logic Array Output Element 1.
32	IRQ1/P0.5/ADC_{BUSY}/PLAO[2]	Multifunction I/O Pin. External Interrupt Request 1, Active High/General-Purpose Input and Output Port 0.5/ADC_{BUSY} Signal Output/Programmable Logic Array Output Element 2.
33	P2.0/SPM9/PLAO[5]/$\overline{CONV_{START}}$	Serial Port Multiplexed. General-Purpose Input and Output Port 2.0/UART/Programmable Logic Array Output Element 5/Start Conversion Input Signal for ADC.
34	P0.7/ECLK/XCLK/SPM8/PLAO[4]	Serial Port Multiplexed. General-Purpose Input and Output Port 0.7/Output for External Clock Signal/Input to the Internal Clock Generator Circuits/UART/Programmable Logic Array Output Element 4.
35	XCLKO	Output from the Crystal Oscillator Inverter.

Pin No.	Mnemonic	Description
36	XCLKI	Input to the Crystal Oscillator Inverter and Input to the Internal Clock Generator Circuits.
37	P3.6/PWM_{TRIP}/PLAI[14]	General-Purpose Input and Output Port 3.6/PWM Safety Cut Off/Programmable Logic Array Input Element 14.
38	P3.7/PWM_{SYNC}/PLAI[15]	General-Purpose Input and Output Port 3.7/PWM Synchronization Input Output/Programmable Logic Array Input Element 15.
39	P1.7/SPM7/PLAO[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.7/UART, SPI/Programmable Logic Array Output Element 0.
40	P1.6/SPM6/PLAI[6]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.6/UART, SPI/Programmable Logic Array Input Element 6.
41	IOGND	Ground for GPIO. Typically connected to DGND.
42	IOV_{DD}	3.3 V Supply for GPIO and Input of the On-Chip Voltage Regulator.
43	P4.0/PLAO[8]	General-Purpose Input and Output Port 4.0/Programmable Logic Array Output Element 8.
44	P4.1/PLAO[9]	General-Purpose Input and Output Port 4.1/Programmable Logic Array Output Element 9.
45	P1.5/SPM5/PLAI[5]/IRQ3	Serial Port Multiplexed. General-Purpose Input and Output Port 1.5/UART, SPI/Programmable Logic Array Input Element 5/External Interrupt Request 3, Active High.
46	P1.4/SPM4/PLAI[4]/IRQ2	Serial Port Multiplexed. General-Purpose Input and Output Port 1.4/UART, SPI/Programmable Logic Array Input Element 4/External Interrupt Request 2, Active High.
47	P1.3/SPM3/PLAI[3]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.3/UART, I2C1/Programmable Logic Array Input Element 3.
48	P1.2/SPM2/PLAI[2]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.2/UART, I2C1/Programmable Logic Array Input Element 2.
49	P1.1/SPM1/PLAI[1]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.1/UART, I2C0/Programmable Logic Array Input Element 1.
50	P1.0/T1/SPM0/PLAI[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.0/Timer1 Input/UART, I^2C0/Programmable Logic Array Input Element 0.
51	P4.2/PLAO[10]	General-Purpose Input and Output Port 4.2/Programmable Logic Array Output Element 10.
52	P4.3/PLAO[11]	General-Purpose Input and Output Port 4.3/Programmable Logic Array Output Element 11.
53	P4.4/PLAO[12]	General-Purpose Input and Output Port 4.4/Programmable Logic Array Output Element 12.
54	P4.5/PLAO[13]	General-Purpose Input and Output Port 4.5/Programmable Logic Array Output Element 13.
55	V_{REF}	2.5 V Internal Voltage Reference. Must be connected to a 0.47 µF capacitor when using the internal reference.
56	DAC_{REF}	External Voltage Reference for the DACs. Range: DACGND to $DACV_{DD}$.
57	DACGND	Ground for the DAC. Typically connected to AGND.
58	AGND	Analog Ground. Ground reference point for the analog circuitry.
59	AV_{DD}	3.3 V Analog Power.
60	$DACV_{DD}$	3.3 V Power Supply for the DACs. Typically connected to AV_{DD}.
61	ADC0	Single-Ended or Differential Analog Input 0.
62	ADC1	Single-Ended or Differential Analog Input 1.
63	ADC2/CMP0	Single-Ended or Differential Analog Input 2/Comparator Positive Input.
64	ADC3/CMP1	Single-Ended or Differential Analog Input 3/Comparator Negative Input.

ADuC7026/ADuC7027

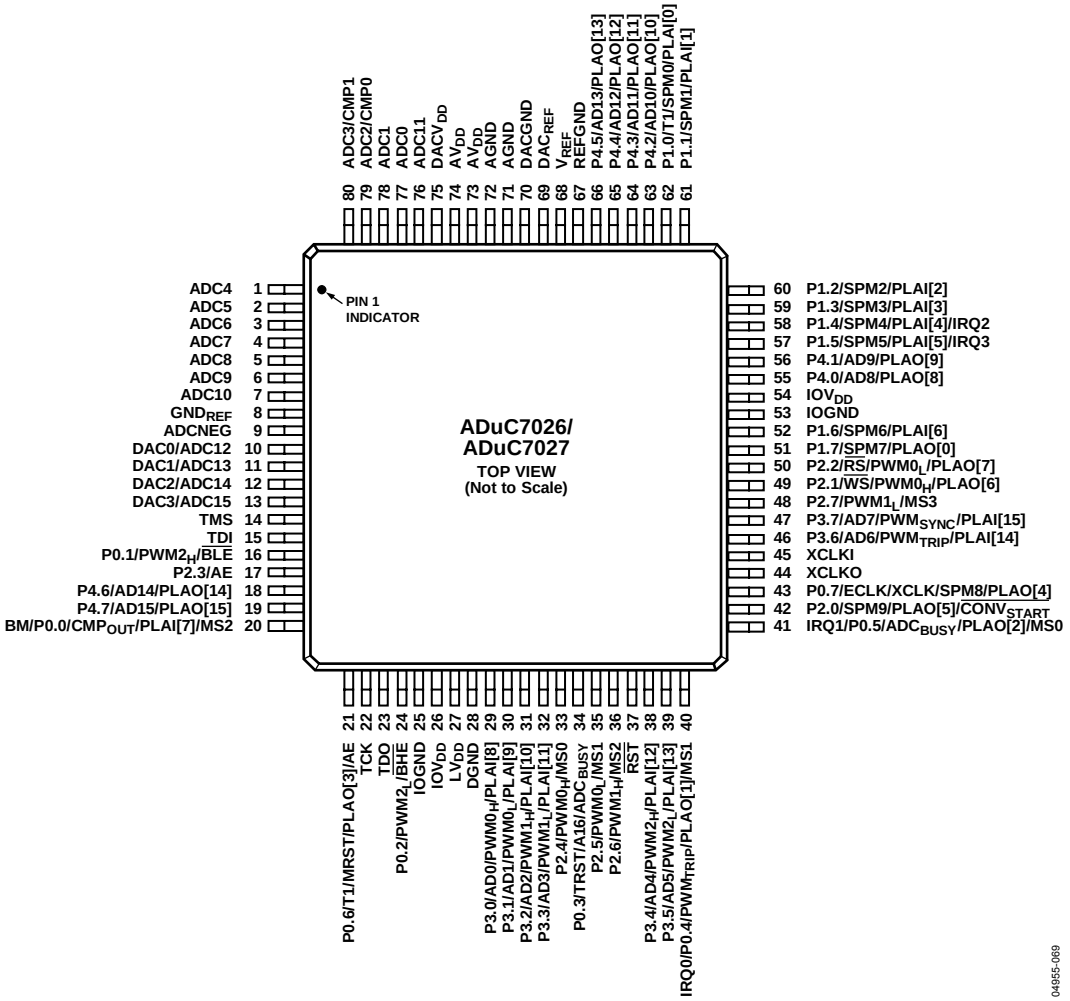


Figure 15. ADuC7026/ADuC7027 80-Lead LQFP Pin Configuration

Table 12. Pin Function Descriptions (ADuC7026/ADuC7027)

Pin No.	Mnemonic	Description
1	ADC4	Single-Ended or Differential Analog Input 4.
2	ADC5	Single-Ended or Differential Analog Input 5.
3	ADC6	Single-Ended or Differential Analog Input 6.
4	ADC7	Single-Ended or Differential Analog Input 7.
5	ADC8	Single-Ended or Differential Analog Input 8.
6	ADC9	Single-Ended or Differential Analog Input 9.
7	ADC10	Single-Ended or Differential Analog Input 10.
8	GND_{REF}	Ground Voltage Reference for the ADC. For optimal performance, the analog power supply should be separated from IOGND and DGND.
9	ADCNEG	Bias Point or Negative Analog Input of the ADC in Pseudo Differential Mode. Must be connected to the ground of the signal to convert. This bias point must be between 0 V and 1 V.
10	DAC0/ADC12	DAC0 Voltage Output/Single-Ended or Differential Analog Input 12. DAC outputs are not present on the ADuC7027.
11	DAC1/ADC13	DAC1 Voltage Output/Single-Ended or Differential Analog Input 13. DAC outputs are not present on the ADuC7027.
12	DAC2/ADC14	DAC2 Voltage Output/Single-Ended or Differential Analog Input 14. DAC outputs are not present on the ADuC7027.
13	DAC3/ADC15	DAC3 Voltage Output/Single-Ended or Differential Analog Input 15. DAC outputs are not present on the ADuC7027.

Pin No.	Mnemonic	Description
14	TMS	JTAG Test Port Input, Test Mode Select. Debug and download access.
15	TDI	JTAG Test Port Input, Test Data In. Debug and download access.
16	P0.1/PWM2$_H$/$\overline{BLE}$	General-Purpose Input and Output Port 0.1/PWM Phase 2 High-Side Output/External Memory Byte Low Enable.
17	P2.3/AE	General-Purpose Input and Output Port 2.3/External Memory Access Enable.
18	P4.6/AD14/PLAO[14]	General-Purpose Input and Output Port 4.6/External Memory Interface/Programmable Logic Array Output Element 14.
19	P4.7/AD15/PLAO[15]	General-Purpose Input and Output Port 4.7/External Memory Interface/Programmable Logic Array Output Element 15.
20	BM/P0.0/CMP$_{OUT}$/PLAI[7]/MS2	Multifunction I/O Pin. Boot Mode. The ADuC7026/ADuC7027 enter UART download mode if BM is low at reset and execute code if BM is pulled high at reset through a 1 kΩ resistor/ General-Purpose Input and Output Port 0.0/Voltage Comparator Output/Programmable Logic Array Input Element 7/External Memory Select 2.
21	P0.6/T1/MRST/PLAO[3]/AE	Multifunction Pin, Driven Low After Reset. General-Purpose Output Port 0.6/Timer1 Input/Power-On Reset Output/Programmable Logic Array Output Element 3.
22	TCK	JTAG Test Port Input, Test Clock. Debug and download access.
23	TDO	JTAG Test Port Output, Test Data Out. Debug and download access.
24	P0.2/ PWM2$_L$/$\overline{BHE}$	General-Purpose Input and Output Port 0.2/PWM Phase 2 Low-Side Output/External Memory Byte High Enable.
25	IOGND	Ground for GPIO. Typically connected to DGND.
26	IOV$_{DD}$	3.3 V Supply for GPIO and Input of the On-Chip Voltage Regulator.
27	LV$_{DD}$	2.6 V Output of the On-Chip Voltage Regulator. This output must be connected to a 0.47 µF capacitor to DGND only.
28	DGND	Ground for Core Logic.
29	P3.0/AD0/PWM0$_H$/PLAI[8]	General-Purpose Input and Output Port 3.0/External Memory Interface/PWM Phase 0 High-Side Output/Programmable Logic Array Input Element 8.
30	P3.1/AD1/PWM0$_L$/PLAI[9]	General-Purpose Input and Output Port 3.1/External Memory Interface/PWM Phase 0 Low-Side Output/Programmable Logic Array Input Element 9.
31	P3.2/AD2/PWM1$_H$/PLAI[10]	General-Purpose Input and Output Port 3.2/External Memory Interface/PWM Phase 1 High-Side Output/Programmable Logic Array Input Element 10.
32	P3.3/AD3/PWM1$_L$/PLAI[11]	General-Purpose Input and Output Port 3.3/External Memory Interface/PWM Phase 1 Low-Side Output/Programmable Logic Array Input Element 11.
33	P2.4/PWM0$_H$/MS0	General-Purpose Input and Output Port 2.4/PWM Phase 0 High-Side Output/External Memory Select 0.
34	P0.3/TRST/A16/ADC$_{BUSY}$	General-Purpose Input and Output Port 0.3/JTAG Test Port Input, Test Reset/ADC$_{BUSY}$ Signal Output.
35	P2.5/PWM0$_L$/MS1	General-Purpose Input and Output Port 2.5/PWM Phase 0 Low-Side Output/External Memory Select 1.
36	P2.6/PWM1$_H$/MS2	General-Purpose Input and Output Port 2.6/PWM Phase 1 High-Side Output/External Memory Select 2.
37	$\overline{RST}$	Reset Input, Active Low.
38	P3.4/AD4/PWM2$_H$/PLAI[12]	General-Purpose Input and Output Port 3.4/External Memory Interface/PWM Phase 2 High-Side Output/Programmable Logic Array Input 12.
39	P3.5/AD5/PWM2$_L$/PLAI[13]	General-Purpose Input and Output Port 3.5/External Memory Interface/PWM Phase 2 Low-Side Output/Programmable Logic Array Input Element 13.
40	IRQ0/P0.4/PWM$_{TRIP}$/PLAO[1]/MS1	Multifunction I/O Pin. External Interrupt Request 0, Active High/General-Purpose Input and Output Port 0.4/PWM Trip External Input/Programmable Logic Array Output Element 1/External Memory Select 1.
41	IRQ1/P0.5/ADC$_{BUSY}$/PLAO[2]/MS0	Multifunction I/O Pin. External Interrupt Request 1, Active High/General-Purpose Input and Output Port 0.5/ADC$_{BUSY}$ Signal Output/Programmable Logic Array Output Element 2/External Memory Select 0.
42	P2.0/SPM9/PLAO[5]/$\overline{CONV_{START}}$	Serial Port Multiplexed. General-Purpose Input and Output Port 2.0/UART/Programmable Logic Array Output Element 5/Start Conversion Input Signal for ADC.

Pin No.	Mnemonic	Description
43	P0.7/ECLK/XCLK/SPM8/ PLAO[4]	Serial Port Multiplexed. General-Purpose Input and Output Port 0.7/Output for External Clock Signal/Input to the Internal Clock Generator Circuits/UART/Programmable Logic Array Output Element 4.
44	XCLKO	Output from the Crystal Oscillator Inverter.
45	XCLKI	Input to the Crystal Oscillator Inverter and Input to the Internal Clock Generator Circuits.
46	P3.6/AD6/PWM_{TRIP}/PLAI[14]	General-Purpose Input and Output Port 3.6/External Memory Interface/PWM Safety Cut Off/Programmable Logic Array Input Element 14.
47	P3.7/AD7/PWM_{SYNC}/PLAI[15]	General-Purpose Input and Output Port 3.7/External Memory Interface/PWM Synchronization/Programmable Logic Array Input Element 15.
48	P2.7/$PWM1_L$/MS3	General-Purpose Input and Output Port 2.7/PWM Phase 1 Low-Side Output/External Memory Select 3.
49	P2.1/$\overline{WS}$/$PWM0_H$/PLAO[6]	General-Purpose Input and Output Port 2.1/External Memory Write Strobe/PWM Phase 0 High-Side Output/Programmable Logic Array Output Element 6.
50	P2.2/$\overline{RS}$/$PWM0_L$/PLAO[7]	General-Purpose Input and Output Port 2.2/External Memory Read Strobe/PWM Phase 0 Low-Side Output/Programmable Logic Array Output Element 7.
51	P1.7/SPM7/PLAO[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.7/UART, SPI/Programmable Logic Array Output Element 0.
52	P1.6/SPM6/PLAI[6]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.6/UART, SPI/Programmable Logic Array Input Element 6.
53	IOGND	Ground for GPIO. Typically connected to DGND.
54	IOV_{DD}	3.3 V Supply for GPIO and Input of the On-Chip Voltage Regulator.
55	P4.0/AD8/PLAO[8]	General-Purpose Input and Output Port 4.0/External Memory Interface/Programmable Logic Array Output Element 8.
56	P4.1/AD9/PLAO[9]	General-Purpose Input and Output Port 4.1/External Memory Interface/Programmable Logic Array Output Element 9.
57	P1.5/SPM5/PLAI[5]/IRQ3	Serial Port Multiplexed. General-Purpose Input and Output Port 1.5/UART, SPI/Programmable Logic Array Input Element 5/External Interrupt Request 3, Active High.
58	P1.4/SPM4/PLAI[4]/IRQ2	Serial Port Multiplexed. General-Purpose Input and Output Port 1.4/UART, SPI/Programmable Logic Array Input Element 4/External Interrupt Request 2, Active High.
59	P1.3/SPM3/PLAI[3]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.3/UART, I2C1/Programmable Logic Array Input Element 3.
60	P1.2/SPM2/PLAI[2]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.2/UART, I2C1/Programmable Logic Array Input Element 2.
61	P1.1/SPM1/PLAI[1]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.1/UART, I2C0/Programmable Logic Array Input Element 1.
62	P1.0/T1/SPM0/PLAI[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.0/Timer1 Input/UART, I^2C0/Programmable Logic Array Input Element 0.
63	P4.2/AD10/PLAO[10]	General-Purpose Input and Output Port 4.2/External Memory Interface/Programmable Logic Array Output Element 10.
64	P4.3/AD11/PLAO[11]	General-Purpose Input and Output Port 4.3/External Memory Interface/Programmable Logic Array Output Element 11.
65	P4.4/AD12/PLAO[12]	General-Purpose Input and Output Port 4.4/External Memory Interface/Programmable Logic Array Output Element 12.
66	P4.5/AD13/PLAO[13]	General-Purpose Input and Output Port 4.5/External Memory Interface/Programmable Logic Array Output Element 13.
67	REFGND	Ground for the Reference. Typically connected to AGND.
68	V_{REF}	2.5 V Internal Voltage Reference. Must be connected to a 0.47 µF capacitor when using the internal reference.
69	DAC_{REF}	External Voltage Reference for the DACs. Range: DACGND to $DACV_{DD}$.
70	DACGND	Ground for the DAC. Typically connected to AGND.
71, 72	AGND	Analog Ground. Ground reference point for the analog circuitry.
73, 74	AV_{DD}	3.3 V Analog Power.

Pin No.	Mnemonic	Description
75	$DACV_{DD}$	3.3 V Power Supply for the DACs. Typically connected to AV_{DD}.
76	ADC11	Single-Ended or Differential Analog Input 11.
77	ADC0	Single-Ended or Differential Analog Input 0.
78	ADC1	Single-Ended or Differential Analog Input 1.
79	ADC2/CMP0	Single-Ended or Differential Analog Input 2/Comparator Positive Input.
80	ADC3/CMP1	Single-Ended or Differential Analog Input 3/Comparator Negative Input.

TYPICAL PERFORMANCE CHARACTERISTICS

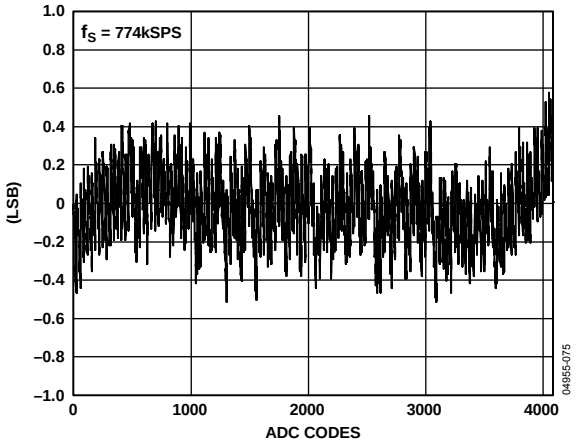


Figure 16. Typical INL Error, f_S = 774 kSPS

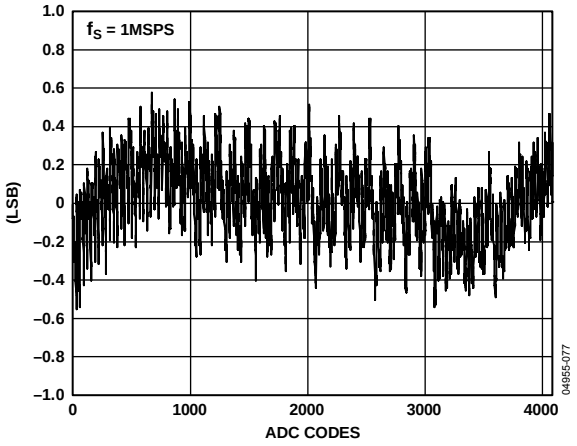


Figure 17. Typical INL Error, f_S = 1 MSPS

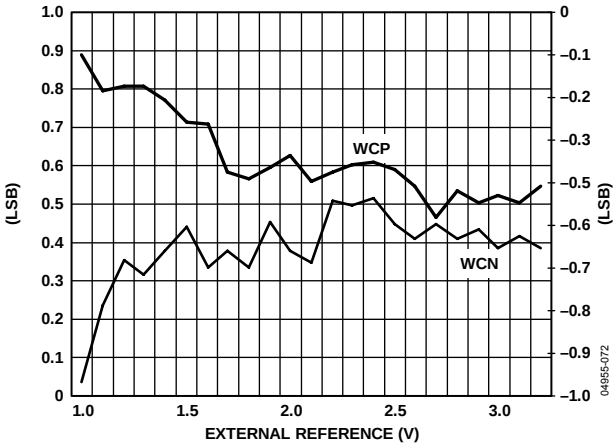


Figure 18. Typical Worst Case INL Error vs. V_{REF}, f_S = 774 kSPS

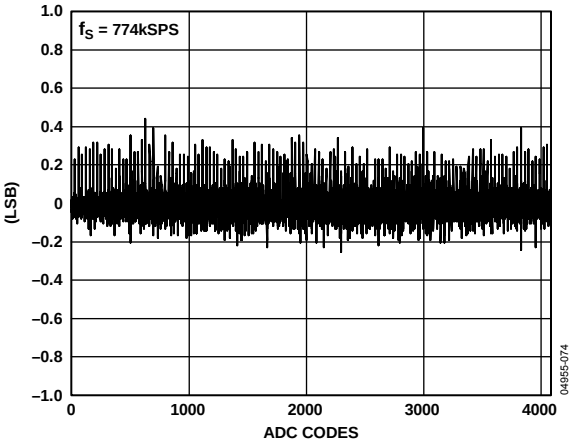


Figure 19. Typical DNL Error, f_S = 774 kSPS

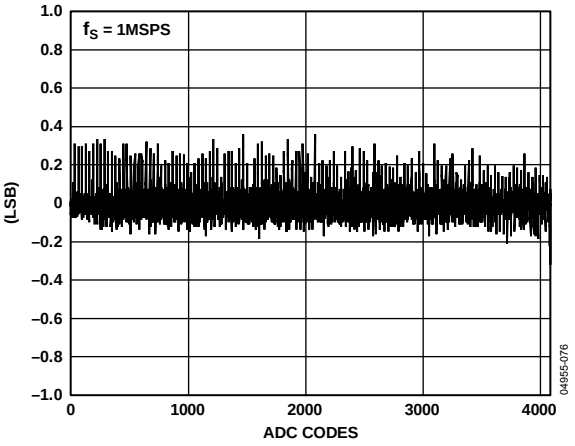


Figure 20. Typical DNL Error, f_S = 1 MSPS

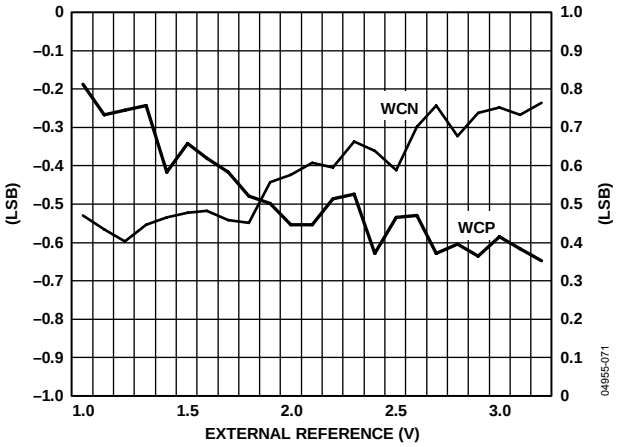


Figure 21. Typical Worst Case DNL Error vs. V_{REF}, f_S = 774 kSPS

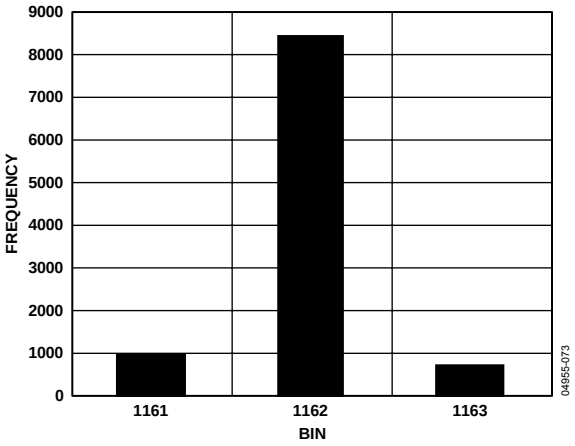


Figure 22. Code Histogram Plot, fs = 774 kSPS, V_{IN} = 0.7 V

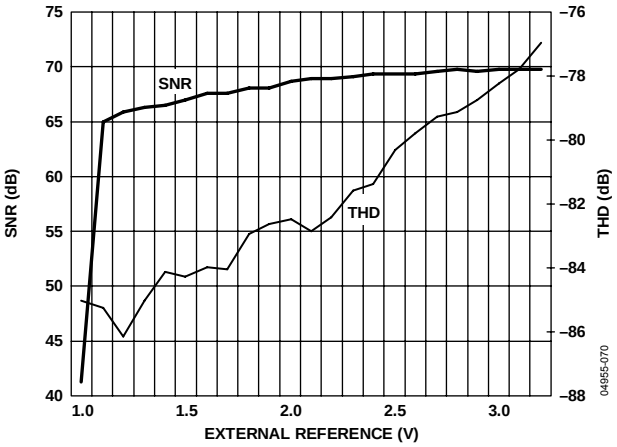


Figure 25. Typical Dynamic Performance vs. V_{REF}

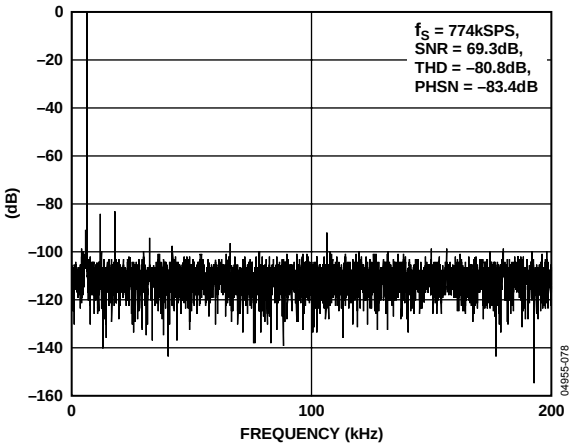


Figure 23. Dynamic Performance, f_S = 774 kSPS

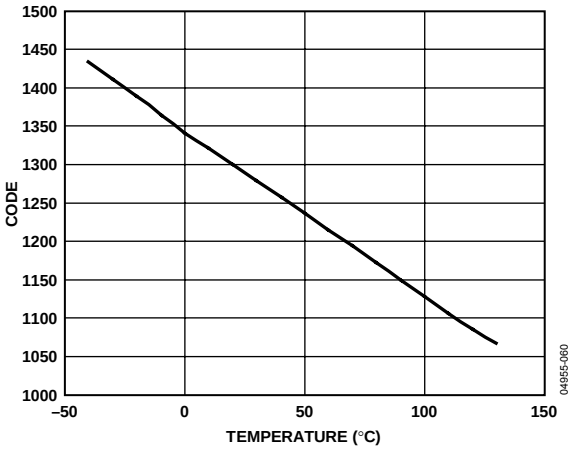


Figure 26. On-Chip Temperature Sensor Voltage Output vs. Temperature

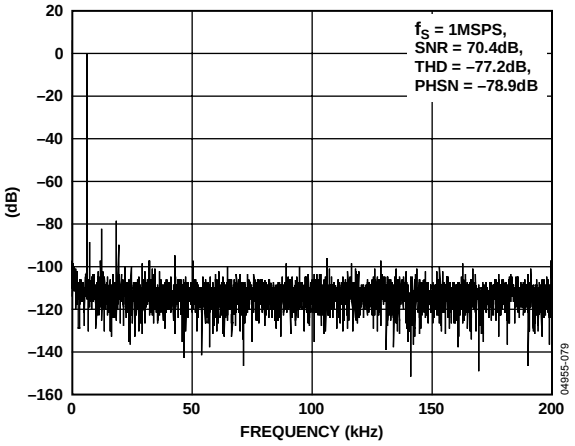


Figure 24. Dynamic Performance, f_S = 1 MSPS

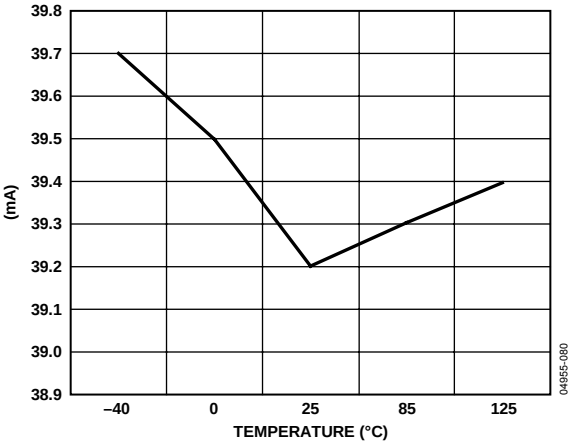


Figure 27. Current Consumption vs. Temperature @ CD = 0

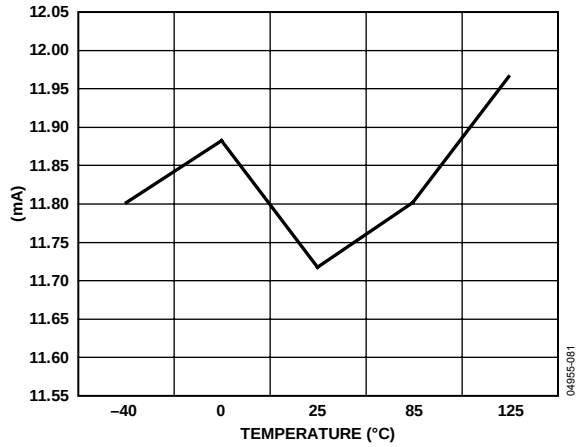


Figure 28. Current Consumption vs. Temperature @ CD = 3

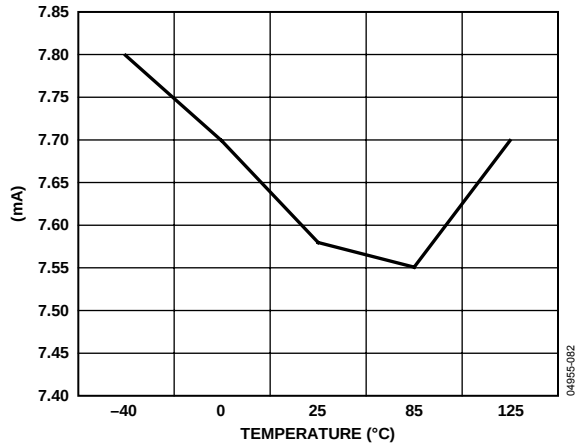


Figure 29. Current Consumption vs. Temperature @ CD = 7

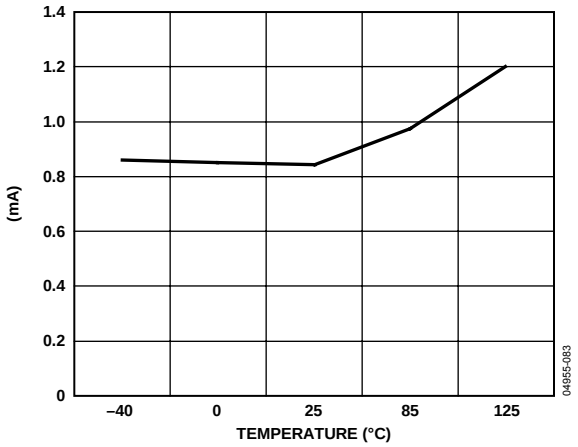


Figure 30. Current Consumption vs. Temperature in Sleep Mode

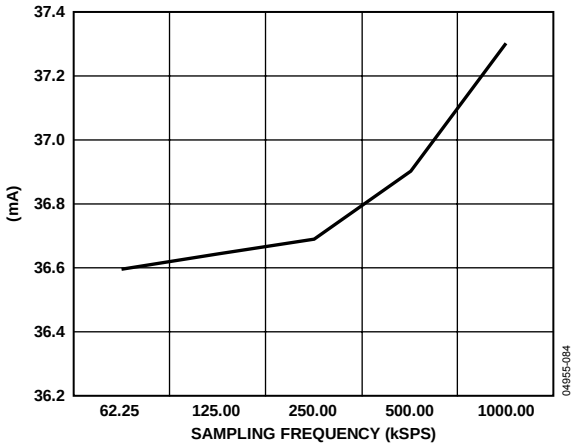


Figure 31. Current Consumption vs. ADC Speed

TERMINOLOGY

ADC SPECIFICATIONS

Integral Nonlinearity
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
The difference between the measured and the ideal 1 LSB change between any two adjacent codes in the ADC.

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

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

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

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

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

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

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

DAC SPECIFICATIONS

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

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

OVERVIEW OF THE ARM7TDMI CORE

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

- 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:

- Thumb code usually uses more instructions for the same job. As a result, ARM code is usually best for maximizing the performance of the 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 64-bit result, and 32-bit by 32-bit multiplication-accumulation (MAC) with 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. Once in a debug state, the processor registers can be inspected as well as the Flash/EE, the SRAM, and the 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 transfer or communication channel 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 six status registers. Each operating mode has dedicated banked registers.

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

When an exception occurs, some of the standard registers are replaced with registers specific to the exception mode. All exception modes have replacement banked registers for the stack pointer (R13) and the link register (R14) as represented in Figure 32. The fast interrupt mode has more registers (R8 to R12) for fast interrupt processing. This means the interrupt processing can begin without the need to save or restore these registers, and thus save critical time in the interrupt handling process.

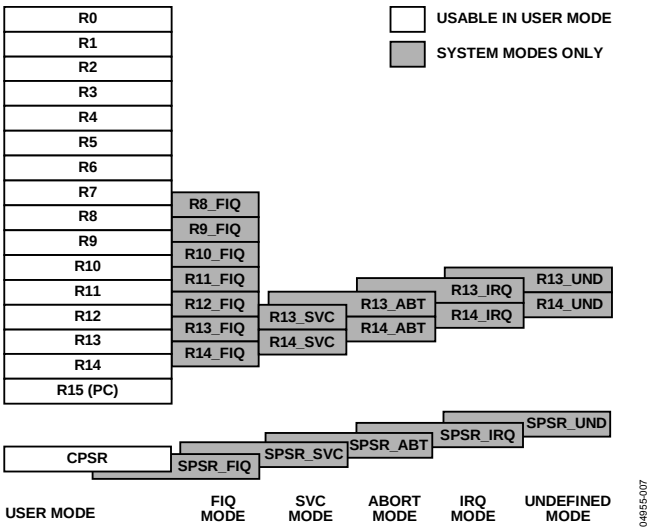


Figure 32. Register Organization

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

- DDI0029G, *ARM7TDMI Technical Reference Manual*
- DDI0100E, *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 could delay entry into the IRQ handling routine for an arbitrary length of time. This time can be reduced to 42 cycles if the LDM command is not used. Some compilers have an option to compile without using this command. Another option is to run the part in thumb mode, where the time is reduced to 22 cycles.

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

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

MEMORY ORGANIZATION

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 incorporate two separate blocks of memory: 8 kB of SRAM and 64 kB of on-chip Flash/EE memory. Sixty-two kilobytes of on-chip Flash/EE memory is 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 33.

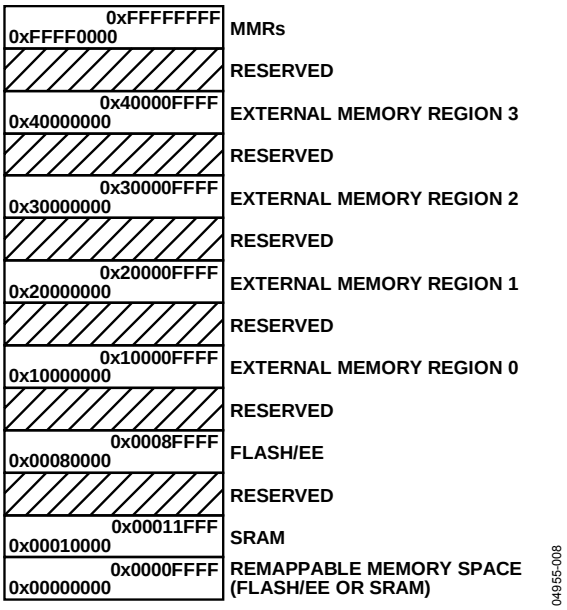


Figure 33. Physical Memory Map

Note that by default, after a reset, the Flash/EE memory is mirrored at address 0×00000000. It is possible to remap the SRAM at address 0×00000000 by clearing Bit 0 of the REMAP MMR. This remap function is described in more detail in the Flash/EE Memory section.

MEMORY ACCESS

The ARM7 core sees memory as a linear array of 2^{32} byte location where the different blocks of memory are mapped as outlined in Figure 33.

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 memory organizations are configured in little endian format, which means that the least significant byte is located in the lowest byte address, and the most significant byte is in the highest byte address.

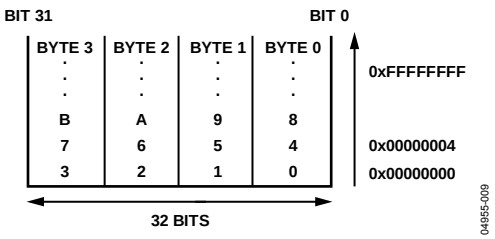


Figure 34. Little Endian Format

FLASH/EE MEMORY

The total 64 kB of Flash/EE memory is organized as 32 k × 16 bits (31 k × 16 bits is user space and 1 k × 16 bits is reserved for the on-chip kernel). The page size of this Flash/EE memory is 512 bytes.

Sixty-two kilobytes of Flash/EE memory are available to the user as code and nonvolatile data memory. There is no distinction between data and program as ARM code shares the same space. The real width of the Flash/EE memory is 16 bits, which means that in ARM mode (32-bit instruction), two accesses to the Flash/EE are necessary for each instruction fetch. It is therefore recommended to use thumb mode 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. More details about Flash/EE access time are outlined later in the Execution Time from SRAM and Flash/EE section of this data sheet.

SRAM

Eight kilobytes of SRAM are available to the user, organized as 2 k × 32 bits, that is, two 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. More details about SRAM access time are outlined later in the Execution Time from SRAM and Flash/EE section of this datasheet.

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 35 are unoccupied or reserved locations, and should not be accessed by user software. Table 13 shows the full MMR memory map.

The access time for reading from or writing to an MMR depends on the advanced microcontroller bus architecture (AMBA) bus used to access the peripheral. The processor has two AMBA busses: advanced high performance bus (AHB) used for system modules, and advanced peripheral bus (APB) used for lower performance peripheral. Access to the AHB is one cycle, and access to the APB is two cycles. All peripherals on the ADuC7019/7020/7021/7022/7024/7025/7026/7027 are on the APB except the Flash/EE memory, the GPIOs, and the PWM.

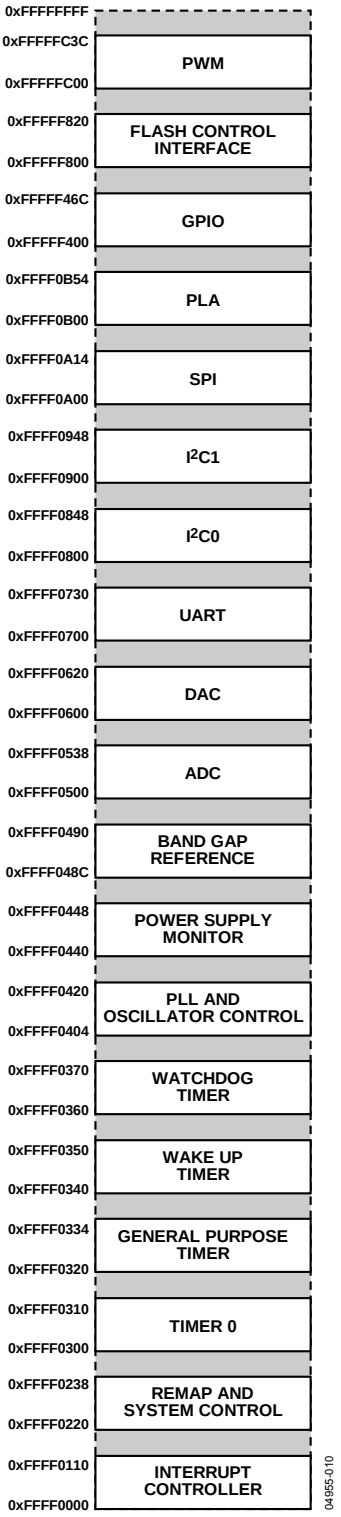


Figure 35. Memory Mapped Registers

Table 13. Complete MMR List

Address	Name	Byte	Access Type	Default Value	Page
IRQ address base = 0xFFFF0000					
0x0000	IRQSTA	4	R	0x00000000	74
0x0004	IRQSIG[1]	4	R	0x00XXX000	74
0x0008	IRQEN	4	R/W	0x00000000	74
0x000C	IRQCLR	4	W	0x00000000	74
0x0010	SWICFG	4	W	0x00000000	75
0x0100	FIQSTA	4	R	0x00000000	74
0x0104	FIQSIG[1]	4	R	0x00XXX000	75
0x0108	FIQEN	4	R/W	0x00000000	75
0x010C	FIQCLR	4	W	0x00000000	75
[1] Depends on the level on the external interrupt pins (P0.4, P0.5, P1.4, and P1.5).					
System control address base = 0xFFFF0200					
0x0220	REMAP[1]	1	R/W	0x00	47
0x0230	RSTSTA	1	R/W	0x01	47
0x0234	RSTCLR	1	W	0x00	47
[1] Depends on model.					
Timer address base = 0xFFFF0300					
0x0300	T0LD	2	R/W	0x0000	76
0x0304	T0VAL	2	R	0xFFFF	76
0x0308	T0CON	2	R/W	0x0000	76
0x030C	T0CLRI	1	W	0xFF	76
0x0320	T1LD	4	R/W	0x00000000	76
0x0324	T1VAL	4	R	0xFFFFFFFF	76
0x0328	T1CON	2	R/W	0x0000	76
0x032C	T1CLRI	1	W	0xFF	77
0x0330	T1CAP	4	R/W	0x00000000	77
0x0340	T2LD	4	R/W	0x00000000	77
0x0344	T2VAL	4	R	0xFFFFFFFF	77
0x0348	T2CON	2	R/W	0x0000	78
0x034C	T2CLRI	1	W	0xFF	78
0x0360	T3LD	2	R/W	0x0000	78
0x0364	T3VAL	2	R	0xFFFF	78
0x0368	T3CON	2	R/W	0x0000	78
0x036C	T3CLRI	1	W	0x00	79
PLL base address = 0xFFFF0400					
0x0404	POWKEY1	2	W	0x0000	52
0x0408	POWCON	2	R/W	0x0003	52
0x040C	POWKEY2	2	W	0x0000	52
0x0410	PLLKEY1	2	W	0x0000	52
0x0414	PLLCON	1	R/W	0x21	52
0x0418	PLLKEY2	2	W	0x0000	52
PSM address base = 0xFFFF0440					
0x0440	PSMCON	2	R/W	0x0008	49
0x0444	CMPCON	2	R/W	0x0000	50
Reference address base = 0xFFFF0480					
0x048C	REFCON	1	R/W	0x00	42

Address	Name	Byte	Access Type	Default Value	Page
ADC address base = 0xFFFF0500					
0x0500	ADCCON	2	R/W	0x0600	39
0x0504	ADCCP	1	R/W	0x00	39
0x0508	ADCCN	1	R/W	0x01	40
0x050C	ADCSTA	1	R	0x00	40
0x0510	ADCDAT	4	R	0x00000000	40
0x0514	ADCRST	1	R/W	0x00	40
0x0530	ADCGN	2	R/W	0x0200	40
0x0534	ADCOF	2	R/W	0x0200	40
DAC address base = 0xFFFF0600					
0x0600	DAC0CON	1	R/W	0x00	48
0x0604	DAC0DAT	4	R/W	0x00000000	48
0x0608	DAC1CON	1	R/W	0x00	48
0x060C	DAC1DAT	4	R/W	0x00000000	48
0x0610	DAC2CON	1	R/W	0x00	48
0x0614	DAC2DAT	4	R/W	0x00000000	48
0x0618	DAC3CON	1	R/W	0x00	48
0x061C	DAC3DAT	4	R/W	0x00000000	48
UART base address = 0xFFFF0700					
0x0700	COMTX	1	R/W	0x00	63
	COMRX	1	R	0x00	63
	COMDIV0	1	R/W	0x00	63
0x0704	COMIEN0	1	R/W	0x00	63
	COMDIV1	1	R/W	0x00	63
0x0708	COMIID0	1	R	0x01	63
0x070C	COMCON0	1	R/W	0x00	63
0x0710	COMCON1	1	R/W	0x00	64
0x0714	COMSTA0	1	R	0x60	64
0x0718	COMSTA1	1	R	0x00	64
0x071C	COMSCR	1	R/W	0x00	64
0x0720	COMIEN1	1	R/W	0x04	65
0x0724	COMIID1	1	R	0x01	65
0x0728	COMADR	1	R/W	0xAA	65
0x072C	COMDIV2	2	R/W	0x0000	64

Address	Name	Byte	Access Type	Default Value	Page
I2C0 base address = 0xFFFF0800					
0x0800	I2C0MSTA	1	R	0x00	68
0x0804	I2C0SSTA	1	R	0x01	68
0x0808	I2C0SRX	1	R	0x00	69
0x080C	I2C0STX	1	W	0x00	69
0x0810	I2C0MRX	1	R	0x00	69
0x0814	I2C0MTX	1	W	0x00	69
0x0818	I2C0CNT	1	R/W	0x00	69
0x081C	I2C0ADR	1	R/W	0x00	69
0x0824	I2C0BYTE	1	R/W	0x00	69
0x0828	I2C0ALT	1	R/W	0x00	69
0x082C	I2C0CFG	1	R/W	0x00	70
0x0830	I2C0DIV	2	R/W	0x1F1F	70
0x0838	I2C0ID0	1	R/W	0x00	70
0x083C	I2C0ID1	1	R/W	0x00	70
0x0840	I2C0ID2	1	R/W	0x00	70
0x0844	I2C0ID3	1	R/W	0x00	70
0x0848	I2C0CCNT	1	R/W	0x01	70
0x084C	I2C0FSTA	2	R	0x0000	71
I2C1 base address = 0xFFFF0900					
0x0900	I2C1MSTA	1	R	0x00	68
0x0904	I2C1SSTA	1	R	0x01	68
0x0908	I2C1SRX	1	R	0x00	69
0x090C	I2C1STX	1	W	0x00	69
0x0910	I2C1MRX	1	R	0x00	69
0x0914	I2C1MTX	1	W	0x00	69
0x0918	I2C1CNT	1	R/W	0x00	69
0x091C	I2C1ADR	1	R/W	0x00	69
0x0924	I2C1BYTE	1	R/W	0x00	69
0x0928	I2C1ALT	1	R/W	0x00	69
0x092C	I2C1CFG	1	R/W	0x00	69
0x0930	I2C1DIV	2	R/W	0x1F1F	70
0x0938	I2C1ID0	1	R/W	0x00	70
0x093C	I2C1ID1	1	R/W	0x00	70
0x0940	I2C1ID2	1	R/W	0x00	70
0x0944	I2C1ID3	1	R/W	0x00	70
0x0948	I2C1CCNT	1	R/W	0x01	70
0x094C	I2C1FSTA	2	R	0x0000	70
SPI base address = 0xFFFF0A00					
0x0A00	SPISTA	1	R	0x00	66
0x0A04	SPIRX	1	R	0x00	66
0x0A08	SPITX	1	W	0x00	66
0x0A0C	SPIDIV	1	R/W	0x1B	66
0x0A10	SPICON	2	R/W	0x0000	66

Address	Name	Byte	Access Type	Default Value	Page
PLA base address = 0xFFFF0B00					
0x0B00	PLAELM0	2	R/W	0x0000	71
0x0B04	PLAELM1	2	R/W	0x0000	71
0x0B08	PLAELM2	2	R/W	0x0000	71
0x0B0C	PLAELM3	2	R/W	0x0000	71
0x0B10	PLAELM4	2	R/W	0x0000	71
0x0B14	PLAELM5	2	R/W	0x0000	71
0x0B18	PLAELM6	2	R/W	0x0000	71
0x0B1C	PLAELM7	2	R/W	0x0000	71
0x0B20	PLAELM8	2	R/W	0x0000	71
0x0B24	PLAELM9	2	R/W	0x0000	71
0x0B28	PLAELM10	2	R/W	0x0000	71
0x0B2C	PLAELM11	2	R/W	0x0000	71
0x0B30	PLAELM12	2	R/W	0x0000	71
0x0B34	PLAELM13	2	R/W	0x0000	71
0x0B38	PLAELM14	2	R/W	0x0000	71
0x0B3C	PLAELM15	2	R/W	0x0000	71
0x0B40	PLACLK	1	R/W	0x00	72
0x0B44	PLAIRQ	4	R/W	0x00000000	73
0x0B48	PLAADC	4	R/W	0x00000000	73
0x0B4C	PLADIN	4	R/W	0x00000000	73
0x0B50	PLADOUT	4	R	0x00000000	73
0x0B54	PLALCK	1	W	0x00	73
External memory base address = 0xFFFFF000					
0xF000	XMCFG	1	R/W	0x00	80
0xF010	XM0CON	1	R/W	0x00	80
0xF014	XM1CON	1	R/W	0x00	80
0xF018	XM2CON	1	R/W	0x00	80
0xF01C	XM3CON	1	R/W	0x00	80
0xF020	XM0PAR	2	R/W	0x70FF	80
0xF024	XM1PAR	2	R/W	0x70FF	80
0xF028	XM2PAR	2	R/W	0x70FF	80
0xF02C	XM3PAR	2	R/W	0x70FF	80

Address	Name	Byte	Access Type	Default Value	Page
GPIO base address = 0xFFFFF400					
0xF400	GP0CON	4	R/W	0x00000000	60
0xF404	GP1CON	4	R/W	0x00000000	60
0xF408	GP2CON	4	R/W	0x00000000	60
0xF40C	GP3CON	4	R/W	0x00000000	60
0xF410	GP4CON	4	R/W	0x00000000	60
0xF420	GP0DAT	4	R/W	0x000000XX	61
0xF424	GP0SET	4	W	0x000000XX	61
0xF428	GP0CLR	4	W	0x000000XX	61
0xF42C	GP0PAR	4	W	0x20000000	61
0xF430	GP1DAT	4	R/W	0x000000XX	61
0xF434	GP1SET	4	W	0x000000XX	61
0xF438	GP1CLR	4	W	0x000000XX	61
0xF43C	GP1PAR	4	W	0x00000000	61
0xF440	GP2DAT	4	R/W	0x000000XX	61
0xF444	GP2SET	4	W	0x000000XX	61
0xF448	GP2CLR	4	W	0x000000XX	61
0xF450	GP3DAT	4	R/W	0x000000XX	61
0xF454	GP3SET	4	W	0x000000XX	61
0xF458	GP3CLR	4	W	0x000000XX	61
0xF45C	GP3PAR	4	W	0x00222222	61
0xF460	GP4DAT	4	R/W	0x000000XX	61
0xF464	GP4SET	4	W	0x000000XX	61
0xF468	GP4CLR	4	W	0x000000XX	61
Flash/EE base address = 0xFFFFF800					
0xF800	FEESTA	1	R	0x20	45
0xF804	FEEMOD	2	R/W	0x0000	45
0xF808	FEECON	1	R/W	0x07	45
0xF80C	FEEDAT	2	R/W	0xXXXX	45
0xF810	FEEADR	2	R/W	0x0000	45
0xF818	FEESIGN	3	R	0xFFFFFF	45
0xF81C	FEEPRO	4	R/W	0x00000000	45
0xF820	FEEHIDE	4	R/W	0xFFFFFFFF	45
PWM base address = 0xFFFFFC00					
0xFC00	PWMCON	2	R/W	0x0000	59
0xFC04	PWMSTA	2	R/W	0x0000	59
0xFC08	PWMDAT0	2	R/W	0x0000	60
0xFC0C	PWMDAT1	2	R/W	0x0000	60
0xFC10	PWMCFG	2	R/W	0x0000	59
0xFC14	PWMCH0	2	R/W	0x0000	60
0xFC18	PWMCH1	2	R/W	0x0000	60
0xFC1C	PWMCH2	2	R/W	0x0000	60
0xFC20	PWMEN	2	R/W	0x0000	59
0xFC24	PWMDAT2	2	R/W	0x0000	60

ADC CIRCUIT OVERVIEW

The analog-to-digital converter (ADC) incorporates a fast, multichannel, 12-bit ADC. It can operate from 2.7 V to 3.6 V supplies and is capable of providing a throughput of up to 1 MSPS when the clock source is 41.78 MHz. This block provides the user with a multichannel multiplexer, differential track-and-hold, on-chip reference, and ADC.

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

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

The converter accepts an analog input range of 0 to V_{REF} when operating in single-ended mode or pseudo differential mode. In fully differential mode, the input signal must be balanced around a common-mode voltage V_{CM}, in the range 0 V to AV_{DD}, and with a maximum amplitude of 2 V_{REF} (see Figure 36).

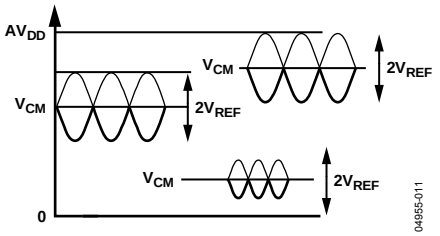


Figure 36. Examples of Balanced Signals in Fully Differential Mode

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 later in the Band Gap Reference section.

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

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

TRANSFER FUNCTION

Pseudo Differential and Single-Ended Modes

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

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

The ideal code transitions occur midway between successive integer LSB values (that is, 1/2 LSB, 3/2 LSB, 5/2 LSB, … , FS – 3/2 LSB). The ideal input/output transfer characteristic is shown in Figure 37.

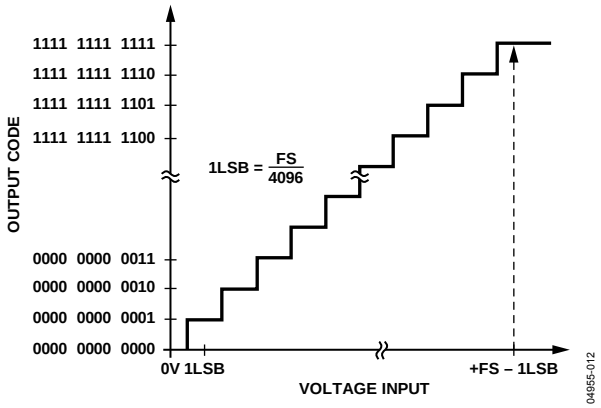


Figure 37. ADC Transfer Function in Pseudo Differential Mode or Single-Ended Mode

Fully Differential Mode

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

The output coding is twos complement in fully differential mode with 1 LSB = 2 $V_{REF}/4096$ or 2 × 2.5 V/4096 = 1.22 mV when V_{REF} = 2.5 V. The designed code transitions occur midway between successive integer LSB values (that is, 1/2 LSB, 3/2 LSB, 5/2 LSB, … , FS – 3/2 LSB). The ideal input/output transfer characteristic is shown in Figure 38.

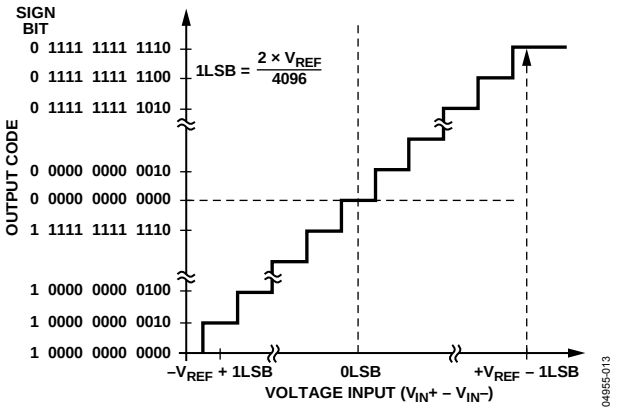


Figure 38. ADC Transfer Function in Differential Mode

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 4 bits are the sign bits. The 12-bit result is placed from Bit 16 to Bit 27 as shown in Figure 39. Again, it should be noted that in fully differential mode, the result is represented in twos complement format, and in pseudo differential and single-ended modes, the result is represented in straight binary format.

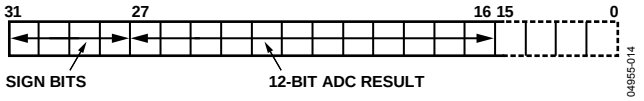


Figure 39. ADC Result Format

The same format is used in DAC×DAT, simplifying the software.

Current Consumption

The ADC in standby mode, that is, powered up but not converting, typically consumes 640 μA. The internal reference adds 140 μA. During conversion, the extra current is 0.3 μA multiplied by the sampling frequency (in kHz). Figure 31 shows the current consumption versus the sampling frequency of the ADC.

Timing

Figure 40 gives details of the ADC timing. Users have control on the ADC clock speed and on 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 extra clocks (such as bit trial or write) is set to 19, which gives a sampling rate of 774 kSPS. For conversion on temperature sensor, the ADC acquisition time is automatically set to 16 clocks and the ADC clock divider is set to 32.

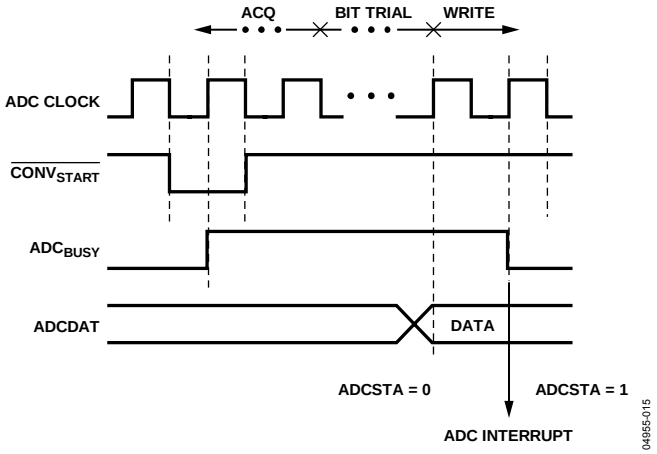


Figure 40. ADC Timing

ADuC7019

The ADuC7019 is identical to the ADuC7020 except for one buffered ADC channel, ADC3, and it has only three DACs. The output buffer of the fourth DAC is internally connected to the ADC3 channel as shown in Figure 41.

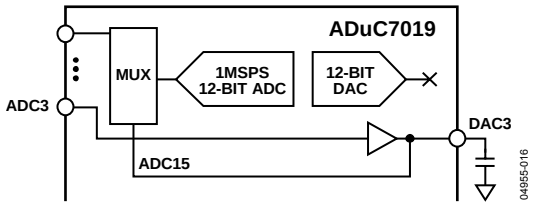


Figure 41. ADC3 Buffered Input

Note that the DAC3 output pin must be connected to a 10 nF capacitor to AGND. This channel should be used to measure dc voltages only. ADC calibration might be necessary on this channel.

MMRS INTERFACE

The ADC is controlled and configured via the eight MMRs described in this section.

ADCCON Register

Name	Address	Default Value	Access
ADCCON	0xFFFF0500	0x0600	R/W

ADCCON is an ADC control register that allows the programmer to enable the ADC peripheral, select the mode of operation of the ADC (either in single-ended mode, pseudo differential mode, or fully differential mode), and select the conversion type. This MMR is described in Table 14.

Table 14. ADCCON MMR Bit Designations

Bit	Value	Description
15:13		Reserved.
12:10		ADC clock speed.
	000	fADC/1. This divider is provided to obtain 1 MSPS ADC with an external clock <41.78 MHz.
	001	fADC/2 (default value).
	010	fADC/4.
	011	fADC/8.
	100	fADC/16.
	101	fADC/32.
9:8		ADC acquisition time.
	00	2 clocks.
	01	4 clocks.
	10	8 clocks (default value).
	11	16 clocks.
7		Enable start conversion.
		Set by the user to start any type of conversion command. *Cleared* by the user to disable a start conversion (clearing this bit does not stop the ADC when continuously converting).
6		Enable ADC_{BUSY}.
		Set by the user to enable the ADC_{BUSY} pin. *Cleared* by the user to disable the ADC_{BUSY} pin.
5		ADC power control.
		Set by the user to place the ADC in normal mode (the ADC must be powered up for at least 5 µs before it converts correctly). *Cleared* by the user to place the ADC in power-down mode.
4:3		Conversion mode.
	00	Single-ended mode.
	01	Differential mode.
	10	Pseudo differential mode.
	11	Reserved.
2:0		Conversion type.
	000	Enable $\overline{CONV}_{START}$ pin as a conversion input.
	001	Enable Timer1 as a conversion input.
	010	Enable Timer0 as a conversion input.
	011	Single software conversion; sets to 000 after conversion (Bit 7 of ADCCON MMR should be cleared after starting a single software conversion to avoid further conversions triggered by the $\overline{CONV}_{START}$ pin).
	100	Continuous software conversion.
	101	PLA conversion.
	Other	Reserved.

ADCCP Register

Name	Address	Default Value	Access
ADCCP	0xFFFF0504	0x00	R/W

ADCCP is an ADC positive channel selection register. This MMR is described in Table 15.

Table 15. ADCCP[1] MMR Bit Designation

Bit	Value	Description
7:5		Reserved
4:0		Positive channel selection bits
	00000	ADC0
	00001	ADC1
	00010	ADC2
	00011	ADC3
	00100	ADC4
	00101	ADC5
	00110	ADC6
	00111	ADC7
	01000	ADC8
	01001	ADC9
	01010	ADC10
	01011	ADC11
	01100	DAC0/ADC12
	01101	DAC1/ADC13
	01110	DAC2/ADC14
	01111	DAC3/ADC15
	10000	Temperature sensor
	10001	AGND (self-diagnostic feature)
	10010	Internal reference (self-diagnostic feature)
	10011	$AV_{DD}/2$
	Others	Reserved

[1] ADC and DAC channel availability depends on part model. See the Ordering Guide for details.

ADCCN Register

Name	Address	Default Value	Access
ADCCN	0xFFFF0508	0x01	R/W

ADCCN is an ADC negative channel selection register. This MMR is described in Table 16.

Table 16. ADCCN MMR Bit Designation

Bit	Value	Description
7:5		Reserved
4:0		Negative channel selection bits
	00000	ADC0
	00001	ADC1
	00010	ADC2
	00011	ADC3
	00100	ADC4
	00101	ADC5
	00110	ADC6
	00111	ADC7
	01000	ADC8
	01001	ADC9
	01010	ADC10
	01011	ADC11
	01100	DAC0/ADC12
	01101	DAC1/ADC13
	01110	DAC2/ADC14
	01111	DAC3/ADC15
	10000	Internal reference (self-diagnostic feature)
	Others	Reserved

ADCSTA Register

Name	Address	Default Value	Access
ADCSTA	0xFFFF050C	0x00	R

ADCSTA is an ADC status register that indicates when an ADC conversion result is ready. The ADCSTA register contains only one bit, ADCReady (Bit 0), representing the status of the ADC. This bit is set at the end of an ADC conversion, generating an ADC interrupt. It is cleared automatically by reading the ADCDAT MMR. When the ADC is performing a conversion, the status of the ADC can be read externally via the ADC_{BUSY} pin. This pin is high during a conversion. When the conversion is finished, ADC_{BUSY} goes back low. This information can be available on P0.5 (see the General-Purpose Input/Output section) if enabled in the ADCCON register.

ADCDAT Register

Name	Address	Default Value	Access
ADCDAT	0xFFFF0510	0x00000000	R

ADCDAT is an ADC data result register. Hold the 12-bit ADC result as shown in Figure 39.

ADCRST Register

Name	Address	Default Value	Access
ADCRST	0xFFFF0514	0x00	R/W

ADCRST resets the digital interface of the ADC. Writing any value to this register resets all the ADC registers to their default value.

ADCGN Register

Name	Address	Default Value	Access
ADCGN	0xFFFF0530	0x0200	R/W

ADCGN is a 10-bit gain calibration register.

ADCOF Register

Name	Address	Default Value	Access
ADCOF	0xFFFF0534	0x0200	R/W

ADCOF is a 10-bit offset calibration register.

CONVERTER OPERATION

The ADC incorporates a successive approximation (SAR) architecture involving a charge-sampled input stage. This architecture can operate in three different modes: differential, pseudo differential, and single-ended.

Differential Mode

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 each contain a successive approximation ADC based on two capacitive DACs. Figure 42 and Figure 43 show simplified schematics of the ADC in acquisition and conversion phase, respectively. The ADC is comprised of control logic, a SAR, and two capacitive DACs. In Figure 42 (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.

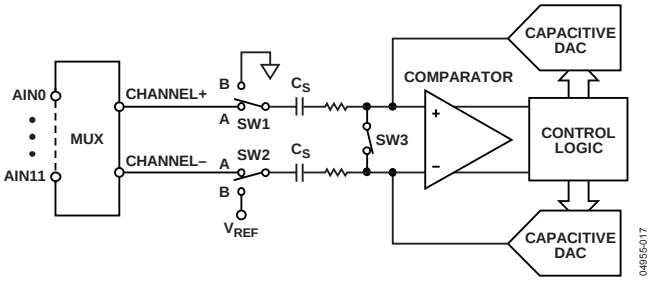


Figure 42. ADC Acquisition Phase

When the ADC starts a conversion, as shown in Figure 43, SW3 opens, and then SW1 and SW2 move to Position B. This causes the comparator to become unbalanced. Both inputs are disconnected once the conversion begins. The control logic and the charge redistribution DACs are used to add and subtract fixed amounts of charge from the sampling capacitor arrays to bring the comparator back into a balanced condition. When the comparator is rebalanced, the conversion is complete. The control logic generates the ADC's output code. The output impedances of the sources driving the V_{IN+} and V_{IN-} pins must be matched; otherwise, the two inputs have different settling times, resulting in errors.

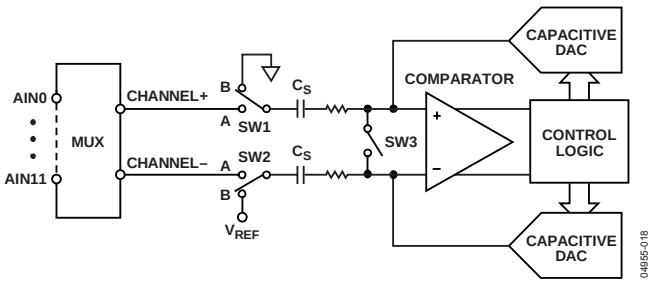


Figure 43. ADC Conversion Phase

Pseudo Differential Mode

In pseudo differential mode, Channel– is linked to the V_{IN-} pin of the ADuC7019/7020/7021/7022/7024/7025/7026/7027. SW2 switches between A (Channel–) and B (V_{REF}). V_{IN-} pin 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 $V_{REF} + V_{IN-}$ do not exceed AV_{DD}.

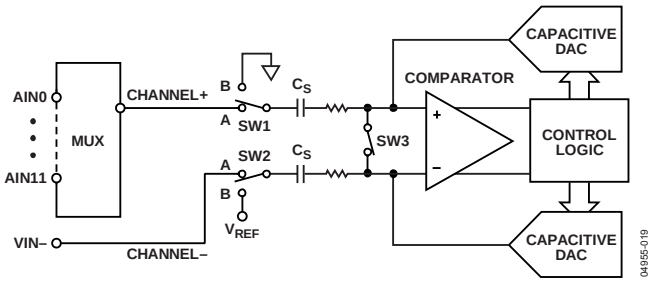


Figure 44. ADC in Pseudo Differential Mode

Single-Ended Mode

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

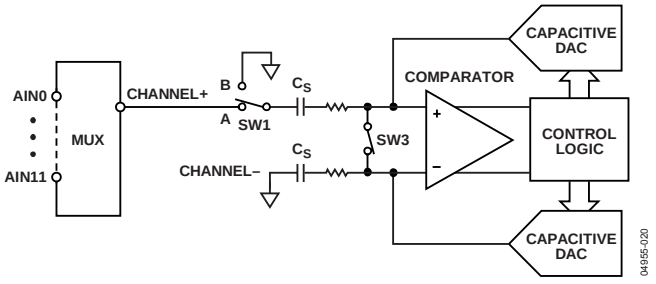


Figure 45. ADC in Single-Ended Mode

Analog Input Structure

Figure 46 shows the equivalent circuit of the analog input structure of the ADC. The four diodes provide ESD protection for the analog inputs. Care must be taken to ensure that the analog input signals never exceed the supply rails by more than 300 mV; this would cause these diodes to become forward biased and start conducting into the substrate. These diodes can conduct up to 10 mA without causing irreversible damage to the part.

The capacitors, C1, in Figure 46 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 capacitors, C2, are the ADC's sampling capacitors and typically have a capacitance of 16 pF.

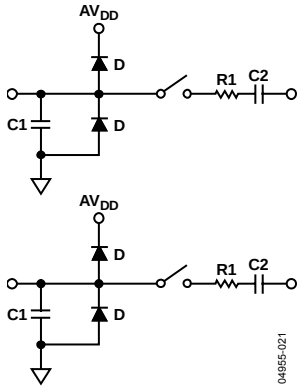


Figure 46. Equivalent Analog Input Circuit Conversion Phase: Switches Open; Track Phase: Switches Closed

For ac applications, removing high frequency components from the analog input signal is recommended by using an RC low-pass filter on the relevant analog input pins. In applications where harmonic distortion and signal-to-noise ratio are critical, the analog input should be driven from a low impedance source. Large source impedances significantly affect the ac performance of the ADC. This can necessitate the use of an input buffer amplifier. The choice of the op amp is a function of the particular application. Figure 47 and Figure 48 give an example of ADC front end.

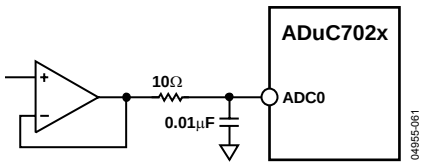


Figure 47. Buffering Single-Ended/Pseudo Differential Input

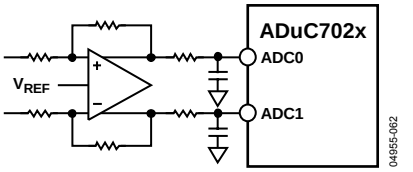


Figure 48. Buffering Differential Inputs

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

DRIVING THE ANALOG INPUTS

Internal or external reference can be used for the ADC. In differential mode of operation, there are restrictions on common-mode input signal (V_{CM}), which is dependent on the reference value and supply voltage used to ensure that the signal remains within the supply rails. Table 17 gives some calculated V_{CM} min and V_{CM} max for some conditions.

Table 17. 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

CALIBRATION

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

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

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

TEMPERATURE SENSOR

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 provide voltage output from on-chip band gap references 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, measuring die temperature to an accuracy of ±3°C.

BAND GAP REFERENCE

Each ADuC7019/7020/7021/7022/7024/7025/7026/7027 provides an on-chip band gap reference of 2.5 V, which can be used for the ADC and DAC. This internal reference also appears on the V_{REF} pin. When using the internal reference, a 0.47 μF capacitor must be connected from the external V_{REF} pin to AGND to ensure stability and fast response during ADC conversions. This reference can also be connected to an external pin (V_{REF}) and used as a reference for other circuits in the system. An external buffer is required because of the low drive capability of the V_{REF} output. A programmable option also allows an external reference input on the V_{REF} pin.

REFCON Register

Name	Address	Default Value	Access
REFCON	0xFFFF048C	0x00	R/W

The band gap reference interface consists of an 8-bit MMR REFCON described in Table 18.

Table 18. REFCON MMR Bit Designations

Bit	Description
7:2	Reserved.
1	Internal Reference Power-Down Enable. *Set* by user to place the internal reference in power-down mode and use as an external reference. *Cleared* by user to place the internal reference in normal mode and use it for ADC conversions.
0	Internal Reference Output Enable. *Set* by user to connect the internal 2.5 V reference to the V_{REF} pin. The reference can be used for external component but needs to be buffered. *Cleared* by user to disconnect the reference from the V_{REF} pin.

NONVOLATILE FLASH/EE MEMORY

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 incorporate Flash/EE memory technology on-chip to provide the user with nonvolatile, in-circuit reprogrammable memory space.

Like EEPROM, flash memory can be programmed in-system at a byte level, although it must first be erased. The erase 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 nonvolatility, in-circuit programmability, high density, and low cost. Incorporated in the ADuC7019/7020/7021/7022/7024/7025/7026/7027, 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.

Each ADuC7019/7020/7021/7022/7024/7025/7026/7027 contains a 64 kB array of Flash/EE memory. The lower 62 kB is available to the user and the upper 2 kB contain permanently embedded firmware, allowing in-circuit serial download. These 2 kB of embedded firmware also contain a power-on configuration routine that downloads factory calibrated coefficients to the various calibrated peripherals (such as ADC, temperature sensor, and band gap references). This 2 kB embedded firmware is hidden from user code.

Flash/EE Memory Reliability

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

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

1. Initial page erase sequence.

2. Read/verify sequence 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 Table 1, the parts' Flash/EE memory endurance qualification is carried out in accordance with JEDEC Retention Lifetime Specification A117 over the industrial temperature range of –40°C to +25°C and +25°C to +125°C. The results allow the specification of a minimum endurance figure over supply temperature of 10,000 cycles.

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. Again, the parts is qualified in accordance with the formal JEDEC Retention Lifetime Specification (A117) at a specific junction temperature (T_J = 85°C). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit, described previously, before data retention is characterized. This means that the Flash/EE memory is guaranteed to retain its data for its fully specified retention lifetime every time the Flash/EE memory is reprogrammed. Also note that retention lifetime, based on an activation energy of 0.6 eV, derates with T_J as shown in Figure 49.

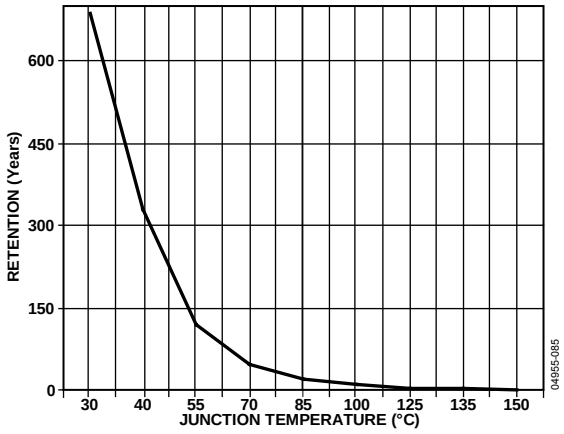


Figure 49. Flash/EE Memory Data Retention

PROGRAMMING

The 62 kB of Flash/EE memory can be programmed in-circuit, using the serial download mode or the JTAG mode provided.

Serial Downloading (In-Circuit Programming)

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 facilitate code download via the standard UART serial port or via the I^2C port. The parts enter serial download mode after a reset or power cycle if the BM pin is pulled low through an external 1 kΩ resistor. Once in serial download mode, the user can download code to the full 62 kB of Flash/EE memory while the device is in-circuit in its target application hardware. An executable PC serial download is provided as part of the development system for serial downloading via the UART. An application note is available at www.analog.com/microconverter describing the protocol for serial downloading via the UART and I^2C.

JTAG Access

The JTAG protocol uses the on-chip JTAG interface to facilitate code download and debug. An application note is available at www.analog.com/microconverter describing the protocol via JTAG.

It is possible to write to a single Flash/EE location address twice. If a single address is written to more than twice, then the data within the Flash/EE memory can be corrupted. That is, it is possible to walk zeros only byte wise.

SECURITY

The 62 kB of Flash/EE memory available to the user can be read and write protected.

Bit 31 of the FEEPRO/FEEHIDE MMR (see Table 22) protects the 62 kB from being read through JTAG and also in parallel programming mode. The other 31 bits of this register protect writing to the flash memory. Each bit protects four pages, that is, 2 kB. Write protection is activated for all types of access.

Three Levels of Protection

- Protection can be set and removed by writing directly into FEEHIDE MMR. This protection does not remain after reset.
- Protection can be set by writing into FEEPRO MMR. It only takes effect after a save protection command (0×0C) and a reset. The FEEPRO MMR is protected by a key to avoid direct access. The key is saved once and must be entered again to modify FEEPRO. A mass erase sets the key back to 0×FFFF but also erases all the user code.
- Flash can be permanently protected by using the FEEPRO MMR and a particular value of key: 0×DEADDEAD. Entering the key again to modify the FEEPRO register is not allowed.

Sequence to Write the Key

1. Write the bit in FEEPRO corresponding to the page to be protected.
2. Enable key protection by setting Bit 6 of FEEMOD (Bit 5 must equal 0).
3. Write a 32-bit key in FEEADR, FEEDAT.
4. Run the write key command 0×0C in FEECON; wait for the read to be successful by monitoring FEESTA.
5. Reset the part.

To remove or modify the protection, the same sequence is used with a modified value of FEEPRO. If the key chosen is the value 0×DEAD, 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 illustrated in the following example (this protects writing Pages 4 to 7 of the Flash):

```
FEEPRO=0xFFFFFFFD;     //Protect pages 4 to 7
FEEMOD=0x48;           //Write key enable
FEEADR=0x1234;         //16 bit key value
FEEDAT=0x5678;         //16 bit key value
FEECON= 0x0C;          // Write key command
```

The same sequence should be followed to protect the part permanently with FEEADR = 0×DEAD and FEEDAT = 0×DEAD.

FLASH/EE CONTROL INTERFACE

Serial, parallel, and JTAG programming use the Flash/EE control interface, which includes eight MMRs outlined in this section.

FEESTA Register

Name	Address	Default Value	Access
FEESTA	0xFFFFF800	0x20	R

FEESTA is a read-only register that reflects the status of the flash control interface as described in Table 19.

Table 19. FEESTA MMR Bit Designations

Bit	Description
15:6	Reserved.
5	Burst Command Enable. *Set* when the command is a burst command: 0x07, 0x08, or 0x09. *Cleared* when another command.
4	Reserved.
3	Flash Interrupt Status Bit. *Set* automatically when an interrupt occurs, that is, when a command is complete and the Flash/EE interrupt enable bit in the FEEMOD register is set. *Cleared* when reading FEESTA register.
2	Flash/EE Controller Busy. *Set* automatically when the controller is busy. *Cleared* automatically when the controller is not busy.
1	Command Fail. *Set* automatically when a command completes unsuccessfully. *Cleared* automatically when reading FEESTA register.
0	Command Pass. *Set* by MicroConverter when a command completes successfully. *Cleared* automatically when reading FEESTA register.

FEEMOD Register

Name	Address	Default Value	Access
FEEMOD	0xFFFFF804	0x0000	R/W

FEEMOD sets the operating mode of the flash control interface. Table 20 shows FEEMOD MMR bit designations.

Table 20. FEEMOD MMR Bit Designations

Bit	Description
15:9	Reserved.
8	Reserved. This bit should always be set to 0.
7:5	Reserved. These bits should always be set to 0 except when writing keys. See the Sequence to Write the Key section.
4	Flash/EE Interrupt Enable. *Set* by user to enable the Flash/EE interrupt. The interrupt occurs when a command is complete. *Cleared* by user to disable the Flash/EE interrupt.
3	Erase/Write Command Protection. *Set* by user to enable the erase and write commands. *Cleared* to protect the Flash against erase/write command.
2:0	Reserved. These bits should always be set to 0.

FEECON Register

Name	Address	Default Value	Access
FEECON	0xFFFFF808	0x07	R/W

FEECON is an 8-bit command register. The commands are described in Table 21.

Table 21. Command Codes in FEECON

Code	Command	Description
0x00[1]	Null	Idle State.
0x01[1]	Single Read	Load FEEDAT with the 16-bit data. Indexed by FEEADR.
0x02[1]	Single Write	Write FEEDAT at the address pointed by FEEADR. This operation takes 20 μs.
0x03[1]	Erase/Write	Erase the page indexed by FEEADR and write FEEDAT at the location pointed by FEEADR. This operation takes 20 ms.
0x04[1]	Single Verify	Compare the contents of the location pointed by FEEADR to the data in FEEDAT. The result of the comparison is returned in FEESTA Bit 1.
0x05[1]	Single Erase	Erase the page indexed by FEEADR.
0x06[1]	Mass Erase	Erase 62 kB of user space. The 2 kB of kernel are protected. This operation takes 2.48 seconds. To prevent accidental execution, a command sequence is required to execute this instruction. See the Command Sequence for Executing a Mass Erase section.
0x07	Burst Read	Default Command. No write is allowed. This operation takes two cycles.
0x08	Burst Read/Write	Write can handle a maximum of 8 data of 16 bits and takes a maximum of 8 x 20 μs.
0x09	Erase Burst Read/Write	Automatically erases the page indexed by the write; writes pages without running an erase command. This command takes 20 ms to erase the page + 20 μs per data to write.
0x0A	Reserved	Reserved.
0x0B	Signature	Give a signature of the 64 kB of Flash/EE in the 24-bit FEESIGN MMR. This operation takes 32,778 clock cycles.
0x0C	Protect	This command can run only once. The value of FEEPRO is saved and removed only with a mass erase (0x06) or the key.
0x0D	Reserved	Reserved.
0x0E	Reserved	Reserved.
0x0F	Ping	No operation; interrupt generated.

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

FEEDAT Register

Name	Address	Default Value	Access
FEEDAT	0xFFFFF80C	0xXXXX	R/W

FEEDAT is a 16-bit data register.

FEEADR Register

Name	Address	Default Value	Access
FEEADR	0xFFFFF810	0x0000	R/W

FEEADR is another 16-bit address register.

FEESIGN Register

Name	Address	Default Value	Access
FEESIGN	0xFFFFF818	0xFFFFFF	R

FEESIGN is a 24-bit code signature.

FEEPRO Register

Name	Address	Default Value	Access
FEEPRO	0xFFFFF81C	0x00000000	R/W

FEEPRO MMR provides immediate protection. It does not require any software keys, see Table 22.

FEEHIDE Register

Name	Address	Default Value	Access
FEEHIDE	0xFFFFF820	0xFFFFFFFF	R/W

FEEHIDE provides protection following subsequent reset of the MMR. It requires a software key. See description in Table 22.

Table 22. FEEPRO and FEEHIDE MMR Bit Designations

Bit	Description
31	Read Protection. *Cleared* by user to protect all code. *Set* by user to allow reading the code.
30:0	Write Protection for Pages 123 to 120, Pages 119 to 116, and Pages 0 to 3. *Cleared* by user to protect the pages in writing. *Set* by user to allow writing the pages.

Command Sequence for Executing a Mass Erase

```
FEEDAT=0x3CFF;
FEEADR = 0xFFC3;
FEEMOD= FEEMOD|0x8; //Erase key enable
FEECON=0x06;        //Mass erase command
```

EXECUTION TIME FROM SRAM AND FLASH/EE

Execution from SRAM

Fetching instructions from SRAM takes one clock cycle as 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 get the 32-bit data from Flash/EE. A control flow instruction (a branch instruction, for example) takes one cycle to fetch but 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 22 ns, execution from Flash/EE cannot be done in one cycle (as can be done from SRAM when CD Bit = 0). Also, 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 the core register does not require any extra clock cycle. However, if it involves data in Flash/EE, an extra clock cycle is needed to decode the address of the data, and two cycles are needed to get 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 23.

Table 23. Execution Cycles in ARM/Thumb Mode

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

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

RESET AND REMAP

The ARM exception vectors are all situated at the bottom of the memory array, from address 0x00000000 to address 0x00000020 as shown in Figure 50.

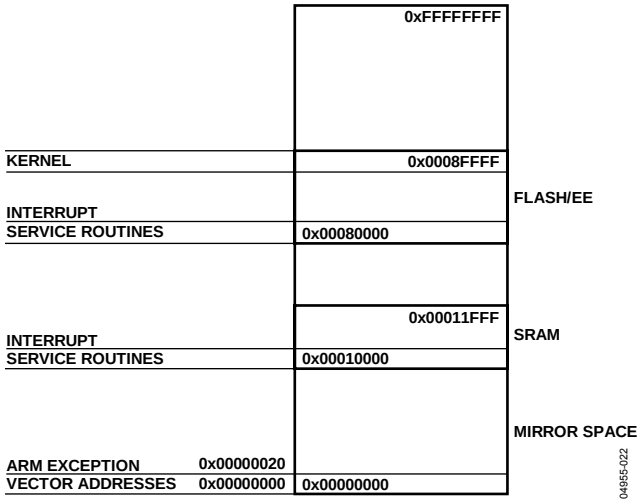


Figure 50. Remap for Exception Execution

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, which facilitates execution of exception routines from SRAM instead of from Flash/EE. This means exceptions are executed twice as fast, being executed in 32-bit ARM mode, and the SRAM being 32-bit wide instead of 16-bit wide Flash/EE memory.

Remap Operation

When a reset occurs on the ADuC7019/7020/7021/7022/7024/7025/7026/7027, execution starts automatically in factory programmed internal configuration code. This kernel is hidden and cannot be accessed by user code. If the ADuC7019/7020/7021/7022/7024/7025/7026/7027 are in normal mode (BM pin is high), then they execute the power-on configuration routine of the kernel and then jump 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 done from Flash/EE by setting Bit 0 of the REMAP register. Precaution must be taken to execute this command from Flash/EE, above address 0x00080020, and not from the bottom of the array as this is replaced by the SRAM.

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

Reset Operation

There are four kinds of reset: external, power-on, watchdog expiation, and software force. The RSTSTA register indicates the source of the last reset, and RSTCLR allows clearing 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 is external.

REMAP Register

Name	Address	Default Value	Access
REMAP	0xFFFF0220	0xXX[1]	R/W

[1] Depends on model.

Table 24. REMAP MMR Bit Designations

Bit	Name	Description
4		Read-Only Bit. Indicates the size of the Flash/EE memory available. If this bit is set, only 32 kB of Flash/EE memory is available.
3		Read-Only Bit. Indicates the size of the SRAM memory available. If this bit is set, only 4 kB of SRAM is available.
2:1		Reserved.
0	Remap	Remap Bit. *Set* by user to remap the SRAM to address 0x00000000. *Cleared* automatically after reset to remap the Flash/EE memory to address 0x00000000.

RSTSTA Register

Name	Address	Default Value	Access
RSTSTA	0xFFFF0230	0x01	R/W

Table 25. RSTSTA MMR Bit Designations

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

RSTCLR Register

Name	Address	Default Value	Access
RSTCLR	0xFFFF0234	0x00	R/W

OTHER ANALOG PERIPHERALS

DAC

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 incorporate two, three, or four 12-bit voltage output DACs on-chip depending on the model. 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 DAC_{REF}, and 0 V to AV_{DD}. DAC_{REF} is equivalent to an external reference for the DAC. The signal range is 0 V to AV_{DD}.

MMRs Interface

Each DAC is configurable independently through a control register and a data register. These two registers are identical for the four DACs. Only DAC0CON (see Table 26) and DAC0DAT (see Table 27) are described in detail in this section.

DACxCON Registers

Name	Address	Default Value	Access
DAC0CON	0xFFFF0600	0x00	R/W
DAC1CON	0xFFFF0608	0x00	R/W
DAC2CON	0xFFFF0610	0x00	R/W
DAC3CON	0xFFFF0618	0x00	R/W

Table 26. DAC0CON MMR Bit Designations

Bit	Value	Name	Description
6			Reserved.
5		DACCLK	DAC Update Rate. *Set* by user to update the DAC using Timer1. *Cleared* by user to update the DAC using HCLK (core clock).
4		DACCLR	DAC Clear Bit. *Set* by user to enable normal DAC operation. *Cleared* by user to reset data register of the DAC to zero.
3			Reserved. This bit should be left at 0.
2			Reserved. This bit should be left at 0.
1:0			DAC Range Bits.
	00		Power-Down Mode. The DAC output is in tri-state.
	01		0 – DAC_{REF} Range.
	10		0 – V_{REF} (2.5 V) Range.
	11		0 – AV_{DD} Range.

DACxDAT Registers

Name	Address	Default Value	Access
DAC0DAT	0xFFFF0604	0x00000000	R/W
DAC1DAT	0xFFFF060C	0x00000000	R/W
DAC2DAT	0xFFFF0614	0x00000000	R/W
DAC3DAT	0xFFFF061C	0x00000000	R/W

Table 27. DAC0DAT MMR Bit Designations

Bit	Description
31:28	Reserved
27:16	12-bit data for DAC0
15:0	Reserved

Using the DACs

The on-chip DAC architecture consists of a resistor string DAC followed by an output buffer amplifier, the functional equivalent of which is shown in Figure 51.

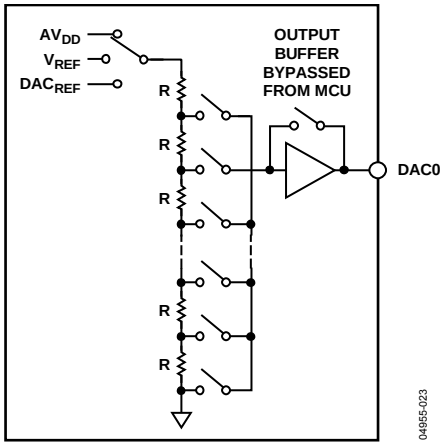


Figure 51. DAC Structure

As illustrated in Figure 51, the reference source for each DAC is user selectable in software. It can be either AV_{DD}, V_{REF}, or DAC_{REF}. In 0-to-AV_{DD} mode, the DAC output transfer function spans from 0 V to the voltage at the AV_{DD} pin. In 0-to-DAC_{REF} mode, the DAC output transfer function spans from 0 V to the voltage at the DAC_{REF} pin. In 0-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, unloaded, each output is capable of swinging to within less than 5 mV of both AV_{DD} and ground. Moreover, the DAC's linearity specification (when driving a 5 kΩ resistive load to ground) is guaranteed through the full transfer function except codes 0 to 100, and, in 0-to-AV_{DD} mode only, codes 3995 to 4095.

Linearity degradation near ground and V_{DD} is caused by saturation of the output amplifier, and a general representation of its effects (neglecting offset and gain error) is illustrated in Figure 52. The dotted line in Figure 52 indicates the ideal transfer function, and the solid line represents what the transfer function might look like with endpoint nonlinearities due to saturation of the output amplifier. Note that Figure 52 represents a transfer function in 0-to-AV_{DD} mode only. In 0-to-V_{REF} or 0-to-DAC_{REF} modes (with $V_{REF} < AV_{DD}$ or $DAC_{REF} < AV_{DD}$), 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.

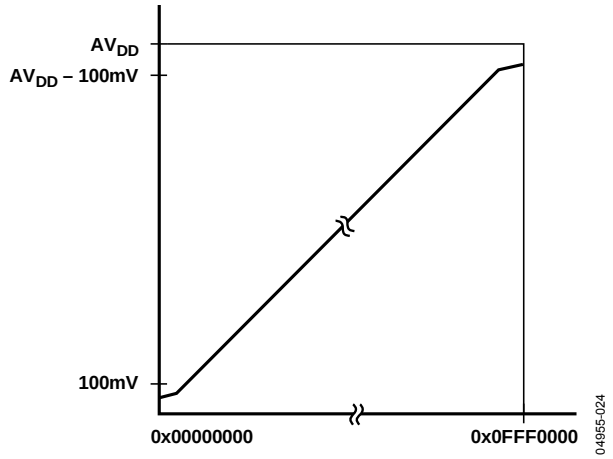


Figure 52. Endpoint Nonlinearities Due to Amplifier Saturation

The endpoint nonlinearities conceptually illustrated in Figure 52 get worse as a function of output loading. Most of the ADuC7019/7020/7021/7022/7024/7025/7026/7027's 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 52 become larger. With larger current demands, this can significantly limit output voltage swing.

POWER SUPPLY MONITOR

The power supply monitor regulates the IOV_{DD} supply on the ADuC7019/7020/7021/7022/7024/7025/7026/7027. It indicates when the IOV_{DD} supply pin drops below one of two supply trip points. The monitor function is controlled via the PSMCON register. If enabled in the IRQEN or FIQEN register, then the monitor interrupts the core using the PSMI bit in the PSMCON MMR. This bit is immediately cleared once CMP goes high.

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

PSMCON Register

Name	Address	Default Value	Access
PSMCON	0xFFFF0440	0x0008	R/W

Table 28. PSMCON MMR Bit Descriptions

Bit	Name	Description
3	CMP	Comparator Bit. This is a read-only bit and directly reflects the state of the comparator. Read 1 indicates that the IOV_{DD} supply is above its selected trip point or the PSM is in power-down mode. Read 0 indicates the IOV_{DD} supply is below its selected trip point. This bit should be set before leaving the interrupt service routine.
2	TP	Trip Point Selection Bits. 0 = 2.79 V, 1 = 3.07 V.
1	PSMEN	Power Supply Monitor Enable Bit. *Set* to 1 to enable the power supply monitor circuit. *Clear* to 0 to disable the power supply monitor circuit.
0	PSMI	Power Supply Monitor Interrupt Bit. This bit is set high by the MicroConverter once when CMP goes low, indicating low I/O supply. The PSMI bit can be used to interrupt the processor. Once CMP returns high, the PSMI bit can be cleared by writing a 1 to this location. A 0 write has no effect. There is no timeout delay; PSMI can be immediately cleared once CMP goes high.

COMPARATOR

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 integrate voltage comparators. The positive input is multiplexed with ADC2 and the negative input has two options: ADC3 or DAC0. The output of the comparator can be configured to generate a system interrupt, can be routed directly to the programmable logic array, can start an ADC conversion, or can be on an external pin, CMP_{OUT}, as shown in Figure 53.

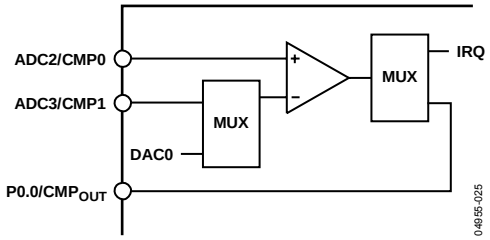


Figure 53. Comparator

Hysteresis

Figure 54 shows how the input offset voltage and hysteresis terms are defined. Input offset voltage (V_{OS}) is the difference between the center of the hysteresis range and the ground level. This can either be positive or negative. The hysteresis voltage (V_H) is ½ the width of the hysteresis range.

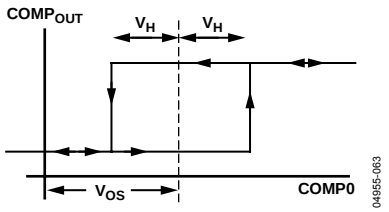


Figure 54. Comparator Hysteresis Transfer Function

Comparator Interface

The comparator interface consists of a 16-bit MMR, CMPCON, which is described in Table 29.

CMPCON Register

Name	Address	Default Value	Access
CMPCON	0xFFFF0444	0x0000	R/W

Table 29. CMPCON MMR Bit Descriptions

Bit	Value	Name	Description
15:11			Reserved.
10		CMPEN	Comparator Enable Bit. *Set* by user to enable the comparator. *Cleared* by user to disable the comparator.
9:8		CMPIN	Comparator Negative Input Select Bits.
	00		AVDD/2.
	01		ADC3 input.
	10		DAC0 output.
	11		Reserved.
7:6		CMPOC	Comparator Output Configuration Bits.
	00		Reserved.
	01		Reserved.
	10		Output on CMP_{OUT}.
	11		IRQ.
5		CMPOL	Comparator Output Logic State Bit. When low, the comparator output is high if the positive input (CMP0) is above the negative input (CMP1). When high, the comparator output is high if the positive input is below the negative input.
4:3		CMPRES	Response Time.
	00		5 μs response time typical for large signals (2.5 V differential).
			17 μs response time typical for small signals (0.65 mV differential).
	11		3 μs typical.
2		CMPHYST	Comparator Hysteresis Bit. *Se*t by user to have a hysteresis of about 7.5 mV. *Cleared* by user to have no hysteresis.
1		CMPORI	Comparator Output Rising Edge Interrupt. *Set* automatically when a rising edge occurs on the monitored voltage (CMP0). *Cleared* by user by writing a 1 to this bit.
0		CMPOFI	Comparator Output Falling Edge Interrupt. *Set* automatically when a falling edge occurs on the monitored voltage (CMP0). *Cleared* by user.

OSCILLATOR AND PLL—POWER CONTROL

Clocking System

Each ADuC7019/7020/7021/7022/7024/7025/7026/7027 integrates a 32.768 kHz ±3% oscillator, a clock divider, and a PLL. The PLL locks onto a multiple (1275) of the internal oscillator or an external 32.768 kHz crystal to provide a stable 41.78 MHz clock for the system referred to as UCLK. To allow power saving, the core can operate at this frequency, or at binary submultiples of it. The actual core operating frequency, UCLK/2^{CD}, is refered to as HCLK. The default core clock is the PLL clock divided by 8 (CD = 3) or 5.22 MHz. The core clock frequency can also come from an external clock on the ECLK pin as described in Figure 55. The core clock can be outputted on the ECLK pin when using an internal oscillator or external crystal.

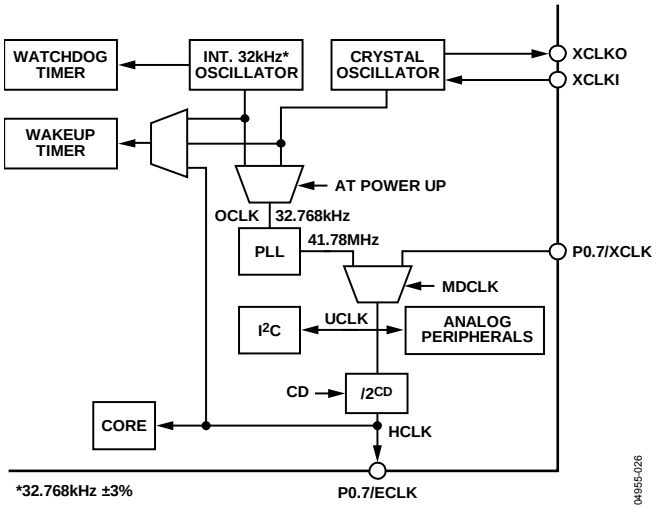


Figure 55. Clocking System

The selection of the clock source is in the PLLCON register. By default, the part uses the internal oscillator feeding the PLL.

External Crystal Selection

To switch to external crystal, clear the OSEL bit in the PLLCON MMR (see Table 32). In noisy environments, noise might couple to the external crystal pins and PLL could lose lock momentarily. A PLL interrupt is provided in the interrupt controller. The core clock is halted immediately and this interrupt is only serviced once the lock has been restored.

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

External Clock Selection

To switch to an external clock on P0.7, configure P0.7 in Mode 1 and MDCLK bits to 11. External clock can be up to 44 MHz providing the tolerance is 1%.

Power Control System

A choice of operating modes is available on the ADuC7019/7020/7021/7022/7024/7025/7026/7027.

Table 30 describes what part is powered on in the different modes and indicates the power-up time. Table 31 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 are measured.

Table 30. Operating Modes

Mode	Core	Peripherals	PLL	XTAL/T2/T3	IRQ0 to IRQ3	Start-up/Power-on Time
Active	X	X	X	X	X	130 ms at CD = 0
Pause		X	X	X	X	24 ns at CD = 0; 3 µs at CD = 7
Nap			X	X	X	24 ns at CD = 0; 3 µs at CD = 7
Sleep				X	X	1.58 ms
Stop					X	1.7 ms

Table 31. 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

The operating mode, clocking mode, and programmable clock divider are controlled via two MMRs, PLLCON (see Table 32) and POWCON (see Table 33). PLLCON controls the operating mode of the clock system, while POWCON controls the core clock frequency and the power-down mode.

To prevent accidental programming, a certain sequence (see Table 34) has to be followed to write to the PLLCON and POWCON registers.

PLLKEYx Registers

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

PLLCON Register

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

POWKEYx Registers

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

POWCON Register

Name	Address	Default Value	Access
POWCON	0xFFFF0408	0x0003	R/W

Table 32. PLLCON MMR Bit Designations

Bit	Value	Name	Description
7:6			Reserved.
5		OSEL	32 kHz PLL Input Selection. *Set* by user to use the internal 32 kHz oscillator. *Set* by default. *Cleared* by 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 P0.7 Pin.

Table 33. POWCON MMR Bit Designations

Bit	Value	Name	Description
7			Reserved.
6:4		PC	Operating Modes.
	000		Active Mode.
	001		Pause Mode.
	010		Nap.
	011		Sleep Mode. IRQ0 to IRQ3 and Timer2 can wake up the ADuC7019/7020/7021/7022/7024/7025/7026/7027.
	100		Stop Mode. IRQ0 to IRQ3 can wake up the ADuC7019/7020/7021/7022/7024/7025/7026/7027.
	Others		Reserved.
3			Reserved.
2:0		CD	CPU Clock Divider Bits.
	000		41.78 MHz.
	001		20.89 MHz.
	010		10.44 MHz.
	011		5.22 MHz.
	100		2.61 MHz.
	101		1.31 MHz.
	110		653 kHz.
	111		326 kHz.

Table 34. PLLCON and POWCON Write Sequence

PLLCON	POWCON
PLLKEY1 = 0xAA	POWKEY1 = 0x01
PLLCON = 0x01	POWCON = User Value
PLLKEY2 = 0x55	POWKEY2 = 0xF4

DIGITAL PERIPHERALS

THREE-PHASE PWM

Each ADuC7019/7020/7021/7022/7024/7025/7026/7027 provides a flexible and programmable, three-phase pulse-width modulation (PWM) waveform generator. It can be programmed to generate the required switching patterns to drive a three-phase voltage source inverter for ac induction motor control (ACIM).

Note that only active high patterns can be produced.

The PWM generator produces three pairs of PWM signals on the six PWM output pins ($PWM0_H$, $PWM0_L$, $PWM1_H$, $PWM1_L$, $PWM2_H$, and $PWM2_L$). The six PWM output signals consist of three high-side drive signals and three low-side drive signals.

The switching frequency and dead time of the generated PWM patterns are programmable using the PWMDAT0 and PWMDAT1 MMRs. In addition, three duty-cycle control registers (PWMCH0, PWMCH1, and PWMCH2) directly control the duty cycles of the three pairs of PWM signals.

Each of the six PWM output signals can be enabled or disabled by separate output enable bits of the PWMEN register. In addition, three control bits of the PWMEN register permit crossover of the two signals of a PWM pair. In crossover mode, the PWM signal destined for the high-side switch is diverted to the complementary low-side output. The signal destined for the low-side switch is diverted to the corresponding high-side output signal.

In many applications, there is a need to provide an isolation barrier in the gate-drive circuits that turn on the power devices of the inverter. In general, there are two common isolation techniques, optical isolation using opto-couplers, and transformer isolation using pulse transformers. The PWM controller permits mixing of the output PWM signals with a high frequency chopping signal to permit easy interface to such pulse transformers. The features of this gate-drive chopping mode can be controlled by the PWMCFG register. An 8-bit value within the PWMCFG register directly controls the chopping frequency. High frequency chopping can be independently enabled for the high-side and low-side outputs using separate control bits in the PWMCFG register.

The PWM generator can operate in one of two distinct modes, single update mode or double update mode. In single update mode, the duty cycle values are programmable only once per PWM period, so that the resulting PWM patterns are symmetrical about the midpoint of the PWM period. In the double update mode, a second updating of the PWM duty cycle values is implemented at the midpoint of the PWM period.

In double update mode, it is also possible to produce asymmetrical PWM patterns that produce lower harmonic distortion in three-phase PWM inverters. This technique permits closed-loop controllers to change the average voltage applied to the machine windings at a faster rate. As a result, faster closed-loop bandwidths are achieved. The operating mode of the PWM block is selected by a control bit in the PWMCON register. In single update mode, a PWMSYNC pulse is produced at the start of each PWM period. In double update mode, an additional PWMSYNC pulse is produced at the midpoint of each PWM period.

The PWM block can also provide an internal synchronization pulse on the PWM_{SYNC} pin that is synchronized to the PWM switching frequency. In single update mode, a pulse is produced at the start of each PWM period. In double update mode, an additional pulse is produced at the mid-point of each PWM period. The width of the pulse is programmable through the PWMDAT2 register. The PWM block can also accept an external synchronization pulse on the PWM_{SYNC} pin. The selection of external synchronization or internal synchronization is in the PWMCON register. The SYNC input timing can be synchronized to the internal peripheral clock, which is selected in the PWMCON register. If the external synchronization pulse from the chip pin is asynchronous to the internal peripheral clock (typical case), the external PWMSYNC is considered asynchronous and should be synchronized. The synchronization logic adds latency and jitter from the external pulse to the actual PWM outputs. The size of the pulse on the PWM_{SYNC} pin must be greater than two core clock periods.

The PWM signals produced by the ADuC7019/7020/7021/7022/7024/7025/7026/7027 can be shut off via a dedicated asynchronous PWM shutdown pin, PWM_{TRIP}. When brought low, PWM_{TRIP} instantaneously places all six PWM outputs in the off state (high). This hardware shutdown mechanism is asynchronous so that the associated PWM disable circuitry does not go through any clocked logic. This ensures correct PWM shutdown even in the event of a core clock loss.

Status information about the PWM system is available to the user in the PWMSTA register. In particular, the state of the PWM_{TRIP} pin is available, as well as a status bit that indicates whether operation is in the first half or the second half of the PWM period.

40-Pin Package Devices

On the 40-pin package devices, the PWM outputs are not directly accessible, as described in the General-Purpose Input/Output section. One channel can be brought out on a GPIO via the PLA as shown in this example:

```
PWMCON = 0x1;       // enables PWM o/p
PWMDAT0 = 0x055F;   // PWM switching freq

// Configure Port Pins
GP4CON = 0x300;     // P4.2 as PLA output
GP3CON = 0x1;       // P3.0 configured as
                    // output of PWM0
                    //(internally)

// PWM0 onto P4.2
PLAELM8 = 0x0035;   // P3.0 (PWM output)
                    // input of element 8
PLAELM10 = 0x0059;  // PWM from element 8
```

Description of the PWM Block

A functional block diagram of the PWM controller is shown in

Figure 56. The generation of the six output PWM signals on pins $PWM0_H$ to $PWM2_L$ is controlled by four important blocks:

- The Three-Phase PWM Timing Unit. The core of the PWM controller, it generates three pairs of complemented and dead-time-adjusted, center-based PWM signals.
- The Output Control Unit. This block can redirect the outputs of the three-phase timing unit for each channel to either the high-side or low-side output. In addition, the output control unit allows individual enabling/disabling of each of the six PWM output signals.
- The Gate Drive Unit. This block can generate the high frequency chopping frequency and its subsequent mixing with the PWM signals.
- The PWM Shutdown Controller. This block takes care of the PWM shutdown via the PWM_{TRIP} pin and generates the correct reset signal for the timing unit.

The PWMSYNC pulse control unit generates the internal synchronization pulse and also controls whether the external PWM_{SYNC} pin is used or not.

The PWM controller is driven by the ADuC7019/7020/7021/7022/7024/7025/7026/7027 core clock frequency and is capable of generating two interrupts to the ARM core. One interrupt is generated on the occurrence of a PWMSYNC pulse, and the other is generated on the occurrence of any PWM shutdown action.

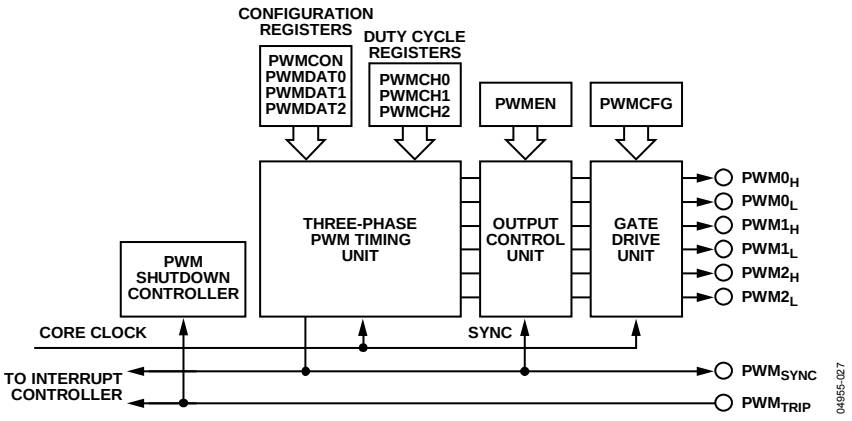


Figure 56. Overview of the PWM Controller

Three-Phase Timing Unit

PWM Switching Frequency (PWMDAT0 MMR)

The PWM switching frequency is controlled by the PWM period register, PWMDAT0. The fundamental timing unit of the PWM controller is

$$t_{CORE} = 1/f_{CORE}$$

where f_{CORE} is the core frequency of the MicroConverter.

Therefore, for a 41.78 MHz f_{CORE}, the fundamental time increment is 24 ns. The value written to the PWMDAT0 register is effectively the number of f_{CORE} clock increments in ½ a PWM period. The required PWMDAT0 value is a function of the desired PWM switching frequency (f_{PWN}) and is given by

$$PWMDAT0 = f_{CORE}/(2 \times f_{PWM})$$

Therefore, the PWM switching period, T_S, can be written as

$$T_S = 2 \times PWMDAT0 \times t_{CORE}$$

The largest value that can be written to the 16-bit PWMDAT0 MMR is 0×FFFF = 65535, which corresponds to a minimum PWM switching frequency of

$$f_{PWM(min)} = 41.78 \times 10^6/(2 \times 65535) = 318.75 \text{ Hz}$$

Note that a PWMDAT0 value of 0 and 1 are not defined and should not be used.

PWM Switching Dead Time (PWMDAT1 MMR)

The second important parameter that must be set up in the initial configuration of the PWM block is the switching dead time. This is a short delay time introduced between turning off one PWM signal (0H, for example) and turning on the complementary signal (0L). This short time delay is introduced to permit the power switch to be turned off (in this case, 0H) to completely recover its blocking capability before the complementary switch is turned on. This time delay prevents a potentially destructive short-circuit condition from developing across the dc link capacitor of a typical voltage source inverter.

The dead time is controlled by the 10-bit, read/write PWMDAT1 register. There is only one dead-time register that controls the dead time inserted into all three pairs of PWM output signals. The dead time, *TD*, is related to the value in the PWMDAT1 register by:

$$TD = PWMDAT1 \times 2 \times t_{CORE}$$

Therefore, a *PWMDAT1* value of 0x00A (= 10), introduces a 426 ns delay between the turn-off on any PWM signal (0H, for example) and the turn-on of its complementary signal (0L). The amount of the dead time can therefore be programmed in increments of $2t_{CORE}$ (or 49 ns for a 41.78 MHz core clock).

The PWMDAT1 register is a 10-bit register with a maximum value of 0x3FF (= 1023), which corresponds to a maximum programmed dead time of

$$TD_{(max)} = 1023 \times 2 \times t_{CORE} = 1023 \times 2 \times 24 \times 10^{-9} = 48.97 \text{ µs}$$

for a core clock of 41.78 MHz

Obviously, the dead time can be programmed to be zero by writing 0 to the PWMDAT1 register.

PWM Operating Mode (PWMCON, PWMSTA MMRs)

As previously discussed, the PWM controller of the ADuC7019/7020/7021/7022/7024/7025/7026/7027 can operate in two distinct modes, single update mode and double update mode. The operating mode of the PWM controller is determined by the state of Bit 2 of the PWMCON register. If this bit is cleared, the PWM operates in the single update mode. Setting Bit 2 places the PWM in the double update mode. The default operating mode is single update mode.

In single update mode, a single PWMSYNC pulse is produced in each PWM period. The rising edge of this signal marks the start of a new PWM cycle, and is used to latch new values from the PWM configuration registers (PWMDAT0 and PWMDAT1) and the PWM duty cycle registers (PWMCH0, PWMCH1, and PWMCH2) into the three-phase timing unit. In addition, the PWMEN register is latched into the output control unit on the rising edge of the PWMSYNC pulse. In effect, this means that the characteristics and resulting duty cycles of the PWM signals can be updated only once per PWM period at the start of each cycle. The result is symmetrical PWM patterns about the midpoint of the switching period.

In double update mode, there is an additional PWMSYNC pulse produced at the midpoint of each PWM period. The rising edge of this new PWMSYNC pulse is again used to latch new values of the PWM configuration registers, duty cycle registers, and the PWMEN register. As a result, it is possible to alter both the characteristics (switching frequency and dead time) as well as the output duty cycles at the midpoint of each PWM cycle. Consequently, it is also possible to produce PWM switching patterns that are no longer symmetrical about the midpoint of the period (asymmetrical PWM patterns). In double update mode, it may be necessary to know whether operation at any point in time is in either the first half or the second half of the PWM cycle. This information is provided by Bit 0 of the PWMSTA register, which is cleared during operation in the first half of each PWM period (between the rising edge of the original PWMSYNC pulse and the rising edge of the new PWMSYNC pulse introduced in double update mode). Bit 0 of the PWMSTA register is set during operation in the second half of each PWM period. This status bit allows the user to make a determination of the particular half-cycle during implementation of the PWMSYNC interrupt service routine, if required.

The advantage of double update mode is that lower harmonic voltages can be produced by the PWM process and faster control bandwidths are possible. However, for a given PWM switching frequency, the PWMSYNC pulses occur at twice the rate in the double update mode. Because new duty cycle values must be computed in each PWMSYNC interrupt service routine, there is a larger computational burden on the ARM core in double update mode.

PWM Duty Cycles (PWMCH0, PWMCH1, PWMCH2 MMRs)

The duty cycles of the six PWM output signals on Pin $PWM0_H$ to Pin $PWM2_L$ are controlled by the three, 16-bit read/write duty cycle registers, PWMCH0, PWMCH1, and PWMCH2. The duty cycle registers are programmed in integer counts of the fundamental time unit, t_{CORE}. They define the desired on-time of the high-side PWM signal produced by the three-phase timing unit over half the PWM period. The switching signals produced by the three-phase timing unit are also adjusted to incorporate the programmed dead time value in the PWMDAT1 register. The three-phase timing unit produces active low signals so that a low level corresponds to a command to turn on the associated power device.

Figure 57 shows a typical pair of PWM outputs (in this case, 0H and 0L) from the timing unit in single update mode. All illustrated time values indicate the integer value in the associated register and can be converted to time by simply multiplying by the fundamental time increment, t_{CORE}. Note that the switching patterns are perfectly symmetrical about the midpoint of the switching period in this mode because the same values of PWMCH0, PWMDAT0, and PWMDAT1 are used to define the signals in both half cycles of the period.

Figure 57 also demonstrates how the programmed duty cycles are adjusted to incorporate the desired dead time into the resulting pair of PWM signals. Clearly, the dead time is incorporated by moving the switching instants of both PWM signals (0H and 0L) away from the instant set by the PWMCH0 register.

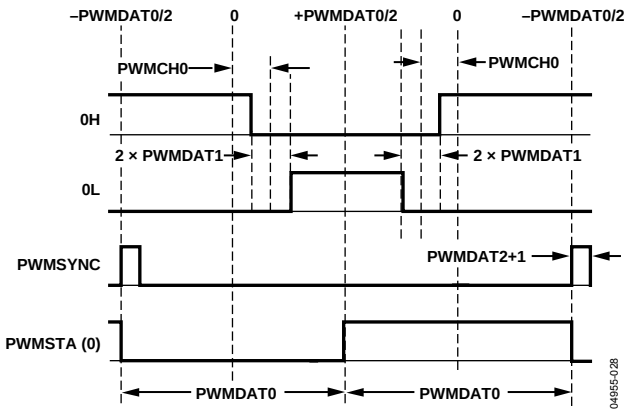


Figure 57. Typical PWM Outputs of Three-Phase Timing Unit in Single Update Mode

Both switching edges are moved by an equal amount ($PWMDAT1 \times t_{CORE}$) to preserve the symmetrical output patterns.

Also shown is the PWMSYNC pulse and Bit 0 of the PWMSTA register, which indicates whether operation is in the first or second half cycle of the PWM period.

The resulting on-times of the PWM signals over the full PWM period (two half periods) produced by the timing unit can be written as follows:

On the high side

$$T_{0HH} = PWMDAT0 + 2(PWMCH0 - PWMDAT1) \times t_{CORE}$$

$$T_{0HL} = PWMDAT0 - 2(PWMCH0 - PWMDAT1) \times t_{CORE}$$

and the corresponding duty cycles (d)

$$d_{0H} = T_{0HH}/T_S = ½ + (PWMCH0 - PWMDAT1)/PWMDAT0$$

and on the low side

$$T_{0LH} = PWMDAT0 - 2(PWMCH0 + PWMDAT1) \times t_{CORE}$$

$$T_{0LL} = PWMDAT0 + 2(PWMCH0 + PWMDAT1) \times t_{CORE}$$

and the corresponding duty cycles (d)

$$d_{OL} = T_{0LH}/T_S = ½ - (PWMCH0 + PWMDAT1)/PWMDAT0$$

The minimum permissible T_{0H} and T_{0L} values are zero, corresponding to a 0% duty cycle. In a similar fashion, the maximum value is T_S, corresponding to a 100% duty cycle.

Figure 58 shows the output signals from the timing unit for operation in double update mode. It illustrates a general case where the switching frequency, dead time, and duty cycle are all changed in the second half of the PWM period. Of course, the same value for any or all of these quantities can be used in both halves of the PWM cycle. However, there is no guarantee that symmetrical PWM signals are produced by the timing unit in double update mode. Figure 58 also shows that the dead time inserted into the PWM signals are done so in the same way as demonstrated in single update mode.

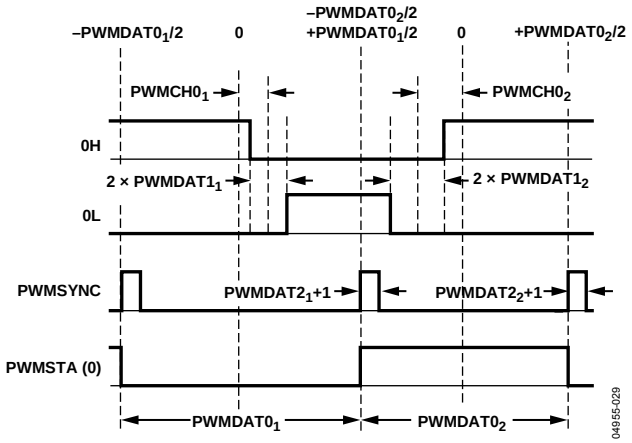


Figure 58. Typical PWM Outputs of the Three-Phase Timing Unit in Double Update Mode

In general, the on-times of the PWM signals in double update mode can be defined as follows:

On the high side

$$T_{0HH} = (PWMDAT0_1/2 + PWMDAT0_2/2 + PWMCH0_1 + PWMCH0_2 - PWMDAT1_1 - PWMDAT1_2) \times t_{CORE}$$

$$T_{0HL} = (PWMDAT0_1/2 + PWMDAT0_2/2 - PWMCH0_1 - PWMCH0_2 + PWMDAT1_1 + PWMDAT1_2) \times t_{CORE}$$

where the subscript *1* refers to the value of that register during the first half cycle, and the subscript 2 refers to the value during the second half cycle.

The corresponding duty cycles (*d*) are

$$d_{0H} = T_{0HH}/T_S = (PWMDAT0_1/2 + PWMDAT0_2/2 + PWMCH0_1 + PWMCH0_2 - PWMDAT1_1 - PWMDAT1_2)/(PWMDAT0_1 + PWMDAT0_2)$$

On the low side

$$T_{0LH} = (PWMDAT0_1/2 + PWMDAT0_2/2 + PWMCH0_1 + PWMCH0_2 + PWMDAT1_1 + PWMDAT1_2) \times t_{CORE}$$

$$T_{0LL} = (PWMDAT0_1/2 + PWMDAT0_2/2 - PWMCH0_1 - PWMCH0_2 - PWMDAT1_1 - PWMDAT1_2) \times t_{CORE}$$

where the subscript *1* refers to the value of that register during the first half cycle, and the subscript 2 refers to the value during the second half cycle.

The corresponding duty cycles (*d*) are

$$d_{0L} = T_{0LH}/T_S = (\mathrm{PWMDAT0}_1/2 + PWMDAT0_2/2 + PWMCH0_1 + PWMCH0_2 + PWMDAT1_1 + PWMDAT1_2)/(PWMDAT0_1 + PWMDAT0_2)$$

For the completely general case in double update mode (see Figure 58), the switching period is given by

$$T_S = (PWMDAT0_1 + PWMDAT0_2) \times t_{CORE}$$

Again, the values of T_{0H} and T_{0L} are constrained to lie between zero and T_S.

PWM signals similar to those illustrated in Figure 57 and Figure 58 can be produced on the 1H, 1L, 2H, and 2L outputs by programming the PWMCH1 and PWMCH2 registers in a manner identical to that described for PWMCH0. The PWM controller does not produce any PWM outputs until all of the PWMDAT0, PWMCH0, PWMCH1, and PWMCH2 registers have been written to at least once. Once these registers have been written, internal counting of the timers in the three-phase timing unit is enabled.

Writing to the PWMDAT0 register starts the internal timing of the main PWM timer. Provided that the PWMDAT0 register is written to prior to the PWMCH0, PWMCH1, and PWMCH2 registers in the initialization, the first PWMSYNC pulse and interrupt (if enabled) appear $1.5 \times t_{CORE} \times PWMDAT0$ seconds after the initial write to the PWMDAT0 register in single update mode. In double update mode, the first PWMSYNC pulse appears after $PWMDAT0 \times t_{CORE}$ seconds.

Output Control Unit

The operation of the output control unit is controlled by the 9-bit read/write PWMEN register. This register controls two distinct features of the output control unit that are directly useful in the control of electronic counter measures (ECM) or binary decimal counter measures (BDCM). The PWMEN register contains three crossover bits, one for each pair of PWM outputs. Setting Bit 8 of the PWMEN register enables the crossover mode for the 0H/0L pair of PWM signals, setting Bit 7 enables crossover on the 1H/1L pair of PWM signals, and setting Bit 6 enables crossover on the 2H/2L pair of PWM signals. If crossover mode is enabled for any pair of PWM signals, the high-side PWM signal from the timing unit (0H, for example) is diverted to the associated low-side output of the output control unit so that the signal ultimately appears at the $PWM0_L$ pin. Of course, the corresponding low-side output of the timing unit is also diverted to the complementary high-side output of the output control unit so that the signal appears at the $PWM0_H$ pin. Following a reset, the three crossover bits are cleared and the crossover mode is disabled on all three pairs of PWM signals. The PWMEN register also contains 6 bits (Bit 0 to Bit 5) that can be used to individually enable or disable each of the six PWM outputs. If the associated bit of the PWMEN register is set, the corresponding PWM output is disabled regardless of corresponding value of the duty cycle register. This PWM output signal remains in the off state as long as the corresponding enable/disable bit of the PWMEN register is set. The implementation of this output enable function is implemented after the crossover function.

Following a reset, all six enable bits of the PWMEN register are cleared, and all PWM outputs are enabled by default. In a manner identical to the duty cycle registers, the PWMEN is latched on the rising edge of the PWMSYNC signal. As a result, changes to this register only become effective at the start of each PWM cycle in single update mode. In double update mode, the PWMEN register can also be updated at the midpoint of the PWM cycle.

In the control of an ECM, only two inverter legs are switched at any time, and often the high-side device in one leg must be switched on at the same time as the low-side driver in a second leg. Therefore, by programming identical duty cycle values for two PWM channels (for example, PWMCH0 = PWMCH1) and setting Bit 7 of the PWMEN register to cross over the 1H/1L pair of PWM signals, it is possible to turn on the high-side switch of Phase A and the low-side switch of Phase B at the same time. In the control of ECM, it is usual for the third inverter leg (Phase C in this example) to be disabled for a number of PWM cycles. This function is implemented by disabling both the 2H and 2L PWM outputs by setting Bit 0 and Bit 1 of the PWMEN register.

This situation is illustrated in Figure 59, where it can be seen that both the 0H and 1L signals are identical, because PWMCH0 = PWMCH1 and the crossover bit for phase B is set.

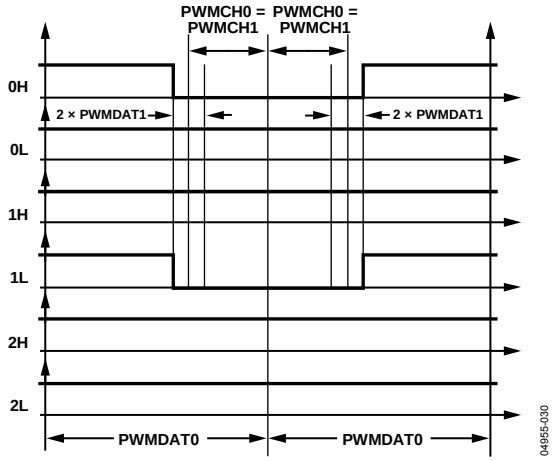


Figure 59. Active LO PWM Signals Suitable for ECM Control, PWMCH0 = PWMCH1, Crossover 1H/1L Pair and Disable 0L, 1H, 2H, and 2L Outputs in Single Update Mode.

In addition, the other four signals (0L, 1H, 2H, and 2L) have been disabled by setting the appropriate enable/disable bits of the PWMEN register. In Figure 59, the appropriate value for the PWMEN register is 0×00A7. In normal ECM operation, each inverter leg is disabled for certain periods of time so that the PWMEN register is changed based on the position of the rotor shaft (motor commutation).

Gate Drive Unit

The gate drive unit of the PWM controller adds features that simplify the design of isolated gate-drive circuits for PWM inverters. If a transformer-coupled, power device, gate-drive amplifier is used, then the active PWM signal must be chopped at a high frequency. The 10-bit read/write PWMCFG register programs this high frequency chopping mode. The chopped active PWM signals can be required for the high-side drivers only, the low-side drivers only, or both the high-side and low-side switches. Therefore, independent control of this mode for both high-side and low-side switches is included with two separate control bits in the PWMCFG register.

Typical PWM output signals with high frequency chopping enabled on both high-side and low-side signals are shown in Figure 60. Chopping of the high-side PWM outputs (0H, 1H, and 2H) is enabled by setting Bit 8 of the PWMCFG register. Chopping of the low-side PWM outputs (0L, 1L, and 2L) is enabled by setting Bit 9 of the PWMCFG register. The high chopping frequency is controlled by the 8-bit word (GDCLK) placed in Bit 0 to Bit 7 of the PWMCFG register. The period of this high frequency carrier is

$$T_{CHOP} = (4 \times (GDCLK + 1)) \times t_{CORE}$$

The chopping frequency is therefore an integral subdivision of the MicroConverter core frequency

$$f_{CHOP} = f_{CORE}/(4 \times (GDCLK + 1))$$

The GDCLK value can range from 0 to 255, corresponding to a programmable chopping frequency rate from 40.8 kHz to 10.44 MHz for a 41.78 MHz core frequency. The gate drive features must be programmed before operation of the PWM controller and are typically not changed during normal operation of the PWM controller. Following a reset, all bits of the PWMCFG register are cleared so that high frequency chopping is disabled, by default.

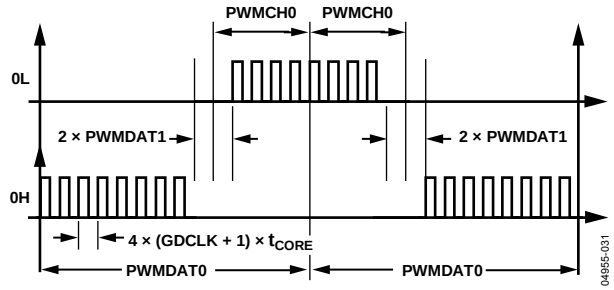


Figure 60. Typical PWM Signals with High Frequency Gate Chopping Enabled on Both High-Side and Low-Side Switches

PWM Shut Down

In the event of external fault conditions, it is essential that the PWM system be instantaneously shut down in a safe fashion. A low level on the PWM_{TRIP} pin provides an instantaneous, asynchronous (independent of the MicroConverter core clock) shutdown of the PWM controller. All six PWM outputs are placed in the off state, that is, high state. In addition, the PWMSYNC pulse is disabled. The PWM_{TRIP} pin has an internal pull-down resistor to disable the PWM if the pin becomes disconnected. The state of the PWM_{TRIP} pin can be read from Bit 3 of the PWMSTA register.

If a PWM shutdown command occurs, a PWMTRIP interrupt is generated, and internal timing of the three-phase timing unit of the PWM controller is stopped. Following a PWM shutdown, the PWM can only be re-enabled (in a PWMTRIP interrupt service routine, for example) by writing to all of the PWMDAT0, PWMCH0, PWMCH1, and PWMCH2 registers. Provided that the external fault is cleared and the PWMTRIP is returned to a high level, the internal timing of the three-phase timing unit resumes, and new duty-cycle values are latched on the next PWMSYNC boundary.

Note that the PWMTRIP interrupt is available in IRQ only, and the PWMSYNC interrupt is available in FIQ only. Both interrupts share the same bit in the interrupt controller. Therefore, only one of the interrupts can be used at once. See the Interrupt System section for further details.

PWM MMRs Interface

The PWM block is controlled via the MMRs described in this section.

PWMCON Register

Name	Address	Default Value	Access
PWMCON	0xFFFFFC00	0x0000	R/W

PWMCON is a control register that enables the PWM and chooses the update rate.

Table 35. PWMCON MMR Bit Descriptions

Bit	Name	Description
7:5		Reserved.
4	PWM_SYNCSEL	External Sync Select. *Set* to use external sync. *Cleared* to use internal sync.
3	PWM_EXTSYNC	External Sync Select. *Set* to select external synchronous sync signal. *Cleared* for asynchronous sync signal.
2	PWMDBL	Double Update Mode. *Set* to 1 by user to enable double update mode. *Cleared* to 0 by the user to enable single update mode.
1	PWM_SYNC_EN	PWM Synchronization Enable. *Set* by user to enable synchronization. *Cleared* by user to disable synchronization.
0	PWMEN	PWM Enable Bit. *Set* to 1 by the user to enable the PWM. *Cleared* to 0 by the user to disable the PWM. Also cleared automatically with PWMTRIP.

PWMSTA Register

Name	Address	Default Value	Access
PWMSTA	0xFFFFFC04	0x0000	R/W

PWMSTA reflects the status of the PWM.

Table 36. PWMSTA MMR Bit Descriptions

Bit	Name	Description
15:10		Reserved.
9	PWMSYNCINT	PWM Sync Interrupt Bit.
8	PWMTRIPINT	PWM Trip Interrupt Bit.
3	PWMTRIP	Raw Signal from the PWM_{TRIP} Pin.
2:1		Reserved.
0	PWMPHASE	PWM Phase Bit. *Set* to 1 by the Micro-Converter when the timer is counting down (1st half). *Cleared* to 0 by the MicroConverter when the timer is counting up (2nd half).

PWMCFG Register

Name	Address	Default Value	Access
PWMCFG	0xFFFFFC10	0x0000	R/W

PWMCFG is a gate chopping register.

Table 37. PWMCFG MMR Bit Descriptions

Bit	Name	Description
15:10		Reserved.
9	CHOPLO	Low-side gate chopping enable bit.
8	CHOPHI	High-side gate chopping enable bit.
7:0	GDCLK	PWM gate chopping period (unsigned).

PWMEN Register

Name	Address	Default Value	Access
PWMEN	0xFFFFFC20	0x0000	R/W

PWMEN allows enabling channel outputs and crossover. See its bit definitions in Table 38.

Table 38. PWMEN MMR Bit Descriptions

Bit	Name	Description
8	0H0L_XOVR	Channel 0 Output Crossover Enable Bit. *Set* to 1 by user to enable Channel 0 output crossover. *Cleared* to 0 by user to disable Channel 0 output crossover.
7	1H1L_XOVR	Channel 1 Output Crossover Enable Bit. *Set* to 1 by user to enable Channel 1 output crossover. *Cleared* to 0 by user to disable Channel 1 output crossover.
6	2H2L_XOVR	Channel 2 Output Crossover Enable Bit. *Set* to 1 by user to enable Channel 2 output crossover. *Cleared* to 0 by user to disable Channel 2 output crossover.
5	0L_EN	0L Output Enable Bit. *Set* to 1 by user to disable the 0L output of the PWM. *Cleared* to 0 by user to enable the 0L output of the PWM.
4	0H_EN	0H Output Enable Bit. *Set* to 1 by user to disable the 0H output of the PWM. *Cleared* to 0 by user to enable the 0H output of the PWM.
3	1L_EN	1L Output Enable Bit. *Set* to 1 by user to disable the 1L output of the PWM. *Cleared* to 0 by user to enable the 1L output of the PWM.
2	1H_EN	1H Output Enable Bit. *Set* to 1 by user to disable the 1H output of the PWM. *Cleared* to 0 by user to enable the 1H output of the PWM.
1	2L_EN	2L Output Enable Bit. *Set* to 1 by user to disable the 2L output of the PWM. *Cleared* to 0 by user to enable the 2L output of the PWM.
0	2H_EN	2H Output Enable Bit. *Set* to 1 by user to disable the 2H output of the PWM. *Cleared* to 0 by user to enable the 2H output of the PWM.

PWMDAT0 Register

Name	Address	Default Value	Access
PWMDAT0	0xFFFFFC08	0x0000	R/W

PWMDAT0 is an unsigned 16-bit register for switching period.

PWMDAT1 Register

Name	Address	Default Value	Access
PWMDAT1	0xFFFFFC0C	0x0000	R/W

PWMDAT1 is an unsigned 10-bit register for dead time.

PWMCHx Registers

Name	Address	Default Value	Access
PWMCH0	0xFFFFFC14	0x0000	R/W
PWMCH1	0xFFFFFC18	0x0000	R/W
PWMCH2	0xFFFFFC1C	0x0000	R/W

PWMCH0, PWMCH1, and PWMCH2 are channel duty cycles for the three phases.

PWMDAT2 Register

Name	Address	Default Value	Access
PWMDAT2	0xFFFFFC24	0x0000	R/W

PWMDAT2 is an unsigned 10-bit register for PWM sync pulse width.

GENERAL-PURPOSE INPUT/OUTPUT

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 provide 40 general-purpose, bi-directional I/O (GPIO) pins. All I/O pins are 5 V tolerant, which means that the GPIOs support an input voltage of 5 V. In general, many of the GPIO pins have multiple functions (see Table 39 for the pin function definitions). 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 GPIO can drive 1.6 mA at the same time. The following GPIO have programmable pull up: P0.0, P0.4, P0.5, P0.6, P0.7, and the 8 GPIOs of P1.

The 40 GPIO are grouped in five ports, Port 0 to Port 4. Each port is controlled by four or five MMRs, x representing the port number.

Note that the kernel changes P0.6 from its default configuration at reset (MRST) to GPIO mode. If MRST is used for external circuitry, an external pull-up resistor should be used to ensure that the level on P0.6 does not drop when the kernel switches mode. For example, if MRST is required for power down, it can be reconfigured in GP0CON MMR.

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

Table 39. GPIO Pin Function Descriptions

		Configuration			
Port	**Pin**	**00**	**01**	**10**	**11**
0	P0.0	GPIO	CMP	MS2	PLAI[7]
	P0.1	GPIO	$PWM2_H$	BLE	
	P0.2	GPIO	$PWM2_L$	BHE	
	P0.3	GPIO	TRST	A16	ADC_{BUSY}
	P0.4	GPIO/IRQ0	PWM_{TRIP}	MS1	PLAO[1]
	P0.5	GPIO/IRQ1	ADC_{BUSY}	MS0	PLAO[2]
	P0.6	GPIO/T1	MRST	AE	PLAO[3]
	P0.7	GPIO	ECLK/XCLK[1]	SIN	PLAO[4]
1	P1.0	GPIO/T1	SIN	SCL0	PLAI[0]
	P1.1	GPIO	SOUT	SDA0	PLAI[1]
	P1.2	GPIO	RTS	SCL1	PLAI[2]
	P1.3	GPIO	CTS	SDA1	PLAI[3]
	P1.4	GPIO/IRQ2	RI	CLK	PLAI[4]
	P1.5	GPIO/IRQ3	DCD	MISO	PLAI[5]
	P1.6	GPIO	DSR	MOSI	PLAI[6]
	P1.7	GPIO	DTR	CSL	PLAO[0]
2	P2.0	GPIO	$\overline{CONV_{START}}$[2]	SOUT	PLAO[5]
	P2.1	GPIO	$PWM0_H$	WS	PLAO[6]
	P2.2	GPIO	$PWM0_L$	RS	PLAO[7]
	P2.3	GPIO		AE	
	P2.4	GPIO	$PWM0_H$	MS0	
	P2.5	GPIO	$PWM0_L$	MS1	
	P2.6	GPIO	$PWM1_H$	MS2	
	P2.7	GPIO	$PWM1_L$	MS3	
3	P3.0	GPIO	$PWM0_H$	AD0	PLAI[8]
	P3.1	GPIO	$PWM0_L$	AD1	PLAI[9]
	P3.2	GPIO	$PWM1_H$	AD2	PLAI[10]
	P3.3	GPIO	$PWM1_L$	AD3	PLAI[11]
	P3.4	GPIO	$PWM2_H$	AD4	PLAI[12]
	P3.5	GPIO	$PWM2_L$	AD5	PLAI[13]
	P3.6	GPIO	PWM_{TRIP}	AD6	PLAI[14]
	P3.7	GPIO	PWM_{SYNC}	AD7	PLAI[15]
4	P4.0	GPIO		AD8	PLAO[8]
	P4.1	GPIO		AD9	PLAO[9]
	P4.2	GPIO		AD10	PLAO[10]
	P4.3	GPIO		AD11	PLAO[11]
	P4.4	GPIO		AD12	PLAO[12]
	P4.5	GPIO		AD13	PLAO[13]
	P4.6	GPIO		AD14	PLAO[14]
	P4.7	GPIO		AD15	PLAO[15]

[1] When configured in Mode 1, P0.7 is ECLK by default, or core clock output. To configure it as a clock input, MDCLK bits in PLLCON must be set to 11.
[2] The $\overline{CONV_{START}}$ signal is active in all modes of P2.0.

GPxCON Registers

Name	Address	Default Value	Access
GP0CON	0xFFFFF400	0x00000000	R/W
GP1CON	0xFFFFF404	0x00000000	R/W
GP2CON	0xFFFFF408	0x00000000	R/W
GP3CON	0xFFFFF40C	0x00000000	R/W
GP4CON	0xFFFFF410	0x00000000	R/W

GPxCON are the port x control registers, which select the function of each pin of port x. as described in Table 40.

Table 40. GPxCON MMR Bit Descriptions

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

GPxPAR Registers

Name	Address	Default Value	Access
GP0PAR	0xFFFFF42C	0x20000000	R/W
GP1PAR	0xFFFFF43C	0x00000000	R/W
GP3PAR	0xFFFFF45C	0x00222222	R/W

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

Table 41. GPxPAR MMR Bit Descriptions

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

GPxDAT Registers

Name	Address	Default Value	Access
GP0DAT	0xFFFFF420	0x000000XX	R/W
GP1DAT	0xFFFFF430	0x000000XX	R/W
GP2DAT	0xFFFFF440	0x000000XX	R/W
GP3DAT	0xFFFFF450	0x000000XX	R/W
GP4DAT	0xFFFFF460	0x000000XX	R/W

GPxDAT are port x configuration and data registers. They configure the direction of the GPIO pins of port x, set the output value for the pins configured as output, and store the input value of the pins configured as input.

Table 42. GPxDAT MMR Bit Descriptions

Bit	Description
31:24	Direction of the Data. *Set* to 1 by user to configure the GPIO pin as an output. *Cleared* to 0 by user to configure the GPIO pin as an input.
23:16	Port x Data Output.
15:8	Reflect the State of Port x Pins at Reset (read only).
7:0	Port x Data Input (read only).

GPxSET Registers

Name	Address	Default Value	Access
GP0SET	0xFFFFF424	0x000000XX	W
GP1SET	0xFFFFF434	0x000000XX	W
GP2SET	0xFFFFF444	0x000000XX	W
GP3SET	0xFFFFF454	0x000000XX	W
GP4SET	0xFFFFF464	0x000000XX	W

GPxSET are data set port x registers.

Table 43. GPxSET MMR Bit Descriptions

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

GPxCLR Registers

Name	Address	Default Value	Access
GP0CLR	0xFFFFF428	0x000000XX	W
GP1CLR	0xFFFFF438	0x000000XX	W
GP2CLR	0xFFFFF448	0x000000XX	W
GP3CLR	0xFFFFF458	0x000000XX	W
GP4CLR	0xFFFFF468	0x000000XX	W

GPxCLR are data clear port x registers.

Table 44. GPxCLR MMR Bit Descriptions

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

SERIAL PORT MUX

The serial port mux multiplexes the serial port peripherals (an SPI, UART, and two I²Cs) and the programmable logic array (PLA) to a set of ten GPIO pins. Each pin must be configured to one of its specific I/O functions as described in Table 45.

Table 45. SPM Configuration

Pin	GPIO (00)	UART (01)	UART/I²C/SPI (10)	PLA (11)
SPM0	P1.0	SIN	I2C0SCL	PLAI[0]
SPM1	P1.1	SOUT	I2C0SDA	PLAI[1]
SPM2	P1.2	RTS	I2C1SCL	PLAI[2]
SPM3	P1.3	CTS	I2C1SDA	PLAI[3]
SPM4	P1.4	RI	SPICLK	PLAI[4]
SPM5	P1.5	DCD	SPIMISO	PLAI[5]
SPM6	P1.6	DSR	SPIMOSI	PLAI[6]
SPM7	P1.7	DTR	SPICSL	PLAO[0]
SPM8	P0.7	ECLK/XCLK	SIN	PLAO[4]
SPM9	P2.0	CONV	SOUT	PLAO[5]

Table 45 also details the mode for each of the SPMUX GPIO pins. This configuration has to be done via the GP0CON, GP1CON, and GP2CON MMRs. By default these ten pins are configured as GPIOs.

UART SERIAL INTERFACE

The UART peripheral is a full-duplex, universal, asynchronous receiver/transmitter. It is fully compatible with the 16450 serial port standard. The UART performs serial-to-parallel conversion on data characters received from a peripheral device or modem, and parallel-to-serial conversion on data characters received from the CPU. The UART includes a fractional divider for baud rate generation and has a network addressable mode. The UART function is made available on the 10 pins of the ADuC7019/7020/7021/7022/7024/7025/7026/7027 (see Table 46).

Table 46. UART Signal Description

Pin	Signal	Description
SPM0 (Mode 1)	SIN	Serial Receive Data
SPM1 (Mode 1)	SOUT	Serial Transmit Data
SPM2 (Mode 1)	RTS	Request to Send
SPM3 (Mode 1)	CTS	Clear to Send
SPM4 (Mode 1)	RI	Ring Indicator
SPM5 (Mode 1)	DCD	Data Carrier Detect
SPM6 (Mode 1)	DSR	Data Set Ready
SPM7 (Mode 1)	DTR	Data Terminal Ready
SPM8 (Mode 2)	SIN	Serial Receive Data
SPM9 (Mode 2)	SOUT	Serial Transmit Data

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

Baud Rate Generation

There are two ways of generating the UART baud rate.

1. 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).

$$Baud\ rate = \frac{41.78\,\text{MHz}}{2^{CD} \times 16 \times 2 \times DL}$$

Table 47 gives some common baud rate values.

Table 47. Baud Rate Using the Normal Baud Rate Generator

Baud Rate	CD	DL	Actual Baud Rate	% Error
9600	0	88 h	9600	0
19200	0	44 h	19200	0
115200	0	0B h	118691	3
9600	3	11 h	9600	0
19200	3	8 h	20400	6.25
115200	3	1 h	163200	41.67

2. Using the Fractional Divider.

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

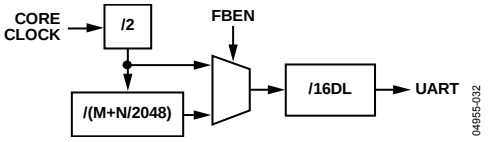


Figure 61. Baud Rate Generation Options

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

$$Baud\ Rate = \frac{41.78\,\text{MHz}}{2^{CD} \times 16 \times DL \times 2 \times \left(M + \frac{N}{2048}\right)}$$

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

For example, generation of 19,200 baud with CD bits = 3 (Table 47 gives DL = 8 h),

$$M + \frac{N}{2048} = \frac{41.78\,\text{MHz}}{19200 \times 2^3 \times 16 \times 8 \times 2}$$

$$M + \frac{N}{2048} = 1.06$$

where:
$M = 1$
$N = 0.06 \times 2048 = 128$

$$Baud\ Rate = \frac{41.78\,\text{MHz}}{2^3 \times 16 \times 8 \times 2 \times \left(\frac{128}{2048}\right)}$$

where:
Baud Rate = 19,200 bps

Error = 0% compared to 6.25% with the normal baud rate generator.

UART Registers Definition

The UART interface consists on 12 registers: COMTX, COMRX, COMDIV0, COMIEN0, COMDIV1, COMIIDO, COMCON0, COMCON1, COMSTA0, COMSTA1, COMSCR, and COMDIV2.

COMTX Register

Name	Address	Default Value	Access
COMTX	0xFFFF0700	0x00	R/W

COMTX is an 8-bit transmit register.

COMRX Register

Name	Address	Default Value	Access
COMRX	0xFFFF0700	0x00	R

COMRX is an 8-bit receive register.

COMDIV0 Register

Name	Address	Default Value	Access
COMDIV0	0xFFFF0700	0x00	R/W

COMDIV0 is a low-byte divisor latch. COMTX, COMRX, and COMDIV0 share the same address location. COMTX and COMRX can be accessed when Bit 7 in COMCON0 register is cleared. COMDIV0 can be accessed when Bit 7 of COMCON0 is set.

COMIEN0 Register

Name	Address	Default Value	Access
COMIEN0	0xFFFF0704	0x00	R/W

COMIEN0 is the interrupt enable register.

Table 48. COMIEN0 MMR Bit Descriptions

Bit	Name	Description
7:4		Reserved.
3	EDSSI	Modem Status Interrupt Enable Bit. *Set* by user to enable generation of an interrupt if any of COMSTA1[3:0] are set. *Cleared* by user.
2	ELSI	Rx Status Interrupt Enable Bit. *Set* by user to enable generation of an interrupt if any of COMSTA0[3:0] are set. *Cleared* by user.
1	ETBEI	Enable Transmit Buffer Empty Interrupt. *Set* by user to enable interrupt when buffer is empty during a transmission. *Cleared* by user.
0	ERBFI	Enable Receive Buffer Full Interrupt. *Set* by user to enable interrupt when buffer is full during a reception. *Cleared* by user.

COMDIV1 Register

Name	Address	Default Value	Access
COMDIV1	0xFFFF0704	0x00	R/W

COMDIV1 is a divisor latch (high byte) register.

COMIID0 Register

Name	Address	Default Value	Access
COMIID0	0xFFFF0708	0x01	R

COMIID0 is the interrupt identification register.

Table 49. COMIID0 MMR Bit Descriptions

Bit 2:1 Status Bits	Bit 0 NINT	Priority	Definition	Clearing Operation
00	1		No interrupt	
11	0	1	Receive line status interrupt	Read COMSTA0
10	0	2	Receive buffer full interrupt	Read COMRX
01	0	3	Transmit buffer empty interrupt	Write data to COMTX or read COMIID0
00	0	4	Modem status interrupt	Read COMSTA1 register

COMCON0 Register

Name	Address	Default Value	Access
COMCON0	0xFFFF070C	0x00	R/W

COMCON0 is the line control register.

Table 50. COMCON0 MMR Bit Descriptions

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

COMCON1 Register

Name	Address	Default Value	Access
COMCON1	0xFFFF0710	0x00	R/W

COMCON1 is the modem control register.

Table 51. COMCON1 MMR Bit Descriptions

Bit	Name	Description
7:5		Reserved.
4	LOOPBACK	Loop Back. *Set* by user to enable loop back mode. In loop back mode, the SOUT is forced high. The modem signals are also directly connected to the status inputs (RTS to CTS, DTR to DSR, OUT1 to RI, and OUT2 to DCD). *Cleared* by user to be in normal mode.
3	PEN	Parity Enable Bit. *Set* by user to transmit and check the parity bit. *Cleared* by user for no parity transmission or checking.
2	STOP	Stop Bit. *Set* by user to transmit 1.5 stop bits if the word length is 5 bits or 2 stop bits if the word length is 6 bits, 7 bits, or 8 bits. The receiver checks the first stop bit only, regardless of the number of stop bits selected. *Cleared* by user to generate 1 stop bit in the transmitted data.
1	RTS	Request To Send. *Set* by user to force the RTS output to 0. *Cleared* by user to force the RTS output to 1.
0	DTR	Data Terminal Ready. *Set* by user to force the DTR output to 0. *Cleared* by user to force the DTR output to 1.

COMSTA0 Register

Name	Address	Default Value	Access
COMSTA0	0xFFFF0714	0x60	R

COMSTA0 is the line status register.

Table 52. COMSTA0 MMR Bit Descriptions

Bit	Name	Description
7		Reserved.
6	TEMT	COMTX Empty Status Bit. *Set* automatically if COMTX is empty. *Cleared* automatically when writing to COMTX.
5	THRE	COMTX and COMRX Empty. *Set* automatically if COMTX and COMRX are empty. *Cleared* automatically when one of the register receives data.
4	BI	Break Error. *Set* when SIN is held low for more than the maximum word length. *Cleared* automatically.
3	FE	Framing Error. *Set* when invalid stop bit. *Cleared* automatically.
2	PE	Parity Error. *Set* when a parity error occurs. *Cleared* automatically.
1	OE	Overrun Error. *Set* automatically if data is overwritten before being read. *Cleared* automatically.
0	DR	Data Ready. *Set* automatically when COMRX is full. *Cleared* by reading COMRX.

COMSTA1 Register

Name	Address	Default Value	Access
COMSTA1	0xFFFF0718	0x00	R

COMSTA1 is a modem status register.

Table 53. COMSTA1 MMR Bit Descriptions

Bit	Name	Description
7	DCD	Data Carrier Eetect.
6	RI	Ring Indicator.
5	DSR	Data Set Ready.
4	CTS	Clear To Send.
3	DDCD	Delta DCD. *Set* automatically if DCD changed state since COMSTA1 last read. *Cleared* automatically by reading COMSTA1.
2	TERI	Trailing Edge RI. *Set* if NRI changed from 0 to 1 since COMSTA1 last read. *Cleared* automatically by reading COMSTA1.
1	DDSR	Delta DSR. *Set* automatically if DSR changed state since COMSTA1 last read. *Cleared* automatically by reading COMSTA1.
0	DCTS	Delta CTS. *Set* automatically if CTS changed state since COMSTA1 last read. *Cleared* automatically by reading COMSTA1.

COMSCR Register

Name	Address	Default Value	Access
COMSCR	0xFFFF071C	0x00	R/W

COMSCR is an 8-bit scratch register used for temporary storage. It is also used in network addressable UART mode.

COMDIV2 Register

Name	Address	Default Value	Access
COMDIV2	0xFFFF072C	0x0000	R/W

COMDIV2 is a 16-bit fractional baud divide register.

Table 54. COMDIV2 MMR Bit Descriptions

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

Network Addressable UART Mode

This mode connects the MicroConverter to a 256-node serial network, either as a hardware single-master or via software in a multimaster network. Bit 7 of COMIEN1 (ENAM bit) must be set to enable UART in network addressable mode. Note that there is no parity check in this mode; the parity bit is used for address.

Network Addressable UART Register Definitions

Four additional registers, COMSCR, COMIEN1, COMIID1, and COMADR are only used in network addressable UART mode.

COMSCR is an 8-bit scratch register used for temporary storage. In network address mode, the least significant bit of the scratch register is the transmitted network address control bit. If set to 1, the device is transmitting an address. If cleared to 0, the device is transmitting data.

COMIEN1 Register

Name	Address	Default Value	Access
COMIEN1	0xFFFF0720	0x04	R/W

COMIEN1 is an 8-bit network enable register.

Table 55. COMIEN1 MMR Bit Descriptions

Bit	Name	Description
7	ENAM	Network Address Mode Enable Bit. *Set* by user to enable network address mode. *Cleared* by user to disable network address mode.
6	E9BT	9-Bit Transmit Enable Bit. *Set* by user to enable 9-bit transmit. ENAM must be set. *Cleared* by user to disable 9-bit transmit.
5	E9BR	9-Bit Receive Enable Bit. *Set* by user to enable 9-bit receive. ENAM must be set. *Cleared* by user to disable 9-bit receive.
4	ENI	Network Interrupt Enable Bit.
3	E9BD	Word Length. *Set* for 9-bit data. E9BT has to be cleared. *Cleared* for 8-bit data.
2	ETD	Transmitter Pin Driver Enable Bit. *Set* by user to enable SOUT pin as an output in slave mode or multimaster mode. *Cleared* by user; SOUT is three-state.
1	NABP	Network Address Bit. Interrupt polarity bit.
0	NAB	Network Address Bit. *Set* by user to transmit the slave's address. *Cleared* by user to transmit data.

COMIID1 Register

Name	Address	Default Value	Access
COMIID1	0xFFFF0724	0x01	R

COMIID1 is an 8-bit network interrupt register. Bit 7 to Bit 4 are reserved (see Table 56).

Table 56. COMIID1 MMR Bit Descriptions

Bit 3:1 Status Bits	Bit 0 NINT	Priority	Definition	Clearing Operation
000	1		No interrupt	
110	0	2	Matching network address	Read COMRX
101	0	3	Address transmitted, buffer empty	Write data to COMTX or read COMIID0
011	0	1	Receive line status interrupt	Read COMSTA0
010	0	2	Receive buffer full interrupt	Read COMRX
001	0	3	Transmit buffer empty interrupt	Write data to COMTX or read COMIID0
000	0	4	Modem status interrupt	Read COMSTA1 register

COMADR Register

Name	Address	Default Value	Access
COMADR	0xFFFF0728	0xAA	R/W

COMADR is an 8-bit, read/write network address register that holds the address that the network addressable UART checks for. Upon receiving this address, the device interrupts the processor and/or sets the appropriate status bit in COMIID1.

SERIAL PERIPHERAL INTERFACE

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 integrate 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 3.48 Mb as shown in Table 57. The SPI interface is not operational with core clock divider bits (CD bits) POWCON[2:0] = 6 or 7 in master mode.

The SPI port can be configured for master or slave operation and typically consists of four pins: MISO, MOSI, SCL, and CS.

MISO (Master In, Slave Out) Pin

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

MOSI (Master Out, Slave In) Pin

The MOSI 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.

SCL (Serial Clock) I/O Pin

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

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

$$f_{serial\ clock} = \frac{f_{UCLK}}{2 \times (1 + SPIDIV)}$$

The maximum speed of the SPI clock is dependant on the clock divider bits and is summarized in Table 57.

Table 57. SPI Speed vs. Clock Divider Bits in Master Mode

CD Bits	0	1	2	3	4	5
SPIDIV in hex	0x05	0x0B	0x17	0x2F	0x5F	0xBF
SPI speed in MHz	3.482	1.741	0.870	0.435	0.218	0.109

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 10.4 Mb at CD = 0. The formula to determine the maximum speed is as follow:

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

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

Chip Select (CS) Input Pin

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

SPI Registers

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

SPISTA Register

Name	Address	Default Value	Access
SPISTA	0xFFFF0A00	0x00	R/W

SPISTA is an 8-bit read-only status register. Only Bit 1 or Bit 4 of this register generates an interrupt. Bit 6 of the SPICON register determines which bit generates the interrupt.

Table 58. SPISTA MMR Bit Descriptions

Bit	Description
7:6	Reserved.
5	SPIRX Data Register Overflow Status Bit. *Set* if SPIRX is overflowing. *Cleared* by reading SPIRX register.
4	SPIRX Data Register IRQ. *Set* automatically if Bit 3 or Bit 5 is set. *Cleared* by reading SPIRX register.
3	SPIRX Data Register Full Status Bit. *Set* automatically if a valid data is present in the SPIRX register. *Cleared* by reading SPIRX register.
2	SPITX Data Register Underflow Status Bit. *Set* automatically if SPITX is under flowing. *Cleared* by writing in the SPITX register.
1	SPITX Data Register IRQ. *Set* automatically if Bit 0 is clear or Bit 2 is set. *Cleared* by writing in the SPITX register or if finished transmission disabling the SPI.
0	SPITX Data Register Empty Status Bit. *Set* by writing to SPITX to send data. This bit is set during transmission of data. *Cleared* when SPITX is empty.

SPIRX Register

Name	Address	Default Value	Access
SPIRX	0xFFFF0A04	0x00	R

SPIRX is an 8-bit read-only receive register.

SPITX Register

Name	Address	Default Value	Access
SPITX	0xFFFF0A08	0x00	W

SPITX is an 8-bit write-only transmit register.

SPIDIV Register

Name	Address	Default Value	Access
SPIDIV	0xFFFF0A0C	0x1B	R/W

SPIDIV is an 8-bit serial clock divider register.

SPICON Register

Name	Address	Default Value	Access
SPICON	0xFFFF0A10	0x0000	R/W

SPICON is a 16-bit control register.

Table 59. SPICON MMR Bit Descriptions

Bit	Description
15:13	Reserved.
12	Continuous Transfer Enable. *Set* by user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the TX register. CS is asserted and remains asserted for the duration of each 8-bit serial transfer until TX is empty. *Cleared* by user to disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPITX register, then a new transfer is initiated after a stall period.
11	Loop Back Enable. *Set* by user to connect MISO to MOSI and test software. *Cleared* by user to be in normal mode.
10	Slave Output Enable. *Set* by user to enable the slave output. *Cleared* by user to disable slave output.
9	Slave Select Input Enable. *Set* by user in master mode to enable the output. *Cleared* by user to disable master output.
8	SPIRX Overflow Overwrite Enable. *Set* by user, the valid data in the RX register is overwritten by the new serial byte received. *Cleared* by user, the new serial byte received is discarded.
7	SPITX Underflow Mode. *Set* by user to transmit 0. *Cleared* by user to transmit the previous data.
6	Transfer and Interrupt Mode. *Set* by user to initiate transfer with a write to the SPITX register. Interrupt only occurs when TX is empty. *Cleared* by user to initiate transfer with a read of the SPIRX register. Interrupt only occurs when RX is full.
5	LSB First Transfer Enable Bit. *Set* by user, the LSB is transmitted first. *Cleared* by user, the MSB is transmitted first.
4	Reserved.
3	Serial Clock Polarity Mode Bit. *Set* by user, the serial clock idles high. *Cleared* by user, the serial clock idles low.
2	Serial Clock Phase Mode Bit. *Set* by user, the serial clock pulses at the beginning of each serial bit transfer. *Cleared* by user, the serial clock pulses at the end of each serial bit transfer.
1	Master Mode Enable Bit. *Set* by user to enable master mode. *Cleared* by user to enable slave mode.
0	SPI Enable Bit. *Set* by user to enable the SPI. *Cleared* by user to disable the SPI.

I²C COMPATIBLE INTERFACES

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 support two fully licensed I²C interfaces. The I²C interfaces are both implemented as a full hardware master and slave interface. Because the two I²C interfaces are identical, this data sheet describes only I2C0 in detail. Note that the two masters and one of the slaves have individual interrupts. See the Interrupt System section.

The two pins used for data transfer, SDA and SCL, are configured in a wired-AND format that allows arbitration in a multimaster system.

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

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

The I²C peripheral master and slave functionality are independent and can be simultaneously active. A slave is activated when a transfer has been initiated on the bus. If it is not addressed, it remains inactive until another transfer is initiated. This also allows a master device, which loses arbitration, to respond as a slave in the same cycle.

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 I2C0DIV MMR as follows:

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

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,

$DIVH = DIVL = 0\times CF$

and for 400 kHz,

$DIVH = DIVL = 0\times 32$

The I2C×DIV register corresponds to DIVH:DIVL.

Slave Addresses

The registers I2C0ID0, I2C0ID1, I2C0ID2, and I2C0ID3 contain the device IDs. The device compares the four I2C0IDx registers to the address byte. The seven most significant bits of either ID register must be identical to that of the seven most significant bits of the first address byte received to be correctly addressed. The LSB of the ID registers, the transfer direction bit, is ignored in the process of address recognition.

I²C Registers

The I²C peripheral interface consists of 18 MMRs, which are discussed in this section.

I2CxMSTA Registers

Name	Address	Default Value	Access
I2C0MSTA	0xFFFF0800	0x00	R
I2C1MSTA	0xFFFF0900	0x00	R

I2CxMSTA are status registers for the master channel.

Table 60. I2C0MSTA MMR Bit Descriptions

Bit	Description
7	Master Transmit FIFO Flush. *Set* by user to flush the master Tx FIFO. *Cleared* automatically once the master Tx FIFO is flushed. This bit also flushes the slave receive FIFO.
6	Master Busy. *Set* automatically if the master is busy. *Cleared* automatically.
5	Arbitration Loss. *Set* in multimaster mode if another master has the bus. *Cleared* when the bus becomes available.
4	No ACK. *Set* automatically if there is no acknowledge of the address by the slave device. *Cleared* automatically by reading the I2C0MSTA register.
3	Master Receive IRQ. *Set* after receiving data. *Cleared* automatically by reading the I2C0MRX register.
2	Master Transmit IRQ. *Set* at the end of a transmission. *Cleared* automatically by writing to the I2C0MTX register.
1	Master Transmit FIFO Underflow. *Set* automatically if the master transmit FIFO is underflowing. *Cleared* automatically by writing to the I2C0MTX register.
0	Master TX FIFO Empty. *Set* automatically if the master transmit FIFO is empty. *Cleared* automatically by writing to the I2C0MTX register.

I2CxSSTA Registers

Name	Address	Default Value	Access
I2C0SSTA	0xFFFF0804	0x01	R
I2C1SSTA	0xFFFF0904	0x01	R

I2CxSSTA are status registers for the slave channel.

Table 61. I2C0SSTA MMR Bit Descriptions

Bit	Value	Description
31:15		Reserved. These bits should be written as 0.
14		START Decode Bit. *Set* by hardware if the device receives a valid START + matching address. *Cleared* by an I²C STOP condition or an I²C general call reset.
13		Repeated START Decode Bit. *Set* by hardware if the device receives a valid repeated START + matching address. *Cleared* by an I²C STOP condition, a read of the I2CSSTA register, or an I²C general call reset.
12:11		ID Decode Bits.
	00	Received Address Natched ID Register 0.
	01	Received Address Matched ID Register 1.
	10	Received Address Matched ID Register 2.
	11	Received Address Matched ID Register 3.
10		Stop After Start and Matching Address Interrupt. *Set* by hardware if the slave device receives an I²C STOP condition after a previous I²C START condition and matching address. *Cleared* by a read of the I2C0SSTA register.
9:8		General Call ID.
	00	No General Call.
	01	General Call Reset and Program Address.
	10	General Call Program Address.
	11	General Call Matching Alternative ID.
7		General Call Interrupt. *Set* if the slave device receives a general call of any type. C*leared* by setting Bit 8 of the I2CxCFG register. If it is a general call reset, then all registers are at their default values. If it is a hardware general call, then the Rx FIFO holds the second byte of the general call. This is similar to the I2C0ALT register (unless it is a general call to reprogram the device address). For more details, see I²C bus specification, version 2.1, Jan. 2000.
6		Slave Busy. *Set* automatically if the slave is busy. *Cleared* automatically.
5		No ACK. *Set* if master asking for data and no data is available. *Cleared* automatically by reading the I2C0SSTA register.
4		Slave Receive FIFO Overflow. *Set* automatically if the slave receive FIFO is overflowing. *Cleared* automatically by reading the I2C0SSTA register.
3		Slave Receive IRQ. *Set* after receiving data. *Cleared* automatically by reading the I2C0SRX register or flushing the FIFO.
2		Slave Transmit IRQ. *Set* at the end of a transmission. *Cleared* automatically by writing to the I2C0STX register.
1		Slave Transmit FIFO Underflow. *Set* automatically if the slave transmit FIFO is underflowing. *Cleared* automatically by writing to the I2C0SSTA MMR.
0		Slave Transmit FIFO Empty. *Set* automatically if the slave transmit FIFO is empty. *Cleared* automatically by writing to the I2C0STX register.

I2CxSRX Registers

Name	Address	Default Value	Access
I2C0SRX	0xFFFF0808	0x00	R
I2C1SRX	0xFFFF0908	0x00	R

I2CxSRX are receive registers for the slave channel.

I2CxSTX Registers

Name	Address	Default Value	Access
I2C0STX	0xFFFF080C	0x00	W
I2C1STX	0xFFFF090C	0x00	W

I2CxSTX are transmit registers for the slave channel.

I2CxMRX Registers

Name	Address	Default Value	Access
I2C0MRX	0xFFFF0810	0x00	R
I2C1MRX	0xFFFF0910	0x00	R

I2CxMRX are receive registers for the master channel.

I2CxMTX Registers

Name	Address	Default Value	Access
I2C0MTX	0xFFFF0814	0x00	W
I2C1MTX	0xFFFF0914	0x00	W

I2CxMTXare transmit registers for the master channel.

I2CxCNT Registers

Name	Address	Default Value	Access
I2C0CNT	0xFFFF0818	0x00	R/W
I2C1CNT	0xFFFF0918	0x00	R/W

I2CxCNT are 3-bit master receive data count registers. If a master read transfer sequence is initiated, then the I2CxCNT registers denote the number of bytes (−1) to be read from the slave device. By default, this counter is 0, which corresponds to 1 byte expected.

I2CxADR Registers

Name	Address	Default Value	Access
I2C0ADR	0xFFFF081C	0x00	R/W
I2C1ADR	0xFFFF091C	0x00	R/W

I2CxADR are master address byte registers. The I2CxADR value is the device address that the master wants to communicate with. It automatically transmits at the start of a master transfer sequence if there is no valid data in the I2CxMTX register when the master enable bit is set.

I2CxBYTE Registers

Name	Address	Default Value	Access
I2C0BYTE	0xFFFF0824	0x00	R/W
I2C1BYTE	0xFFFF0924	0x00	R/W

I2CxBYTE are broadcast byte registers. Data written to these register do not go through the TxFIFO. This data is transmitted at the start of a transfer sequence before the address. Once the byte has been transmitted and acknowledged, the I^2C expects another byte written in I2CxBYTE or an address written to the address register.

I2CxALT Registers

Name	Address	Default Value	Access
I2C0ALT	0xFFFF0828	0x00	R/W
I2C1ALT	0xFFFF0928	0x00	R/W

I2CxALT are hardware general call ID registers used in slave mode.

I2CxCFG Registers

Name	Address	Default Value	Access
I2C0CFG	0xFFFF082C	0x00	R/W
I2C1CFG	0xFFFF092C	0x00	R/W

I2CxCFG are configuration registers.

Table 62. I2C0CFG MMR Bit Descriptions

Bit	Description
31:5	Reserved. These bits should be written by the user as 0.
14	Enable Stop Interrupt. *Set* by the user to generate an interrupt upon receiving a stop condition and after receiving a valid start condition + matching address. *Cleared* by the user to disable the generation of an interrupt upon receiving a stop condition.
13	Reserved.
12	Reserved.
11	Enable Stretch SCL (Holds SCL Low). *Set* by the user to stretch the SCL line. *Cleared* by the user to disable stretching of the SCL line.
10	Reserved.
9	Slave Tx FIFO Request Interrupt Enable. *Set* by the user to disable the slave Tx FIFO request interrupt. *Cleared* by the user to generate an interrupt request just after the negative edge of the clock for the R/W bit. This allows the user to input data into the slave Tx FIFO if it is empty. At 400 ksps and the core clock running at 41.78 MHz, the user has 45 clock cycles to take appropriate action, taking interrupt latency into account.
8	General Call Status Bit Clear. *Set* by the user to clear the general call status bits. *Cleared* automatically by hardware after the general call status bits have been cleared.
7	Master Serial Clock Enable Bit. *Set* by user to enable generation of the serial clock in master mode. *Cleared* by user to disable serial clock in master mode.
6	Loop Back Enable Bit. *Set* by user to internally connect the transition to the reception to test user software. *Cleared* by user to operate in normal mode.
5	Start Back-Off Disable Bit. *Set* by user in multimaster mode. If losing arbitration, the master immediately tries to retransmit. *Cleared* by user to enable start back-off. After losing arbitration, the master waits before trying to retransmit.
4	Hardware General Call Enable. When this bit and the general call enable bit are set, and have received a general call (address 0x00) and a data byte, the device checks the contents of the I2C0ALT against the receive register. If they match, then 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 ADuC7019/7020/7021/7022/7024/7025/7026/7027 watch for these addresses. The device that requires attention embeds its own address into the message. All masters listen and the master that knows how to handle the device contacts its slave and acts appropriately. The LSB of the I2C0ALT register should always be written to a 1, as per I²C January 2000 specification.
3	General Call Enable Bit. *Set* this bit to enable the slave device to ACK an I²C general call, address 0x00 (write). The device then recognizes a data bit. If it receives a 0x06 as the data byte, "Reset and write programmable part of slave address by hardware," then the I²C interface resets as per the I²C January 2000 specification. This command can be used to reset an entire I²C system. The general call interrupt status bit sets on any general call. It is up to the user to take correct action by setting up the I²C interface after a reset. If it receives a 0x04 as the data byte, "Write programmable part of slave address by hardware," then the general call interrupt status bit sets on any general call. It is up to the user to take correct action by reprogramming the device address.
2	Reserved.
1	Master Enable Bit. *Set* by user to enable the master I²C channel. *Cleared* by user to disable the master I²C channel.
0	Slave Enable Bit. *Set* by user to enable the slave I²C channel. A slave transfer sequence is monitored for the device address in I2C0ID0, I2C0ID1, I2C0ID2, and I2C0ID3. If the device address is recognized, the part participates in the slave transfer sequence. *Cleared* by user to disable the slave I²C channel.

I2CxDIV Registers

Name	Address	Default Value	Access
I2C0DIV	0xFFFF0830	0x1F1F	R/W
I2C1DIV	0xFFFF0930	0x1F1F	R/W

I2CxDIV are the clock divider registers.

I2CxIDx Registers

Name	Address	Default Value	Access
I2C0ID0	0xFFFF0838	0x00	R/W
I2C0ID1	0xFFFF083C	0x00	R/W
I2C0ID2	0xFFFF0840	0x00	R/W
I2C0ID3	0xFFFF0844	0x00	R/W
I2C1ID0	0xFFFF0938	0x00	R/W
I2C1ID1	0xFFFF093C	0x00	R/W
I2C1ID2	0xFFFF0940	0x00	R/W
I2C1ID3	0xFFFF0944	0x00	R/W

I2CxID0, I2CxID1, I2CxID2, and I2CxID3 are slave address device ID registers of I2Cx.

I2CxCCNT Registers

Name	Address	Default Value	Access
I2C0CCNT	0xFFFF0848	0x01	R/W
I2C1CCNT	0xFFFF0948	0x01	R/W

I2CxCCNT are 8-bit start/stop generation counters. They hold off SDA low for start and stop conditions.

I2CxFSTA Registers

Name	Address	Default Value	Access
I2C0FSTA	0xFFFF084C	0x0000	R
I2C1FSTA	0xFFFF094C	0x0000	R

I2CxFSTA are FIFO status registers.

Table 63. I2C0FSTA MMR Bit Descriptions

Bit	Value	Description
15:0		Reserved.
9		Master Transmit FIFO Flush. *Set* by the user to flush the master Tx FIFO. *Cleared* automatically once the master Tx FIFO is flushed. This bit also flushes the slave receive FIFO.
8		Slave Transmit FIFO Flush. *Set* by the user to flush the slave Tx FIFO. *Cleared* automatically once the slave Tx FIFO is flushed.
7:6		Master Rx FIFO Status Bits.
	00	FIFO Empty.
	01	Byte Written to FIFO.
	10	1 Byte in FIFO.
	11	FIFO Full.
5:4		Master Tx FIFO Status Bits.
	00	FIFO Empty.
	01	Byte Written to FIFO.
	10	1 Byte in FIFO.
	11	FIFO Full.
3:2		Slave Rx FIFO Status Bits.
	00	FIFO Empty.
	01	Byte Written to FIFO.
	10	1 Byte in FIFO.
	11	FIFO Full.
1:0		Slave Tx FIFO Status Bits.
	00	FIFO Empty.
	01	Byte Written to FIFO.
	10	1 Byte in FIFO.
	11	FIFO Full.

PROGRAMMABLE LOGIC ARRAY (PLA)

Every ADuC7019/7020/7021/7022/7024/7025/7026/7027 integrates a fully programmable logic array (PLA), which consists of two independent but interconnected PLA blocks. Each block consists of eight PLA elements, which gives each part a total of 16 PLA elements.

Each PLA element contains a two-input look-up table that can be configured to generate any logic output function based on two inputs and a flip-flop. This is represented in Figure 62.

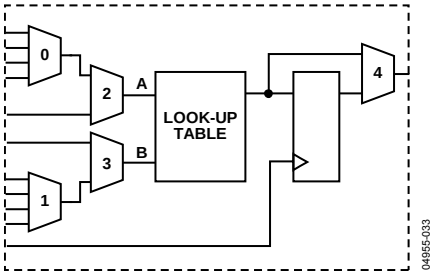


Figure 62. PLA Element

In total, 30 GPIO pins are available on each ADuC7019/7020/7021/7022/7024/7025/7026/7027 for the PLA. These include 16 input pins and 14 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.

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

The two blocks can be interconnected as follows:

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

Table 64. Element Input/Output

PLA Block 0			PLA Block 1		
Element	**Input**	**Output**	**Element**	**Input**	**Output**
0	P1.0	P1.7	8	P3.0	P4.0
1	P1.1	P0.4	9	P3.1	P4.1
2	P1.2	P0.5	10	P3.2	P4.2
3	P1.3	P0.6	11	P3.3	P4.3
4	P1.4	P0.7	12	P3.4	P4.4
5	P1.5	P2.0	13	P3.5	P4.5
6	P1.6	P2.1	14	P3.6	P4.6
7	P0.0	P2.2	15	P3.7	P4.7

PLA MMRs Interface

The PLA peripheral interface consists of the 22 MMRs described in this section.

PLAELMx Registers

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

PLAELMx are Element 0 to Element 15 control registers. They configure the input and output mux of each element, select the function in the look-up table, and bypass/use the flip-flop. See Table 65 and Table 67.

Table 65. PLAELMx MMR Bit Descriptions

Bit	Value	Description
31:11		Reserved.
10:9		Mux (0) Control (see Table 67).
8:7		Mux (1) Control (see Table 67).
6		Mux (2) Control. *Set* by user to select the output of mux (0). *Cleared* by user to select the bit value from PLADIN.
5		Mux (3) Control. *Set* by user to select the input pin of the particular element. *Cleared* by user to select the output of mux (1).
4:1		Look-Up Table Control.
	0000	0.
	0001	NOR.
	0010	B AND NOT A.
	0011	NOT A.
	0100	A AND NOT B.
	0101	NOT B.
	0110	EXOR.
	0111	NAND.
	1000	AND.
	1001	EXNOR.
	1010	B.
	1011	NOT A OR B.
	1100	A.
	1101	A OR NOT B.
	1110	OR.
	1111	1.
0		Mux (4) Control. *Set* by user to bypass the flip-flop. *Cleared* by user to select the flip-flop (cleared by default).

PLACLK Register

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

PLACLK is a clock selection for the flip-flops of Block 0 and clock selection for the flip-flops of Block 1.

Table 66. PLACLK MMR Bit Descriptions

Bit	Value	Description
7		Reserved
6:4		Block 1 Clock Source Selection
	000	GPIO Clock on P0.5
	001	GPIO Clock on P0.0
	010	GPIO Clock on P0.7
	011	HCLK
	100	OCLK (32.768 kHz)
	101	Timer1 Overflow
	Other	Reserved
3		Reserved
2:0		Block 0 Clock Source Selection
	000	GPIO Clock on P0.5
	001	GPIO Clock on P0.0
	010	GPIO Clock on P0.7
	011	HCLK
	100	OCLK (32.768 kHz)
	101	Timer1 Overflow
	Other	Reserved

Table 67. 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

PLAIRQ Register

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

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

Table 68. PLAIRQ MMR Bit Descriptions

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

PLAADC Register

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

PLAADC is a PLA source from the ADC start conversion signal.

Table 69. PLAADC MMR Bit Descriptions

Bit	Value	Description
31:5		Reserved.
4		ADC Start Conversion Enable Bit. *Set* by user to enable ADC start conversion from PLA. *Cleared* by user to disable ADC start conversion from PLA.
3:0		ADC Start Conversion Source.
	0000	PLA Element 0.
	0001	PLA Element 1.
	1111	PLA Element 15.

PLADIN Register

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

PLADIN is a data input MMR for PLA.

Table 70. PLADIN MMR Bit Descriptions

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

PLADOUT Register

Name	Address	Default Value	Access
PLADOUT	0xFFFF0B50	0x00000000	R

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

Table 71. PLADOUT MMR Bit Descriptions

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

PLALCK Register

Name	Address	Default Value	Access
PLALCK	0xFFFF0B54	0x00	W

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

PROCESSOR REFERENCE PERIPHERALS

INTERRUPT SYSTEM

There are 23 interrupt sources on the ADuC7019/7020/7021/7022/7024/7025/7026/7027 that are controlled by the interrupt controller. Most interrupts are generated from the on-chip peripherals, such as ADC and UART. Four additional interrupt sources are generated from external interrupt request pins, IRQ0, IRQ1, IRQ2, and IRQ3. The ARM7TDMI CPU core only recognizes interrupts as one of two types, a normal interrupt request IRQ or a fast interrupt request FIQ. All the interrupts can be masked separately.

The control and configuration of the interrupt system is managed through nine interrupt-related registers, four dedicated to IRQ, and four dedicated to FIQ. An additional MMR is used to select the programmed interrupt source. The bits in each IRQ and FIQ registers (except for Bit 23) represent the same interrupt source as described in Table 72.

Table 72. IRQ/FIQ MMRs Bit Description

Bit	Description
0	All Interrupts OR'ed
1	SWI
2	Timer0
3	Timer1
4	Wake-Up Timer – Timer2
5	Watchdog Timer – Timer3
6	Flash Control
7	ADC Channel
8	PLL Lock
9	I2C0 Slave
10	I2C0 Master
11	I2C1 Master
12	SPI Slave
13	SPI Master
14	UART
15	External IRQ0
16	Comparator
17	PSM
18	External IRQ1
19	PLA IRQ0
20	PLA IRQ1
21	External IRQ2
22	External IRQ3
23	PWM Trip (IRQ only)/ PWM Sync (FIQ only)

IRQ

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

The four 32-bit registers dedicated to IRQ are: IRQSTA, IRQSIG, IRQEN, and IRQCLR.

IRQSTA Register

Name	Address	Default Value	Access
IRQSTA	0xFFFF0000	0x00000000	R

IRQSTA (read-only register) provides the current enabled IRQ source status. When set to 1 that source should generate an active IRQ request to the ARM7TDMI core. There is no priority encoder or interrupt vector generation. This function is implemented in software in a common interrupt handler routine. All 32 bits are logically OR'ed to create the IRQ signal to the ARM7TDMI core.

IRQSIG Register

Name	Address	Default Value	Access
IRQSIG	0xFFFF0004	0x00XXX000	R

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

IRQEN Register

Name	Address	Default Value	Access
IRQEN	0xFFFF0008	0x00000000	R/W

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

IRQCLR Register

Name	Address	Default Value	Access
IRQCLR	0xFFFF000C	0x00000000	W

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

FIQ

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

FIQSTA Register

Name	Address	Default Value	Access
FIQSTA	0xFFFF0100	0x00000000	R

FIQSIG Register

Name	Address	Default Value	Access
FIQSIG	0xFFFF0104	0x00XXX000	R

FIQEN Register

Name	Address	Default Value	Access
FIQEN	0xFFFF0108	0x00000000	R/W

FIQCLR Register

Name	Address	Default Value	Access
FIQCLR	0xFFFF010C	0x00000000	W

Bit 31 to Bit 1 of FIQSTA are logically OR'ed to create the FIQ signal to the core and 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 does, as a side effect, clear the same bit in IRQEN. Also, a bit set to 1 in IRQEN does, as a side effect, clear the same bit in FIQEN. An interrupt source can be disabled in both IRQEN and FIQEN masks.

Programmed Interrupts

Because the programmed interrupts are nonmaskable, they are controlled by another register, SWICFG, which simultaneously writes into the IRQSTA and IRQSIG registers, and/or the FIQSTA and FIQSIG registers. The 32-bit register dedicated to software interrupt is SWICFG described in Table 73. This MMR allows the control of programmed source interrupt.

SWICFG Register

Name	Address	Default Value	Access
SWICFG	0xFFFF0010	0x00000000	W

Table 73. SWICFG MMR Bit Descriptions

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

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

TIMERS

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 have four general-purpose timer/counters:

- Timer0
- Timer1
- Timer2 or Wake-Up Timer
- Timer3 or Watchdog Timer

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

In free-running mode, the counter decreases from the maximum value until zero scale and starts again at the minimum value. (It also increases from the minimum value until full scale and starts again at the maximum value.)

In periodic mode, the counter decrements/increments from the value in the load register (T×LD MMR) until zero/full scale and starts again at the value stored in the load register.

The timer interval is calculated as follow:

$$Interval = \frac{(T \times LD) \times prescaler}{source\ \ clock}$$

The value of a counter can be read at any time by accessing its value register (T×VAL). Note that when a timer is being clocked from a clock other than core clock, an incorrect value could be read (due to asynchronous clock system). In this configuration, T×VAL should always be read twice. If the two readings are different, then it should be read a third time to get the correct value.

Timers are started by writing in the control register of the corresponding timer (T×CON).

In normal mode, an IRQ is generated each time the value of the counter reaches zero when counting down. It is also generated each time the counter value reaches full scale when counting up. An IRQ can be cleared by writing any value to clear the register of that particular timer (T×CLRI).

When using an asynchronous clock-to-clock timer, the interrupt in the timer block could take more time to clear than the time it takes for the code in the interrupt routine to execute. Ensure that the interrupt signal is cleared before leaving the interrupt service routine. This can be done by checking the IRQSTA MMR.

Timer0 (RTOS Timer)

Timer0 is a general-purpose, 16-bit timer (count-down) with a programmable prescaler (see Figure 63). The prescaler source is the core clock frequency (HCLK) and can be scaled by factors of 1, 16, or 256.

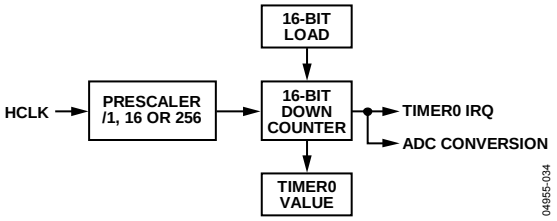


Figure 63. Timer0 Block Diagram

Timer0's interface consists of four MMRS: T0LD, T0VAL, T0CON, and T0CLRI.

T0LD Register

Name	Address	Default Value	Access
T0LD	0xFFFF0300	0x0000	R/W

T0LD is a 16-bit load register.

T0VAL Register

Name	Address	Default Value	Access
T0VAL	0xFFFF0304	0xFFFF	R

T0VAL is a 16-bit read-only register representing the current state of the counter.

T0CON Register

Name	Address	Default Value	Access
T0CON	0xFFFF0308	0x0000	R/W

T0CON is the configuration MMR described in Table 74.

Table 74. T0CON MMR Bit Descriptions

Bit	Value	Description
31:8		Reserved.
7		Timer0 Enable Bit. *Set* by user to enable Timer0. *Cleared* by user to disable Timer0 by default.
6		Timer0 Mode. *Set* by user to operate in periodic mode. *Cleared* by user to operate in free-running mode. Default mode.
5:4		Reserved.
3:2		Prescale.
	00	Core Clock/1. Default value.
	01	Core Clock/16.
	10	Core Clock/256.
	11	Undefined. Equivalent to 00.
1:0		Reserved.

T0CLRI Register

Name	Address	Default Value	Access
T0CLRI	0xFFFF030C	0xFF	W

T0CLRI is an 8-bit register. Writing any value to this register clears the interrupt.

Timer1 (General-Purpose Timer)

Timer1 is a general-purpose, 32-bit timer (count down or count up) with a programmable prescaler. The source can be the 32 kHz external crystal, the core clock frequency, or an external GPIO, P1.0 or P0.6. This source can be scaled by a factor of 1, 16, 256, or 32768.

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

Timer1 has a capture register (T1CAP), which can be triggered by a selected IRQ source initial assertion. This feature can be used to determine the assertion of an event more accurately than the precision allowed by the RTOS timer when the IRQ is serviced.

Timer1 can be used to start ADC conversions as shown in the block diagram in Figure 64.

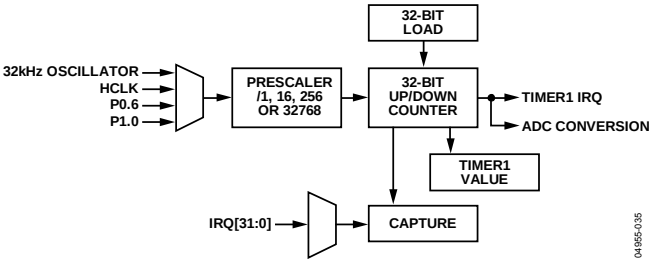


Figure 64. Timer1 Block Diagram

Timer1's interface consists of five MMRS: T1LD, T1VAL, T1CON, T1CLRI, and T1CAP.

T1LD Register

Name	Address	Default Value	Access
T1LD	0xFFFF0320	0x00000000	R/W

T1LD is a 16-bit load register.

T1VAL Register

Name	Address	Default Value	Access
T1VAL	0xFFFF0324	0xFFFFFFFF	R

T1VAL is a 16-bit read-only register that represents the current state of the counter.

T1CON Register

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

T1CON is the configuration MMR described in Table 75.

Table 75. T1CON MMR Bit Descriptions

Bit	Value	Description
31:18		Reserved.
17		Event Select Bit. *Set* by user to enable time capture of an event. *Cleared* by user to disable time capture of an event
16:12		Event Select Range, 0 to 31. These events are as described in Table 72. All events are offset by two, that is, event 2 in Table 72 becomes event zero for the purposes of Timer1.
11:9		Clock Select.
	000	Core Clock (HCLK).
	001	External 32.768 kHz Crystal.
	010	P1.0 Raising Edge Triggered.
	011	P0.6 Raising Edge Triggered.
8		Count Up. *Set* by user for Timer1 to count up. *Cleared* by user for Timer1 to count down by default.
7		Timer1 enable bit. *Set* by user to enable Timer1. *Cleared* by user to disable Timer1 by default.
6		Timer1 Mode. *Set* by user to operate in periodic mode. *Cleared* by user to operate in free-running mode. Default mode.
5:4		Format.
00		Binary.
01		Reserved.
10		Hr:Min:Sec:Hundredths (23 hours to 0 hour).
11		Hr:Min:Sec:Hundredths (255 hours to 0 hour).
3:0		Prescale:
	0000	Source Clock/1.
	0100	Source Clock/16.
	1000	Source Clock/256.
	1111	Source Clock/32768.

T1CLRI Register

Name	Address	Default Value	Access
T1CLRI	0xFFFF032C	0xFF	W

T1CLRI is an 8-bit register. Writing any value to this register clears the Timer1 interrupt.

T1CAP Register

Name	Address	Default Value	Access
T1CAP	0xFFFF0330	0x00000000	R

T1CAP is a 32-bit register. It holds the value contained in T1VAL when a particular event occurred. This event must be selected in T1CON.

Timer2 (Wake-Up Timer)

Timer2 is a 32-bit wake-up timer (count-down or count-up) with a programmable prescaler. The source can be the 32 kHz external crystal, the core clock frequency, or the internal 32 kHz oscillator. The 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.

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

Timer2 can be used to start ADC conversions as shown in the block diagram Figure 65.

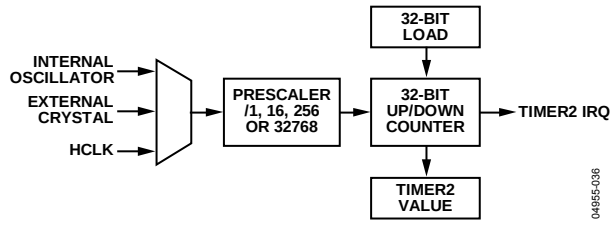


Figure 65. Timer2 Block Diagram

Timer2 interface consists in four MMRS: T2LD, T2VAL, T2CON, and T2CLRI.

T2LD Register

Name	Address	Default Value	Access
T2LD	0xFFFF0340	0x00000000	R/W

T2LD is a 16-bit register load register.

T2VAL Register

Name	Address	Default Value	Access
T2VAL	0xFFFF0344	0xFFFFFFFF	R

T2VAL is a 16-bit read-only register that represents the current state of the counter.

T2CON Register

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

T2CON is the configuration MMR described in Table 76.

Table 76. T2CON MMR Bit Descriptions

Bit	Value	Description
31:11		Reserved.
10:9		Clock Source.
	00	External Crystal.
	01	External Crystal.
	10	Internal Oscillator.
	11	Core Clock (41 MHz/2^{CD}).
8		Count Up. *Set* by user for Timer2 to count up. *Cleared* by user for Timer2 to count down by default.
7		Timer2 Enable Bit. *Set* by user to enable Timer2. *Cleared* by user to disable Timer2 by default.
6		Timer2 Mode. *Set* by user to operate in periodic mode. *Cleared* by user to operate in free-running mode. Default mode.
5:4		Format.
	00	Binary.
	01	Reserved.
	10	Hr:Min:Sec:Hundredths (23 hours to 0 hour).
	11	Hr:Min:Sec:Hundredths (255 hours to 0 hour).
3:0		Prescale:
	0000	Source Clock/1 by Default.
	0100	Source Clock/16.
	1000	Source Clock/256 Expected for Format 2 and 3.
	1111	Source Clock/32768.

T2CLRI Register

Name	Address	Default Value	Access
T2CLRI	0xFFFF034C	0xFF	W

T2CLRI is an 8-bit register. Writing any value to this register clears the Timer2 interrupt.

Timer3 (Watchdog Time)

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

Normal Mode

Timer3 in normal mode is identical to Timer0, except for the clock source and the count-up functionality. The clock source is 32 kHz from the PLL and can be scaled by a factor of 1, 16, or 256 (see Figure 66).

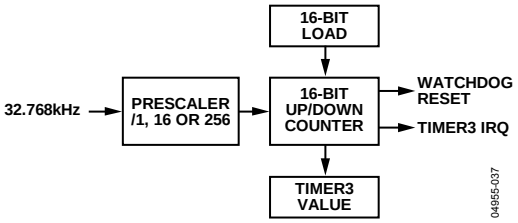


Figure 66. Timer3 Block Diagram

Watchdog Mode

Watchdog mode is entered by setting Bit 5 in T3CON MMR. Timer3 decreases from the value present in T3LD register until zero. T3LD is used as timeout. The maximum timeout can be 512 seconds using the prescaler/256, and full-scale in T3LD. Timer3 is clocked by the internal 32 kHz crystal when operating in the watchdog mode. Note that to enter watchdog mode successfully, Bit 5 in the T3CON MMR must be set after writing to the T3LD MMR.

If the timer reaches 0, a reset or an interrupt occurs, depending on Bit 1 in T3CON register. To avoid reset or interrupt, any value must be written to T3ICLR before the expiration period. This reloads the counter with T3LD and begins a new timeout period.

As soonas watchdog mode is entered, T3LD and T3CON are write-protected. These two registers cannot be modified until a reset clears the watchdog enable bit, which causes Timer3 to exit watchdog mode.

The Timer3 interface consists of four MMRS: T3LD, T3VAL, T3CON, and T3CLRI.

T3LD Register

Name	Address	Default Value	Access
T3LD	0xFFFF0360	0x0000	R/W

T3LD is a 16-bit register load register.

T3VAL Register

Name	Address	Default Value	Access
T3VAL	0xFFFF0364	0xFFFF	R

T3VAL is a 16-bit read-only register that represents the current state of the counter.

T3CON Register

Name	Address	Default Value	Access
T3CON	0xFFFF0368	0x0000	R/W

T3CON is the configuration MMR described in Table 77.

Table 77. T3CON MMR Bit Descriptions

Bit	Value	Description
31:9		Reserved.
8		Count Up. *Set* by user for Timer3 to count up. *Cleared* by user for Timer3 to count down by default.
7		Timer3 Enable Bit. *Set* by user to enable Timer3. *Cleared* by user to disable Timer3 by default.
6		Timer3 Mode. *Set* by user to operate in periodic mode. *Cleared* by user to operate in free-running mode. Default mode.
5		Watchdog Mode Enable Bit. *Set* by user to enable watchdog mode. *Cleared* by user to disable watchdog mode by default.
4		Secure Clear Bit. *Set* by user to use the secure clear option. *Cleared* by user to disable the secure clear option by default.
3:2		Prescale:
	00	Source Clock/1 by Default.
	01	Source Clock/16.
	10	Source Clock/256.
	11	Undefined. Equivalent to 00.
1		Watchdog IRQ Option Bit. *Set* by user to produce an IRQ instead of a reset when the watchdog reaches 0. *Cleared* by user to disable the IRQ option.
0		Reserved.

T3CLRI Register

Name	Address	Default Value	Access
T3CLRI	0xFFFF036C	0x00	W

T3CLRI is an 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.

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 T3ICLR 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 as shown in Figure 67.

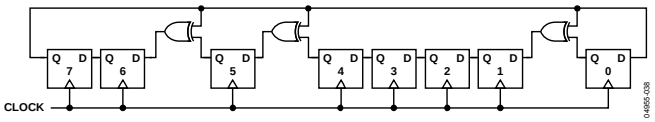


Figure 67. 8-Bit LFSR

The initial value or seed is written to T3ICLR before entering watchdog mode. After entering watchdog mode, a write to T3ICLR 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.

The value 0×00 should not be used as an initial seed due to the properties of the polynomial. The value 0×00 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, 0×AA, in T3ICLR before starting Timer3 in watchdog mode.
2. Enter 0×AA in T3ICLR; Timer3 is reloaded.
3. Enter 0×37 in T3ICLR; Timer3 is reloaded.
4. Enter 0×6E in T3ICLR; Timer3 is reloaded.
5. Enter 0×66. 0×DC was expected; the watchdog reset the chip.

EXTERNAL MEMORY INTERFACING

The ADuC7026 and ADuC7027 are the only models in their series that feature an external memory interface The external memory interface requires a larger number of pins. This is why it is only available on larger pin count packages. The XMCFG MMR must be set to 1 to use the external port.

Although 32-bit addresses are supported internally, only the lower 16 bits of the address are on external pins.

The memory interface can address up to four 128 kB of asynchronous memory (SRAM or/and EEPROM).

The pins required for interfacing to an external memory are shown in Table 78.

Table 78. External Memory Interfacing Pins

Pin	Function
AD[15:0]	Address/Data Bus
A16	Extended Addressing for 8-Bit Memory Only
MS[3:0]	Memory Select Pins
WR	Write Strobe
RS	Read Strobe
AE	Address Latch Enable
BHE, BLE	Byte Write Capability

There are four external memory regions available as described in Table 79. Associated with each region are the pins MS[3:0]. These signals allow access to the particular region of external memory. The size of each memory region can be 128 kB maximum, 64 k × 16 or 128 k × 8. To access 128 k with an 8-bit memory, an extra address line (A16) is provided. (See the example in Figure 68.) The four regions are configured independently.

Table 79. Memory Regions

Address Start	Address End	Contents
0x10000000	0x1000FFFF	External Memory 0
0x20000000	0x2000FFFF	External Memory 1
0x30000000	0x3000FFFF	External Memory 2
0x40000000	0x4000FFFF	External Memory 3

Each external memory region can be controlled through three MMRs: XMCFG, XMxCON, and XMxPAR.

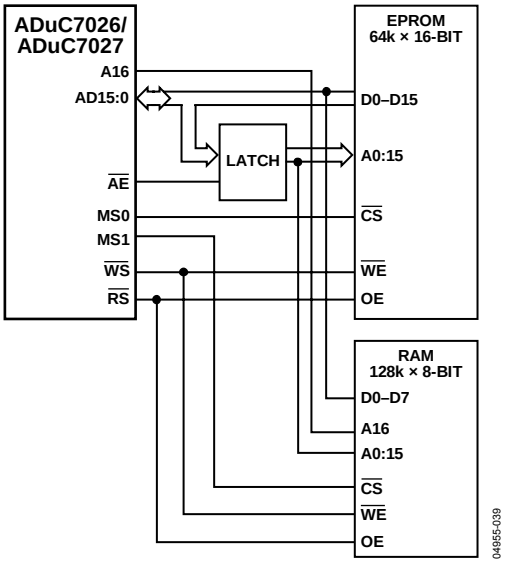


Figure 68. Interfacing to External EPROM/RAM

XMCFG Register

Name	Address	Default Value	Access
XMCFG	0xFFFFF000	0x00	R/W

XMCFG is set to 1 to enable external memory access. This must be set to 1 before any port pins function as external memory access pins. The port pins must also be individually enabled via the GPxCON MMR.

XMxCON Registers

Name	Address	Default Value	Access
XM0CON	0xFFFFF010	0x00	R/W
XM1CON	0xFFFFF014	0x00	R/W
XM2CON	0xFFFFF018	0x00	R/W
XM3CON	0xFFFFF01C	0x00	R/W

XMxCON are the control registers for each memory region. They allow the enabling/disabling of a memory region and control the data bus width of the memory region.

Table 80. XMxCON MMR Bit Descriptions

Bit	Description
1	Selects Between 8-Bit and 16-Bit Data Bus Width. *Set* by the user to select a 16-bit data bus. *Cleared* by the user to select an 8-bit data bus.
0	Enables Memory Region. *Set* by the user to enable memory region. *Cleared* by the user to disable the memory region.

XMxPAR Registers

Name	Address	Default Value	Access
XM0PAR	0xFFFFF020	0x70FF	R/W
XM1PAR	0xFFFFF024	0x70FF	R/W
XM2PAR	0xFFFFF028	0x70FF	R/W
XM3PAR	0xFFFFF02C	0x70FF	R/W

XMxPAR are registers that define the protocol used for accessing the external memory for each memory region.

Table 81. XMxPAR MMR Bit Descriptions

Bit	Description
15	Enable Byte Write Strobe. This bit is only used for two, 8-bit memory sharing the same memory region. *Set* by the user to gate the A0 output with the WR output. This allows byte write capability without using BHE and BLE signals. *Cleared* by user to use BHE and BLE signals.
14:12	Number of wait states on the address latch enable strobe.
11	Reserved.
10	Extra Address Hold Time. *Set* by the user to disable extra hold time. *Cleared* by the user to enable one clock cycle of hold on the address in read and write.
9	Extra bus transition time on read. *Set* by the user to disable extra bus transition time. *Cleared* by the user to enable one extra clock before and after the read strobe (RS).
8	Extra Bus Transition Time On Write. *Set* by the user to disable extra bus transition time. *Cleared* by the user to enable one extra clock before and after the write strobe (WS).
7:4	Number of Write Wait States. Select the number of wait states added to the length of the WS pulse. 0x0 is 1clock; 0xF is 16 clock cycles (default value).
3:0	Number of Read Wait States. Select the number of wait states added to the length of the RS pulse. 0x0 is 1 clock; 0xF is 16 clock cycles (default value).

Figure 69, Figure 70, Figure 71, and Figure 72 show the timing for a read cycle, a read cycle with address hold and bus turn cycles, a write cycle with address and write hold cycles, and a write cycle with wait sates, respectively.

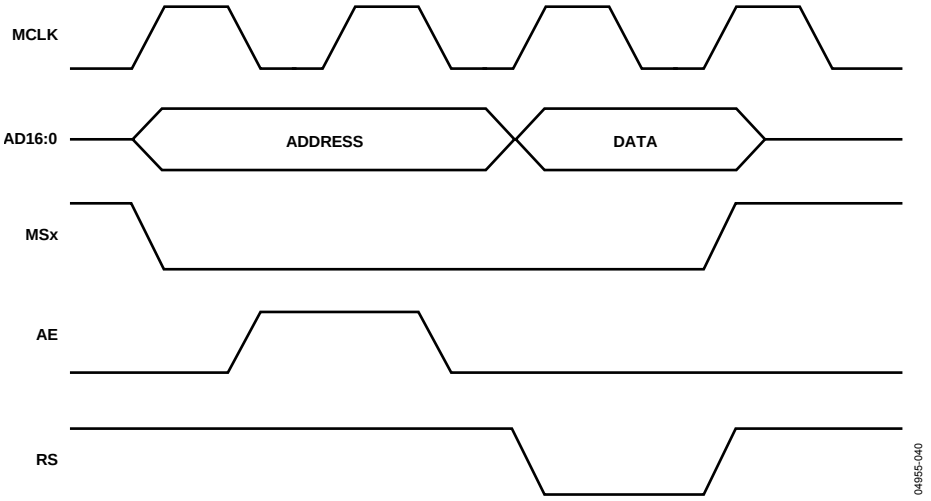


Figure 69. External Memory Read Cycle

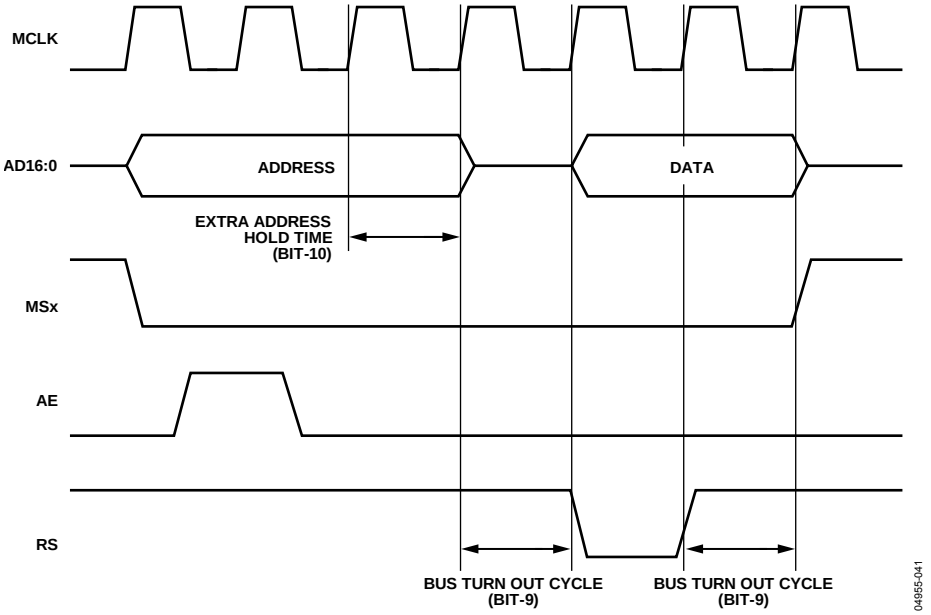


Figure 70. External Memory Read Cycle with Address Hold and Bus Turn Cycles

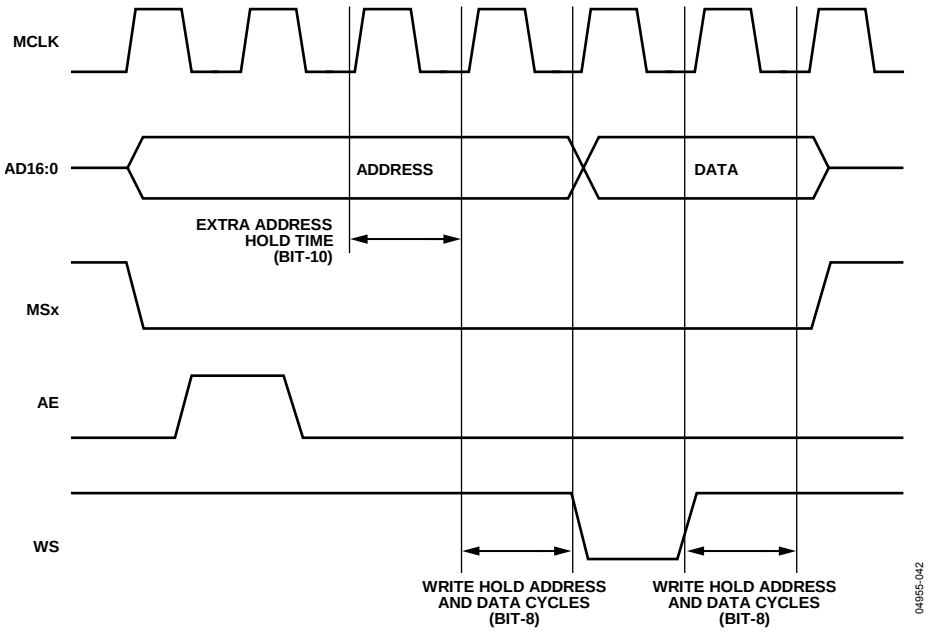


Figure 71. External Memory Write Cycle with Address and Write Hold Cycles

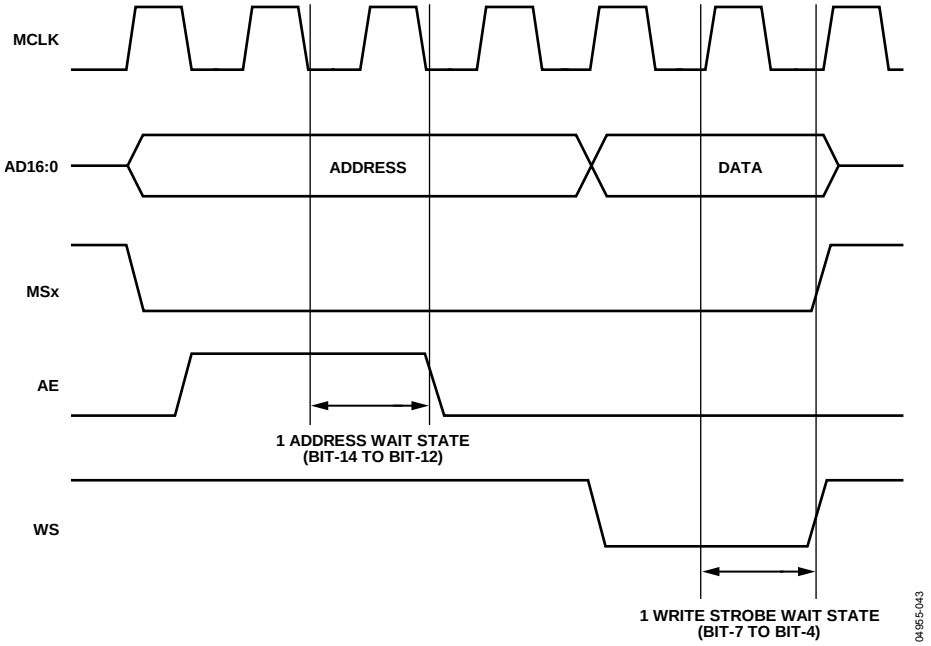


Figure 72. External Memory Write Cycle with Wait States

HARDWARE DESIGN CONSIDERATIONS

POWER SUPPLIES

The ADuC7019/7020/7021/7022/7024/7025/7026/7027 operational power supply voltage range is 2.7 V to 3.6 V. Separate analog and digital power supply pins (AV_{DD} and IOV_{DD}, respectively) allow AV_{DD} to be kept relatively free of noisy digital signals often present on the system IOV_{DD} 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 IOV_{DD} voltage level of 3.3 V while the AV_{DD} level can be at 3 V, or vice versa. A typical split supply configuration is shown in Figure 73.

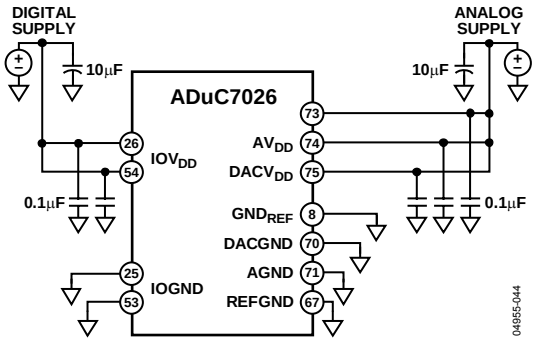


Figure 73. External Dual Supply Connections

As an alternative to providing two separate power supplies, the user can reduce noise on AV_{DD} by placing a small series resistor and/or ferrite bead between AV_{DD} and IOV_{DD}, and then decouple AV_{DD} separately to ground. An example of this configuration is shown in Figure 74. With this configuration, other analog circuitry (such as op amps, voltage reference, and others) can be powered from the AV_{DD} supply line as well.

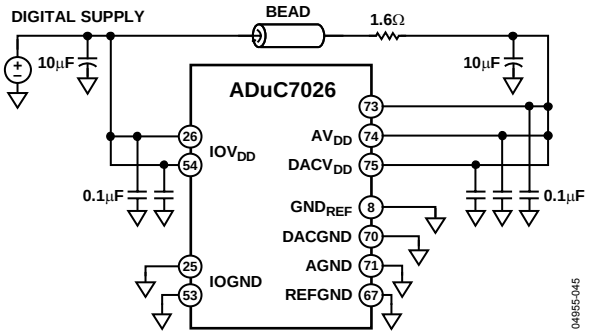


Figure 74. External Single Supply Connections

Notice that in both Figure 73 and Figure 74, a large value (10 µF) reservoir capacitor sits on IOV_{DD}, and a separate 10 µF capacitor sits on AV_{DD}. In addition, local small-value (0.1 µF) capacitors are located at each AV_{DD} and IOV_{DD} 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 AV_{DD} 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 ADuC7019/7020/7021/7022/7024/7025/7026/7027 must be referenced to the same system ground reference point at all times.

Linear Voltage Regulator

Each ADuC7019/7020/7021/7022/7024/7025/7026/7027 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 IOV_{DD} for the core logic. LV_{DD} Pin 21 is the 2.6 V supply for the core logic. An external compensation capacitor of 0.47 µF must be connected between LV_{DD} and DGND (as close as possible to these pins) to act as a tank of charge as shown Figure 75.

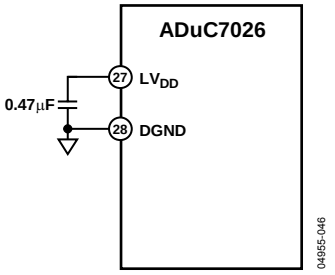


Figure 75. Voltage Regulator Connections

The LV_{DD} pin should not be used for any other chip. It is also recommended to use excellent power supply decoupling on IOV_{DD} 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 ADuC7019/7020/7021/7022/7024/7025/7026/7027-based designs in order to achieve optimum performance from the ADCs and DAC.

Although the ADuC7019/7020/7021/7022/7024/7025/7026/7027 have 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 76a. In systems where digital and analog ground planes are connected together somewhere else (at the system's power supply, for example), the planes cannot be reconnected near the part, because a ground loop would result. In these cases, tie all the ADuC7019/7020/7021/7022/7024/7025/7026/7027's AGND and IOGND pins to the analog ground plane, as illustrated in Figure 76b. 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 ADuC7019/7020/7021/7022/7024/7025/7026/7027 can then be placed between the digital and analog sections, as illustrated in Figure 76c.

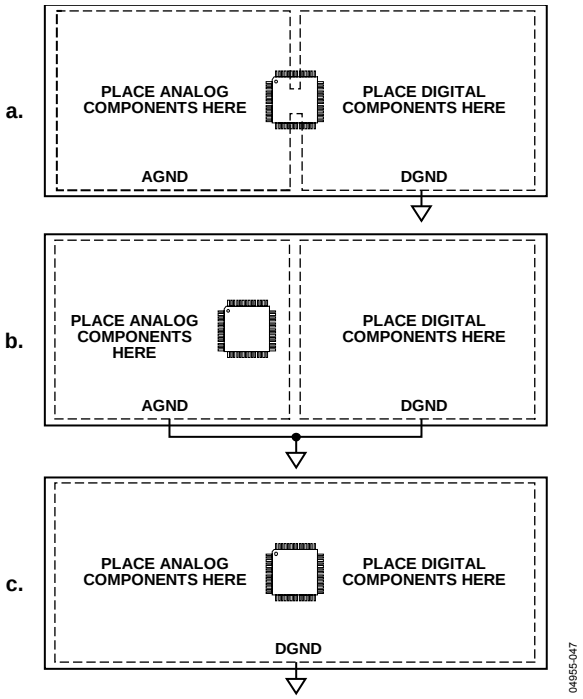


Figure 76. System Grounding Schemes

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 76b, with IOV_{DD} because that would force return currents from IOV_{DD} 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 76c. 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 ADuC7019/7020/7021/7022/7024/7025/7026/7027's digital inputs, add a series resistor to each relevant line to keep rise and fall times longer than 5 ns at the ADuC7019/7020/7021/7022/7024/7025/7026/7027 input pins. 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 ADuC7019/7020/7021/7022/7024/7025/7026/7027 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 XCLKI and XCLKO, and connect a capacitor from each pin to ground as shown Figure 77. 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.

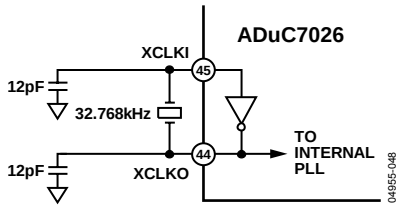


Figure 77. External Parallel Resonant Crystal Connections

To use an external source clock input instead of the PLL (see Figure 78), Bit 1 and Bit 0 of PLLCON must be modified.The external clock uses P0.7 and XCLK.

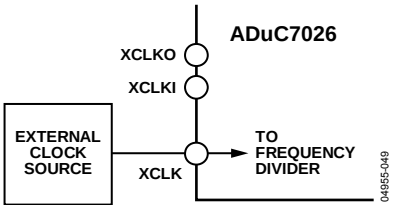


Figure 78. Connecting an External Clock Source

Using an external clock source, the ADuC7019/7020/7021/7022/7024/7025/7026/7027's specified operational clock speed range is 50 kHz to 44 MHz ±1% to ensure correct operation of the analog peripherals and Flash/EE.

POWER-ON RESET OPERATION

An internal power-on reset (POR) is implemented on the ADuC7019/7020/7021/7022/7024/7025/7026/7027. For LV_{DD} below 2.35 V typical, the internal POR holds the part in reset. As LV_{DD} rises above 2.35 V, an internal timer times out for typically 128 ms before the part is released from reset. The user must ensure that the power supply IOV_{DD} has reached a stable 2.7 V minimum level by this time. Likewise, on power-down, the internal POR holds the ADuC7019/7020/7021/7022/7024/7025/7026/7027 in reset until LV_{DD} has dropped below 2.35 V.

Figure 79 illustrates the operation of the internal POR in detail.

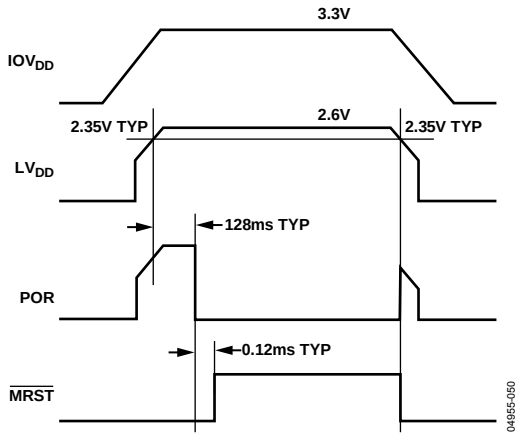


Figure 79. ADuC7019/7020/7021/7022/7024/7025/7026/7027 Internal Power-on Reset Operation

TYPICAL SYSTEM CONFIGURATION

A typical ADuC7020 configuration is shown in Figure 80. It summarizes some of the hardware considerations discussed in the previous sections. The bottom of the CSP package has an exposed pad that needs to be soldered to a metal plate on the board for mechanical reasons. The metal plate of the board can be connected to ground.

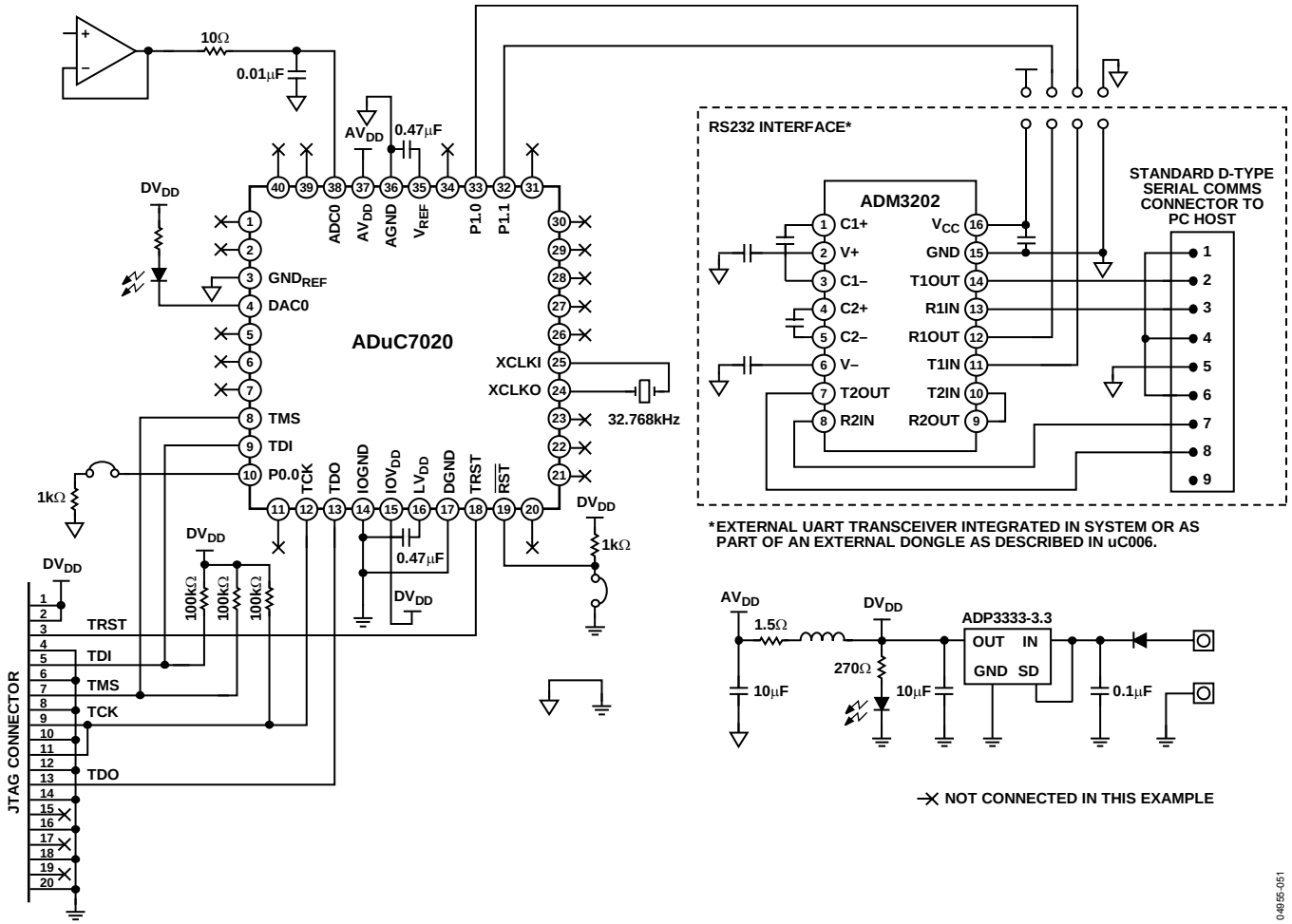


Figure 80. Typical System Configuration

DEVELOPMENT TOOLS

PC-BASED TOOLS

Four types of development systems are available for the ADuC7019/7020/7021/7022/7024/7025/7026/7027 family:

- The ADuC7026 QuickStart Plus is intended for new users who want to have a comprehensive hardware development environment. Since the ADuC7026 contains the superset of functions available on the ADuC7019/7020/7021/7022/7024/7025/7026/7027 family, it is suitable for users who wish to develop on any of the parts in this family. All of the parts are fully code compatible.
- The ADuC7020, ADuC7024, and ADuC7026 QuickStart are intended for users who already have an emulator.

These systems consist of the following PC-based (Windows® compatible) hardware and software development tools:

Hardware

- ADuC7019/7020/7021/7022/7024/7025/7026/7027 evaluation board
- Serial port programming cable
- RDI compliant JTAG emulator (included in the ADuC7026 QuickStart Plus only)

Software

- Integrated development environment, incorporating assembler, compiler, and nonintrusive JTAG-based debugger
- Serial downloader software
- Example code

Miscellaneous

- CD-ROM documentation

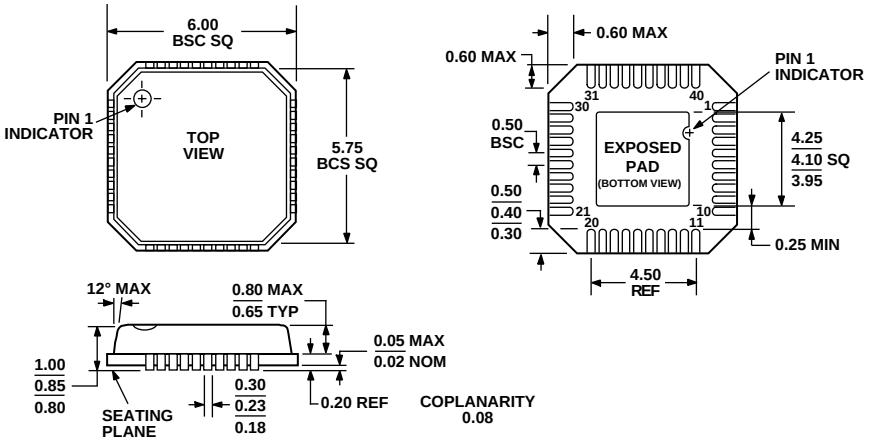
IN-CIRCUIT SERIAL DOWNLOADER

The serial downloader is a Windows application that allows the user to serially download an assembled program to the on-chip program Flash/EE memory via the serial port on a standard PC.

The UART based serial downloader is included in all the development systems and is usable with ADuC7019/7020/7021/7022/7024/7025/7026/7027 that do not contain the "I" suffix in the ordering guide.

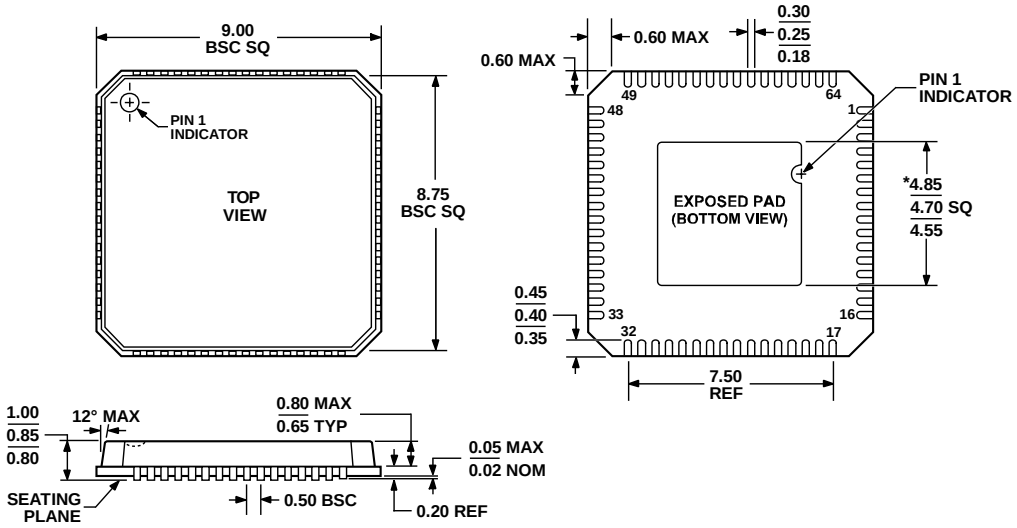
An I^2C based serial downloader is also available at www.analog.com. This software requires an USB to I^2C adaptor board available from http://www.fh-pforzheim.de/stw-svs/texte/Dongle.html. The I^2C based serial downloader is only usable with the part models containing the "I" suffix in the ordering guide.

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-220-VJJD-2

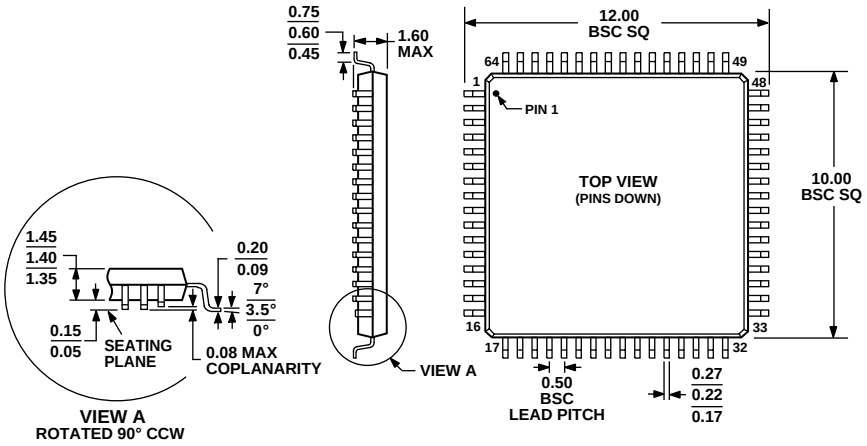
Figure 81. 40-Lead Frame Chip Scale Package [LFCSP_VQ]
6 mm x 6 mm Body, Very Thin Quad
(CP-40)
Dimensions shown in millimeters



***COMPLIANT TO JEDEC STANDARDS MO-220-VMMD-4 EXCEPT FOR EXPOSED PAD DIMENSION**

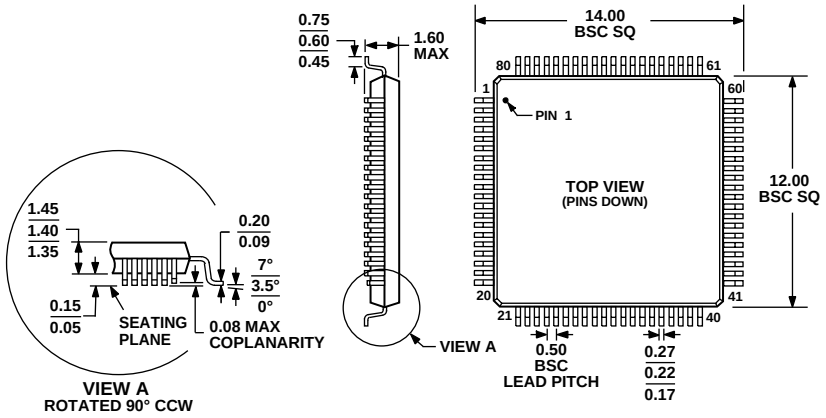
112805-0

Figure 82. 64-Lead Frame Chip Scale Package [LFCSP_VQ]
9 mm x 9 mm Body, Very Thin Quad
(CP-64-1)
Dimensions shown in millimeters



COMPLIANT TO JEDEC STANDARDS MS-026-BCD

Figure 83. 64-Lead Low Profile Quad Flat Package [LQFP]
(ST-64-2)
Dimensions shown in millimeters



COMPLIANT TO JEDEC STANDARDS MS-026-BDD

Figure 84. 80-Lead Low Profile Quad Flat Package [LQFP]
(ST-80-1)
Dimensions shown in millimeters

ORDERING GUIDE

Model	ADC Channels	DAC Channels	FLASH/RAM	GPIO	Downloader	Temperature Range	Package Description	Package Option	Ordering Quantity
ADuC7019BCPZ62I[1]	5[2]	3	62 kB/8 kB	14	I²C	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	
ADuC7019BCPZ62I-RL[1]	5[2]	3	62 kB/8 kB	14	I²C	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	2,500
ADuC7019BCPZ62IRL7[1]	5[2]	3	62 kB/8 kB	14	I²C	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	750
ADuC7020BCPZ62[1]	5	4	62 kB/8 kB	14	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	
ADuC7020BCPZ62-RL[1]	5	4	62 kB/8 kB	14	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	2,500
ADuC7020BCPZ62-RL7[1]	5	4	62 kB/8 kB	14	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	750
ADuC7020BCPZ62I[1]	5	4	62 kB/8 kB	14	I²C	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	
ADuC7020BCPZ62I-RL[1]	5	4	62 kB/8 kB	14	I²C	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	2,500
ADuC7020BCPZ62IRL7[1]	5	4	62 kB/8 kB	14	I²C	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	750
ADuC7021BCPZ62[1]	8	2	62 kB/8 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	
ADuC7021BCPZ62-RL[1]	8	2	62 kB/8 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	2,500
ADuC7021BCPZ62-RL7[1]	8	2	62 kB/8 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	750
ADuC7021BCPZ62I[1]	8	2	62 kB/8 kB	13	I²C	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	
ADuC7021BCPZ62I-RL[1]	8	2	62 kB/8 kB	13	I²C	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	2,500
ADuC7021BCPZ62IRL7[1]	8	2	62 kB/8 kB	13	I²C	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	750
ADuC7021BCPZ32[1]	8	2	32 kB/4 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	
ADuC7021BCPZ32-RL[1]	8	2	32 kB/4 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	2,500
ADuC7021BCPZ32-RL7[1]	8	2	32 kB/4 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	750
ADuC7022BCPZ62[1]	10		62 kB/8 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	
ADuC7022BCPZ62-RL[1]	10		62 kB/8 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	2,500
ADuC7022BCPZ62-RL7[1]	10		62 kB/8 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	750
ADuC7022BCPZ32[1]	10		32 kB/4 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	
ADuC7022BCPZ32-RL[1]	10		32 kB/4 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	2,500
ADuC7022BCPZ32-RL7[1]	10		32 kB/4 kB	13	UART	−40°C to +125°C	40-Lead LFCSP_VQ	CP-40	750
ADuC7024BCPZ62[1]	10	2	62 kB/8 kB	30	UART	−40°C to +125°C	64-Lead LFCSP_VQ	CP-64-1	
ADuC7024BCPZ62-RL[1]	10	2	62 kB/8 kB	30	UART	−40°C to +125°C	64-Lead LFCSP_VQ	CP-64-1	2,500
ADuC7024BCPZ62-RL7[1]	10	2	62 kB/8 kB	30	UART	−40°C to +125°C	64-Lead LFCSP_VQ	CP-64-1	750
ADuC7024BSTZ62[1]	10	2	62 kB/8 kB	30	UART	−40°C to +125°C	64-Lead LQFP	ST-64-2	
ADuC7024BSTZ62-RL[1]	10	2	62 kB/8 kB	30	UART	−40°C to +125°C	64-Lead LQFP	ST-64-2	1,000

Model	ADC Channels	DAC Channels	FLASH/RAM	GPIO	Downloader	Temperature Range	Package Description	Package Option	Ordering Quantity
ADuC7025BCPZ62[1]	12		62 kB/8 kB	30	UART	–40°C to +125°C	64-Lead LFCSP_VQ	CP-64-1	
ADuC7025BCPZ62-RL[1]	12		62 kB/8 kB	30	UART	–40°C to +125°C	64-Lead LFCSP_VQ	CP-64-1	2,500
ADuC7025BCPZ62-RL7[1]	12		62 kB/8 kB	30	UART	–40°C to +125°C	64-Lead LFCSP_VQ	CP-64-1	750
ADuC7025BCPZ32[1]	12		32 kB/4 kB	30	UART	–40°C to +125°C	64-Lead LFCSP_VQ	CP-64-1	
ADuC7025BCPZ32-RL[1]	12		32 kB/4 kB	30	UART	–40°C to +125°C	64-Lead LFCSP_VQ	CP-64-1	2,500
ADuC7025BCPZ32-RL7[1]	12		32 kB/4 kB	30	UART	–40°C to +125°C	64-Lead LFCSP_VQ	CP-64-1	750
ADuC7025BSTZ62[1]	12		62 kB/8 kB	30	UART	–40°C to +125°C	64-Lead LQFP	ST-64-2	
ADuC7025BSTZ62-RL[1]	12		62 kB/8 kB	30	UART	–40°C to +125°C	64-Lead LQFP	ST-64-2	1,000
ADuC7026BSTZ62[1, 3]	12	4	62 kB/8 kB	40	UART	–40°C to +125°C	80-Lead LQFP	ST-80-1	
ADuC7026BSTZ62-RL[1, 3]	12	4	62 kB/8 kB	40	UART	–40°C to +125°C	80-Lead LQFP	ST-80-1	1,000
ADuC7026BSTZ62I[1, 3]	12	4	62 kB/8 kB	40	I^2C	–40°C to +125°C	80-Lead LQFP	ST-80-1	
ADuC7026BSTZ62I-RL[1, 3]	12	4	62 kB/8 kB	40	I^2C	–40°C to +125°C	80-Lead LQFP	ST-80-1	1,000
ADuC7027BSTZ62[1, 3]	16		62 kB/8 kB	40	UART	–40°C to +125°C	80-Lead LQFP	ST-80-1	
ADuC7027BSTZ62-RL[1, 3]	16		62 kB/8 kB	40	UART	–40°C to +125°C	80-Lead LQFP	ST-80-1	1,000
EVAL-ADuC7020MK							ADuC7020 MiniKit		
EVAL-ADuC7020QS							ADuC7020 QuickStart Development System		
EVAL-ADuC7024QS							ADuC7024 QuickStart Development System		
EVAL-ADuC7026QS							ADuC7026 QuickStart Development System		
EVAL-ADuC7026QSP							ADuC7026 QuickStart Plus Development System		

[1] Z = Pb-free part.
[2] One of the ADC channels is internally buffered.
[3] Includes external memory interface.

NOTES

NOTES