ANALOG DEVICES

Precision Analog Microcontroller
ARM7TDMI® MCU with 12-bit ADC & DDS DAC

Preliminary Technical Data

ADuC7128

FEATURES

Analog I/O

- Multi-Channel, 12-bit, 1MSPS ADC
  - 10 ADC channels
  - Fully differential and single-ended modes
  - 0 to V_{REF} Analog Input Range
- 10-bit DAC
  - 32-bit 21MHz DDS
  - Current-to-Voltage (I/V) Conversion
  - Integrated 2nd order LPF
  - DDS Input to DAC
  - 100ohm Line Driver
- On-Chip Voltage Reference
- On-Chip Temperature Sensor (±3°C)
- Uncommitted Voltage Comparator

Microcontroller

- ARM7TDMI Core, 16/32-bit RISC architecture
- JTAG Port supports code download and debug
- External Watch crystal/ Clock Source
  - 41.78 MHz PLL with 8 way Programmable Divider
  - Optional Trimmed On-Chip Oscillator

Memory

- 126k Bytes Flash/EE Memory, 8k Bytes SRAM
- In-Circuit Download, JTAG based Debug
- Software triggered in-circuit re-programmability

On-Chip Peripherals

- 2 x UART, 2 x I^{2}C® and SPI Serial I/O
- 28-Pin GPIO Port
- 5 X General Purpose Timers
- Wake-up and Watchdog Timers
- Power Supply Monitor
- 16-bit PWM generator
- Quadrature Encoder
- PLA – Programmable Logic (Array)

Power

- Specified for 3V operation
- Active Mode: 11mA (@5MHz)
  45mA (@41.78 MHz)

Packages and Temperature Range

- 64 lead LFCSP (9mm x 9mm) package –40°C to 85°C

Tools

- Low-Cost QuickStart Development System
- Full Third-Party Support

FUNCTIONAL BLOCK DIAGRAM

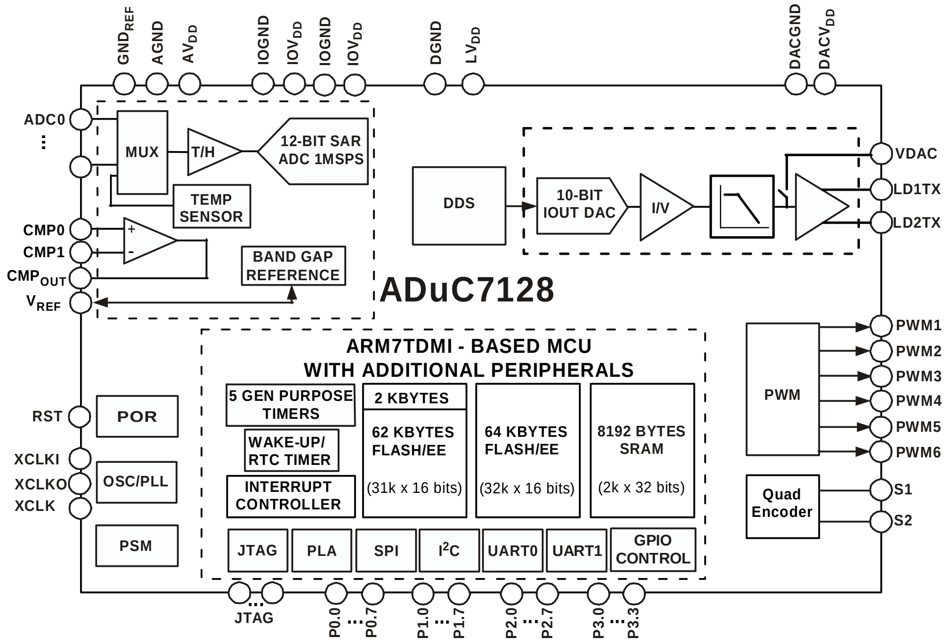


Figure 1. Basic Block Diagram

Rev. PrA

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective companies.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.326.8703

GENERAL DESCRIPTION

The ADuC7128 is a fully integrated, 1MSPS, 12-bit data acquisition system incorporating a high performance multi-channel ADC, DDS with line driver, a 16/32-bit MCU and Flash/EE Memory on a single chip.

The ADC consists of up to 10 single-ended inputs. 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.

The ADuC7128 also integrates a differential line driver output. This line driver transmits a sine wave whose values are calculated by an on chip DDS or a voltage output determined by the DACDAT MMR.

The device operates from an on-chip oscillator and 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, 16/32-bit RISC machine, offering up to 41 MIPS peak performance. 126k Bytes of non-volatile Flash/EE are provided on-chip as well as 8k Bytes of SRAM. 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 and JTAG serial interface ports while non-intrusive 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 3.0V to 3.6V and are specified over an industrial temperature range of -40°C to 85°C. When operating at 41.78 MHz the power dissipation is 150mW. The line driver output if enabled consumes and additional 30mW.

ADUC7128—SPECIFICATIONS

Table 1. (AV_{DD} = IOV_{DD} = 3.0 V to 3.6 V, V_{REF} = 2.5 V Internal Reference, f_{CORE} = 41.78MHz, All specifications T_A = T_{MAX} to T_{MIN}, unless otherwise noted.)

Parameter	ADuC7128	Unit	Test Conditions/Comments
ADC CHANNEL SPECIFICATIONS			
ADC Powerup Time	5	us	
DC Accuracy[1,2]			Eight acquisition clocks and Fadc/2
Resolution	12	Bits	
Integral Nonlinearity	±1.5	LSB max	2.5V internal reference
	±0.6	LSB typ	2.5V internal reference
Integral Nonlinearity[3]	±2.0	LSB typ	1.0V external reference
Differential Nonlinearity	+1/-0.9	LSB max	2.5V internal reference
	±0.5	LSB typ	2.5V internal reference
Differential Nonlinearity[3]	+0.7/-0.6	LSB typ	1.0V external reference
DC Code Distribution	1	LSB typ	ADC input is a dc voltage
ENDPOINT ERRORS[4]			
Offset Error	±5	LSB max	
Offset Error Match	±1	LSB typ	
Gain Error	±5	LSB max	
Gain Error Match	±1	LSB typ	
DYNAMIC PERFORMANCE			Fin = 10kHz Sine Wave, f_{SAMPLE} = 1MSPS
Signal-to-Noise Ratio (SNR)	69	dB typ	
Total Harmonic Distortion (THD)	-78	dB typ	
Peak Harmonic or Spurious Noise	-75	dB typ	
Channel-to-Channel Crosstalk	-80	dB typ	
ANALOG INPUT			
Input Voltage Ranges			
Differential mode[5]	V_{CM}±V_{REF}/2	Volts	
Single-ended mode	0 to V_{REF}	Volts	
Leakage Current	±6	µA max	
	±1	µA typ	
Input Capacitance	20	pF typ	During ADC Acquisition
ON-CHIP VOLTAGE REFERENCE			0.47µF from V_{REF} to AGND
Output Voltage	2.5	V	
Accuracy	±5	mV max	Measured at T_A = 25°C
Reference Temperature Coefficient	±40	ppm/°C typ	
Power Supply Rejection Ratio	75	dB typ	
Output Impedance	70	Ω typ	
Internal V_{REF} Power-On Time	1	ms typ	
EXTERNAL REFERENCE INPUT[6]			
Input Voltage Range	0.625	V min	
	AV_{DD}	V max	
Input Impedance	65	KΩ typ	
DAC CHANNEL SPECIFICATIONS			
VDAC Output			R_L = 5kΩ, C_L = 100pF
Voltage Swing	0.33*V_{REF} ± 0.2*V_{REF}		V_{REF} is the internal 2.5V reference
I/V output resistance	500	Ω max	V mode selected
Low Pass Filter 3db point	1	MHz min	2-pole.
	1.5	MHz typ	
	2	MHz max	
Resolution	10	Bits	

Parameter	ADuC7128	Unit	Test Conditions/Comments
Relative Accuracy	±2	LSB typ	
Differential Nonlinearity, +'ve	0.25	LSB Typ	
Differential Nonlinearity, -'ve	1.5	LSB Typ	
Offset Error	TBD	mV max	
Gain Error	TBD	mV max	DDS Mode
	TBD	mV typ	DAC Mode
Voltage Output Settling Time to 0.1%	600	ns max	
Line Driver Output			As measured into a range of specified loads (see Figure 2) at PL1/LD2TX unless otherwise noted
Total Harmonic Distortion	0.30 TBD	% typ % max	DDS operating at 691.2kHz.
Output Voltage Swing	±1.753 ±1.768 ±1.782	V min RMS V typ RMS V max RMS	
Common Mode	TBD	V typ	AC Mode Each output has a common mode of 0.5*AV_{DD} and swings 0.5*V_{REF} above and below this. V_{REF} is the internal 2.5V reference
	TBD	V typ	DC Mode Each output has a common mode of 0.5*V_{REF} and swings 0.5*V_{REF} above and below this. V_{REF} is the internal 2.5V reference
Differential Input Impedance	10	kΩ min	Line Driver Buffer disabled
	12.5	kΩ typ	
Leakage current LD1TX, LD2TX	5	uA max	Line Driver Buffer disabled
Leakage current LDIN	5	uA max	
Short Circuit Current	±50	mA	
Digital to Analog Glitch Energy	TBD	nVsec typ	I LSB change at major carry
Line Driver Tx Powerup time	20	µs max	
COMPARATOR			
Input Offset Voltage	±15	mV typ	
Input Bias Current	1	µA typ	
Input Voltage Range	AGND to AV_{DD}-1.2	Vmin/Vmax	
Input Capacitance	7	pF typ	
Hysteresis[3,5]	2 15	mV min mv max	Hysteresis can be turned on or off via the CMPHYST bit in the CMPCON register
Response Time	1	µs typ	Response time may be modified via the CMPRES bits in the CMPCON register
TEMPERATURE SENSOR			
Voltage Output at 25°C	780	mV typ	
Voltage TC	-1.3	mV/°C typ	
Accuracy	±3	°C typ	
POWER SUPPLY MONITOR (PSM)			
IOV_{DD} Trip Point Selection	2.79 3.07	V V	Two selectable Trip Points
Power Supply Trip Point Accuracy	±2.5	% typ	Of the selected nominal Trip Point Voltage
Glitch Immunity on RESET Pin[3]	50	µs typ	
Watchdog Timer (WDT)			
Timeout Period	0	ms min	
	512	s max	

Parameter	ADuC7128	Unit	Test Conditions/Comments
Flash/EE MEMORY[7,8]			
Endurance	10,000	Cycles min	
Data Retention	20	Years min	T_J = 85°C
Digital Inputs			All digital inputs including XCLKI and XCLKO
Logic 1 Input Current	±1	µA max	V_{INH} = V_{DD} or V_{INH} = 5V
(leakage Current)	±0.2	µA typ	
Logic 0 Input Current	-60	µA max	V_{INL} = 0V, except TDI
(leakage Current)	-40	µA typ	
	-120	µA max	V_{INL} = 0V, TDI Only
	-80	µA typ	
Input Capacitance	10	pF typ	
Logic Inputs[3]			All Logic inputs including XCLKI and XCLKO
VINL, Input Low Voltage	0.8	V max	
VINH, Input High Voltage	2.0	V min	
Quadrature Encoder Inputs S1/S2/CLR (Schmitt-Triggered Inputs)			
V_{T+}	1.9	V min	
	2.1	V max	
V_{T-}	0.9	V min	
	1.1	V max	
V_{T+} -V_{T-}	0.9	V min	
	1.1	V max	
Logic Outputs[9]			
VOH, Output High Voltage	IOV_{DD} – 400mV	V min	I_{SOURCE} = 1.6mA
VOL, Output Low Voltage	0.4	V max	I_{SINK} = 1.6mA
CRYSTAL INPUTS XCLKI and XCLKO			
Logic Inputs, XCLKI Only			
VINL, Input Low Voltage	1.1	V	
VINH, Input High Voltage	1.7	V	
XCLKI Input Capacitance	20	pF	
XCLKO Output Capacitance	20	pF	
MCU CLOCK RATE (PLL)			8 programmable core clock selections within this range.
	326.4	kHz min	(32.768kHz x 1275)/128
	41.779200	MHz max	(32.768kHz x 1275)/1
INTERNAL OSCILLATOR	32.768	kHz typ	
Tolerance	±3	% max	
STARTUP TIME			Core Clock = 41.78 MHz
At Power-On	TBD		
From Pause/Nap Mode	TBD		
From Sleep Mode	TBD		
From Stop Mode	TBD		
Programmable Logic Array (PLA)			
Pin Propagation Delay	12	ns typ	From input pin to output pin

Parameter	ADuC7128	Unit	Test Conditions/Comments
Element Propagation Delay	2.5	ns typ	
POWER REQUIREMENTS			
Power Supply Voltage Range			
IOV_{DD}, AV_{DD} and $DACV_{DD}$	3.0	V min	
(Supply Voltage to Chip)	3.6	V max	
LV_{DD}	2.5	V min	
(Regulator Output from Chip)	2.6	V typ	
	2.7	V max	
Power Supply Current[10,11]			
Normal Mode	15	mA typ	5.52MHz clock
	17	mA max	5.52MHz clock
	42	mA typ	41.78MHz clock
	45	mA max	41.78MHz clock
Additional Line Driver Tx Supply Current	30	mA max	691kHz, max load (Fig. 2)
Pause Mode	30	mA max	44.2MHz clock
Sleep Mode	250	µA typ	External Crystal or Internal Osc ON
	400	µA max	

[1] All ADC channel specifications are guaranteed during normal MicroConverter core operation.
[2] Apply to all ADC input channels.
[3] Not production tested but supported by design and/or characterization data on production release.
[4] Measured using an external AD845 op amp as an input buffer stage as shown in Figure 38. Based on external ADC system components.
[5] 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.
[6] 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.
[7] Endurance is qualified as per JEDEC Std. 22 method A117 and measured at −40°C, +25°C, and +85°C.
[8] Retention lifetime equivalent at junction temperature (Tj) = 85°C as per JEDEC Std. 22 method A117. Retention lifetime derates with junction temperature.
[9] Test carried out with a maximum of 8 I/O set to a low output level.
[10] 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
[11] IOV_{DD} power supply current decreases typically by 2 mA during a Flash/EE erase cycle.

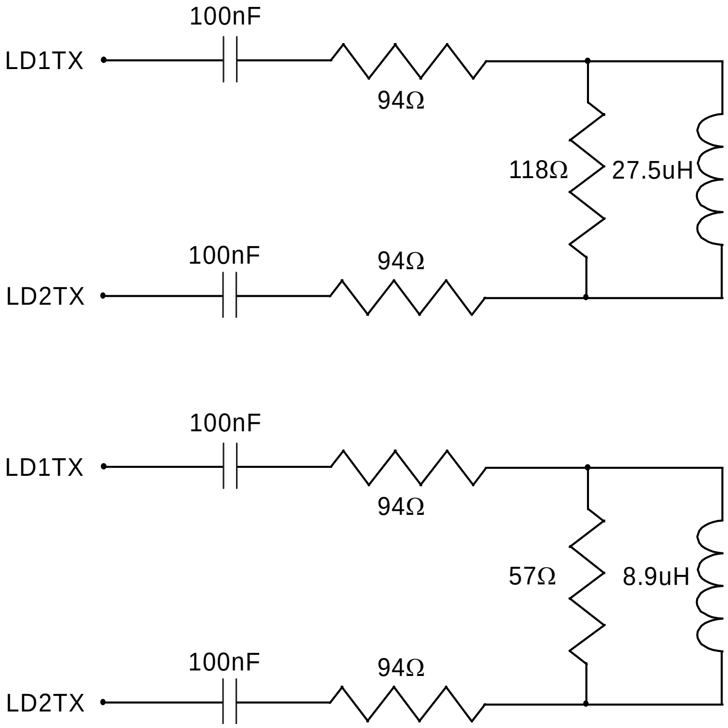


Figure 2. Line Driver Load min (top) and max (bottom)

Table 2. I^2C Timing in Fast Mode (400 kHz)

		Slave			
Parameter	Description	Min	Max	Master Typ	Unit
t_{LOW}	SCLOCK low pulsewidth[1]	200		1360	ns
t_{HIGH}	SCLOCK high pulsewidth[1]	100		1140	ns
$t_{HD;STA}$	Start condition hold time	300		251350	ns
$t_{SU;DAT}$	Data setup time	100		740	ns
$t_{HD;DAT}$	Data hold time	50		400	ns
$t_{SU;STA}$	Setup time for repeated start	100		12.51350	ns
$t_{SU;STO}$	STOP condition setup time	100		400	
t_{BUF}	Bus-free time between a STOP condition and a START condition	1.3			
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}	Pulsewidth of spike suppressed		50		ns

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

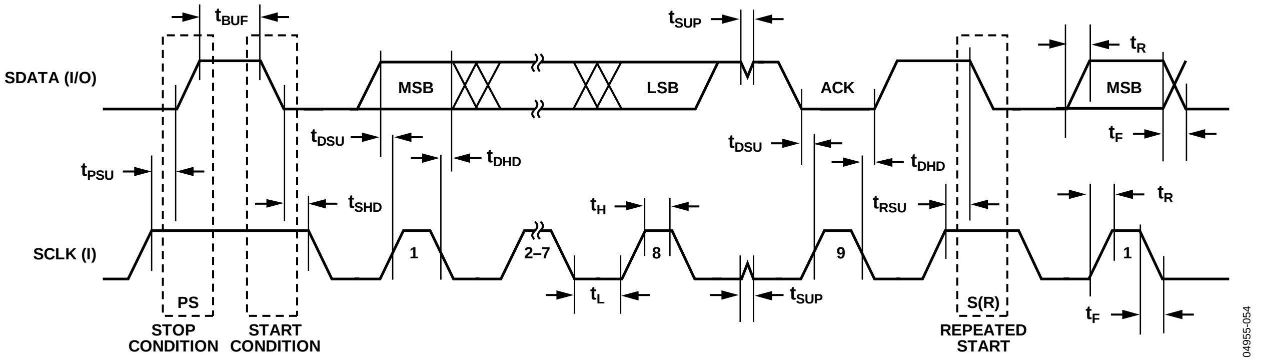


Figure 3. I^2C Compatible Interface Timing

Table 3. SPI Master Mode Timing (PHASE Mode = 1)

Parameter	Description	Min	Typ	Max	Unit
t_{SL}	SCLOCK low pulsewidth[1]		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLOCK high pulsewidth[1]		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge			2 x t_{HCLK} + 2 × t_{UCLK}	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.

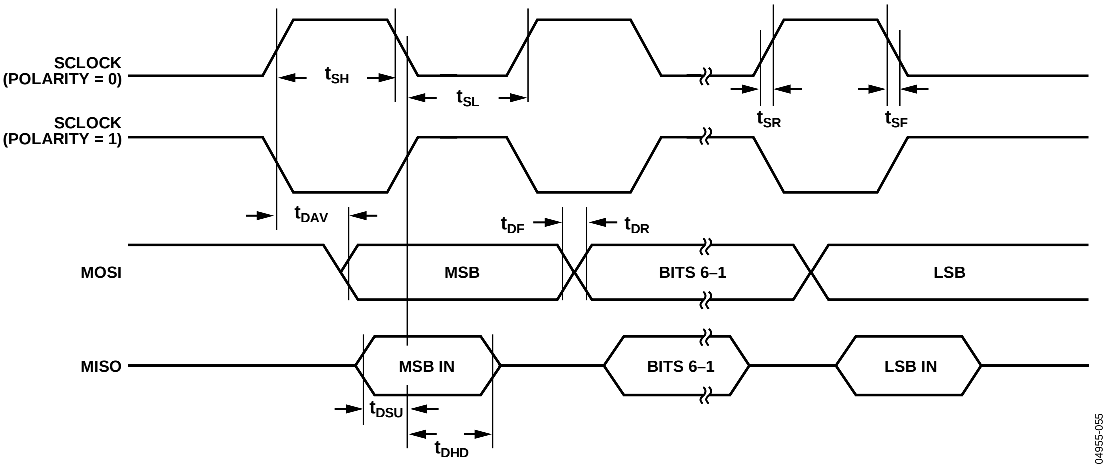


Figure 4. SPI Master Mode Timing (PHASE Mode = 1)

Table 4. SPI Master Mode Timing (PHASE Mode = 0)

Parameter	Description	Min	Typ	Max	Unit
t_{SL}	SCLOCK low pulsewidth[1]		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLOCK high pulsewidth[1]		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge			2 x t_{HCLK} + 2 × t_{UCLK}	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.

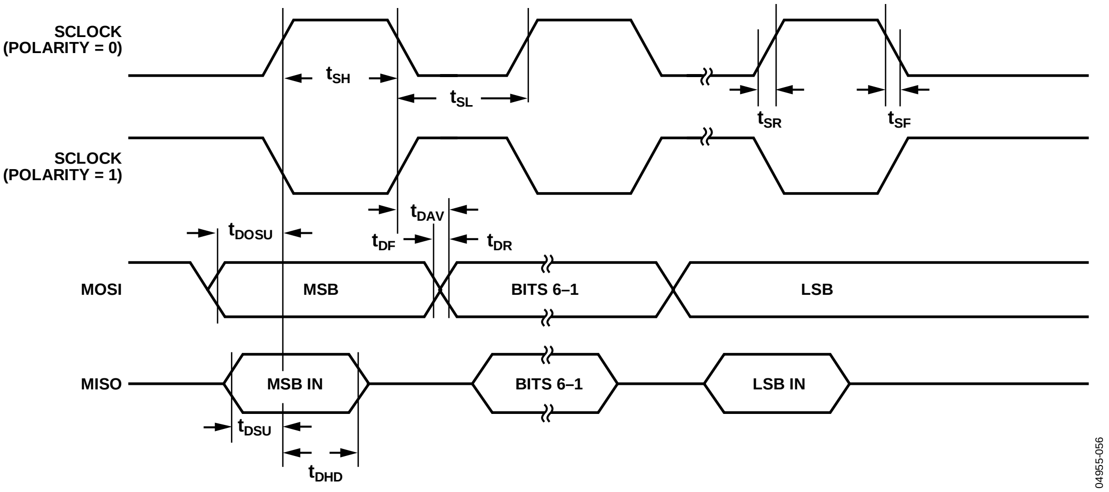


Figure 5. SPI Master Mode Timing (PHASE Mode = 0)

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

Parameter	Description	Min	Typ	Max	Unit
t_{CS}	CS to SCLOCK edge[1]	2 × t_{UCLK}			ns
t_{SL}	SCLOCK low pulsewidth[2]		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLOCK high pulsewidth[2]		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge			2 x t_{HCLK} + 2 × t_{UCLK}	ns
t_{DSU}	Data input setup time before SCLOCK edge[1]	1 × t_{UCLK}			ns
t_{DHD}	Data input hold time after SCLOCK edge[1]	2 × t_{UCLK}			ns
t_{DF}	Data output fall time		5	12.5	ns
t_{DR}	Data output rise time		5	12.5	ns
t_{SR}	SCLOCK rise time		5	12.5	ns
t_{SF}	SCLOCK fall time		5	12.5	ns
t_{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. $T_{HCLK} = t_{UCLK}/2^{CD}$.

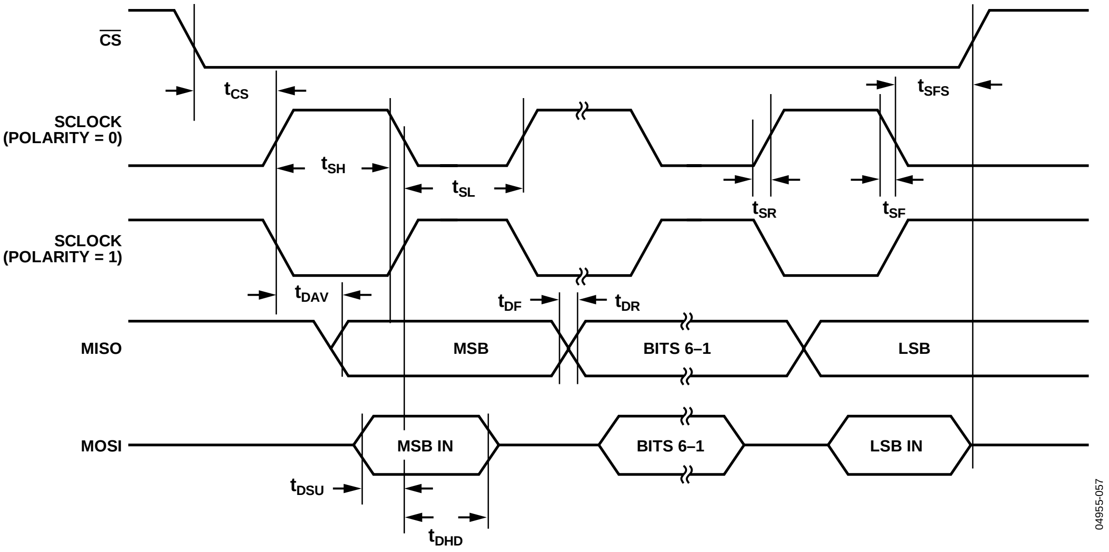


Figure 6. SPI Slave Mode Timing (PHASE Mode = 1)

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

Parameter	Description	Min	Typ	Max	Unit
t_{CS}	CS to SCLOCK edge[1]	2 × t_{UCLK}			ns
t_{SL}	SCLOCK low pulsewidth[2]		(SPIDIV + 1) × t_{HCLK}		ns
t_{SH}	SCLOCK high pulsewidth[2]		(SPIDIV + 1) × t_{HCLK}		ns
t_{DAV}	Data output valid after SCLOCK edge			2 x t_{HCLK} + 2 × t_{UCLK}	ns
t_{DSU}	Data input setup time before SCLOCK edge[1]	1 × t_{UCLK}			ns
t_{DHD}	Data input hold time after SCLOCK edge[1]	2 × t_{UCLK}			ns
t_{DF}	Data output fall time		5	12.5	ns
t_{DR}	Data output rise time		5	12.5	ns
t_{SR}	SCLOCK rise time		5	12.5	ns
t_{SF}	SCLOCK fall time		5	12.5	ns
t_{DOCS}	Data output valid after 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. $T_{HCLK} = t_{UCLK}/2^{CD}$.

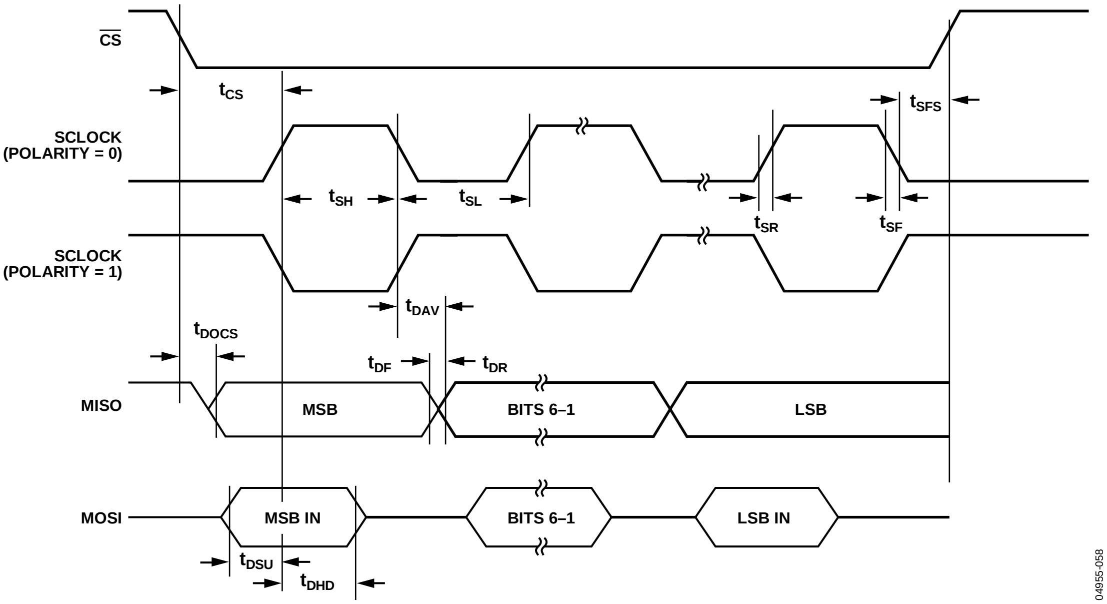


Figure 7. SPI Slave Mode Timing (PHASE Mode = 0)

ABSOLUTE MAXIMUM RATINGS

T_A = 25°C unless otherwise noted. DV$_{DD}$ = IOV$_{DD}$,
AGND = REFGND = DACGND = GND$_{REF}$.

Table 7.

Parameter	Rating
AV$_{DD}$ to DV$_{DD}$	−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 IOV$_{DD}$ + 0.3 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 Output to AGND	−0.3 V to AV$_{DD}$ + 0.3 V
Operating Temperature Range Industrial ADuC7128	–40°C to +85°C
Storage Temperature Range	–65°C to +150°C
Junction Temperature	125°C
θ_{JA} Thermal Impedance (64-pin CSP)	24°C/W
Peak Solder Reflow Temperature	
SnPb Assemblies (10 sec to 30 sec)	240°C
PbFree 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 may 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.

WARNING!

ESD SENSITIVE DEVICE

PIN FUNCTION DESCRIPTIONS - ADUC7128

Pin#	Mnemonic	Type	Function
1	ADC5	I	Single-ended or differential Analog input 5 / Line Driver input
2	VDACout	I	Output from DAC buffer
3	ADC9	I	Single-ended or differential Analog input 9
4	ADC10	I	Single-ended or differential Analog input 10
5	GND$_{REF}$	S	Ground voltage reference for the ADC. For optimal performance the analog power supply should be separated from IOGND and DGND
6	ADCNEG	I	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 0V and 1V
7	AV$_{DD}$	S	Analog Power
8	ADC12/LD1TX	I/O	Single-ended or differential Analog input 12 / DAC differential negative output
9	ADC13/ LD2TX	I/O	Single-ended or differential Analog input 13 / DAC differential Positive output
10	AGND	S	Analog Ground. Ground reference point for the analog circuitry
11	TMS	I	JTAG Test Port Input - Test Mode Select. Debug and download access
12	TDI	I	JTAG Test Port Input – Test Data In. Debug and download access
13	P4.6/SPM10	I.O	General Purpose Input-Output Port 4.6 / Serial Port Mux pin 10
14	P4.7/SPM11	I.O	General Purpose Input-Output Port 4.7 / Serial Port Mux pin 11
15	P0.0/BM/CMP$_{OUT}$	I/O	General Purpose Input-Output Port 0.0 /Boot Mode. The ADuC7128 will enter download mode if BM is low at reset and will execute code if BM is pulled high at reset through a 1kOhm resistor/ Voltage Comparator Output
16	P0.6/T1/MRST	O	General Purpose Output Port 0.6 / Timer 1 Input / Power on reset output
17	TCK	I	JTAG Test Port Input - Test Clock. Debug and download access
18	TDO	O	JTAG Test Port Output - Test Data Out. Debug and download access
19	IOGND	S	Ground for GPIO. Typically connected to DGND
20	IOV$_{DD}$	S	3.3V Supply for GPIO and input of the on-chip voltage regulator.
21	LV$_{DD}$	S	2.5V. Output of the on-chip voltage regulator. Must be connected to a 0.47µF capacitor to DGND
22	DGND	S	Ground for core logic.
23	P3.0/PWM1	I/O	General Purpose Input-Output Port 3.0/ PWM 1 output
24	P3.1/PWM2	I/O	General Purpose Input-Output Port 3.1/ PWM 2 output
25	P3.2/PWM3	I/O	General Purpose Input-Output Port 3.2/ PWM 3 output
26	P3.3/PWM4	I/O	General Purpose Input-Output Port 3.3/ PWM 4 output
27	P0.3/ADC$_{BUSY}$/TRST	I/O	General Purpose Input-Output Port 3.3/ ADC$_{BUSY}$ signal / JTAG Test Port Input – Test Reset. Debug and download access
28	RST	I	Reset Input. (active low)
29	P3.4/PWM5	I/O	General Purpose Input-Output Port 3.4/ PWM 5 output
30	P3.5/PWM6	I/O	General Purpose Input-Output Port 3.5/ PWM 6 output
31	P0.4/IRQ0/CONVST	I/O	General Purpose Input-Output Port 0.5 / External Interrupt Request 0, active high / Start conversion input signal for ADC
32	P0.5/IRQ1/ADC$_{BUSY}$	I/O	General Purpose Input-Output Port 0.6 / External Interrupt Request 1, active high / ADC$_{BUSY}$ signal
33	P2.0/SPM9	I/O	General Purpose Input-Output Port 2.0 / Serial Port Mux pin 9
34	P0.7/SPM8/ECLK/XCLK	I/O	General Purpose Input-Output Port 0.7 / Serial Port Mux pin 8 / Output for External Clock signal/ Input to the internal clock generator circuits
35	XCLKO	O	Output from the crystal oscillator inverter
36	XCLKI	I	Input to the crystal oscillator inverter and input to the internal clock generator circuits
37	PV$_{DD}$	S	2.5V.PLL supply. Must be connected to a 0.1µF capacitor to DGND Should be connected to 2.5V LDO output.

Pin#	Mnemonic	Type	Function
38	DGND	S	Ground for PLL.
39	P1.7/SPM7	I/O	General Purpose Input-Output Port 1.7/Serial Port Mux pin 7
40	P1.6/SPM6	I/O	General Purpose Input-Output Port 1.6/Serial Port Mux pin 6
41	IOGND	S	Ground for GPIO. Typically connected to DGND
42	IOV_{DD}	S	3.3V Supply for GPIO and input of the on-chip voltage regulator.
43	P4.0/S1	I/O	General Purpose Input-Output Port 4.0/ Quadrature Input 1
44	P4.1/S2	I/O	General Purpose Input-Output Port 4.1 / Quadrature Input 2
45	P1.5/SPM5	I/O	General Purpose Input-Output Port 1.5/Serial Port Mux pin 5
46	P1.4/SPM4	I/O	General Purpose Input-Output Port 1.4/Serial Port Mux pin 4
47	P1.3/SPM3	I/O	General Purpose Input-Output Port 1.3/Serial Port Mux pin 3
48	P1.2/SPM2	I/O	General Purpose Input-Output Port 1.2/Serial Port Mux pin 2
49	P1.1/SPM1	I/O	General Purpose Input-Output Port 1.1/Serial Port Mux pin 1
50	P1.0/SPM0	I/O	General Purpose Input-Output Port 1.0/Serial Port Mux pin 0
51	P4.2	I/O	General Purpose Input-Output Port 4.2
52	P4.3/ PWM_{TRIP}	I/O	General Purpose Input-Output Port 4.3/ PWM safety cut off
53	P4.4	I/O	General Purpose Input-Output Port 4.4
54	P4.5	I/O	General Purpose Input-Output Port 4.5
55	V_{REF}	I/O	2.5V internal Voltage Reference. Must be connected to a 0.47uF capacitor when using the internal reference.
56	DACGND	S	Ground for the DAC. Typically connected to AGND
57	AGND	S	Analog Ground. Ground reference point for the analog circuitry
58	AV_{DD}	S	Analog Power
59	$DACV_{DD}$	S	Power Supply for the DAC, This must be supplied with 2.5V. This can be connected to the LDO output.
60	ADC0	I	Single-ended or differential Analog input 0
61	ADC1	I	Single-ended or differential Analog input 1
62	ADC2/CMP0	I	Single-ended or differential Analog input 2/ Comparator positive input
63	ADC3/CMP1	I	Single-ended or differential Analog input 3/ Comparator negative input
64	ADC4	I	Single-ended or differential Analog input 4

TYPICAL PERFORMANCE CHARACTERISTICS

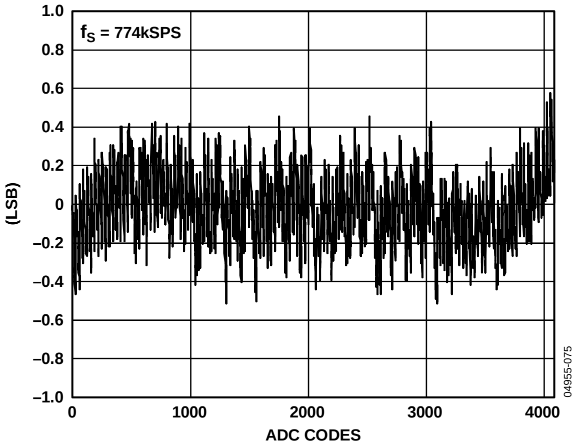


Figure 8. Typical INL Error, f_S = 774 kSPS

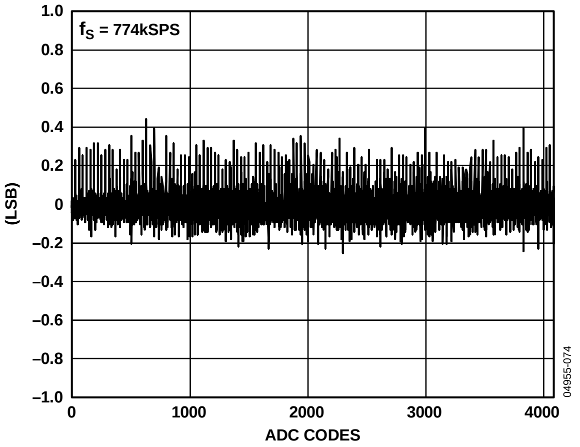


Figure 11. Typical DNL Error, fs = 774 kSPS

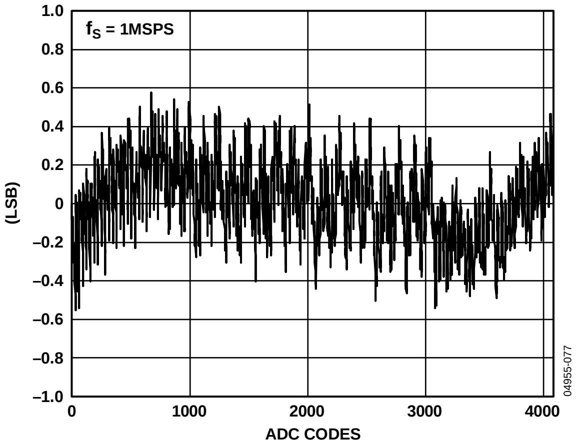


Figure 9. Typical INL Error, f_S = 1 MSPS

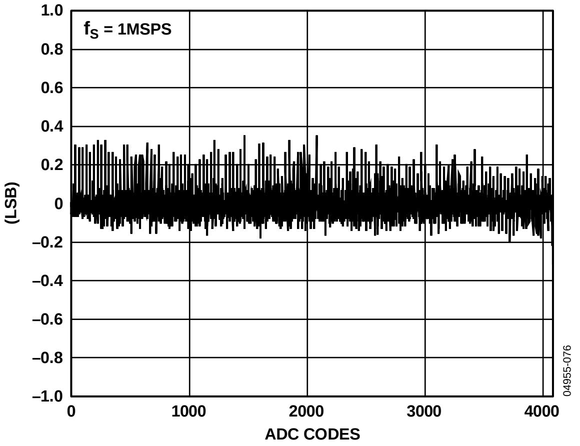


Figure 12. Typical DNL Error, f_S = 1 MSPS

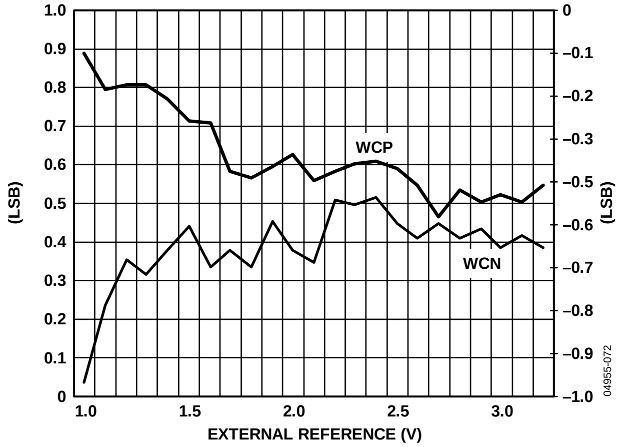


Figure 10. Typical Worse Case INL Error vs. V_{REF}, f_S = 774 kSPS

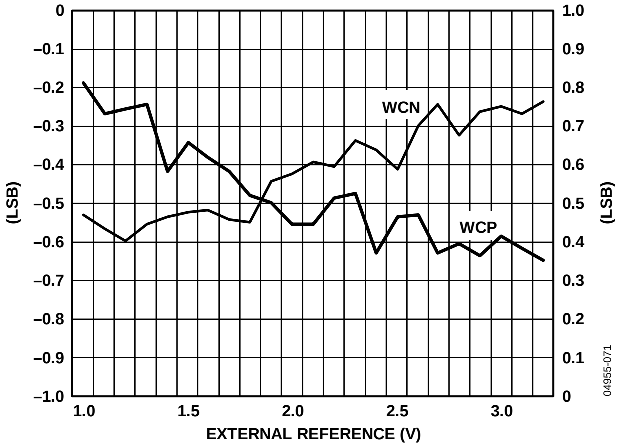


Figure 13. Typical Worse Case DNL Error vs. V_{REF}, f_S = 774 kSPS

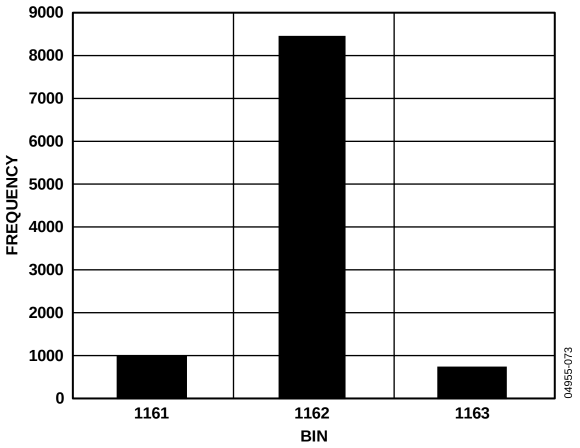


Figure 14. Code Histogram Plot

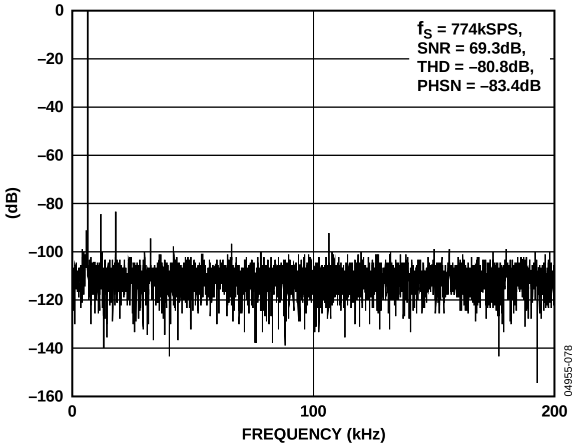


Figure 15. Dynamic Performance, f_S = 774 kSPS

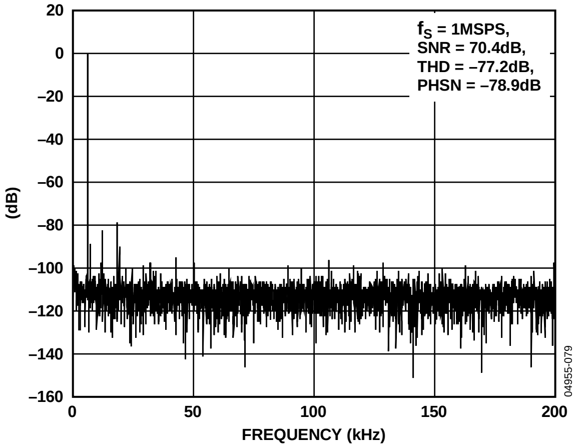


Figure 16. Dynamic Performance, f_S = 1 MSPS

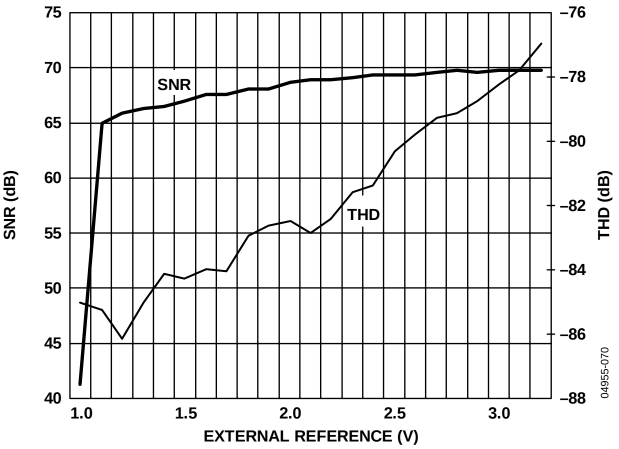


Figure 17. Typical Dynamic Performance vs. V_{REF}

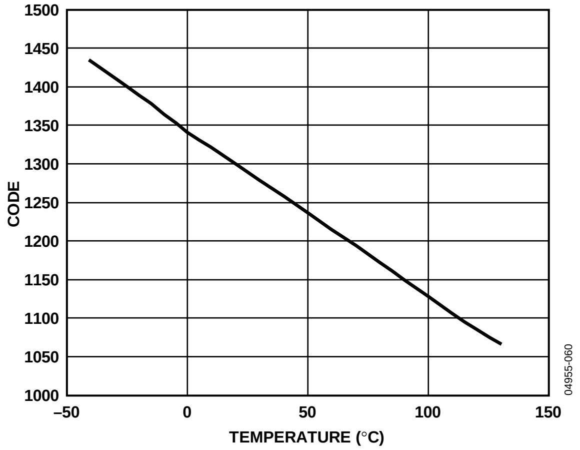


Figure 18. On-Chip Temperature Sensor Voltage Output vs. Temperature

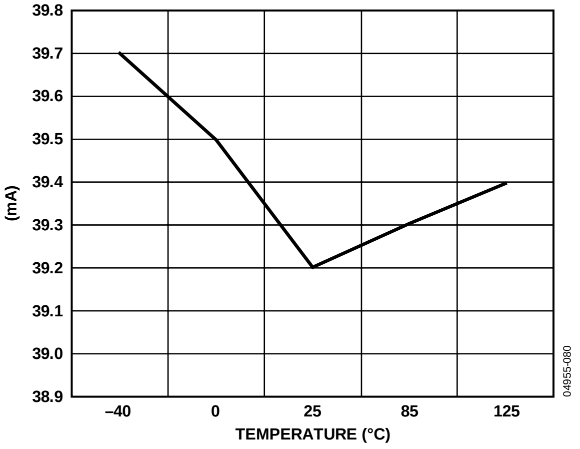


Figure 19. Current Consumption vs. Temperature @ CD = 0

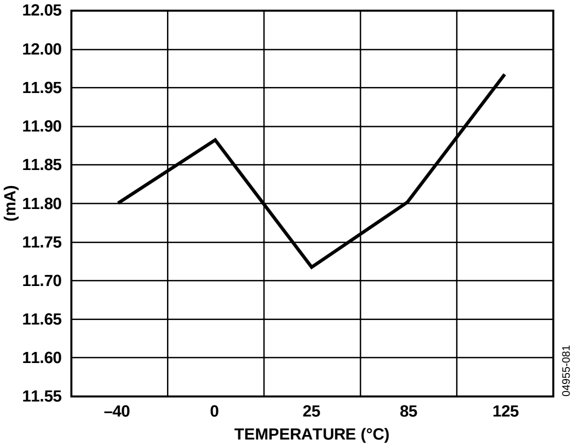


Figure 20. Current Consumption vs. Temperature @ CD = 3

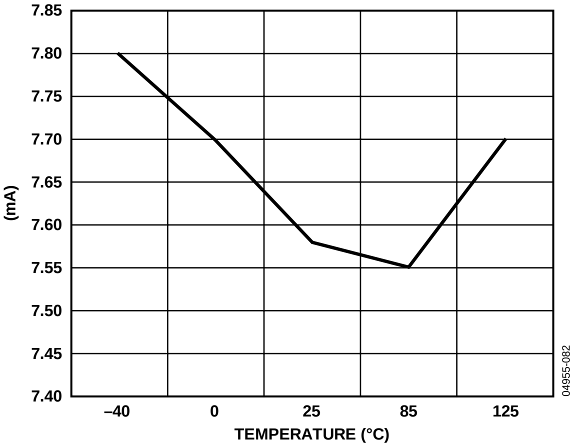


Figure 21. Current Consumption vs. Temperature@t CD = 7

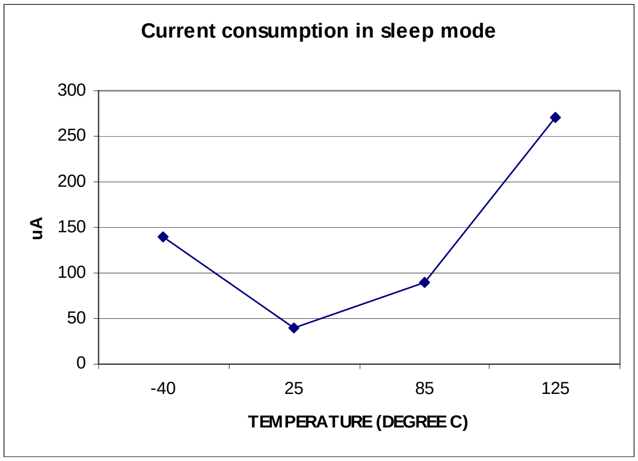


Figure 22. Current Consumption vs. Temperature in Sleep Mode

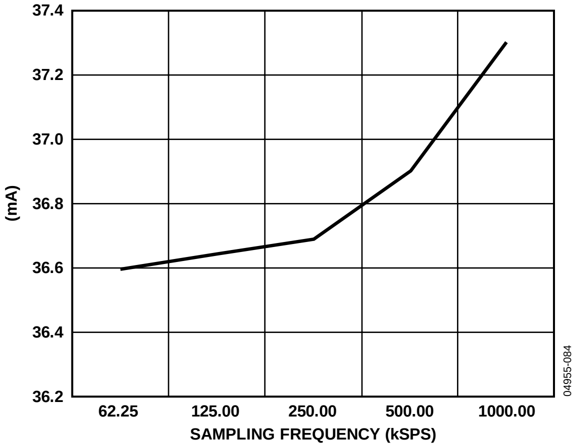


Figure 23. Current Consumption vs. ADC Speed

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 (fS/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:

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

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, 16 or 32 bits. The length of the instruction word is 32 bits.

The ARM7TDMI is an ARM7 core with 4 additional features:

- T support for the Thumb (16 bit) instruction set.
- D support for debug
- M support for long multiplies
- I include 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:

1. Thumb code usually uses more instructions for the same job. As a result, ARM code is usually best for maximising the performance of the time-critical code.
2. 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. This result is 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 de-bugging 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:

1. Normal interrupt or IRQ. This is provided to service general-purpose interrupt handling of internal and external events.
2. Fast interrupt or FIQ. This is provided to service data transfer or communication channel with low latency. FIQ has priority over IRQ.
3. Memory abort.
4. Attempted execution of an undefined instruction.
5. 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 24. 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.

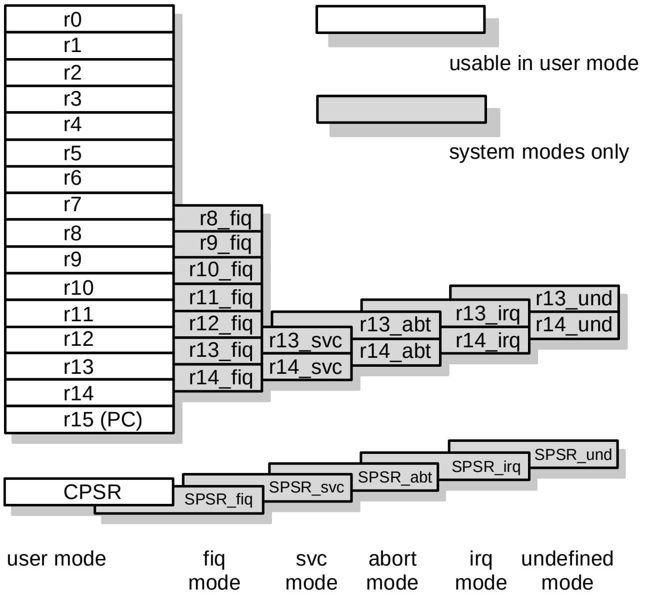


Figure 24: register organisation

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 an FIQ consists of the longest time the request can take to pass through the synchronizer, plus the time for the longest instruction to complete (the longest instruction is an LDM) which loads all the registers including the PC, plus the time for the data abort entry, plus the time for FIQ entry. At the end of this time, the ARM7TDMI will be executing the instruction at 0x1C (FIQ interrupt vector address). The maximum total time is 50 processor cycles, which is just over 1.1µS in a system using a continuous 41.78 MHz processor clock. The maximum IRQ latency calculation is similar, but must allow for the fact that FIQ has higher priority and could delay entry into the IRQ handling routine for an arbitrary length of time. This time 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 this is reduced to 22 cycles.

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

Note that the ARM7TDMI will always be run in ARM (32-bit) mode when in privileged modes, i.e. when executing interrupt service routines.

Memory organisation

The part incorporates three separate blocks of memory, 8kByte of SRAM and two 64kByte of On-Chip Flash/EE memory. 126kByte of On-Chip Flash/EE memory are available to the user, and the remaining 2kBytes are reserved for the factory configured boot page. These two blocks are mapped as shown in Figure 25.

Note that by default, after a reset, the Flash/EE memory is mirrored at address 0x00000000. It is possible to remap the SRAM at address 0x00000000 by clearing bit 0 of the REMAP MMR. This remap function is described in more details in the Flash/EE memory chapter.

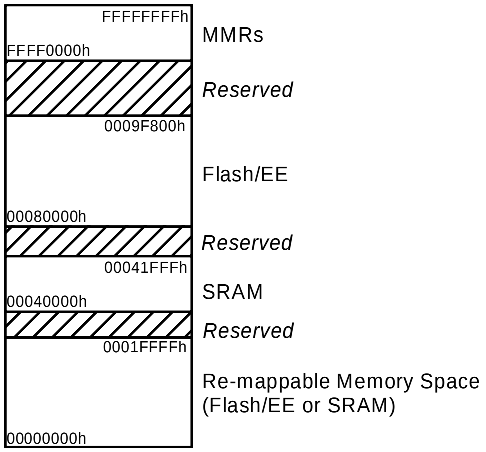


Figure 25: Physical memory map

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 25.

The ADuC7128 memory organisation is configured in little endian format: the least significant byte is located in the lowest byte address and the most significant byte in the highest byte address.

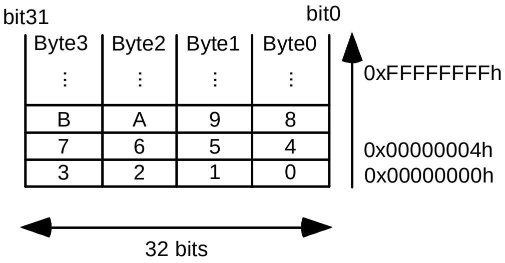


Figure 26: little endian format

Flash/EE Memory

The 128kBytes of Flash/EE are organised as two banks of 32k X 16 bits. In the first block 31k X 16 bits are user space and 1k X 16 bits is reserved for the factory configured boot page.. The page size of this Flash/EE memory is 512Bytes.

The second 64kByte block is organized in a similar manner. It is arranged in 32k x 16 bits. All of this is available as user space.

126 kBytes of Flash/EE are available to the user as code and non-volatile data memory. There is no distinction between data and program as ARM code shares the same space. The real width of the Flash/EE memory is 16 bits, 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 41.78MHz in Thumb mode and 20.89MHz in full ARM mode. More details on Flash/EE access time are outlined later in 'Execution from SRAM and Flash/EE' section of this datasheet.

SRAM

8kBytes of SRAM are available to the user, organized as 2k X 32 bits, i.e. 2kWords. ARM code can run directly from SRAM at 41.78MHz , given that the SRAM array is configured as a 32-bit wide memory array. More details on SRAM access time are outlined later in 'Execution from SRAM and Flash/EE' section of this datasheet.

Memory Mapped Registers

The Memory Mapped Register (MMR) space is mapped into the upper 2 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 6 are unoccupied or reserved locations and should not be accessed by user software. Table 8 shows a full MMR memory map.

The access time reading or writing a MMR depends on the advanced microcontroller bus architecture (AMBA) bus used to access the peripheral. The processor has two AMBA 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 ADuC7128 are on the APB except the Flash/EE memory and the GPIOs.

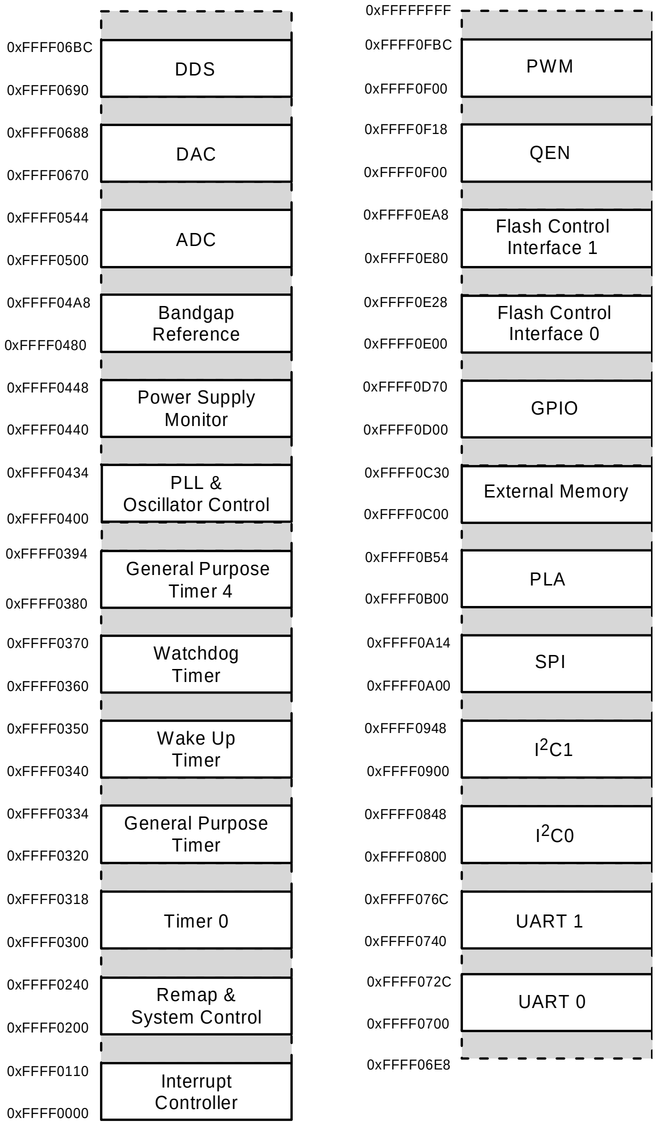


Figure 27: Memory Mapped

Table 8. Complete MMRs list

Address	Name	Byte	Access		Page
			Type	Cycle	
IRQ address base = 0xFFFF0000					
0x0000	IRQSTA	4	R	1	
0x0004	IRQSIG	4	R	1	
0x0008	IRQEN	4	RW	1	
0x000C	IRQCLR	4	W	1	
0x0010	SWICFG	4	W	1	
0x0100	FIQSTA	4	R	1	
0x0104	FIQSIG	4	R	1	
0x0108	FIQEN	4	RW	1	
0x010C	FIQCLR	4	W	1	
System Control address base = 0xFFFF0200					
0x0220	REMAP	1	RW	1	
0x0230	RSTSTA	1	R	1	
0x0234	RSTCLR	1	W	1	
Timer address base = 0xFFFF0300					
0x0300	T0LD	2	RW	2	
0x0304	T0VAL0	2	R	2	
0x0308	T0VAL1	4	R	2	
0x030C	T0CON	4	RW	2	
0x0310	T0ICLR	1	W	2	
0x0314	T0CAP	2	R	2	
0x0320	T1LD	4	RW	2	
0x0324	T1VAL	4	R	2	
0x0328	T1CON	4	RW	2	
0x032C	T1ICLR	1	W	2	
0x0330	T1CAP	4	R	2	
0x0340	T2LD	4	RW	2	
0x0344	T2VAL	4	R	2	
0x0348	T2CON	4	RW	2	
0x034C	T2ICLR	1	W	2	
0x0360	T3LD	2	RW	2	
0x0364	T3VAL	2	R	2	
0x0368	T3CON	2	RW	2	
0x036C	T3ICLR	1	W	2	
0x0380	T4LD	4	RW	2	
0x0384	T4VAL	4	R	2	
0x0388	T4CON	4	RW	2	
0x038C	T4ICLR	1	W	2	
0x0390	T4CAP	4	R	2	
PLL base address = 0xFFFF0400					
0x0404	POWKEY1	2	W	2	
0x0408	POWCON	2	RW	2	
0x040C	POWKEY2	2	W	2	
0x0410	PLLKEY1	2	W	2	
0x0414	PLLCON	2	RW	2	
0x0418	PLLKEY2	2	W	2	
PSM address base = 0xFFFF0440					
0x0440	PSMCON	2	RW	2	
0x0444	CMPCON	2	RW	2	
Reference address base = 0xFFFF0480					
0x048C	REFCON	1	RW	2	
ADC address base = 0xFFFF0500					
0x0500	ADCCON	2	RW	2	
0x0504	ADCCP	1	RW	2	
0x0508	ADCCN	1	RW	2	
0x050C	ADCSTA	1	R	2	
0x0510	ADCDAT	4	R	2	
0x0514	ADCRST	1	W	2	
0x0530	ADCGN	2	RW	2	
0x0534	ADCOF	2	RW	2	
DAC and **DDS** address base = 0xFFFF0670					
0x0670	DACCON	2	RW	2	
0x0690	DDSCON	1	RW	2	
0x0694	DDSFRQ	4	RW	2	
0x0698	DDSPHS	2	RW	2	
0x06A4	DACKEY0	1	RW	2	
0x06B4	DACDAT	2	RW	2	
0x06B8	DACEN	1	RW	2	
0x06BC	DACKEY1	1	RW	2	
UART 0 base address = 0xFFFF0700					
0x0700	COM0TX	1	RW	2	
	COM0RX	1	R	2	
	COM0DIV0	1	RW	2	
0x0704	COM0IEN0	1	RW	2	
	COM0DIV1	1	R/W	2	
0x0708	COM0IID0	1	R	2	

Address	Name	Byte	Access		Page
			Type	Cycle	
0x070C	COM0CON0	1	RW	2	
0x0710	COM0CON1	1	RW	2	
0x0714	COM0STA0	1	R	2	
0x0718	COM0STA1	1	R	2	
0x071C	COM0SCR	1	RW	2	
0x0720	COM0IEN1	1	RW	2	
0x0724	COM0IID1	1	R	2	
0x0728	COM0ADR	1	RW	2	
0X072C	COM0DIV2	2	RW	2	
UART 1 base address = 0xFFFF0740					
0x0740	COM1TX	1	RW	2	
	COM1RX	1	R	2	
	COM1DIV0	1	RW	2	
0x0744	COM1IEN0	1	RW	2	
	COM1DIV1	1	R/W	2	
0x0748	COM1IID0	1	R	2	
0x074C	COM1CON0	1	RW	2	
0x0750	COM1CON1	1	RW	2	
0x0754	COM1STA0	1	R	2	
0x0758	COM1STA1	1	R	2	
0x075C	COM1SCR	1	RW	2	
0x0760	COM1IEN1	1	RW	2	
0x0764	COM1IID1	1	R	2	
0x0768	COM1ADR	1	RW	2	
0X076C	COM1DIV2	2	RW	2	
I2C0 base address = 0xFFFF0800					
0x0800	I2C0MSTA	1	R	2	
0x0804	I2C0SSTA	1	R	2	
0x0808	I2C0SRX	1	R	2	
0x080C	I2C0STX	1	W	2	
0x0810	I2C0MRX	1	R	2	
0x0814	I2C0MTX	1	W	2	
0x0818	I2C0CNT	1	RW	2	
0x081C	I2C0ADR	1	RW	2	
0x0824	I2C0BYT	1	RW	2	
0x0828	I2C0ALT	1	RW	2	
0x082C	I2C0CFG	1	RW	2	
0x0830	I2C0DIV	2	RW	2	
0x0838	I2C0ID0	1	RW	2	
0x083C	I2C0ID1	1	RW	2	
0x0840	I2C0ID2	1	RW	2	
0x0844	I2C0ID3	1	RW	2	
0x0848	I2C0SSC	1	RW	2	
0x084C	I2C0FIF	1	RW	2	
I2C1 base address = 0xFFFF0900					
0x0900	I2C1MSTA	1	R	2	
0x0904	I2C1SSTA	1	R	2	
0x0908	I2C1SRX	1	R	2	
0x090C	I2C1STX	1	W	2	
0x0910	I2C1MRX	1	R	2	
0x0914	I2C1MTX	1	W	2	
0x0918	I2C1CNT	1	RW	2	
0x091C	I2C1ADR	1	RW	2	
0x0924	I2C1BYT	1	RW	2	
0x0928	I2C1ALT	1	RW	2	
0x092C	I2C1CFG	1	RW	2	
0x0930	I2C1DIV	2	RW	2	
0x0938	I2C1ID0	1	RW	2	
0x093C	I2C1ID1	1	RW	2	
0x0940	I2C1ID2	1	RW	2	
0x0944	I2C1ID3	1	RW	2	
0x0948	I2C1SSC	1	RW	2	
0x094C	I2C1FIF	1	RW	2	
SPI base address = 0xFFFF0A00					
0x0A00	SPISTA	1	R	2	
0x0A04	SPIRX	1	R	2	
0x0A08	SPITX	1	W	2	
0x0A0C	SPIDIV	1	RW	2	
0x0A10	SPICON	2	RW	2	
PLA base address = 0xFFFF0B00					
0x0B00	PLAELM0	2	RW	2	
0x0B04	PLAELM1	2	RW	2	
0x0B08	PLAELM2	2	RW	2	
0x0B0C	PLAELM3	2	RW	2	
0x0B10	PLAELM4	2	RW	2	
0x0B14	PLAELM5	2	RW	2	

Address	Name	Byte	Access		Page
			Type	Cycle	
0x0B18	PLAELM6	2	RW	2	
0x0B1C	PLAELM7	2	RW	2	
0x0B20	PLAELM8	2	RW	2	
0x0B24	PLAELM9	2	RW	2	
0x0B28	PLAELM10	2	RW	2	
0x0B2C	PLAELM11	2	RW	2	
0x0B30	PLAELM12	2	RW	2	
0x0B34	PLAELM13	2	RW	2	
0x0B38	PLAELM14	2	RW	2	
0x0B3C	PLAELM15	2	RW	2	
0x0B40	PLACLK	1	RW	2	
0x0B44	PLAIRQ	4	RW	2	
0x0B48	PLAADC	4	RW	2	
0x0B4C	PLADIN	4	RW	2	
0x0B50	PLAOUT	4	R	2	
GPIO base address = 0xFFFF0D00					
0x0D00	GP0CON	4	RW	1	
0x0D04	GP1CON	4	RW	1	
0x0D08	GP2CON	4	RW	1	
0x0D0C	GP3CON	4	RW	1	
0x0D10	GP4CON	4	RW	1	
0x0D20	GP0DAT	4	RW	1	
0x0D24	GP0SET	1	W	1	
0x0D28	GP0CLR	1	W	1	
0x0D2C	GP0PAR	4	RW	1	
0x0D30	GP1DAT	4	RW	1	
0x0D34	GP1SET	1	W	1	
0x0D38	GP1CLR	1	W	1	
0x0D3C	GP1PAR	4	RW	1	
0x0D40	GP2DAT	4	RW	1	
0x0D44	GP2SET	1	W	1	
0x0D48	GP2CLR	1	W	1	
0x0D50	GP3DAT	4	RW	1	
0x0D54	GP3SET	1	W	1	
0x0D58	GP3CLR	1	W	1	
0x0D5C	GP3PAR	4	RW	1	
0x0D60	GP4DAT	4	RW	1	
0x0D64	GP4SET	1	W	1	

Address	Name	Byte	Access		Page
			Type	Cycle	
0x0D68	GP4CLR	1	W	1	
0x0D6C	GP4PAR	1	W	1	
Flash/EE Block 0 base address = 0xFFFF0E00					
0x0E00	FEE0STA	1	R	1	
0x0E04	FEE0MOD	1	RW	1	
0x0E08	FEE0CON	1	RW	1	
0x0E0C	FEE0DAT	2	RW	1	
0x0E10	FEE0ADR	2	RW	1	
0x0E18	FEE0SGN	3	R	1	
0x0E1C	FEE0PRO	4	RW	1	
0x0E20	FEE0HID	4	RW	1	
Flash/EE Block 1 base address = 0xFFFF0E80					
0x0E80	FEE1STA	1	R	1	
0x0E84	FEE1MOD	1	RW	1	
0x0E88	FEE1CON	1	RW	1	
0x0E8C	FEE1DAT	2	RW	1	
0x0E90	FEE1ADR	2	RW	1	
0x0E98	FEE1SGN	3	R	1	
0x0E9C	FEE1PRO	4	RW	1	
0x0EA0	FEE1HID	4	RW	1	
QEN base address= 0xFFFF0F00					
0x0F00	QENCON	2	RW	2	
0x0F04	QENSTA	1	R	2	
0x0F08	QENDAT	2	RW	2	
0x0F0C	QENVAL	2	R	2	
0x0F14	QENCLR	1	W	2	
0x0F18	QENSET	1	W	2	
PWM base address= 0xFFFF0F80					
0x0F80	PWMCON1	2	RW	2	
0x0F84	PWM1COM1	2	RW	2	
0x0F88	PWM1COM2	2	RW	2	
0x0F8C	PWM1COM3	2	RW	2	
0x0F90	PWM1LEN	2	RW	2	
0x0F94	PWM2COM1	2	RW	2	
0x0F98	PWM2COM2	2	RW	2	
0x0F9C	PWM2COM3	2	RW	2	
0x0FA0	PWM2LEN	2	RW	2	
0x0FA4	PWM3COM1	2	RW	2	

Address	Name	Byte	Access		Page
			Type	Cycle	
0x0FA8	PWM3COM2	2	RW	2	
0x0FAC	PWM3COM3	2	RW	2	
0x0FB0	PWM3LEN	2	RW	2	
0x0FB4	PWMCON2	2	RW	2	
0x0FB8	PWMICLR	2	W	2	

The ‘Access’ column corresponds to the access time reading or writing a MMR. It depends on the AMBA (Advanced Microcontroller Bus Architecture) bus used to access the peripheral. The processor has two AMBA busses, AHB (Advanced High-performance Bus) used for system modules and APB (Advanced Peripheral Bus) used for lower performance peripheral.

ADC CIRCUIT INFORMATION

GENERAL OVERVIEW

The Analog Digital Converter (ADC) incorporates a fast, multi-channel, 12-bit ADC. It can operate from 3.0V to 3.6V supplies and is capable of providing a throughput of up to 1MSPS when the clock source is 41.78MHz. This block provides the user with multi-channel 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

1. Fully differential mode, for small and balanced signals.
2. Single-ended mode, for any single-ended signals.
3. 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 0V to AV_{DD} and with a maximum amplitude of 2 V_{REF} (see Figure 28).

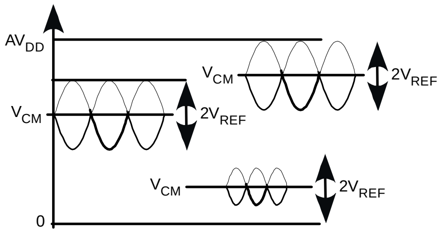


Figure 28: examples of balanced signals for fully differential mode

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

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

If the signal has not been de asserted by the time the ADC conversion is complete then a second conversion will begin automatically.

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

ADC TRANSFER FUNCTION

Pseudo-differential and single-ended modes

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

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

The ideal code transitions occur midway between successive integer LSB values (i.e. 1/2 LSB, 3/2 LSBs, 5/2 LSBs, . . ., FS –3/2 LSBs). The ideal input/output transfer characteristic is shown in Figure 29.

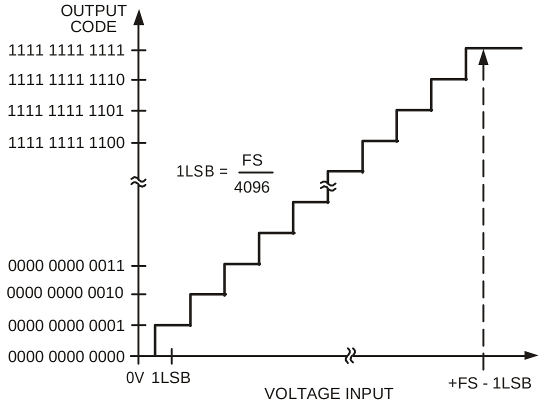


Figure 29: ADC transfer function in pseudo differential mode or single-ended mode

Fully differential mode

The amplitude of the differential signal is the difference between the signals applied to the V_{IN+} and V_{IN-} pins (i.e., V_{IN+} – V_{IN-}). The maximum amplitude of the differential signal is therefore –V_{REF} to +V_{REF} p-p (i.e. 2 X V_{REF}). This is regardless of the common mode (CM). The common mode is the average of the two signals, i.e. (V_{IN+} + V_{IN-})/2 and is therefore the voltage that the two inputs are centred 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 driving the ADC).

The output coding is two's complement in fully differential mode with 1 LSB = $2V_{REF}$/4096 or 2x2.5 V/4096 = 1.22 mV when V_{REF} = 2.5 V. The output result is +/- 11 bits but this is shifted by one to the right. This allows the result in ADCDAT to be declared as a signed integer when writing 'c' code. The designed code transitions occur midway between successive integer LSB values (i.e., 1/2 LSB, 3/2 LSBs, 5/2 LSBs, . . ., FS –3/2 LSBs). The ideal input/output transfer characteristic is shown in Figure 30.

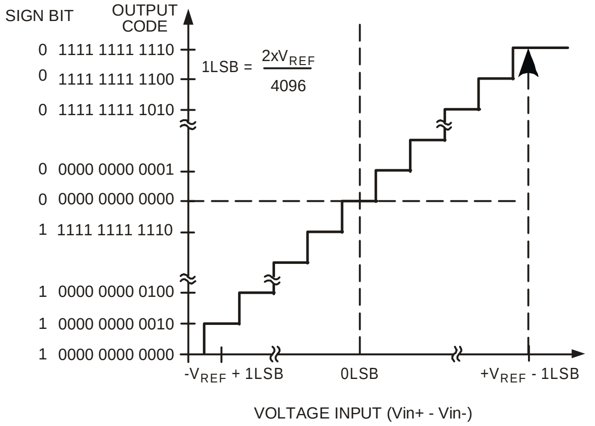


Figure 30: ADC transfer function in differential mode

TYPICAL OPERATION

Once configured via the ADC control and channel selection registers, the ADC will convert the analog input and provide a 11-bit result in the ADC data register.

The top 4 bits are the sign bits and the 11-bit result is placed from bit 16 to 27 as shown in Figure 31. Again, it should be noted that in fully differential mode, the result is represented in two's complement format shifted one bit to the right , and in pseudo differential and single-ended mode, the result is represented in straight binary format.

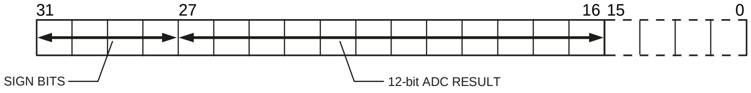


Figure 31: ADC Result Format

Timing

Figure 32 gives details of the ADC timing. Users have control on the ADC clock speed and on the number of acquisition clock in the ADCCON MMR. By default, the acquisition time is eight clocks and the clock divider is two. The number of extra clocks (such as bit trial or write) is set to 19, which gives a sampling rate of 819 kSPS. For conversion on temperature sensor, the ADC acquisition time is automatically set to 16 clocks and the ADC clock divider to 32.

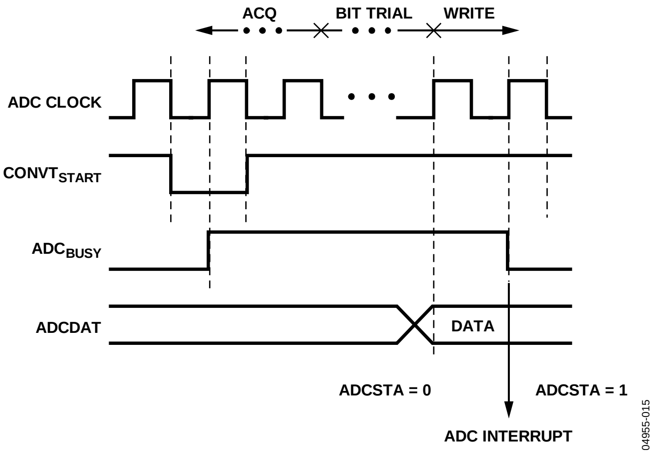


Figure 32. ADC Timing

ADC MMRS interface

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

- **ADCCON:** ADC Control Register allows the programmer to enable the ADC peripheral, to select the mode of operation of the ADC, either Single-ended, pseudo-differential or fully differential mode and the conversion type. This MMR is described Table 9.
- **ADCCP:** ADC positive Channel selection Register
- **ADCCN:** ADC negative Channel selection Register

  ADCSTA: ADC Status Register, indicates when an ADC conversion result is ready. The ADCSTA register contains only one bit, ADCReady, bit (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 ADCBusy pin. This pin is high during a conversion. When the conversion is finished, ADCBusy goes back low. This information can be available on P0.5 (see chapter on GPIO) if enabled in GP0CON register.

  ADCDAT: ADC Data Result Register, hold the 12-bit ADC result as shown Figure 31
- **ADCRST:** ADC Reset Register. Resets all the ADC registers to their default value.

Table 9: ADCCON MMR Bit Designations

Bit	Description
12-10	ADC Speed (Fadc = Fcore, conversion = 14 ADC clocks + Acquisition time) 000 – Fadc / 1 001 – Fadc / 2 010 – Fadc / 4 011 – Fadc / 8 100 – Fadc / 16 101 – Fadc / 32
9-8	ADC Acquisition Time (number of ADC clocks) 00 – 2 01 – 4 10 – 8 11 – 16
7	Enable Conversion *Set* by the user to enable conversion mode *Cleared* by the user to disable conversion mode
6	*Reserved* *This bit should be set to 0 by the user.*
5	ADC power control: *Set* by the user to place the ADC in normal mode, the ADC must be powered up for at least 500uS before it will convert 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 $\text{CONV}_{\text{START}}$ pin as a conversion input 001 Enable timer 1 as a conversion input 010 Enable timer 0 as a conversion input 011 Single software conversion, will be set 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 CONVSTART pin).

	100	Continuous software conversion
	101	PLA conversion
	110	PWM conversion
	Other	*Reserved*

Table 10: ADCCP* MMR bit designation

Bit	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	ADC12/LD2TX
	01101	ADC13/LD1TX
	01110	Reserved
	01111	Reserved
	10000	Temperature sensor
	10001	AGND
	10010	Reference
	10011	AVDD/2
	Others	*Reserved*

Since ADC12 and ADC13 are shared with the Line Driver TX pins a high level of crosstalk will be seen on these pins when used in ADC mode.

Table 11: ADCCN* MMR bit designation

Bit	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	ADC12/LD2TX
	01101	ADC13/LD1TX
	01110	Reserved
	01111	Reserved
	10000	Temperature sensor
	Others	*Reserved*

* ADC channel availability depends on part model.

CONVERTER OPERATION

The ADC incorporates a successive approximation (SAR) architecture involving a charge-sampled input stage. This architecture is described below for the three different modes of operation.

Differential mode

The ADuC7128 contains a successive approximation ADC based on two capacitive DACs. Figure 33 and Figure 34 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 33 (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.

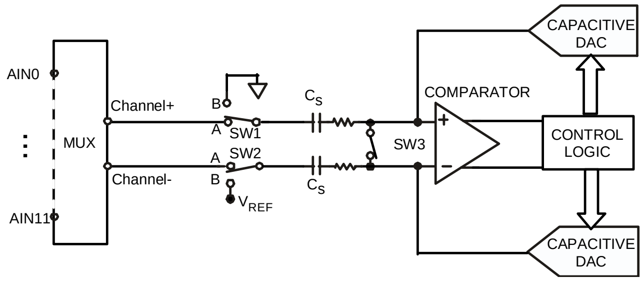


Figure 33: ADC acquisition phase

When the ADC starts a conversion (Figure 34), SW3 will open and SW1 and SW2 will move to Position B, causing 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 VIN+ and VIN– pins must be matched; otherwise, the two inputs will have different settling times, resulting in errors.

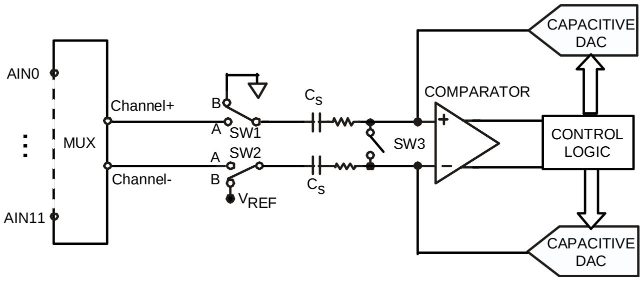


Figure 34: ADC conversion phase

Pseudo-differential mode

In pseudo-differential mode, Channel- is linked to the VIN- pin of the ADuC7128 and SW2 switches between A (Channel-) and B (VREF). VIN- 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 V_{IN}- must be chosen so that V_{REF} + V_{IN}- does not exceed AV_{DD}.

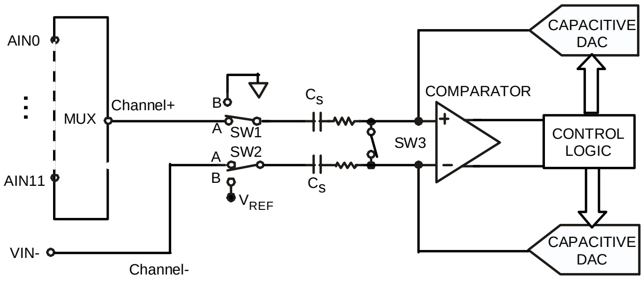


Figure 35: ADC in pseudo-differential mode

Single-ended mode

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

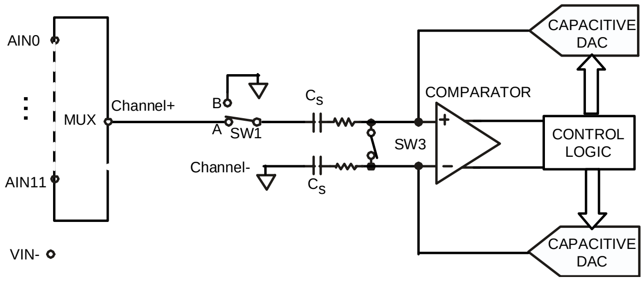


Figure 36: ADC in single-ended mode

Analog Input Structure

Figure 37 shows the equivalent circuit of the analog input structure of the ADC. The four diodes provides 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 37 are typically 4 pF and can primarily be 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 have a capacitance of 16 pF typically.

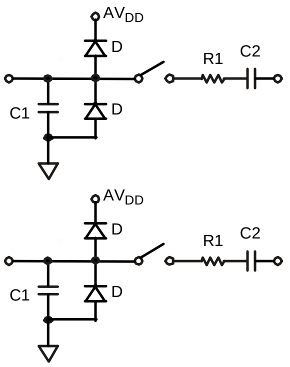


Figure 37: Equivalent Analog Input Circuit Conversion Phase: Switches Open Track Phase: Switches Closed

For AC applications, removing high-frequency components from the analog input signal is recommended by the use of an RC low-pass filter on the relevant analog input pins. In applications where harmonic distortion and signal-to-noise ratio are critical, the analog input should be driven from a low impedance source. Large source impedances will significantly affect the AC performance of the ADC. This may necessitate the use of an input buffer amplifier. The choice of the op amp will be a function of the particular application. Figure 38 and Figure 39 give an example of ADC front end.

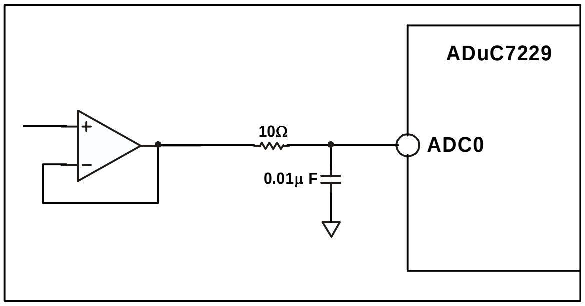


Figure 38. Buffering Single-Ended/Pseudo Differential Input

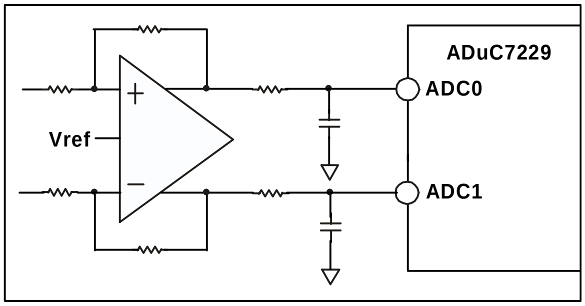


Figure 39. Buffering Differential Inputs

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 will depend on the amount of total harmonic distortion (THD) that can be tolerated. The THD will increase as the source impedance increases and the performance will degrade.

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}) that are dependant on reference value and supply voltage used to ensure that the signal remains within the supply rails. Table 12 gives some calculated V_{CM} min V_{CM} max for some conditions.

Table 12: V_{CM} ranges

AVDD	VREF	V_{CM} min	V_{CM} max	Signal Peak-Peak
3.3V	2.5V	1.25V	2.05V	2.5V
	2.048V	1.024V	2.276V	2.048V
	1.25	0.75V	2.55V	1.25
3.0V	2.5V	1.25V	1.75V	2.5V
	2.048V	1.024V	1.976V	2.048V
	1.25	0.75V	2.25V	1.25

TEMPERATURE SENSOR

The ADuC7128 provides a voltage output from an on-chip bandgap reference proportional to absolute temperature. It 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.

BANDGAP REFERENCE

The ADuC7128 provides an on-chip bandgap reference of 2.5V, which can be used for the ADC and for the DAC. This internal reference also appears on the V_{REF} pin. When using the internal reference, a capacitor of 0.47µF 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 (VREF) and used as a reference for other circuits in the system. An external buffer would be required because of the low drive capability of the VREF output. A programmable option also allows an external reference input on the V_{REF} pin.

The bandgap reference interface consists on a 8-bit MMR, REFCON described in the following table.

Table 13: REFCON MMR bit designations

Bit	Description
7-2	*Reserved*
1	Internal reference powerdown enable *Set* by user to place the internal reference in power-down mode and use 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 2.5V reference to the VREF pin. The reference can be used for external component but will need to be buffered. *Cleared* by user to disconnect the reference from the VREF pin. Note: The on chip DAC is only functional with the internal reference output enable bit set. It will not work with an external reference.

NONVOLATILE FLASH/EE MEMORY

FLASH/EE MEMORY OVERVIEW

The ADuC7128 incorporates Flash/EE memory technology on-chip to provide the user with non-volatile, 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 non-volatility, in-circuit programmability, high density, and low cost. Incorporated in the ADuC7128, Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one time programmable (OTP) devices at remote operating nodes.

FLASH/EE MEMORY AND THE ADUC7128

The ADuC7128 contains two 64 kByte arrays of Flash/EE Memory. In the first block the lower 62 Kbytes is available to the user and the upper 2 kBytes of this Flash/EE program memory array contain permanently embedded firmware, allowing in circuit serial download. These 2 Kbytes of embedded firmware also contain a power-on configuration routine that downloads factory calibrated coefficients to the various calibrated peripherals (bandgap references and so on). This 2 kByte embedded firmware is hidden from user code. It is not possible for the user to read, write or erase this page. In the second block all 64kB of Flash/EE memory are available to the user.

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

(1) Serial Downloading (In-Circuit Programming)

The ADuC7128 facilitates code download via the standard UART serial port or via the I2C port. The ADuC7128 enters serial download mode after a reset or power cycle if the BM pin is pulled low through an external 1kOhm resistor. Once in serial download mode, the user can download code to the full 126kBytes of Flash/EE memory while the device is in circuit in its target application hardware. A PC serial download executable 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 I2C.

(2) 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 could be corrupted. That is, it is possible to walk zeros only byte wise.

FLASH/EE MEMORY SECURITY

The 126 kB of Flash/EE memory available to the user can be read and write protected.

Bit 31 of the FEE0PRO/FEE0HID MMR protects the 126 kB from being read through JTAG and also in parallel programming mode. The other 31 bits of this register protect writing to the flash memory; each bit protects 4 pages, that is, 2 kB. Write protection is activated for all type of access. FEE1PRO and FEE1HID similarly protect the second 64kB block. All 32 bits of this are used to protect 4 pages at a time.

Three Levels of Protection

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

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

3. The 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 FEExPRO register is not allowed

Sequence to Write the Key

1. Write the bit in FEExPRO corresponding to the page to be protected.

2. Enable key protection by setting Bit 6 of FEExMOD (Bit 5 must be = 0).

3. Write a 32-bit key in FEExADR, FEExDAT.

4. Run the write key command 0×0C in FEExCON; wait for the read to be successful by monitoring FEExSTA.

5. Reset the part.

To remove or modify the protection, the same sequence is used with a modified value of FEExPRO. If the key chosen is the value 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:

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

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

FLASH/EE CONTROL INTERFACE

FEE0DAT Register

Name	Address	Default Value	Access
FEE0DAT	0xFFFF0E0C	0xXXXX	RW

FEE0DAT is a 16-bit data register.

FEE0ADR Register

Name	Address	Default Value	Access
FEE0ADR	0xFFFF0E10	0x0000	RW

FEE0ADR is another 16-bit address register.

FEE0SGN Register

Name	Address	Default Value	Access
FEE0SGN	0xFFFF0E18	0xFFFFFF	R

FEE0SGN is a 24-bit code signature.

FEE0PRO Register

Name	Address	Default Value	Access
FEE0PRO	0xFFFF0E1C	0x00000000	RW

FEE0PRO provides immediate protection MMR. It does not require any software keys. See description in Table 17.

FEE0HID Register

Name	Address	Default Value	Access
FEE0HID	0xFFFF0E20	0xFFFFFFFF	RW

FEE0HID provides protection following subsequent reset MMR. It requires a software key. See description in Table 18.

Command Sequence for Executing a Mass Erase

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

FEE1DAT Register

Name	Address	Default Value	Access
FEE1DAT	0xFFFF0E8C	0xXXXX	RW

FEE1DAT is a 16-bit data register.

FEE1ADR Register

Name	Address	Default Value	Access
FEE1ADR	0xFFFF0E90	0x0000	RW

FEE1ADR is another 16-bit address register.

FEE1SGN Register

Name	Address	Default Value	Access
FEE1SGN	0xFFFF0E98	0xFFFFFF	R

FEE1SGN is a 24-bit code signature.

FEE1PRO Register

Name	Address	Default Value	Access
FEE1PRO	0xFFFF0E9C	0x00000000	RW

FEE1PRO provides immediate protection MMR. It does not require any software keys. See description in Table 17.

FEE1HID Register

Name	Address	Default Value	Access
FEE1HID	0xFFFF0EA0	0xFFFFFFFF	RW

FEE1HID provides protection following subsequent reset MMR. It requires a software key. See description in Table 18.

Table 14: FEExSTA 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 other command
4	*Reserved*
3	Flash interrupt status bit *Set* automatically when an interrupt occurs, i.e. when a command is complete and the Flash/EE interrupt enable bit in the FEExMOD register is set *Cleared* when reading FEExSTA register
2	Flash/EE controller busy *Set* automatically when the controller is busy *Cleared* automatically when the controller is not busy
1	Command fail *Set* automatically when a command completes unsuccessfully *Cleared* automatically when reading FEExSTA register
0	Command complete *Set* by MicroConverter when a command is complete *Cleared* automatically when reading FEExSTA register

Table 15: FEExMOD MMR bit designations

Bit	Description
7-5	*Reserved*
4	Flash/EE interrupt enable: *Set* by user to enable the Flash/EE interrupt. The interrupt will occur 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. Clear to protect the Flash against erase/write command.
2	*Reserved*
1-0	Flash waitstates, when the kernel exits this with be set to 1. The user should first switch to the external 32kHz crystal before setting the waitstates to 0. Both flash blocks must have the same wait state value for any change to take effect.

Table 16: command codes in FEExCON

Code	command	Description
0x00*	Null	Idle state
0x01*	Single Read	Load FEExDAT with the 16-bit data indexed by FEExADR
0x02*	Single Write	Write FEExDAT at the address pointed by FEExADR. This operation takes 20µs.
0x03*	Erase-Write	Erase the page indexed by FEExADR and write FEExDAT at the location pointed by FEExADR. This operation takes 20ms
0x04*	Single Verify	Compare the contents of the location pointed by FEExADR to the data in FEExDAT. The result of the comparison is returned in FEExSTA bit 1
0x05*	Single Erase	Erase the page indexed by FEExADR
0x06*	Mass erase	Erase user space. The 2kByte of kernel are protected in block 0. This operation takes 2.48s To prevent accidental execution a command sequence is required to execute this instruction, this is described below.
0x07	Burst read	Default command. No write is allowed. This operation takes 2 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	Will automatically erase the page indexed by the write, allow to write pages without running an erase command. This command takes 20 ms to erase the page + 20 µs per data to write

0x0A	Burst termination	Stops the running burst to allow execution from Flash/EE immediately
0x0B	Signature	Give a signature of the 64kBytes of Flash/EE in the 24-bit FEExSIGN MMR. This operation takes 32778 clock cycles.
0x0C	Protect	This command can be run only once. The value of FEExPRO is saved and can be removed only with a mass erase (0x06) or with the key
0x0D	*Reserved*	*Reserved*
0x0E	*Reserved*	*Reserved*
0x0F	Ping	No operation, interrupt generated

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

Table 17: FEE0PRO and FEE0HID MMR bit designations

Bit	Description
31	Read protection *Cleared* by user to protect block 0. *Set* by user to allow reading block 0.
30-0	Write protection for pages 123 to 120, for pages 119 to 116… and for pages 0 to 3 *Cleared* by user to protect the pages in writing *Set* by user to allow writing the pages

Table 18: FEE1PRO and FEE1HID MMR bit designations

Bit	Description
31	Read protection *Cleared* by user to protect block 1. *Set* by user to allow reading block 1.
30	Write Protection for pages 127 to 120 *Cleared* by user to protect the pages in writing *Set* by user to allow writing the pages
31-0	Write protection for pages for pages 119 to 116… and for pages 0 to 3 *Cleared* by user to protect the pages in writing *Set* by user to allow writing the pages

Execution time from SRAM and FLASH/EE

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

Execution from SRAM

Fetching instructions from SRAM takes one clock cycle as the access time of the SRAM is 2ns and a clock cycle is 23ns minimum. However, if the instruction involve reading or writing data to memory, one extra cycle must be added if the data is in SRAM, or three cycle if the data is in Flash/EE, one cycle to execute the instruction and two cycles to get the 32-bit data from Flash/EE. A control flow instruction, for example a branch instruction will take one cycle to fetch but also two cycle to fill the pipeline with the new instructions.

Execution from Flash/EE

Because the Flash/EE width is 16-bit and access time for 16-bit words is 23ns, execution from Flash/EE cannot be done in one cycle as 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 and in Thumb mode, where instructions are 16 bits, one cycle is needed to fetch any instruction.

Timing is identical in both mode 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 pipe-line. A data processing instruction involving only core register doesn't require any extra clock cycle but if it involves data in Flash/EE, an extra clock cycle is needed to decode the address of the data and two cycles to get the 32-bit data from Flash/EE. An extra cycle must also be added before fetching another instruction. Data transfer instruction are more complex and are summarised Table 19.

Table 19: execution cycles in ARM/Thumb mode

Instructions	Fetch cycles	Dead time	Data access	Dead time
LD	2/1	1	2	1
LDH	2/1	1	1	1
LDM/PUSH	2/1	N	2 x n	N
STR	2/1	1	2 x 20µs	1
STRH	2/1	1	20µs	1
STRM/POP	2/1	N	2 x N x 20µs	N

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

The SWAP instruction combine a LD and STR instruction with only one fetch giving a total of 8 cycles plus 40µs.

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 Figure 40.

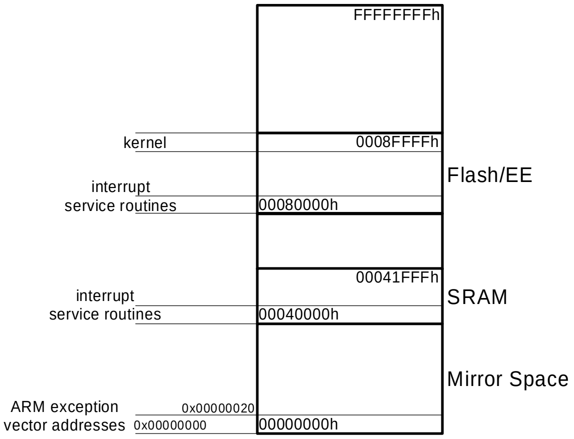


Figure 40: remap for exception execution

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

Remap operation

When a reset occurs on the ADuC7128, execution starts automatically in factory programmed internal configuration code. This so called kernel is hidden and cannot be accessed by user code. If the ADuC7128 is in normal mode (BM pin is high), it will execute the power-on configuration routine of the kernel and then jump to the reset vector address, 0x00000000, to execute the users reset exception routine.

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 bit0 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 will be replaced by the SRAM.

This operation is reversible: the Flash/EE can be remapped at address 0x00000000 by clearing Bit0 of the REMAP MMR. Precaution must again be taken to execute the remap function from outside the mirrored area. Any kind of reset will remap the Flash /EE memory at the bottom of the array.

Reset

There are four kinds of reset: external reset, Power-on-reset, watchdog expiation and software force. The RSTSTA register indicates the source of the last reset and RSTCLR allows to clear the RSTSTA register. These registers can be used during a reset exception service routine to identify the source of the reset. If RSTSTA is null, the reset was external. Note: When clearing RSTSTA all bits that are currently '1' must be cleared, otherwise a reset event will occur.

Table 20: REMAP MMR bit designations

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

Table 21: 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

OTHER ANALOG PERIPHERALS

DAC

The ADuC7128 features a 10-bit current DAC which can be used to generate user defined waveforms or sine waves generated by the DDS. The DAC consists of a 10 bit IDAC followed by a current to voltage conversion.

The current output of the IDAC is passed through a resistor and capacitor network where it is both filtered and converted to a voltage. This voltage is then buffered by an op-amp and passed to the line driver.

The user may optionally disable the internal filter and place an external filter between VDAC and AIN5.

For the DAC to function the internal 2.5v voltage reference must be enabled and driven out onto an external capacitor, i.e. REFCON = 0x01.

Once the DAC is enabled users will see a 5mV drop in the internal reference value. This is due to bias currents drawn from the reference used in the DAC circuitry. It is recommended that if using the DAC then it is left powered on to avoid seeing variations in ADC results.

Table 22: DACCON MMR bit designations

Bit	Description
10-9	00 Shuffle one increment at a time. 01 Shuffle based on an internal counter. 10 Shuffle based on the input data. 11 Reserved.
8	MSB Shuffle Enable Control *Set* by user to enable MSB Shuffling *Cleared* by user to disable MSB Shuffling
7	LSB Shuffle Enable Control *Set* by user to enable LSB Shuffling *Cleared* by user to disable LSB Shuffling
6	Power Reduction Control *Set* by user to reduce power consumption of DAC and line driver, this will also reduce the performance of the circuit. *Cleared* by user to operate in normal power mode.
5	Output Enable. This bit operates is all modes *Set* by user to enable the Line Drive output. *Cleared* by user to disable the Line Driver output. In this mode the line driver output is high impedance.
4	Single ended or Differential output Control. *Set* by user to operated in differential mode, the output is the differential voltage between LD1TX and LD2TX *The voltage output range will be Vref/2 +/- Vref/2* *Cleared* by user to reference the LD1TX output to AGND *The voltage output rage will be AVdd/2 +/- Vref/2*
3	Reserved, This bit should be set to ‘0’ by the user.
2-1	Operation Mode Control. This bit selects the mode of operation of the DAC. 00 Powerdown 01 Reserved 10 Reserved 11 DDS and DAC mode, selected by DACEN
0	DAC update rate control This bit has no effect when in DDS or PLM mode. *Set* by user to update the DAC on the negative edge of Timer 1. This allows the user to use any one of the core clk, osc clk, baud clks or user clk and divide these down by 1, 16, 256 or 32768. A user can do waveform generation by writing to the Dac Data Register from ram and update the dac at regular intervals via timer1. *Cleared* by user to update the DAC on the negative edge of HCLK.

DACEN Register

Name	Address	Default Value	Access
DACEN	0xFFFF06B8	0x00	RW

DACEN MMR Bit Designations

Bit	Description
7 to 1	Reserved.
0	Set to '1' by the user to enable DAC mode Set to '0' by the user to enable DDS mode

DACDAT Register

Name	Address	Default Value	Access
DACDAT	0xFFFF06B4	0x0000	RW

Table 23. DACDAT MMR Bit Designations

Bit	Description
15 to 10	Reserved.
9 to 0	10-bit data for DAC

The DACDAT MMR controls the output of the DAC. The data written to this register is a +/- 9 bits signed value. This means that 0x0000 represents midscale, 0x0200 represents zero scale and 0x01FF full scale.

DACEN and DACDAT require key access, the write to these MMRs follow the sequence below,

DACEN	DACDAT

DACKEY0 = 0x07
DACEN = User Value
DACKEY1 = 0xB9
	DACKEY0 = 0x07
	DACDAT = User Value
	DACKEY1 = 0xB9

DDS

The DDS is used to generate a digital sine wave signal for the DAC on the ADuC7128. It can be enabled into a free running mode by the user.

Both the phase and frequency can be controlled.

Table 24: DDSCON MMR bit designations

Bit	Description	
7-6	*Reserved*	
5	DDS Output Enable *Set* by user to enable the DDS output, this only has effect if the DDS is selected in DACCON *Cleared* by user to disable the DDS output.	
4	*Reserved*	
0	Binary Divide Control.	
	DIV	**Scale Ratio**
	0000	0.000
	0001	0.125
	0010	0.250
	0011	0.375
	0100	0.500
	0101	0.625
	0110	0.750
	0111	0.875
	1xxx	1.000

DDSFRQ Register

Name	Address	Default Value	Access
DDSFRQ	0xFFFF0694	0x00000000	RW

Table 25. DACDAT MMR Bit Designations

Bit	Description
31 to 0	FSW

The DDS Frequency is controlled via the DDSFRQ MMR. This MMR contains a 32 bit word (FSW, frequency select word) which controls the frequency according to the following formula:

$$Frequency = \frac{FSW \times 20.8896MHz}{2^{32}}$$

DDSPHS Register

Name	Address	Default Value	Access
DDSPHS	0xFFFF0698	0x00000000	RW

Table 26. DDSPHS MMR Bit Designations

Bit	Description
31 to 12	Reserved.
11 to 0	Phase

The DDS Phase offset is controlled via the DDSPHS MMR. This MMR contains a 12 bit value which controls the phase of the DDS output according to the following formula:

$$Phaseoffset = \frac{2 \times \pi \times Phase}{2^{12}}$$

POWER SUPPLY MONITOR

The Power Supply Monitor monitors the IOV_{DD} supply on the ADuC7128. It indicate when IOV_{DD} supply pin drops below one of two supply trip points. The monitor function is controlled via the PSMCON register. If enabled in the IRQEN or FIQEN register, the monitor will interrupt the core using the PSMI bit in the PSMCON MMR. This bit will be cleared immediately once CMP goes high. Note that if the interrupt generated is exited before CMP goes high (i.e. IOVdd above the trip point) then no further interrupts will be generated until CMP returns high. The user should ensure that code execution remains within the ISR until CMP returns high.

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

The PSM will not operate correctly when using JTAG debug. It should be disabled in this mode.

Table 27: 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 the IOV_{DD} supply is above its selected trip point. Read '0' indicates the IOV_{DD} supply is below its selected trip point.
2	TP	Trip Point Selection Bits 0 - 2.79V 1 - 3.07V
1	PSMEN	Power Supply Monitor Enable Bit *Set* to '1' by the user to enable the Power Supply Monitor circuit *Clear* to '0' by the user to disable the Power Supply Monitor circuit
0	PSMI	Power Supply Monitor Interrupt Bit. This bit will be set high by the MicroConverter if CMP is low, indicating low I/O supply. The PSMI Bit can be used to interrupt the processor. Once CMP returns high, the PSMI bit may be cleared by writing a '1' to this location. A write of '0' has no effect. There is no timeout delay, PSMI may be cleared immediately once CMP goes high.

COMPARATOR

The ADuC7128 also integrates an uncommitted voltage comparator.

The positive input is multiplexed with ADC2 and the negative input has two options: ADC3 or the internal reference. 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 be on an external pin, CMP_{OUT}.

Hysteresis

Figure 41 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 one-half the width of the hysteresis range.

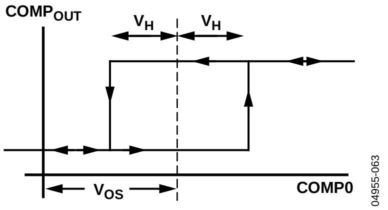


Figure 41. Comparator Hysteresis Transfer Function

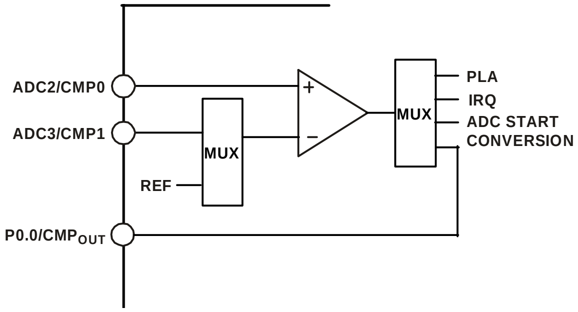


Figure 42: Comparator

The comparator interface consists on a 16-bit MMR, CMPCON described below.

Table 28: CMPCON MMR bit descriptions

Bit	Name	Description
15-11		*Reserved*
10	CMPEN	Comparator enable bit: *Set* by user to enable the comparator, Note: A comparator interrupt will be generated on the enable of the comparator, this should be cleared in the user software. *Cleared* by user to disable the comparator
9-8	CMPIN	Comparator negative input select bits: 00 *AVDD/2* 01 ADC3 input 10 Vref * 0.6 11 *Reserved*
7-6	CMPOC	Comparator output configuration bits: 00 IRQ and PLA connections disabled 01 IRQ and PLA connections disabled 10 PLA Connections enabled 11 IRQ Connections enabled
5	CMPOL	Comparator output logic state bit When low the comparator output is high when the positive input (CMP0) is above the negative input (CMP1). When high, the comparator output is high when 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). 01 *Reserved* 10 *Reserved* 11 3 µs response time typical for any signal type.

2	CMPHYST	Comparator hysteresis bit: *Set* by user to have an hysteresis of about 7.5mV *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

The ADuC7128 integrates a 32.768kHz oscillator, a clock divider and a PLL. The PLL locks onto a multiple (1275) of the internal oscillator to provide a stable 41.78 MHz clock for the system. The core can operate at this frequency, or at binary submultiples of it, to allow power saving. The default core clock is the PLL clock divided by 8 (CD = 3) or 5.2 MHz. The core clock frequency can be outputted on the ECLK pin as described in the GPIO section. A power down mode is available on the ADuC7128.

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

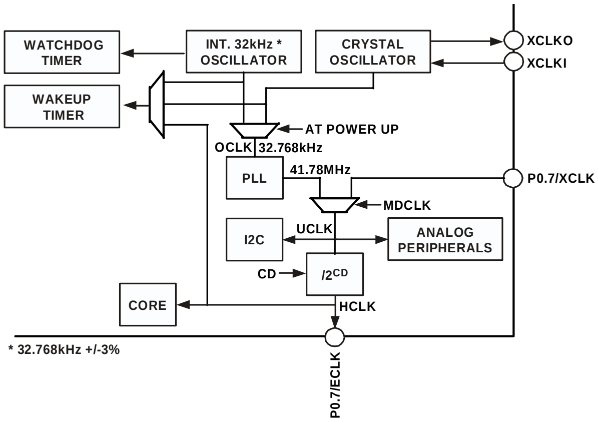


Figure 43: clocking system

External Crystal Selection

To switch to external crystal, clear the OSEL bit in the PLLCON MMR (see Table 31). In noisy environments, noise might couple to the external crystal pins and PLL could loose lock momentary. 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 initialisation 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 ADuC7128. Table 29 describes what part of the ADuC7128 is powered on in the different modes and indicates the power-up time. Table 30 gives some typical values of total the current consumption (analog + digital supply currents) in the different modes depending on the clock divider bits. The ADC is turned off. Note that these values also include current consumption of the regulator and other parts on the test board on which these values were measured.

Table 29. Operating Modes

Mode	Core	Peripherals	PLL	XTAL/T2/T3	XIRQ	Start up /power on Time
Active	X	X	X	X	X	TBD at CD = 0
Pause		X	X	X	X	TBD at CD = 0. 3.06 µs at CD = 7
Nap			X	X	X	TBD at CD = 0. 3.06 µs at CD = 7
Sleep				X	X	TBD
Stop					X	TBD

Table 30. 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								
001	Pause								
010	Nap								
011	Sleep								
100	Stop								

MMRs and Keys

The operating mode, clocking mode and programmable clock divider are controlled via two MMRs, PLLCON (see Table 31) and POWCON (see Table 32). PLLCON controls 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, shown in Table 33, has to be followed to write in the PLLCON and POWCON registers.

PLLKEYx Register

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

PLLCON Register

Name	Address	Default Value	Access
PLLCON	0xFFFF0414	0x21	RW

POWKEYx Register

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

POWCON Register

Name	Address	Default Value	Access
POWCON	0xFFFF0408	0x0003	RW

Table 31. PLLCON MMR Bit Designations

Bit	Value	Name	Description
7, 6			Reserved.
5		OSEL	32k Hz PLL input selection. Set by the user to use the internal 32 kHz oscillator. Set by default. Cleared by user to use the external 32 kHz crystal.
4, 3, 2			Reserved.
1, 0		MDCLK	Clocking modes.
	00		Reserved.
	01		PLL. Default configuration.
	10		Reserved.
	11		External clock on P0.7 pin.

Table 32. POWCON MMR Bit Designations

Bit	Value	Name	Description
7			Reserved.
6. 5. 4		PC	Operating modes.
	000		Active mode.
	001		Pause mode.
	010		Nap.
	011		Sleep mode. XIRQ0, XIRQ1, Timer2, and Timer3 can wake-up the ADuC7128
	100		Stop mode.
	Others		Reserved.
3		RSVD	Reserved
2, 1, 0		CD	CPU clock divider bits.
	000		41.779200 MHz
	001		20.889600 MHz
	010		10.444800 MHz
	011		5.222400 MHz
	100		2.611200 MHz
	101		1.305600 MHz
	110		654.800 kHz
	111		326.400 kHz

Table 33. PLLCON and POWCON Write Sequence

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

DIGITAL PERIPHERALS

PWM

General overview

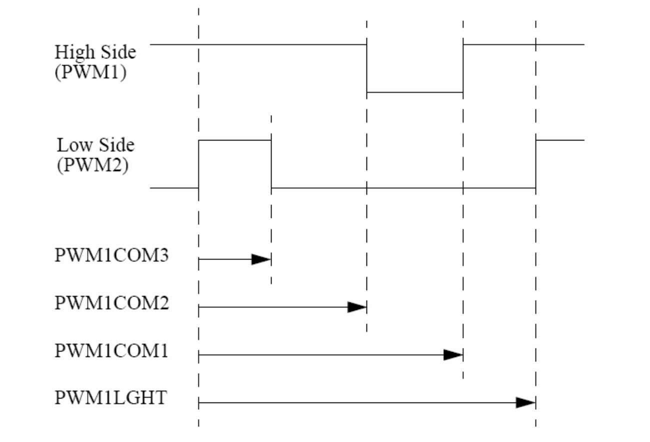
The ADuC7128 integrates a six channel PWM interface. The PWM outputs can be configured to drive a H-Bridge or can be used as standard PWM outputs. On power up the PWM outputs default to H-Bridge mode. This ensures that the motor is turned off by default. In standard PWM mode the outputs are arranged as 3 pairs of PWM pins. Users have control over the period of each pair of outputs and of the duty cycle of each individual output.

The PWM has the following MMRs

Name	Function
PWMCON1	PWM Control
PWM1COM1	Compare register 1 for PWM o/ps 1 and 2
PWM1COM2	Compare register 2 for PWM o/ps 1 and 2
PWM1COM3	Compare register 3 for PWM o/ps 1 and 2
PWM1LEN	Frequency Control for PWM o/ps 1 and 2
PWM2COM1	Compare register 1 for PWM o/ps 3 and 4
PWM2COM2	Compare register 2 for PWM o/ps 3 and 4
PWM2COM3	Compare register 3 for PWM o/ps 3 and 4
PWM2LEN	Frequency Control for PWM o/ps 3 and 4
PWM3COM1	Compare register 1 for PWM o/ps 5 and 6
PWM3COM2	Compare register 2 for PWM o/ps 5 and 6
PWM3COM3	Compare register 3 for PWM o/ps 5 and 6
PWM3LEN	Frequency Control for PWM o/ps 5 and 6
PWMCON2	PWM Convert Start Control
PWMICLR	PWM Interrupt Clear

In all modes the PWMxCOMx MMRs controls the point at which the PWM outputs change state. An example of the first pair of PWM outputs (PWM1 and PWM2) is shown below.

Figure 44 PWM Timing.



The PWM clock is selectable via PWMCON1 with one of the following values, UCLK /2, 4, 8, 16, 32, 64, 128 or 256.

The length of a PWM period is defined by PWMxLEN.

The PWM waveforms are set by the count value of the 16-bit timer and the compare registers contents as shown with the PWM1 and PWM2 waveforms above.

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

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

Table 34. PWMCON1 MMR Bit Designations

Bit	Value	Name	Description
14		SYNC	Enables PWM synchronization *Set* to '1' by the user so that all PWM counters are reset on the next clock edge after the detection of a high to low transition on the SYNC pin. *Cleared* by the user to ignore transitions on the SYNC pin.
13		PWM6INV	*Set* to '1' by the user to invert PWM6 *Cleared* by the user to use PWM6 in normal mode.
12		PWM4NV	*Set* to '1' by the user to invert PWM4 *Cleared* by the user to use PWM4 in normal mode.
11		PWM2INV	*Set* to '1' by the user to invert PWM2 *Cleared* by the user to use PWM2 in normal mode.
10		PWMTRIP	Set to '1' by the user to enable PWM trip interrupt. When the PWMTRIP input is low the PWMEN bit is cleared and an interrupt is generated.

			Cleared by the user to diable the PWMTRIP interrupt.
9		ENA	If HOFF = 0 and HMODE = 1 Set to ‘1’ by the user to enable PWM outputs Cleared by the user to disable PWM outputs. If HOFF=1 and HMODE=1, see table 51 In not H-Bridge mode this bit has no effect.
8		PWMCP2	PWM Clock prescaler bits.
7		PWMCP1	Sets UCLCK divider.
6		PWMCP0	
	000		2
	001		4
	010		8
	011		16
	100		32
	101		64
	110		128
	111		256
5		POINV	Set to ‘1’ by the user to invert all PWM outputs Cleared by the user to use PWM outputs as normal.
4		HOFF	High side off. Set to ‘1’ by the user to force PWM1 and PWM3 outputs high, this also forces PWM2 and PWM4 low. Cleared by the user to use the PWM outputs as normal
3		LCOMP	Load compare regiseters. Set to ‘1’ by the user to load the internal compare registers with the values in PWMxCOMx on the next transition of the PWM timer from 0x00 to 0x01. Cleared by the user to use the values previously stored in the internal compare registers.
2		DIR	Direction Control Set to ‘1’ by the user to enalble PWM1 and PWM2 as the output signals while PWM3 and PWM4 are held low. Cleared by the user to enable PWM3 and PWM4 as the output signals while PWM 1 and PWM2 are held low.
1		HMODE	Enables H-Bridge mode Set to ‘1’ by the user to enable H-Bridge mode and bits 1-5 of PWMCON1. Cleared by the user to operate the PWMs in standard mode.
0		PWMEN	Set to ‘1’ by the user to enable all PWM outputs.

In H-Bridge mdoe i.e. HMODE = 1 then the table below determines the PWM outputs.

HS= High side, LS= Low side

Table 35.

PWMCOM1 MMR				PWM Outputs			
ENA	HOFF	POINV	DIR	PWM1	PWM2	PWMR3	PWM4
0	0	x	x	1	1	1	1
x	1	x	x	1	0	1	0
1	0	0	0	0	0	HS	LS
1	0	0	1	HS	LS	0	0
1	0	1	0	!HS	!LS	1	1
1	0	1	1	1	1	!HS	!LS

Note on POWERUP PWMCON1 defaults to 0x12 i.e. HOFF = 1 and HMODE = 1. All GPIO pins associated with the PWM are configured in PWM mode by default.

The compare registers are detailed below.

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

The PWM Trip interrupt can be cleared by writing to the PWMICLR MMR. Note: When using the PWM trip interrupt users should make sure that the PWM interrupt has been cleared before exiting the ISR. This avoids multiple interrupts being generated.

PWM Convert Start Control

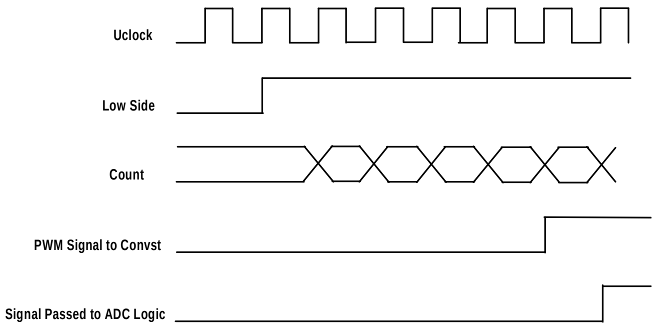
The PWM can be configured to generated an ADC convert start signal after the active low side signal goes high. There is a programmable delay between when the low side signal goes high and the Convert Start signal is generated.

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

Table 36. PWMCON2 MMR Bit Designations

Bit	Value	Name	Description
7		CSEN	Set to '1' by the user to enable the PWM to generate a convert start signal. Cleared by the user to diable the PWM convert start signal.
3-0		CSD3 CSD2 CSD1 CSD0	Convert Start Delays. Delay the convert start signal by a number of clock pulses.
	0000		4
	0001		8
	0010		12
	0011		16
	0100		20
	0101		24
	0110		28
	0111		32
	1000		36
	1001		40
	1010		44
	1011		48
	1100		52
	1101		56
	1110		60
	1111		64

When calculating the time from the convert start delay to the start of an ADC conversion the user needs to take account of internal delays. The example below shows the case for a delay of 4 clocks. One additional clock is required to pass the convert start signal to the ADC logic. Once the ADC logic receives the convert start signal an ADC conversion will begin on the next ADC clock edge. See figure 32 in the ADC section.



Quadrature Encoder

A quadrature encoder is used to determine both the speed and direction of a rotating shaft. In it's most common form there are two digital outputs S1 and S2. As the shaft rotates both S1 and S2 will toggle, however they will be 90° out of phase. The leading output determines the direction of rotation. The time between each transition indicates the speed of rotation.

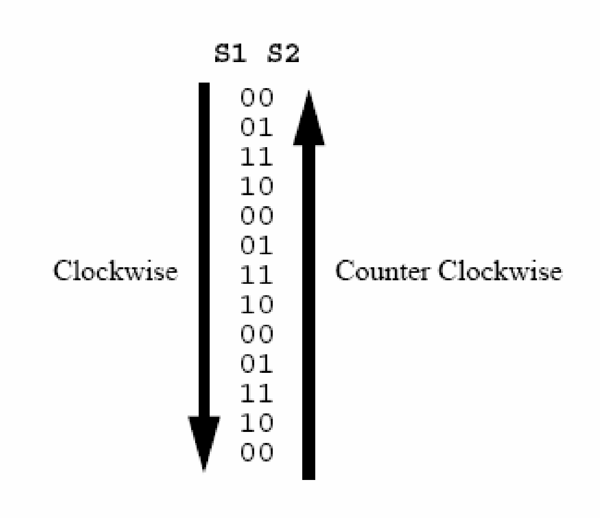


Figure 45. Quadrate Encoder Input Values

The quadrature encoder takes the incremental input shown above and increments or decrements a counter depending on the direction and speed of the rotating shaft.

On the ADuC7128, the internal counter is clocked on the rising edge of the S1 input, the S2 input indicates the direction of rotation/count. The counter increments when S2 is high and decrements when it is low.

In addition, if the software has prior knowledge of the direction of rotation. Then one input can be ignored (S2) and the other can act as a clock (S1).

For added flexibility, all inputs can be internally inverted prior to use.

The Quadrature Encoder operates asynchronously from the system clock.

Input Filtering

Filtering can be applied to the S1 input by setting the FILTEN bit in QENCON. S1 normally acts as the clock to the counter, however the filter can be used to ignore positive edges on S1 unless there has been a high or a low pulse on S2 between two positive edges on S1. (See figure).

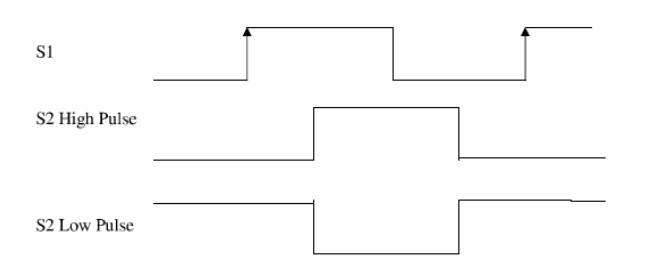


Figure 46. S1 Input Filtering

Table 37. QENCON MMR Bit Designations

Bit	Value	Name	Description
15-11			Reserved.
10		FILTEN	Set to '1' by the user to enable filtering on the S1 pin Cleared by the user do disable filtering on the S1 pin.
9		RSVD	Reserved, This bit should be set to '0' by the user.
8		S2INV	Set to '1' by the user to invert the S2 intput. Cleared by the user to use the S2 input as normal. If the DIRCON bit is set, then S2INV controls the direction fo the counter. In this case: Set to '1' by the user to operate the counter in increment mode. Cleared by the user to operate the counter in decrement mode.
7		S1INV	Set to '1' by the user to invert the S1 intput. Cleared by the user to use the S1 input as normal.
6		DIRCON	Direction Control. Set to '1' by the user to enable S1 as the input to the counter clock. The driection of the counter is controlled via the S2INV bit. Cleared by the user to operated in normal mode.
5		S1IRQEN	Set to '1' by the user to geneate an IRQ when a low to high transition is

			detected on S1. Cleared by the user to disable the interrupt.
4		RSVD	This bit should be set to '0' by the user.
3		UIRQEN	Underflow IRQ enable. Set to '1' by the user to generate an interrupt if QENVAL underflows. Cleared by the user to disable the interrupt.
2		OIREQEN	Overflow IRQ enable. Set to '1' by the user to generate an interrupt if QENVAL overflows. Cleared by the user to disable the interrupt.
1		RSVD	This bit should be set to '0' by the user.
0		ENQEN	Enable Quadrature Encoder Set to '1' by the user to enable the quadrature encoder. Cleared by the user to diable the quadrature encoder.

Table 38. QENSTA MMR Bit Designations

Bit	Value	Name	Description
7-5			Reserved.
4		S1EDGE	S1 rising edge. This bit is set automatically on a rising edge of S1 Cleared by reading QENSTA
3		RSVD	Reserved.
2		UNDER	Underflow flag This bit is set automatically if an underflow occurs. Cleared by read QENSTA
1		OVER	This bit is set automatically if an overflow has occurred. Cleared by reading QENSTA
0		DIR	Direction of the counter Set to '1' by hardare to idicate that the counter Is incrementing. Set to '0' by hardware to indicate that the counter is decrementing.

Table 39. QENDAT

Name	Address	Default Value	Access
QENDAT	0xFFFF0F08	0Xffff	RW

This register holds the maximum value allowed for the QENVAL register. If the QENVAL register increments past the value in this register then an overflow condition occurs. When an overflow occurs the QENVAL register is reset to 0x0000. When the QENVAL register decrements past zero during an underflow it will be loaded with the value in QENDAT.

Table 40. QENVAL

Name	Address	Default Value	Access
QENVAL	0xFFFF0F0C	0x0000	RW

This register contains the current value of the Quadrature Encoder counter.

Table 41. QENCLR

Name	Address	Default Value	Access
QENCLR	0xFFFF0F14	0x00000000	RW

Writing any value to this register clears the QENVAL register to 0x0000. The bits in this register are undefined.

Table 42. **QENSET**

Name	Address	Default Value	Access
QENSET	0xFFFF0F18	0x00000000	RW

Writing any value to this register loads the QENVAL register with the value in QENDAT. The bits in this register are undefined.

Note:

The interrupt conditions are ORed together to form one interrupt to the interrupt controller. The Interrupt Service Routine should check the QENSTA register to find out the cause of the interrupt.

- The S1 and S2 inputs shall appear as the QENS1 and QENS2 inputs in the GPIO list.
- The motor speed can be measured by using the capture facility in Timer 0 or Timer 1.
- An overflow of either timer can be checked by using an ISR or by checking IRQSIG.

  The counter with the quadrature encoder is gray encoded to ensure reliable data transfer across
  clock boundaries. When an underflow or overflow occurs the count value does not jump to the
  other end of the scale, instead the direction of count changes. When this happens the value in QENDAT is subtracted from the value derived from the gray count. When the value in QENDAT changes the value read back from QENVAL changes, however the gray encoded value will not change.

  This will only occur after an underflow or overflow.
  If the value in QENDAT changes then there must be a write to QENSET or QENCLR to ensure
  that a valid number is read back from QENVAL.

GENERAL PURPOSE I/O

The ADuC7128 provides 28 general purpose, bi-directional I/O (GPIO) pind. 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 the pin function definitions on page 14). By default, the GPIO pins are configured in GPIO mode.

All GPIO pins have 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 5 ports, Port 0 to Port 4. Each port is controlled by four or five MMRs, x representing the port number.

GPxCON Register

Name	Address	Default Value	Access
GP0CON	0xFFFF0D00	0x00000000	RW
GP1CON	0xFFFF0D04	0x00000000	RW
GP2CON	0xFFFF0D08	0x00000000	RW
GP3CON	0xFFFF0D0C	0x11111111	RW
GP4CON	0xFFFF0D10	0x00000000	RW

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 insure 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.

GPxCON is the port x control register, and selects the function of each pin of port X. as described in Table 43.

Table 43. GPIO Pin Function Descriptions

		Configuration			
Port	Pin	00	01	10	11
0	P0.0	GPIO	CMP	MS2	PLAI[7]
	P0.1[2]	GPIO	-	BLE	-
	P0.2[2]	GPIO	-	BHE	
	P0.3	GPIO	TRST	A16	ADC$_{\text{BUSY}}$
	P0.4	GPIO/IRQ0	$\overline{\text{CONV}}_{\text{START}}$	MS1	PLAO[1]
	P0.5	GPIO/IRQ1	ADC$_{\text{BUSY}}$	PLM_COMP	PLAO[2]
	P0.6	GPIO/T1	MRST	AE	PLAO[3]
	P0.7	GPIO	ECLK/XCLK[1]	SIN0	PLAO[4]
1	P1.0	GPIO/T1	SIN0	SCL0	PLAI[0]
	P1.1	GPIO	SOUT0	SDA0	PLAI[1]
	P1.2	GPIO	RTS0	SCL1	PLAI[2]
	P1.3	GPIO	CTS0	SDA1	PLAI[3]
	P1.4	GPIO/IRQ2	RI0	CLK	PLAI[4]
	P1.5	GPIO/IRQ3	DCD0	MISO	PLAI[5]
	P1.6	GPIO	DSR0	MOSI	PLAI[6]
	P1.7	GPIO	DTR0	CSL	PLAO[0]
2	P2.0	GPIO	SYNC	SOUT	PLAO[5]
	P2.1[2]	GPIO	-	WS	PLAO[6]
	P2.2[2]	GPIO	RTS1	RS	PLAO[7]
	P2.3[2]	GPIO	CTS1	AE	
	P2.4[2]	GPIO	RI1	MS0	
	P2.5[2]	GPIO	DCD1	MS1	
	P2.6[2]	GPIO	DSR1	MS2	
	P2.7[2]	GPIO	DTR1	MS3	
3	P3.0	GPIO	PWM1	AD0	PLAI[8]
	P3.1	GPIO	PWM2	AD1	PLAI[9]
	P3.2	GPIO	PWM3	AD2	PLAI[10]
	P3.3	GPIO	PWM4	AD3	PLAI[11]
	P3.4	GPIO	PWM5	AD4	PLAI[12]
	P3.5	GPIO	PWM6	AD5	PLAI[13]
	P3.6[2]	GPIO	PWM1	AD6	PLAI[14]
	P3.7[2]	GPIO	PWM3	AD7	PLAI[15]
4	P4.0	GPIO	QENS1	AD8	PLAO[8]
	P4.1	GPIO	QENS2	AD9	PLAO[9]
	P4.2	GPIO	RSVD	AD10	PLAO[10]
	P4.3	GPIO	Trip (Shutdown)	AD11	PLAO[11]
	P4.4	GPIO	PLMIN	AD12	PLAO[12]
	P4.5	GPIO	PLMOUT	AD13	PLAO[13]
	P4.6	GPIO	SIN1	AD14	PLAO[14]
	P4.7	GPIO	SOUT1	AD15	PLAO[15]

[1] When configured in Mode 1, PO.7 is ECLK by default, or core clock output. To configure it as a clock output, MDCLK bits in PLLCON must be set to 11.

[2] Only available on 80 pin ADuC7129 part.

Table 44. 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 Register

Name	Address	Default Value	Access
GP0PAR	0xFFFF0D2C	0x20000000	RW
GP1PAR	0xFFFF0D3C	0x00000000	RW
GP3PAR	0xFFFF0D5C	0x00222222	RW
GP4PAR	0xFFFF0D6C	0x00000000	RW

GPxPAR programs the parameters for Port 0, Port 1,,Port 3 and Port 4. Note that the GPxDAT MMR must always be written after changing the GPxPAR MMR.

Table 45. GPxPAR MMR Bit Descriptions

Bit	Description
31 to 29	Reserved
28	Pull up disable Px.7
27, 26, 25	Reserved
24	Pull up disable Px.6
23, 22, 21	Reserved
20	Pull up disable Px.5
19 , 18, 17	Reserved
16	Pull up disable Px.4
15, 14, 13	Reserved
12	Pull up disable Px.3
11, 10, 9	Reserved
8	Pull up disable Px.2
7, 6, 5	Reserved
4	Pull up disable Px.1
3, 2, 1	Reserved
0	Pull up disable Px.0

GPxDAT Register

Name	Address	Default Value	Access
GP0DAT	0xFFFF0D20	0x000000XX	RW
GP1DAT	0xFFFF0D30	0x000000XX	RW
GP2DAT	0xFFFF0D40	0x000000XX	RW
GP3DAT	0xFFFF0D50	0x000000XX	RW
GP4DAT	0xFFFF0D60	0x000000XX	RW

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

Table 46. GPxDAT MMR Bit Descriptions

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

GPxSET Register

Name	Address	Default Value	Access
GP0SET	0xFFFF0D24	0x000000XX	W
GP1SET	0xFFFF0D34	0x000000XX	W
GP2SET	0xFFFF0D44	0x000000XX	W
GP3SET	0xFFFF0D54	0x000000XX	W
GP4SET	0xFFFF0D64	0x000000XX	W

GPxSET is a data set port x register.

Table 47. GPxSET MMR Bit Descriptions

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

GPxCLR Register

Name	Address	Default Value	Access
GP0CLR	0xFFFF0D28	0x000000XX	W
GP1CLR	0xFFFF0D38	0x000000XX	W
GP2CLR	0xFFFF0D48	0x000000XX	W
GP3CLR	0xFFFF0D58	0x000000XX	W
GP4CLR	0xFFFF0D68	0x000000XX	W

GPxCLR is a data clear port x register.

Table 48. GPxCLR MMR Bit Descriptions

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

SERIAL PORT MUX

The Serial Port Mux multiplexes the serial port peripherals (two I^{2}C, SPI, two UARTs) 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 function as described in Table 49.

	GPIO	**UART**	**UART/I^{2}C/SPI**	PLA
	00	**01**	**10**	11
SPM0	P1.0	**SIN0**	**I2C0SCL**	PLAI[0]
SPM1	P1.1	**SOUT0**	**I2C0SDA**	PLAI[1]
SPM2	P1.2	**RTS0**	**I2C1SCL**	PLAI[2]
SPM3	P1.3	**CTS0**	**I2C1SDA**	PLAI[3]
SPM4	P1.4	**RI0**	**SPICLK**	PLAI[4]
SPM5	P1.5	**DCD0**	**SPIMISO**	PLAI[5]
SPM6	P1.6	**DSR0**	**SPIMOSI**	PLAI[6]
SPM7	P1.7	**DTR0**	**SPICSL**	PLAO[0]
SPM8	P0.7	ECLK	**SIN0**	PLAO[4]
SPM9	P2.0[1]	PWMSYNC	**SOUT0**	PLAO[5]
SPM10	P2.2[1]	**RTS1**	RS	PLAO[7]
SPM11	P2.3[1]	**CTS1**	AE	
SPM12	P2.4[1]	**RI1**	MS0	
SPM13	P2.5[1]	**DCD1**	MS1	
SPM14	P2.6[1]	**DSR1**	MS2	
SPM15	P2.7[1]	**DTR1**	MS3	
SPM16	P4.6	**SIN1**	AD14	PLAO[14]
SPM17	P4.7	**SOUT1**	AD15	PLAO[15]

Table 49: SPM configuration

[1] Only available on 80 pin part.

Table 49 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 17 pins are configured as GPIOs.

UART SERIAL INTERFACE

The ADuC7128 contains two identical UART blocks. Only UART0 is described here, UART1 functions in the exact same manner.

The UART peripheral is a full-duplex Universal Asynchronous Receiver/Transmitter, fully compatible with the 16450 serial port standard. The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a MODEM, and parallel-to-serial conversion on data characters received from the CPU. The UART includes a fractional divider for baudrate generation and has a network addressable mode. The UART function is made available on the following 10 pins of the ADuC7128:

Pin	**Signal**	**Description**
SPM0 (mode 1)	SIN0	Serial Receive Data
SPM1 (mode 1)	SOUT0	Serial Transmit Data
SPM2 (mode 1)	RTS0	Request To Send
SPM3 (mode 1)	CTS0	Clear To Send
SPM4 (mode 1)	RI0	Ring Indicator
SPM5 (mode 1)	DCD0	Data Carrier Detect
SPM6 (mode 1)	DSR0	Data Set Ready
SPM7 (mode 1)	DTR0	Data Terminal Ready
SPM8 (mode 2)	SIN0	Serial Receive Data
SPM9 (mode 2)	SOUT0	Serial Transmit Data

Table 50: UART signal description

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

Baud rate generation

There is two way of generating the UART baudrate.

- Normal 450 UART baudrate generation:

The baudrate is a divided version of the core clock using the value in COM0DIV0 and COM0DIV1 MMRs (16-bit value, DL).

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

The following table gives some common baudrate values:

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%

Table 51: baudrate using the normal baudrate generator

- Using the fractional divider:

The fractional divider combined with the normal baudrate generator allows the generating of a wider range of more accurate baudrates.

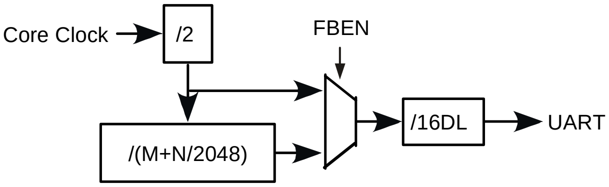


Figure 47: baudrate generation options

Calculation of the baudrate using fractional divider is as follow:

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

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

For example:
Generation of 19200 baud with CD bits = 3 (Table 51 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:
$M = 1$
$N = 0.06 \times 2048 = 128$

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

where:
Baud rate = 19200 bps
Error = 0% compared to 6.25% with the normal baud rate generator.

UART registers definition

The UART interface consists on 12 registers namely:

- **COMxTX:** 8-bit transmit register
- **COMxRX:** 8-bit receive register
- **COMxDIV0:** divisor latch (low byte)
  COMTX, COMRX and COMDIV0 share the same address location. COMTX and COMTX can be accessed when bit 7 in COMCON0 register is cleared. COMDIV0 can be accessed when bit 7 of COMCON0 is set.
- **COMxDIV1:** divisor latch (high byte)
- **COMxCON0:** line control register
- **COMxSTA0:** line status register
- **COMxIEN0:** interrupt enable register
- **COMxIID0:** interrupt identification register
- **COMxCON1:** modem control register
- **COMxSTA1:** modem status register
- **COMxDIV2:** 16-bit fractional baud divide register
- **COMxSCR:** 8-bit scratch register used for temporary storage. Also used in network addressable UART mode.

Table 52: COMxCON0 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 bit if the Word Length is 5 bits or 2 Stop bits if the word length is 6, 7 or 8 bits. The receiver checks the first Stop bit only, regardless of the number of Stop bits selected Cleared by user to generate 1 Stop bit in the transmitted data
1-0	WLS	Word length select: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits

Table 53: COMxSTA0 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 are overwrite before been read Cleared automatically
0	DR	Data ready Set automatically when COMRX is full Cleared by reading COMRX

Table 54: COMxIEN0 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

Table 55: COMxIID0 MMR Bit Descriptions

Bit 2-1	Bit 0	Priority	Definition	Clearing operation
Status bits	NINT			
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

Table 56: COMxCON1 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. Also the modem signals are directly connected to the status inputs (RTS to CTS, DTR to DSR, OUT1 to RI and OUT2 to DCD)
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

Table 57: COMxSTA1 MMR Bit Descriptions

Bit	Name	Description
7	DCD	Data carrier detect
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

Table 58: COMxDIV2 MMR Bit Descriptions

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

Network addressable UART mode

This mode allows connecting the MicroConverter on a 256-node serial network, either as a hardware single-master or via software in a multi-master network. Bit 7 of COMxIEN1 (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

Three additional register:

- **COMxSCR:** 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.
- **COMxIEN1:** 8-bit network enable register.
- **COMxIID1:** 8-bit network interrupt register. Bit 7 to 4 are reserved. See Table 54.
- **COMxADR:** 8-bit read and write network address register. Holds the address the network addressable UART checks for. On receiving this address the device interrupts the processor and/or sets the appropriate status bit in COMIID1.

COMIEN1, COMIID1 and COMADR are used only in network addressable UART mode.

Table 59: COMxIEN1 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 multi-master 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

Table 60: COMxIID1 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 COMxRX
101	0	3	Address transmitted, buffer empty	Write data to COMTX or read COMxIID0
011	0	1	Receive line status interrupt	Read COMxSTA0
010	0	2	Receive buffer full interrupt	Read COMxRX
001	0	3	Transmit buffer empty interrupt	Write data to COMxTX or read COMxIID0
000	0	4	Modem status interrupt	Read COMxSTA1 register

SERIAL PERIPHERAL INTERFACE

The ADuC7128 integrates a complete hardware serial peripheral interface (SPI) on-chip. SPI is an industry standard synchronous serial interface that allows eight bits of data to be synchronously transmitted and simultaneously received, that is, full duplex up to a maximum bit rate of 1.6 Mb. The SPI interface is only operational with core clock divider bits POWCON[2:0] = 0, 1, or 2.

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

MISO (Master In, Slave Out) Data I/O 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:

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

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 1.6 Mbs at CD = 0.

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 desassertion 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	RW

SPISTA is an 8-bit read only status register.

Table 61. SPISTA MMR Bit Descriptions

Bit	Description
7, 6	Reserved.
5	SPIRX data register overflow status bit. Set if SPIRX is overflowing. Cleared by reading SPISRX register.
4	SPIRX data register IRQ. Set automatically if Bit 3 or 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	RW

SPIDIV is an 8-bit serial clock divider register.

SPICON Register

Name	Address	Default Value	Access
SPICON	0xFFFF0A10	0x0000	RW

SPICON is a 16-bit control register.

Table 62. 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.
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 (Master Mode). Set by user to initiate transfer with a write to the SPITX register. Interrupt occurs when TX is empty. Cleared by user to initiate transfer with a read of the SPIRX register. Interrupt 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.
	Should be set to ‘0’
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 to disable the SPI.

I²C COMPATIBLE INTERFACES

The ADuC7128 supports 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 document describes only I2C0 in detail. Note that the two masters and slaves have individual interrupts.

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

The I²C bus peripheral's addresses 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 loose 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 looses 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, 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 Register

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

I2CxMSTA is a status register for the master channel.

Table 63. I2C0MSTA MMR Bit Descriptions

Bit	Description
7	Master busy. Set automatically if the master is busy. Cleared automatically.
6	Arbitration loss. Set in multi-master mode if another master has the bus. Cleared when the bus becomes available.
5	No ACK. Set automatically. If the master receive FIFO is full, the master does not acknowledge the data received. Cleared automatically.
4	Master receive FIFO overflow. Set automatically if the master receive FIFO is overflowing. Cleared automatically by reading I2C0MRX.
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 Register

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

I2CxSSTA is a status register for the slave channel.

Table 64. I2CxSSTA MMR Bit Descriptions

Bit	Value	Description
31 to 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 matched 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 I2CxSSTA 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.
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.
4		Slave receive FIFO overflow. Set automatically if the slave receive FIFO is overflowing. Cleared automatically by reading I2C0SRX.
3		Slave receive IRQ. Set after receiving data. Cleared automatically by reading the I2C0SRX register.
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 I2C0STX register.
0		Slave transmit FIFO empty. Set automatically if the slave transmit FIFO is empty. Cleared automatically by writing to the I2C0STX register.

I2CxSRX Register

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

I2CxSRX is a receive register for the slave channel.

I2CxSTX Register

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

I2CxSTX is a transmit register for the slave channel.

I2CxMRX Register

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

I2CxMRX is a receive register for the master channel.

I2CxMTX Register

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

I2CxSTX is a transmit register for the master channel.

I2CxCNT Register

Name	Address	Default Value	Access
I2C0CNT	0xFFFF0818	0x00	RW
I2C1CNT	0xFFFF0918	0x00	RW

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

I2CxADR Register

Name	Address	Default Value	Access
I2C0ADR	0xFFFF081C	0x00	RW
I2C1ADR	0xFFFF091C	0x00	RW

I2CxADR is a master address byte register. The I2CxADR value is the device address that the master wants to communicate with. It is automatically transmitted 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 Register

Name	Address	Default Value	Access
I2C0BYT	0xFFFF0824	0x00	RW
I2C1BYT	0xFFFF0924	0x00	RW

I2CxBYTE is a broadcast byte register.

I2CxALT Register

Name	Address	Default Value	Access
I2C0ALT	0xFFFF0828	0x00	RW
I2C1ALT	0xFFFF0928	0x00	RW

I2CxALT is a hardware general call ID register used in slave mode

I2CxCFG Register

Name	Address	Default Value	Access
I2C0CFG	0xFFFF082C	0x00	RW
I2C1CFG	0xFFFF092C	0x00	RW

I2CxCFG is a configuration register.

Table 65. I2C0CFG MMR Bit Descriptions

Bit	Description
31 to 15	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. This bit should be written by the user as 0.
12	Reserved. This bit should be written by the user as 0.
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. This bit should be written by the user as 0.
9	Slave Tx FIFO request interrupt enable. 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 55 clock cycles to take appropriate action, taking interrupt latency into account. Set by the user to disable the slave Tx FIFO request interrupt.
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 multi-master 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. (Bit 3 must be set.) Set by user to enable hardware general call. Cleared by user to disable hardware general call.
3	General call enable bit. Set by user to address every device on the I^{2}C bus. Cleared by user to operate in normal mode.
2	Reserved.
1	Master enable bit. Set by user to enable the master I^{2}C channel. Cleared by user to disable the master I^{2}C channel.
0	Slave enable bit. Set by user to enable the slave I^{2}C 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 I^{2}C channel.

I2CxDIV Register

Name	Address	Default Value	Access
I2C0DIV	0xFFFF0830	0x1F1F	RW
I2C1DIV	0xFFFF0930	0x1F1F	RW

I2CxDIV are the clock divider registers.

I2CxIDx Register

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

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

I2CxCCNT Register

Name	Address	Default Value	Access
I2C0SSC	0xFFFF0848	0x01	RW
I2C1SSC	0xFFFF0948	0x01	RW

I2CxCCNT is an 8-bit start/stop generation counter. It holds off SDA low for start and stop conditions.

I2CxFIF Register

Name	Address	Default Value	Access
I2C0FIF	0xFFFF084C	0x0000	R
I2C1FIF	0xFFFF094C	0x0000	R

I2C0FIF is an FIFO status register.

Table 66. I2C0FIF MMR Bit Descriptions

Bit	Value	Description
15 to 10		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.
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.

[*] Purchase of licensed I^2C components of Analog Devices or one of its sublicensed Associated Companies conveys a license for the purchaser under the Philips I^2C Patent Rights to use the ADuC7128 in an I^2C system, provided that the system conforms to the I^2C Standard Specification as defined by Philips.

PROGRAMMABLE LOGIC ARRAY (PLA)

The ADuC7128 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 a total of 16 PLA elements.

A PLA element contains a two-input lookup table that can be configured to generate any logic output function based on two inputs and a flip-flop as represented in Figure 48 below.

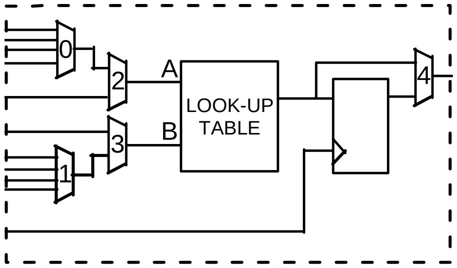


Figure 48: PLA element

In total, 30 GPIO pins are available on the ADuC7128 for the PLA. These include 16 input pins and 14 output pins. They 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 and the output(s) of the PLA can be routed to the internal interrupt system, to the CONV$_{\text{START}}$ signal of the ADC, to a MMR or to any of the 16 PLA output pins.

The interconnection between the two blocks is supported by connecting output of element 7 of block 1 fed back to the input 0 of mux 0 of element 0 of block 0, and the output of element 7 of block 0 is fed back to the input 0 of mux 0 of element 0 of block 1.

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

Table 67: element input/output

PLA MMRs interface

The PLA peripheral interface consists on 21 MMRs:

- **PLAELMx**: element0 to element 15 control registers, configure the input and output mux of each element, select the function in the lookup table and bypass/use the flip-flop.
- **PLACLK**: clock selection for the flip-flops of block 0 and clock selection for the flip-flops of block 1
- **PLAIRQ**: enable IRQ0 or/and IRQ1 and select the source of the IRQ
- **PLAADC**: PLA source fro ADC start conversion signal
- **PLADIN**: data input MMR for PLA
- **PLAOUT**: data output MMR for PLA. This register is always updated.

A PLA tool is provided in the development system to easily configure the PLA.

Table 68: PLAELMx MMR Bit Descriptions

Bit	Description	PLAELM0	PLAELM1 - 7	PLAELM8	PLAELM9-15
31-11	*Reserved*				
10-9	Mux (0) control, select feedback from:	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	Mux (1) control, select feedback from:	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
6	Mux (3) control *Set* by user to select the output of mux (1) *Cleared* by user to select the bit value from PLADIN				
5	Mux (2) control *Set* by user to select the input pin of the particular element *Cleared* by user to select the output of mux (0)				
4-1	Look-up table control 0000 – 0				
	0001 – NOR				
	0010 – B AND NOT A				

	0011 – NOT A 0100 – A AND NOT B 0101 – NOT B 0110 – EXOR 0111 – NAND 1000 – AND 1001 – EXNOR 1010 – B 1011 – NOT A OR B 1100 – A 1101 – A OR NOT B 1110 – OR 1111 – 1
0	Mux (4) control *Set* by user to bypass the flip-flop *Cleared* by user to select the flip-flop. Cleared by default

Table 69: PLACLK MMR Bit Descriptions

Bit	Description
7	*Reserved*
6-4	Block1 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 101 - Timer 1 overflow 110 - Timer 4 overflow Other – *Reserved*
3	*Reserved*
2-0	Block0 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 101 - Timer 1 overflow 110 - Timer 4 overflow Other – *Reserved*

Table 70: PLAIRQ MMR Bit Descriptions

Bit	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

Table 71: PLAADC MMR Bit Descriptions

Bit	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

Table 72: PLADIN MMR Bit Descriptions

Bit	Description
31-16	*Reserved*
15-0	Input Bit to element 15-0

Table 73: PLAOUT MMR Bit Descriptions

Bit	Description
31-16	*Reserved*
15-0	Output Bit from element 15-0

PROCESSOR REFERENCE PERIPHERALS

INTERRUPT SYSTEM

There are 30 interrupt sources on the ADuC7128 which are controlled by the Interrupt Controller. Most interrupts are generated from the on-chip peripherals like ADC, UART, etc. and two additional interrupt sources are generated from external interrupt request pins, XIRQ0 and XIRQ1. The ARM7TDMI CPU core will only recognise interrupts as one of two types, a normal interrupt request IRQ and a fast interrupt request FIQ. All the interrupts can be masked separately.

The control and configuration of the interrupt system is managed through nine interrupt-related registers, four dedicated to IRQ, four dedicated to FIQ. An additional MMR is used to select the programmed interrupt source. The bits in each IRQ and FIQ registers represent the same interrupt source as described in Table 74.

Table 74: IRQ/FIQ MMRs bit description

Bit	Description
0	FIQ source
1	SWI: not used in IRQEN/CLR and FIQEN/CLR
2	Timer 0
3	Timer 1
4	Wake Up timer – Timer 2
5	Watchdog timer – Timer 3
6	Timer 4
7	Flash Controler 0
8	Flash Controler 1
9	ADC
10	Quadrature Encoder
11	I^2C0 Slave
12	I^2C1 Slave
13	I^2C0 Master
14	I^2C1 Master
15	SPI Slave
16	SPI Master
17	UART 0
18	UART 1
19	External IRQ0
20	Comparator
21	PSM
22	External IRQ1
23	PLA IRQ0
24	PLA IRQ1
25	External IRQ2
26	External IRQ3
27	PWM Trip
28	PLL Lock
29	PLM RX
30	PLM TX

IRQ

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

The four 32-bit registers dedicated to IRQ are:

- **IRQSIG**, reflects the status of the different IRQ sources. If a peripheral generate an IRQ signal, the corresponding bit in the IRQSIG will be set, otherwise it is cleared. The IRQSIG bits are cleared when the interrupt in the particular peripheral is cleared. All IRQ sources can be masked in the IRQEN MMR. IRQSIG is read-only.
- **IRQEN**, provides the value of the current enable mask. When bit is set to 1, the source request is enabled to create an IRQ exception. When bit is set to 0, the source request is disabled or masked which will not create an IRQ exception. To clear a bit in IRQEN, use the IRQCLR MMR.
- **IRQCLR**, (write-only register) allows clearing the IRQEN register in order to mask an interrupt source. Each bit set to 1 will clear the corresponding bit in the IRQEN register without affecting the remaining bits. The pair of registers IRQEN and IRQCLR allows independent manipulation of the enable mask without requiring an atomic read-modify-write.
- **IRQSTA**, (read-only register) provides the current enabled IRQ source status. 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.

FIQ

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

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

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

FIQEN. An interrupt source can be disabled in both IRQEN and FIQEN masks.

Programmed interrupts

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

The 32-bit register dedicated to software interrupt is SWICFG described in Table 75. This MMR allows the control of programmed source interrupt.

Table 75: SWICFG MMR Bit Descriptions

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

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

TIMERS

The ADuC7128 has five general purpose Timer/Counters:

- Timer0,
- Timer1,
- Timer2 or Wake-up Timer,
- Timer3 or Watchdog Timer.
- Timer4

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

- In free-running mode the counter decrements/increments from the maximum/minimum value until zero/full scale and starts again at the maximum /minimum value.
- In periodic mode the counter decrements/increments from the value in the Load Register(TxLD MMR,) until zero/full scale and starts again at the value stored in the Load Register.

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

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

Table 76. : Event Selection Numbers

ES	Interrupt Number	Name
00000	2	RTOS Timer (Timer 0)
00001	3	GP Timer 0 (Timer 1)
00010	4	Wake Up Timer (Timer 2)
00011	5	Watchdog Timer (Timer 3)
00100	6	GP Timer 1 (Timer 4)
00101	7	Flash Control 0
00110	8	Flash Control 1
00111	9	ADC Channel
01000	10	Quadrature Encoder
01001	11	I2C Slave0
01010	12	I2C Slave1
01011	13	I2C Master0
01100	14	I2C Master1
01101	15	SPI Slave
01110	16	SPI Master
01111	17	UART0
10000	18	UART1
10001	19	External irq0

TIMER0 – LIFE-TIME TIMER

Timer0 is a general purpose 48-bit count-up, or a 16-bit count up/down timer with a programmable prescalar. Timer0 is clocked from the core clock, with a prescalar of 1,16, 256 or 32768. This gives a minimum resolution of 22ns when the core is operating at 41.78MHz, and with a prescalar of 1.

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

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

Timer0 reloads the value from T0LD either when TIMER0 overflows, or immediately when T0ICLR is written.

Timer0 interface consists of six MMRS:

- **T0LD** is a 16-bit register which holds the 16 bit value that is loaded into the counter. Only available in 16-bit mode.
- **T0CAP** is a 16-bit register which holds the 16-bit value captured by an enabled IRQ event. Only available in 16-bit mode.
- **T0VAL0/T0VAL1** are 16-bit and 32-bit registers which hold the 16 least significant bits and 32 most significant bits respectively. T0VAL0 and T0VAL1 is read-only. In 16-bit mode 16-bit T0VAL0 is used. In 48-bit mode both 16-bit T0VAL0 and 32-bit T0VAL1 are used.
- **T0ICLR** is an 8-bit register. Writing any value to this register will clear the interrupt. Only available in 16-bit mode.
- **T0CON** is the configuration MMR described in Table 77.

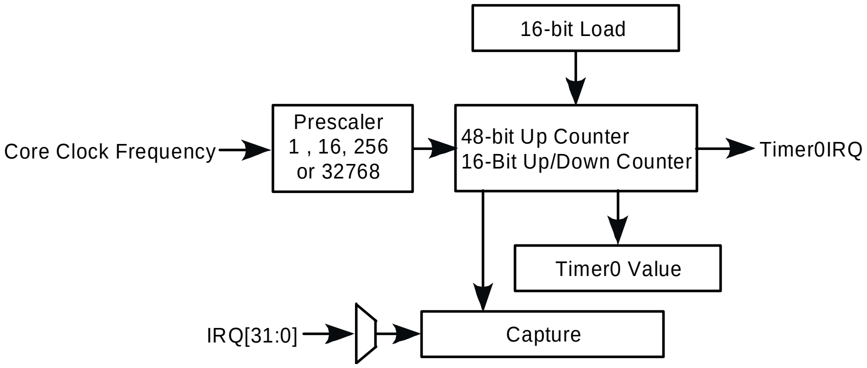


Figure 49 : Timer 0 block diagram

Timer0 Value Register :

Name : T0VAL0/T0VAL1
Address : 0xFFFF0304, 0xFFFF0308
Default Value : 0x00, 0x00
Access : Read Only
Function : T0VAL0 and T0VAL1 are 16-bit and 32-bit registers which hold the 16 least significant bits and 32 most significant bits respectively. T0VAL0 and T0VAL1 is read-only. In 16-bit mode 16-bit T0VAL0 is used. In 48-bit mode both 16-bit T0VAL0 and 32-bit T0VAL1 are used.

Timer0 Capture Register :

Name : T0CAP
Address : 0xFFFF0314

Default Value : 0x00
Access : Read Only
Function : This is a 16-bit register which holds the 16-bit value captured by an enabled IRQ event. Only available in 16-bit mode.

Timer0 Control Register :

Name : T0CON
Address : 0xFFFF030C

Default Value : 0x00
Access : Read/Write

Function : The 17-bit MMR configures the mode of operation of Timer0

Table 77 : T0CON MMR Bit Descriptions

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

Timer0 Load Registers:

Name : T0LD
Address : 0xFFFF0300

Default Value : 0x00
Access : Read/Write

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

Timer0 Clear Register :

Name : T0ICLR
Address : 0xFFFF0310

Default Value : 0x00
Access : Write Only
Function : This 8-bit, write-only MMR is written (with any value) by user code to refresh(reload) Timer0.

TIMER1

Timer1 is a 32-bit general purpose timer, count-down or count-up, with a programmable pre-scalar. The pre-scalar source can the 32kHz oscillator, the core clock, or from one of two external GPIO. This source can be scaled by a factor of 1, 16, 256 or 32768. This gives a minimum resolution of 42ns when operating at CD zero, the core is operating at 41.78MHz, and with a pre-scalar of 1 (Ignoring external GPIO).

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

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

Timer1 interface consists of five MMRS:

- **T1LD**, **T1VAL** and **T1CAP** are 32-bit registers and hold 32-bit unsigned integers. T1VAL and T1CAP are read-only.
- **T1ICLR** is an 8-bit register. Writing any value to this register will clear the timer1 interrupt.
- **T1CON** is the configuration MMR described in below.

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

Timer1 reloads the value from T1LD either when TIMER01 overflows, or immediately when T1ICLR is written.

Timer1 Load Registers:

Name : T1LD
Address : 0xFFFF0320
Default Value : 0x00000
Access : Read/Write
Function : T1LD is a 32 bit register which holds the 32 bit value that is loaded into the counter.

Timer1 Clear Register :

Name : T1ICLR
Address : 0xFFFF032C
Default Value : 0x00
Access : Write Only
Function : This 8-bit, write-only MMR is written (with any value) by user code to refresh(reload) Timer1.

Timer1 Value Register :

Name : T1VAL
Address : 0xFFFF0324
Default Value : 0x0000
Access : Read Only
Function : T1VAL is a 32-bit register which holds the current value of Timer1

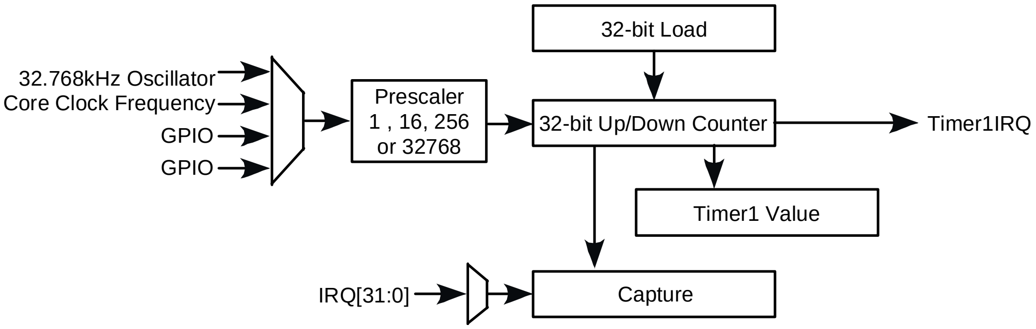


Figure 50 : Timer 1 Block Diagram

Timer1 Capture Register :

Name : T1CAP
Address : 0xFFFF0330
Default Value : 0x00
Access : Read Only
Function : This is a 32-bit register which holds the 32-bit value captured by an enabled IRQ event.

Timer1 Control Register :

Name : T1CON
Address : 0xFFFF0328
Default Value : 0x0000
Access : Read/Write
Function : This 32-bit MMR configures the mode of operation of Timer1

Table 78 : T1CON MMR Bit Descriptions

Bit	Description
31-18	Reserved Should be set to '0' by the user
17	Event Select bit: *Set* by user to enable time capture of an event *Cleared* by user to disable time capture of an event
16-12	Event select range, 0 to 31 The events are as described in the introduction to the timers.
11-9	Clock select: 000 Core clock (Default) 001 32.768kHz Oscillator 010 P1.0 011 P0.6
8	Count up: *Set* by user for timer 1 to count up *Cleared* by user for timer 1 to count down. (Default)
7	Timer1 enable bit: Set by user to enable timer 1 Cleared by user to disable timer 1. (Default)
6	Timer 1 mode: Set by user to operate in periodic mode Cleared by user to operate in free-running mode. (Default)
5-4	Format: 00 Binary (Default) 01 *Reserved* 10 Hr:Min:Sec:Hundredths – 23 hours to 0 hour 11 Hr:Min:Sec:Hundredths – 255 hours to 0 hour
3-0	Pre-Scalar: 0000 Source clock / 1 (Default) 0100 Source clock / 16 1000 Source clock / 256 1111 Source clock / 32768

TIMER2 - WAKE-UP TIMER

Timer2 is a 32-bit wake-up timer, count-down or count-up, with a programmable prescalar. The pre-scalar is clocked directly from 1 of 4 clock sources, namely, the Core Clock (default selection), the internal 32.768kHz Oscillator, External 32.768kHz Watch Crystal, or the core clock. The selected clock source can be scaled by a factor of 1, 16, 256 or 32768. The wake-up timer will continue to run when the core clock is disabled. This gives a minimum resolution of 22ns when operating at CD zero, the core is operating at 41.78MHz, and with a prescalar of 1. Capture of the current timer value is enabled if the Timer2 interrupt is enabled via IRQEN[4].

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

Timer2 reloads the value from T2LD either when TIMER2 overflows, or immediately when T2ICLR is written.

Timer2 interface consists of four MMRS:

- **T2LD** and **T2VAL** are 32-bit registers and hold 32-bit unsigned integers. T2VAL is read-only.
- **T2ICLR** is an 8-bit register. Writing any value to this register will clear the timer2 interrupt.
- **T2CON** is the configuration MMR described in Table 79.

Timer2 Load Registers:

Name : T2LD
Address : 0xFFFF0340
Default Value : 0x00000
Access : Read/Write

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

Timer2 Clear Register :

Name : T2ICLR
Address : 0xFFFF034C
Default Value : 0x00
Access : Write Only
Function : This 8-bit, write-only MMR is written (with any value) by user code to refresh(reload) Timer2.

Timer2 Value Register :

Name : T2VAL
Address : 0xFFFF0344
Default Value : 0x0000
Access : Read Only
Function : T2VAL is a 32-bit register which holds the current value of Timer2

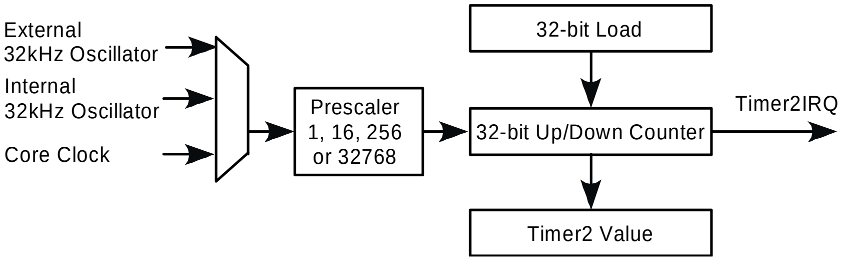


Figure 51 : Timer 2 block diagram

Timer2 Control Register :

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

Table 79 : T2CON MMR Bit Descriptions

Bit	Description
31-11	*Reserved*
10-9	Clock Source Select: 00 Core Clock (Default) 01 Inernal 32.768kHz Oscillator 10 External 32.768kHz Watch Crystal 11 External 32.768kHz Watch Crystal
8	Count up: Set by user for timer 2 to count up Cleared by user for timer 2 to count down. (Default)
7	Timer2 enable bit: Set by user to enable timer 2 Cleared by user to disable timer 2. (Default)
6	Timer 2 mode: Set by user to operate in periodic mode Cleared by user to operate in free-running mode. (Default)
5-4	Format: 00 Binary (Default) 01 *Reserved* 10 Hr:Min:Sec:Hundredths – 23 hours to 0 hour 11 Hr:Min:Sec:Hundredths – 255 hours to 0 hour
3-0	Prescalar: 0000 Source clock / 1 (Default) 0100 Source clock / 16 1000 Source clock / 256 (This setting should be used in conjunction Timer2 Formats 1,0 and 1,1) 1111 Source clock / 32768

TIMER3 - WATCHDOG TIMER

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

Timer3 reloads the value from T3LD either when TIMER3 overflows, or immediately when T3ICLR is written.

Normal mode:

The Timer3 in normal mode is identical to Timer0, in 16-bit mode of operation, except for the clock source. The clock source is the 32.768kHz oscillator and can be scaled by a factor of 1, 16, or 256. Timer3 also features a capture facility, which allows the capture of the current timer value if the Timer2 interrupt is enabled via IRQEN[5].

Watchdog mode:

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

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

If T3VAL reaches 0, a reset or an interrupt occurs, depending on T3CON[1]. To avoid a reset or an interrupt event, any value must be written to T3ICLR before T3VAL reaches zero. This reloads the counter with T3LD and begins a new timeout period.

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

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

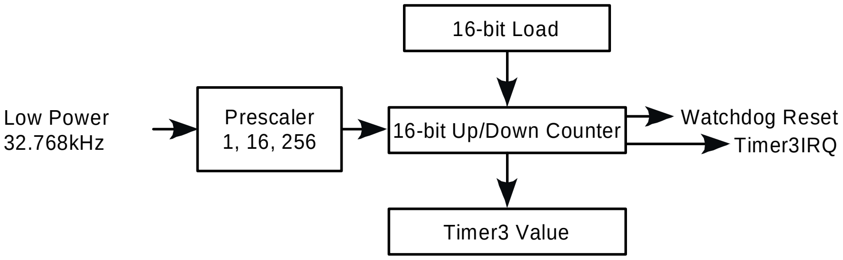


Figure 52 : Timer3 Block Diagram

Timer3 Interface:

Timer3 interface consists of four MMRS:

- **T3CON** is the configuration MMR described in Table 80
- **T3LD** and **T3VAL** are 16-bit registers (bit 0 to 15) and hold 16-bit unsigned integers. T3VAL is read-only.
- **T3ICLR** is an 8-bit register. Writing any value to this register will clear the Timer3 interrupt in normal mode or will reset a new timeout period in watchdog mode

Timer3 Load Register :

Name :	T3LD
Address :	0xFFFF0360
Default Value :	0x03D7
Access :	Read/Write
Function :	This 16-bit MMR holds the Timer3 reload value.

Timer3 Value Register :

Name : T3VAL
Address : 0xFFFF0364
Default Value : 0x03D7
Access : Read Only
Function : This 16-bit, read-only MMR holds the currentTimer3 count value.

Timer3 Clear Register :

Name : T3ICLR
Address : 0xFFFF036C
Default Value : 0x00
Access : Write Only
Function : This 8-bit, write-only MMR is written (with any value) by user code to refresh(reload) Timer3 in watchdog mode to prevent a watchdog timer reset event.

Timer3 Control Register :

Name : T3CON
Address : 0xFFFF0368
Default Value : 0x00
Access : Read/Write Once Only
Function : The 16-bit MMR configures the mode of operation of Timer3 as is described in detail in Table 80

Table 80 : T3CON MMR Bit Definition

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

Secure Bit Clear (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 Figure 53.

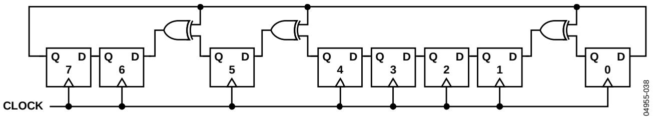


Figure 53. 8-Bit LFSR

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 can not 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.

TIMER4

Timer4 is a 32-bit general purpose timer, count-down or count-up, with a programmable pre-scalar. The pre-scalar source can the 32kHz oscillator, the core clock, or from one of two external GPIO. This source can be scaled by a factor of 1, 16, 256 or 32768. This gives a minimum resolution of 42ns when operating at CD zero, the core is operating at 41.78MHz, and with a pre-scalar of 1 (Ignoring external GPIO).

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

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

Timer4 interface consists of five MMRS:

- **T4LD**, **T4VAL** and **T4CAP** are 32-bit registers and hold 32-bit unsigned integers. T4VAL and T4CAP are read-only.
- **T4ICLR** is an 8-bit register. Writing any value to this register will clear the timer1 interrupt.
- **T4CON** is the configuration MMR described in Table 81.

NOTE: If the part is in a low power mode, and Timer4 is clocked from the GPIO or oscillator source then, Timer4 will continue to be operate.

Timer4 reloads the value from T4LD either when TIMER04 overflows, or immediately when T4ICLR is written.

Timer4 Load Registers:

Name : T4LD
Address : 0xFFFF0380
Default Value : 0x00000
Access : Read/Write

Function : T4LD is a 32 bit register which holds the 32 bit value that is loaded into the counter.

Timer4 Clear Register :

Name : T4ICLR
Address : 0xFFFF038C
Default Value : 0x00
Access : Write Only
Function : This 8-bit, write-only MMR is written (with any value) by user code to refresh(reload) Timer4.

Timer4Value Register :

Name : T4VAL
Address : 0xFFFF0384
Default Value : 0x0000
Access : Read Only
Function : T4VAL is a 32-bit register which holds the current value of Timer4

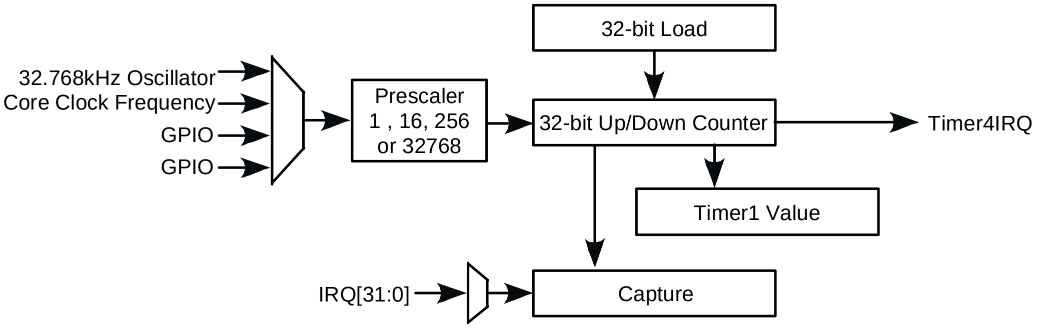


Figure 54 : Timer 4 Block Diagram

Timer4 Capture Register :

Name : T4CAP
Address : 0xFFFF0390
Default Value : 0x00
Access : Read Only
Function : This is a 32-bit register which holds the 32-bit value captured by an enabled IRQ event.

Timer4 Control Register :

Name : T4CON
Address : 0xFFFF0388
Default Value : 0x0000
Access : Read/Write
Function : This 32-bit MMR configures the mode of operation of Timer4

Table 81 : T4CON MMR Bit Descriptions

Bit	Description
31-18	Reserved Should be set to '0' by the user
17	Event Select bit: *Set* by user to enable time capture of an event *Cleared* by user to disable time capture of an event
16-12	Event select range, 0 to 31 The events are as described in the introduction to the timers.
11-9	Clock select: 000 Core clock (Default) 001 32.768kHz Oscillator 010 P4.6 011 P4.7
8	Count up: *Set* by user for timer 4 to count up *Cleared* by user for timer 4 to count down. (Default)
7	Timer 4 enable bit: Set by user to enable timer 4 Cleared by user to disable timer 4. (Default)
6	Timer 4 mode: Set by user to operate in periodic mode Cleared by user to operate in free-running mode. (Default)
5-4	Format: 00 Binary (Default) 01 *Reserved* 10 Hr:Min:Sec:Hundredths – 23 hours to 0 hour 11 Hr:Min:Sec:Hundredths – 255 hours to 0 hour
3-0	Pre-Scalar: 0000 Source clock / 1 (Default) 0100 Source clock / 16 1000 Source clock / 256 1111 Source clock / 32768

ADuC7128 Hardware Design considerations

POWER SUPPLIES

The ADuC7128 operational power supply voltage range is 3.0V to 3.6V. Separate analog and digital power supply pins (AVDD and IOVDD, respectively) allow AVDD to be kept relatively free of noisy digital signals often present on the system IOVDD line. In this mode, the part can also operate with split supplies; that is, using different voltage supply levels for each supply. For example, this means that the system can be designed to operate with a IOVDD voltage level of 3.3 V while the AVDD level can be at 3 V, or vice versa if required. A typical split supply configuration is shown in Figure 55.

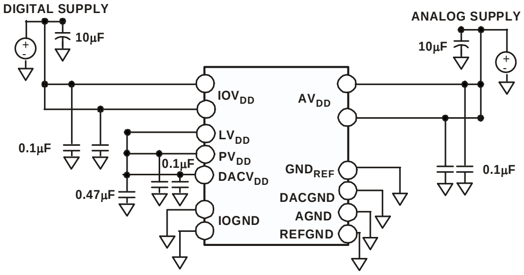


Figure 55: External dual supply connections

As an alternative to providing two separate power supplies, the user can help keep AVDD quiet by placing a small series resistor and/or ferrite bead between it and IOVDD, and then decoupling AVDD separately to ground. An example of this configuration is shown in Figure 56. With this configuration other analog circuitry (such as op amps, voltage reference, and so on) can be powered from the AVDD supply line as well.

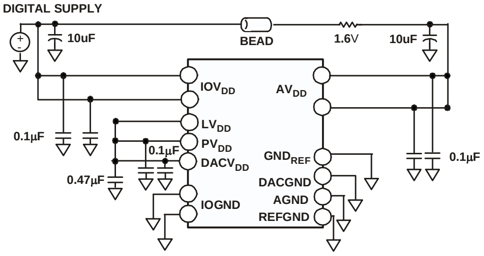


Figure 56: external single supply connections

Notice that in both Figure 56 and Figure 57, a large value (10 µF) reservoir capacitor sits on IOVDD and a separate 10 µF capacitor sits on AVDD. Also, local small-value (0.1 µF) capacitors are located at each AVDD and IOVDD pin of the chip. As per standard design practice, be sure to include all of these capacitors, and ensure the smaller capacitors are close to each AVDD pin with trace lengths as short as possible. Connect the ground terminal of each of these capacitors directly to the underlying ground plane. It should also be noted that, at all times, the analog and digital ground pins on the ADuC7128 must be referenced to the same system ground reference point.

Finally, on the CSP package, the paddle on the bottom of the package should be soldered to a metal plate to provide mechanical stability. This metal plate should be connected to ground.

Linear Voltage regulator

The ADuC7128 requires a single 3.3V supply but the core logic requires a 2.5V supply. An on-chip linear regulator generates the 2.5V from IOVDD for the core logic. LVDD pin 21 is the 2.5V supply for the core logic. The DAC logic and PLL logic also require a 2.5V supply, this must be connected externally from the LVdd pin to the DACVdd and PVdd pins. An external compensation capacitor of 0.47 µF must be connected between LVDD and DGND (as close as possible to these pins) to act as a tank of charge as shown inFigure 57. 0.1 µF decoupling capacitors also must be placed as close as possible to the PVdd and DACVdd pins.

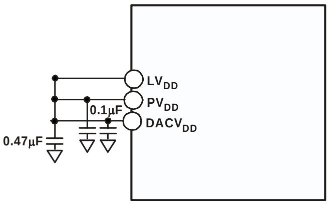


Figure 57: voltage regulator connections

The LVDD pin should not be used for any other chip. It is also recommended that the IOVDD has excellent power supply decoupling this to help improving 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 ADuC7128-based designs in order to achieve optimum performance from the ADCs and DAC.

Although the ADuC7128 has separate pins for analog and digital ground (AGND and IOGND), the user must not tie these to two separate ground planes unless the two ground planes are connected together very close to the ADuC7128, as illustrated in the simplified example of Figure 58a. In systems where digital and analog ground planes are connected together somewhere else (at the system's power supply for example), they cannot be connected again near the ADuC7128 since a ground loop would result. In these cases, tie the ADuC7128's AGND and IOGND Pins all to the analog ground plane, as illustrated in Figure 58b. In systems with only one ground plane, ensure that the digital and analog components are physically separated onto separate halves of the board such that digital return currents do not flow near analog circuitry and vice versa. The ADuC7128 can then be placed between the digital and analog sections, as illustrated in Figure 58c.

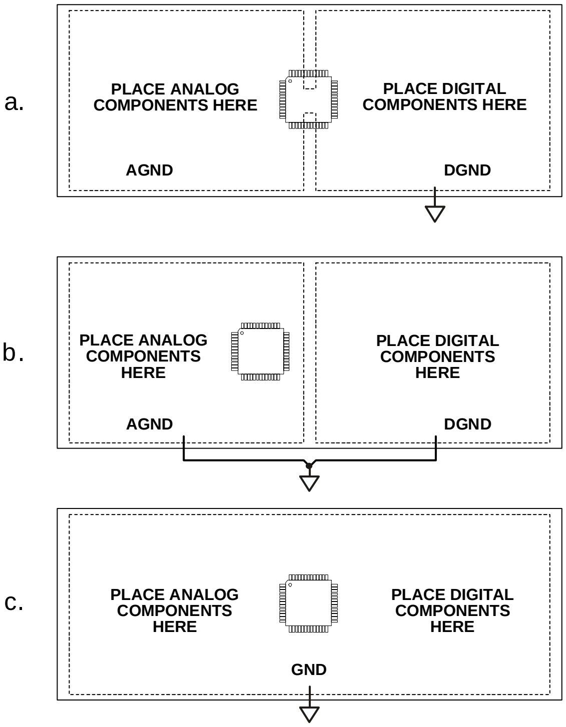

a.
PLACE ANALOG COMPONENTS HERE
PLACE DIGITAL COMPONENTS HERE
AGND
DGND
b.
PLACE ANALOG COMPONENTS HERE
PLACE DIGITAL COMPONENTS HERE
AGND
DGND
c.
PLACE ANALOG COMPONENTS HERE
PLACE DIGITAL COMPONENTS HERE
GND

Figure 58:. System grounding schemes

In all of these scenarios, and in more complicated real-life applications, keep in mind 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 of Figure 58b with IOVDD since that would force return currents from IOVDD to flow through AGND. Also, try to avoid digital currents flowing under analog circuitry, which could happen if the user placed a noisy digital chip on the left half of the board in Figure 58c. Whenever possible, avoid large discontinuities in the ground plane(s) (such as are formed by a long trace on the same layer), since they force return signals to travel a longer path. And of course, make all connections to the ground plane directly, with little or no trace separating the pin from its via to ground.

If the user plans to connect fast logic signals (rise/fall time < 5 ns) to any of the ADuC7128's digital inputs, add a series resistor to each relevant line to keep rise and fall times longer than 5 ns at the ADuC7128 input pins. A value of 100Ω or 200Ω is usually sufficient to prevent high speed signals from coupling capacitively into the ADuC7128 and affecting the accuracy of ADC conversions.

CLOCK OSCILLATOR

The clock source for the ADuC7128 can be generated by the internal PLL or by an external clock input. To use the internal PLL, connect a 32.768kHz parallel resonant crystal between XCLKI and XCLKO as shown Figure 59. External capacitors should be connected as per the crystal manufacturer's recommendations. Note that the crystal pads already have an internal capacitance of typically 10pF. User should ensure that the total capacitance (10pF internal + external capacitance) doesn't exceed the manufacturer rating.

This 32kHz crystal allows the PLL to lock correctly to give a frequency of 41.78 MHz. If no external crystal is present, the internal oscillator will be used to give a frequency of 41.78MHz ±3% typically.

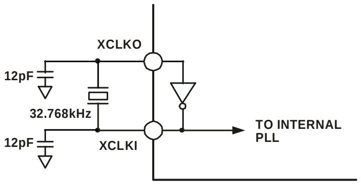


Figure 59: external parallel resonant crystal connections

To use an external source clock input instead of the PLL, bit 1 and bit 0 of PLLCON must be modified. The external clock uses pin 17, XCLK.

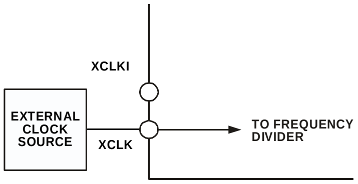


Figure 60:connecting an external clock source

Whether using the internal PLL or an external clock source, the ADuC7128's specified operational clock speed range is 50kHz to 41.78 MHz to ensure correct operation of the analog peripherals and Flash/EE.

POWER-ON RESET OPERATION

An internal POR (Power-On Reset) is implemented on the ADuC7128. For LVDD below 2.45 V, the internal POR will hold the ADuC7128 in reset. As LVDD rises above 2.45 V, an internal timer will time out for typically 64 ms before the part is released from reset. The user must ensure that the power supply IOVDD has reached a stable 3.0 V minimum level by this time. Likewise on power-down, the internal POR will hold the ADuC7128 in reset until LVDD has dropped below 2.45V. Figure 61 illustrates the operation of the internal POR in detail.

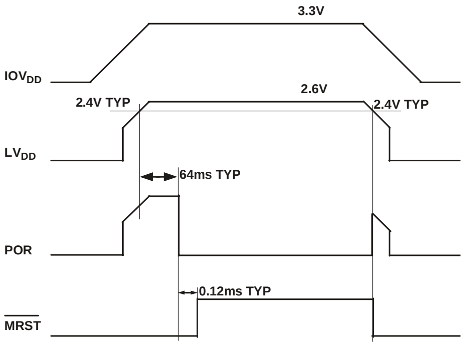


Figure 61:. ADuC7128 Internal Power-on-Reset operation

TYPICAL SYSEM CONFIGURATION

A typical ADuC7128 configuration is shown in Figure 62. It summarizes some of the hardware considerations discussed in the previous paragraphs.

Figure 62:. Typical System Configuration

DEVELOPMENT TOOLS

An entry level, low cost development system is available for the ADuC7128 family. This system consists of the following PC-based (Windows® compatible) hardware and software development tools:

Hardware:

- ADuC7128 Evaluation board
- Serial Port programming cable
- JTAG emulator

Software:

- Integrated Development Environment, incorporating assembler, compiler and non intrusive 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.

OUTLINE DIMENSIONS

On the CSP package, the paddle on the bottom of the package should be soldered to a metal plate to provide mechanical stability. This metal plate should be connected to ground.

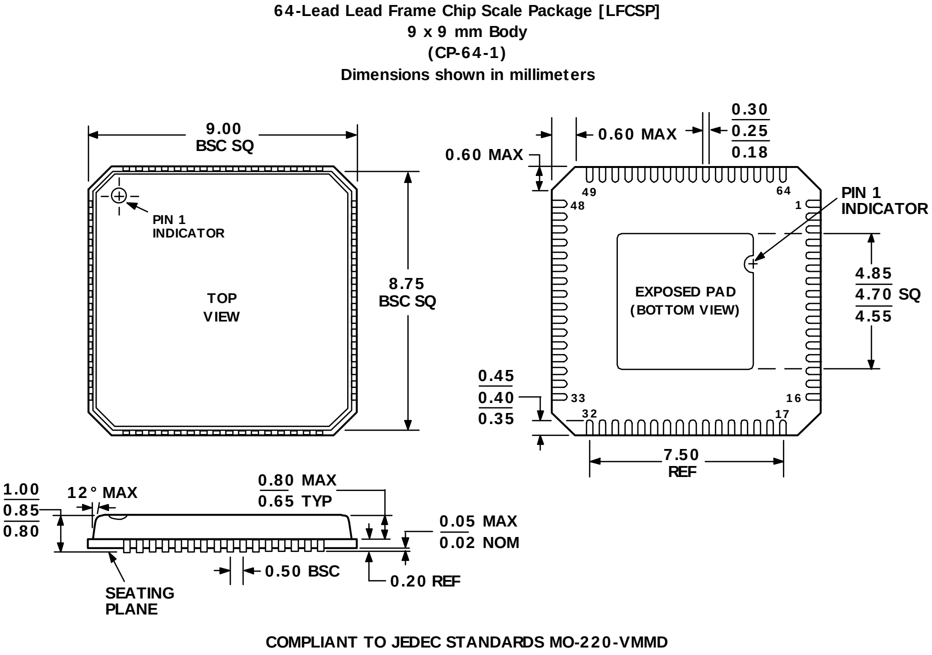


Figure 63. 64-Lead Frame Chip Scale Package [LFCSP] (CP-64-1)—Dimensions shown in millimetres

NOTES

ANALOG DEVICES

www.analog.com