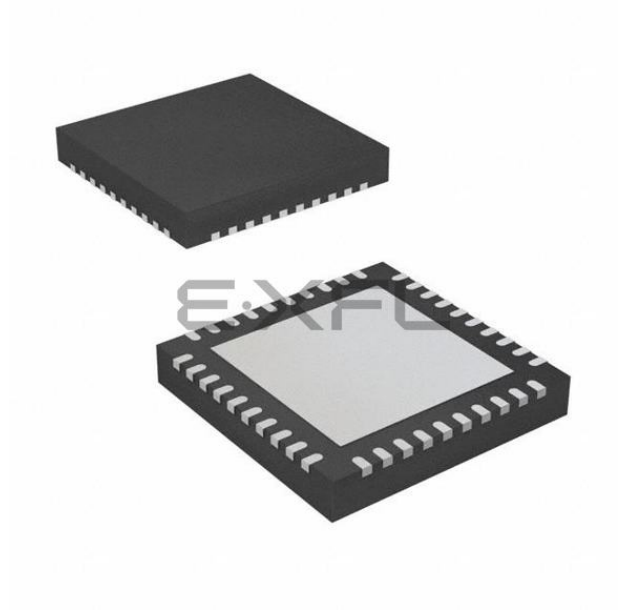


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	ARM7®
Core Size	16/32-Bit
Speed	44MHz
Connectivity	EBI/EMI, I²C, SPI, UART/USART
Peripherals	PLA, PWM, PSM, Temp Sensor, WDT
Number of I/O	14
Program Memory Size	62KB (31K x16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	2K x 32
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 5x12b; D/A 4x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	40-WFQFN Exposed Pad, CSP
Supplier Device Package	40-LFCSP-WQ (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7020bcpz62i-rl

TABLE OF CONTENTS

GENERAL DESCRIPTION

The ADuC7019/20/21/22/24/25/26/27/28/29 are fully integrated, 1 MSPS, 12-bit data acquisition systems incorporating high performance multichannel ADCs, 16-bit/32-bit MCUs, and Flash®/EE memory on a single chip.

The ADC consists of up to 12 single-ended inputs. An additional four inputs are available but are multiplexed with the four DAC output pins. The four DAC outputs are available only on certain models (ADuC7020, ADuC7026, ADuC7028, and ADuC7029). However, in many cases where the DAC outputs are not present, these pins can still be used as additional ADC inputs, giving a maximum of 16 ADC input channels. The ADC can operate in single-ended or differential input mode. The ADC input voltage is 0 V to V_{REF}. A low drift band gap reference, temperature sensor, and voltage comparator complete the ADC peripheral set.

Depending on the part model, up to four buffered voltage output DACs are available on-chip. The DAC output range is programmable to one of three voltage ranges.

The devices operate from an on-chip oscillator and a PLL generating an internal high frequency clock of 41.78 MHz (UCLK). This clock is routed through a programmable clock divider from which the MCU core clock operating frequency is generated. The microcontroller core is an ARM7TDMI®, 16-bit/32-bit RISC machine, which offers up to 41 MIPS peak performance. Eight kilobytes of SRAM and 62 kilobytes of nonvolatile Flash/EE memory are provided on-chip. The ARM7TDMI core views all memory and registers as a single linear array.

On-chip factory firmware supports in-circuit serial download via the UART or I²C serial interface port; nonintrusive emulation is also supported via the JTAG interface. These features are incorporated into a low cost QuickStart™ development system supporting this MicroConverter® family.

The parts operate from 2.7 V to 3.6 V and are specified over an industrial temperature range of −40°C to +125°C. When operating at 41.78 MHz, the power dissipation is typically 120 mW. The ADuC7019/20/21/22/24/25/26/27/28/29 are available in a variety of memory models and packages (see Ordering Guide).

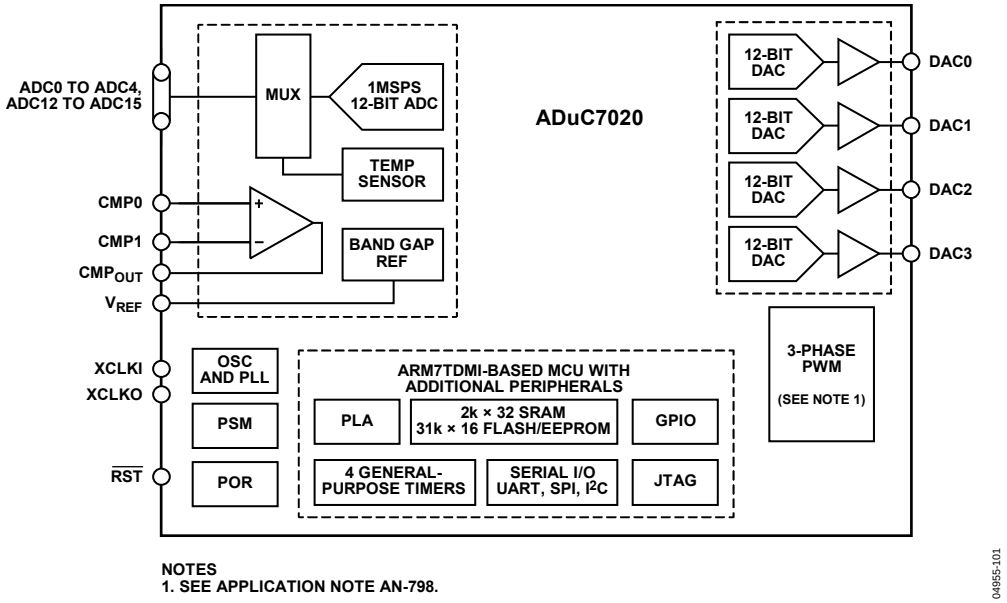


Figure 2.

Table 3. External Memory Read Cycle

Parameter	Min	Typ	Max	Unit
CLK[1]	1/MD clock	ns typ × (POWCON[2:0] + 1)		
$t_{MS\_AFTER\_CLKH}$	4		8	ns
$t_{ADDR\_AFTER\_CLKH}$	4		16	ns
$t_{AE\_H\_AFTER\_MS}$		½ CLK		
t_{AE}		(XMxPAR[14:12] + 1) × CLK		
$t_{HOLD\_ADDR\_AFTER\_AE\_L}$		½ CLK + (! XMxPAR[10]) × CLK		
$t_{RD\_L\_AFTER\_AE\_L}$		½ CLK + (! XMxPAR[10]+ ! XMxPAR[9]) × CLK		
$t_{RD\_H\_AFTER\_CLKH}$	0		4	
t_{RD}		(XMxPAR[3:0] + 1) × CLK		
$t_{DATA\_BEFORE\_RD\_H}$	16			ns
$t_{DATA\_AFTER\_RD\_H}$	8	+ (! XMxPAR[9]) × CLK		
$t_{RELEASE\_MS\_AFTER\_RD\_H}$		1 × CLK		

[1] See Table 78.

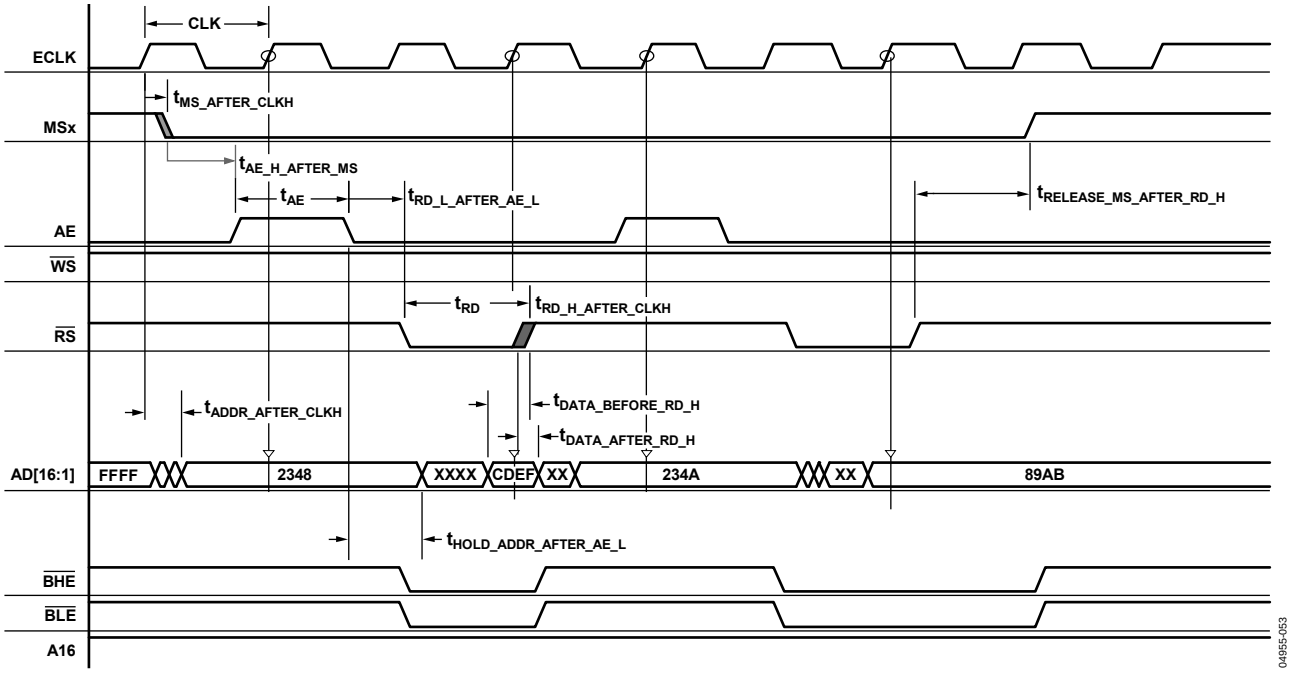


Figure 13. External Memory Read Cycle (See Table 78)

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

		Slave		Master	
Parameter	**Description**	**Min**	**Max**	**Typ**	**Unit**
t_L	SCL low pulse width[1]	200		1360	ns
t_H	SCL high pulse width[1]	100		1140	ns
t_{SHD}	Start condition hold time	300			ns
t_{DSU}	Data setup time	100		740	ns
t_{DHD}	Data hold time	0		400	ns
t_{RSU}	Setup time for repeated start	100			ns
t_{PSU}	Stop condition setup time	100		400	ns
t_{BUF}	Bus-free time between a stop condition and a start condition	1.3			µs
t_R	Rise time for both SCL and SDA		300	200	ns
t_F	Fall time for both SCL and SDA		300		ns
t_{SUP}	Pulse width of spike suppressed		50		ns

[1] t_{HCLK} depends on the clock divider or CD bits in the POWCON MMR. $t_{HCLK} = t_{UCLK}/2^{CD}$; see Figure 67.

Table 5. I²C Timing in Standard Mode (100 kHz)

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

[1] t_{HCLK} depends on the clock divider or CD bits in the POWCON MMR. $t_{HCLK} = t_{UCLK}/2^{CD}$; see Figure 67.

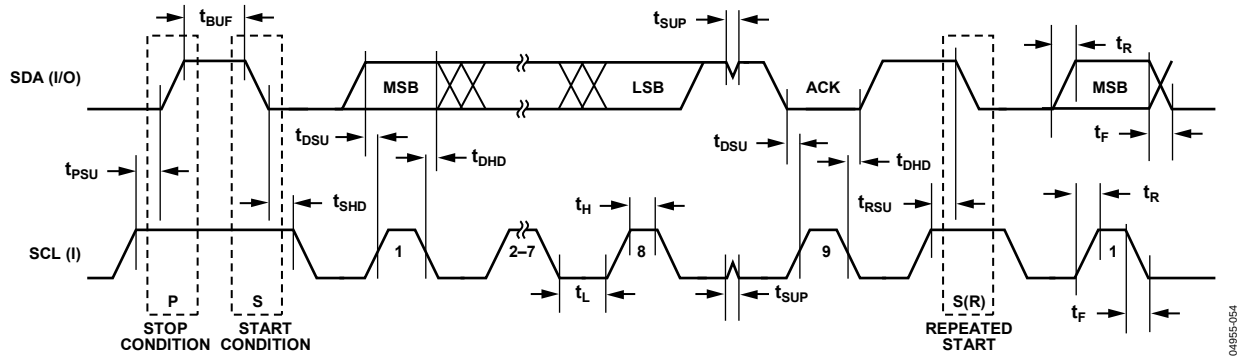


Figure 14. I²C Compatible Interface Timing

ABSOLUTE MAXIMUM RATINGS

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

Table 10.

Parameter	Rating
AV_{DD} to IOV_{DD}	−0.3 V to +0.3 V
AGND to DGND	−0.3 V to +0.3 V
IOV_{DD} to IOGND, AV_{DD} to AGND	−0.3 V to +6 V
Digital Input Voltage to IOGND	−0.3 V to +5.3 V
Digital Output Voltage to IOGND	−0.3 V to IOV_{DD} + 0.3 V
V_{REF} to AGND	−0.3 V to AV_{DD} + 0.3 V
Analog Inputs to AGND	−0.3 V to AV_{DD} + 0.3 V
Analog Outputs to AGND	−0.3 V to AV_{DD} + 0.3 V
Operating Temperature Range, Industrial	–40°C to +125°C
Storage Temperature Range	–65°C to +150°C
Junction Temperature	150°C
θ_{JA} Thermal Impedance	
40-Lead LFCSP	26°C/W
49-Ball CSP_BGA	80°C/W
64-Lead LFCSP	24°C/W
64-Ball CSP_BGA	75°C/W
64-Lead LQFP	47°C/W
80-Lead LQFP	38°C/W
Peak Solder Reflow Temperature	
SnPb Assemblies (10 sec to 30 sec)	240°C
RoHS Compliant Assemblies (20 sec to 40 sec)	260°C

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

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

ESD CAUTION

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

Pin No.				
7019/7020	7021	7022	Mnemonic	Description
22	22	21	P2.0/SPM9/PLAO[5]/$\overline{CONV_{START}}$	Serial Port Multiplexed. General-Purpose Input and Output Port 2.0/UART/ Programmable Logic Array Output Element 5/Start Conversion Input Signal for ADC.
23	23	22	P0.7/ECLK/XCLK/SPM8/PLAO[4]	Serial Port Multiplexed. General-Purpose Input and Output Port 0.7/ Output for External Clock Signal/Input to the Internal Clock Generator Circuits/UART/ Programmable Logic Array Output Element 4.
24	24	23	XCLKO	Output from the Crystal Oscillator Inverter.
25	25	24	XCLKI	Input to the Crystal Oscillator Inverter and Input to the Internal Clock Generator Circuits.
26	26	25	P1.7/SPM7/PLAO[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.7/UART, SPI/Programmable Logic Array Output Element 0.
27	27	26	P1.6/SPM6/PLAI[6]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.6/UART, SPI/Programmable Logic Array Input Element 6.
28	28	27	P1.5/SPM5/PLAI[5]/IRQ3	Serial Port Multiplexed. General-Purpose Input and Output Port 1.5/UART, SPI/Programmable Logic Array Input Element 5/External Interrupt Request 3, Active High.
29	29	28	P1.4/SPM4/PLAI[4]/IRQ2	Serial Port Multiplexed. General-Purpose Input and Output Port 1.4/UART, SPI/Programmable Logic Array Input Element 4/External Interrupt Request 2, Active High.
30	30	29	P1.3/SPM3/PLAI[3]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.3/UART, I2C1/Programmable Logic Array Input Element 3.
31	31	30	P1.2/SPM2/PLAI[2]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.2/UART, I2C1/Programmable Logic Array Input Element 2.
32	32	31	P1.1/SPM1/PLAI[1]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.1/UART, I2C0/Programmable Logic Array Input Element 1.
33	33	32	P1.0/T1/SPM0/PLAI[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.0/ Timer1 Input/UART, I2C0/Programmable Logic Array Input Element 0.
34	–	–	P4.2/PLAO[10]	General-Purpose Input and Output Port 4.2/Programmable Logic Array Output Element 10.
35	34	33	V_{REF}	2.5 V Internal Voltage Reference. Must be connected to a 0.47 µF capacitor when using the internal reference.
36	35	34	AGND	Analog Ground. Ground reference point for the analog circuitry.
37	36	35	AV_{DD}	3.3 V Analog Power.
0	0	0	EP	Exposed Pad. The pin configuration for the ADuC7019/ADuC7020/ ADuC7021/ADuC7022 has an exposed pad that must be soldered for mechanical purposes and left unconnected.

Pin No.	Mnemonic	Description
46	P3.6/AD6/PWM_{TRIP}/PLAI[14]	General-Purpose Input and Output Port 3.6/External Memory Interface/PWM Safety Cutoff/ Programmable Logic Array Input Element 14.
47	P3.7/AD7/PWM_{SYNC}/PLAI[15]	General-Purpose Input and Output Port 3.7/External Memory Interface/PWM Synchronization/ Programmable Logic Array Input Element 15.
48	P2.7/$PWM1_L$/MS3	General-Purpose Input and Output Port 2.7/PWM Phase 1 Low-Side Output/External Memory Select 3.
49	P2.1/$\overline{WS}$/$PWM0_H$/PLAO[6]	General-Purpose Input and Output Port 2.1/External Memory Write Strobe/PWM Phase 0 High-Side Output/Programmable Logic Array Output Element 6.
50	P2.2/$\overline{RS}$/$PWM0_L$/PLAO[7]	General-Purpose Input and Output Port 2.2/External Memory Read Strobe/PWM Phase 0 Low-Side Output/Programmable Logic Array Output Element 7.
51	P1.7/SPM7/PLAO[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.7/UART, SPI/Programmable Logic Array Output Element 0.
52	P1.6/SPM6/PLAI[6]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.6/UART, SPI/Programmable Logic Array Input Element 6.
53	IOGND	Ground for GPIO (see Table 78). Typically connected to DGND.
54	IOV_{DD}	3.3 V Supply for GPIO (see Table 78) and Input of the On-Chip Voltage Regulator.
55	P4.0/AD8/PLAO[8]	General-Purpose Input and Output Port 4.0/External Memory Interface/Programmable Logic Array Output Element 8.
56	P4.1/AD9/PLAO[9]	General-Purpose Input and Output Port 4.1/External Memory Interface/Programmable Logic Array Output Element 9.
57	P1.5/SPM5/PLAI[5]/IRQ3	Serial Port Multiplexed. General-Purpose Input and Output Port 1.5/UART, SPI/Programmable Logic Array Input Element 5/External Interrupt Request 3, Active High.
58	P1.4/SPM4/PLAI[4]/IRQ2	Serial Port Multiplexed. General-Purpose Input and Output Port 1.4/UART, SPI/Programmable Logic Array Input Element 4/External Interrupt Request 2, Active High.
59	P1.3/SPM3/PLAI[3]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.3/UART, I2C1/Programmable Logic Array Input Element 3.
60	P1.2/SPM2/PLAI[2]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.2/UART, I2C1/Programmable Logic Array Input Element 2.
61	P1.1/SPM1/PLAI[1]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.1/UART, I2C0/Programmable Logic Array Input Element 1.
62	P1.0/T1/SPM0/PLAI[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.0/Timer1 Input/UART, I2C0/ Programmable Logic Array Input Element 0.
63	P4.2/AD10/PLAO[10]	General-Purpose Input and Output Port 4.2/External Memory Interface/Programmable Logic Array Output Element 10.
64	P4.3/AD11/PLAO[11]	General-Purpose Input and Output Port 4.3/External Memory Interface/Programmable Logic Array Output Element 11.
65	P4.4/AD12/PLAO[12]	General-Purpose Input and Output Port 4.4/External Memory Interface/Programmable Logic Array Output Element 12.
66	P4.5/AD13/PLAO[13]	General-Purpose Input and Output Port 4.5/External Memory Interface/Programmable Logic Array Output Element 13.
67	REFGND	Ground for the Reference. Typically connected to AGND.
68	V_{REF}	2.5 V Internal Voltage Reference. Must be connected to a 0.47 µF capacitor when using the internal reference.
69	DAC_{REF}	External Voltage Reference for the DACs. Range: DACGND to $DACV_{DD}$.
70	DACGND	Ground for the DAC. Typically connected to AGND.
71, 72	AGND	Analog Ground. Ground reference point for the analog circuitry.
73, 74	AV_{DD}	3.3 V Analog Power.
75	$DACV_{DD}$	3.3 V Power Supply for the DACs. Must be connected to AV_{DD}.
76	ADC11	Single-Ended or Differential Analog Input 11.
77	ADC0	Single-Ended or Differential Analog Input 0.
78	ADC1	Single-Ended or Differential Analog Input 1.
79	ADC2/CMP0	Single-Ended or Differential Analog Input 2/Comparator Positive Input.
80	ADC3/CMP1	Single-Ended or Differential Analog Input 3/Comparator Negative Input.

Pin No.	Mnemonic	Description
D7	P1.6/SPM6/PLAI[6]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.6/UART, SPI/Programmable Logic Array Input Element 6.
D8	IOV_{DD}	3.3 V Supply for GPIO (see Table 78) and Input of the On-Chip Voltage Regulator.
E1	DAC3/ADC15	DAC3 Voltage Output/ADC Input 15.
E2	DAC2/ADC14	DAC2 Voltage Output/ADC Input 14.
E3	DAC1/ADC13	DAC1 Voltage Output/ADC Input 13.
E4	P3.0/$PWM0_H$/PLAI[8]	General-Purpose Input and Output Port 3.0/PWM Phase 0 High-Side Output/Programmable Logic Array Input Element 8.
E5	P3.2/$PWM1_H$/PLAI[10]	General-Purpose Input and Output Port 3.2/PWM Phase 1 High-Side Output/Programmable Logic Array Input Element 10.
E6	P1.5/SPM5/PLAI[5]/IRQ3	Serial Port Multiplexed. General-Purpose Input and Output Port 1.5/UART, SPI/Programmable Logic Array Input Element 5/External Interrupt Request 3, Active High.
E7	P3.7/PWM_{SYNC}/PLAI[15]	General-Purpose Input and Output Port 3.7/PWM Synchronization/Programmable Logic Array Input Element 15.
E8	XCLKI	Input to the Crystal Oscillator Inverter and Input to the Internal Clock Generator Circuits.
F1	P4.6/PLAO[14]	General-Purpose Input and Output Port 4.6/Programmable Logic Array Output Element 14.
F2	TDI	JTAG Test Port Input, Test Data In. Debug and download access.
F3	DAC0/ADC12	DAC0 Voltage Output/ADC Input 12.
F4	P3.1/$PWM0_L$/PLAI[9]	General-Purpose Input and Output Port 3.1/PWM Phase 0 Low-Side Output/Programmable Logic Array Input Element 9.
F5	P3.3/$PWM1_L$/PLAI[11]	General-Purpose Input and Output Port 3.3/PWM Phase 1 Low-Side Output/Programmable Logic Array Input Element 11.
F6	$\overline{RST}$	Reset Input, Active Low.
F7	P0.7/ECLK/XCLK/SPM8/PLAO[4]	Serial Port Multiplexed. General-Purpose Input and Output Port 0.7/Output for External Clock Signal/Input to the Internal Clock Generator Circuits/UART/Programmable Logic Array Output Element 4.
F8	XCLKO	Output from the Crystal Oscillator Inverter.
G1	BM/P0.0/CMP_{OUT}/PLAI[7]	Multifunction I/O Pin. Boot mode. The ADuC7028 enters UART download mode if BM is low at reset and executes code if BM is pulled high at reset through a 1 kΩ resistor/General-Purpose Input and Output Port 0.0/Voltage Comparator Output/Programmable Logic Array Input Element 7.
G2	P4.7/PLAO[15]	General-Purpose Input and Output Port 4.7/Programmable Logic Array Output Element 15.
G3	TMS	JTAG Test Port Input, Test Mode Select. Debug and download access.
G4	TDO	JTAG Test Port Output, Test Data Out. Debug and download access.
G5	P0.3/TRST/ADC_{BUSY}	General-Purpose Input and Output Port 0.3/JTAG Test Port Input, Test Reset/ADC_{BUSY} Signal Output.
G6	P3.4/$PWM2_H$/PLAI[12]	General-Purpose Input and Output Port 3.4/PWM Phase 2 High-Side Output/Programmable Logic Array Input 12.
G7	P3.5/$PWM2_L$/PLAI[13]	General-Purpose Input and Output Port 3.5/PWM Phase 2 Low-Side Output/Programmable Logic Array Input Element 13.
G8	P2.0/SPM9/PLAO[5]/$\overline{CONV_{START}}$	Serial Port Multiplexed. General-Purpose Input and Output Port 2.0/UART/Programmable Logic Array Output Element 5/Start Conversion Input Signal for ADC.
H1	P0.6/T1/MRST/PLAO[3]	Multifunction Pin, Driven Low After Reset. General-Purpose Output Port 0.6/Timer1 Input/Power-On Reset Output/Programmable Logic Array Output Element 3.
H2	TCK	JTAG Test Port Input, Test Clock. Debug and download access.
H3	IOGND	Ground for GPIO (see Table 78). Typically connected to DGND.
H4	IOV_{DD}	3.3 V Supply for GPIO (see Table 78) and Input of the On-Chip Voltage Regulator.
H5	LV_{DD}	2.6 V Output of the On-Chip Voltage Regulator. This output must be connected to a 0.47 µF capacitor to DGND only.
H6	DGND	Ground for Core Logic.
H7	IRQ0/P0.4/PWM_{TRIP}/PLAO[1]	Multifunction I/O Pin. External Interrupt Request 0, Active High/General-Purpose Input and Output Port 0.4/PWM Trip External Input/Programmable Logic Array Output Element 1.
H8	IRQ1/P0.5/ADC_{BUSY}/PLAO[2]	Multifunction I/O Pin. External Interrupt Request 1, Active High/General-Purpose Input and Output Port 0.5/ADC_{BUSY} Signal Output/Programmable Logic Array Output Element 2.

TYPICAL PERFORMANCE CHARACTERISTICS

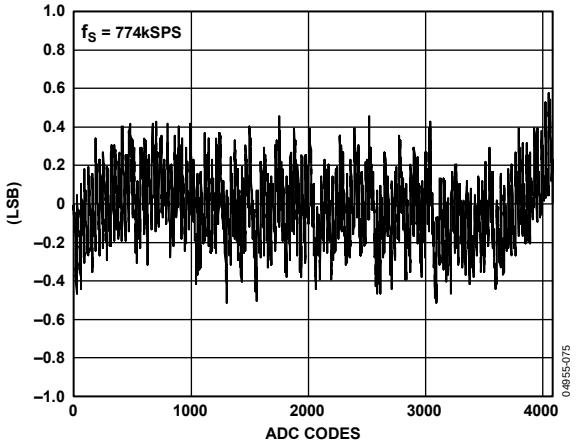


Figure 28. Typical INL Error, f_S = 774 kSPS

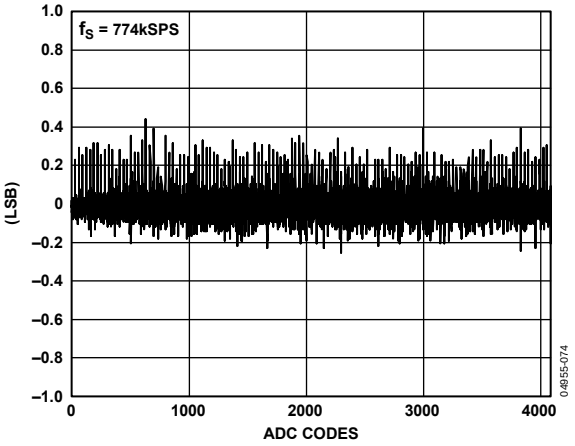


Figure 31. Typical DNL Error, f_S = 774 kSPS

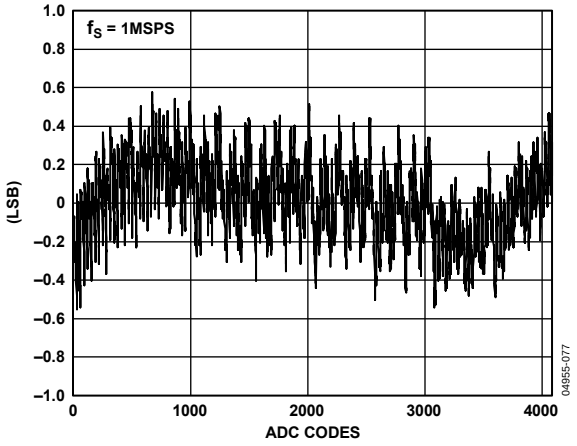


Figure 29. Typical INL Error, f_S = 1 MSPS

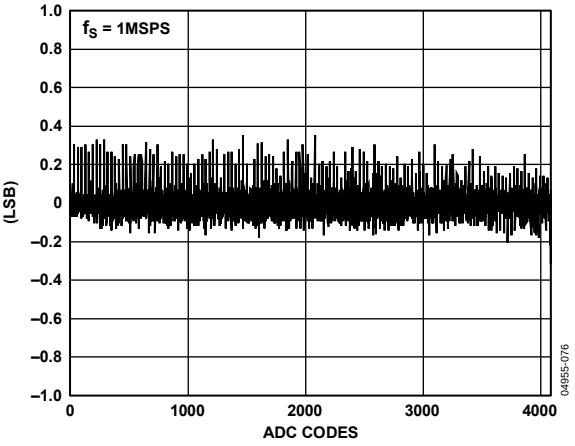


Figure 32. Typical DNL Error, f_S = 1 MSPS

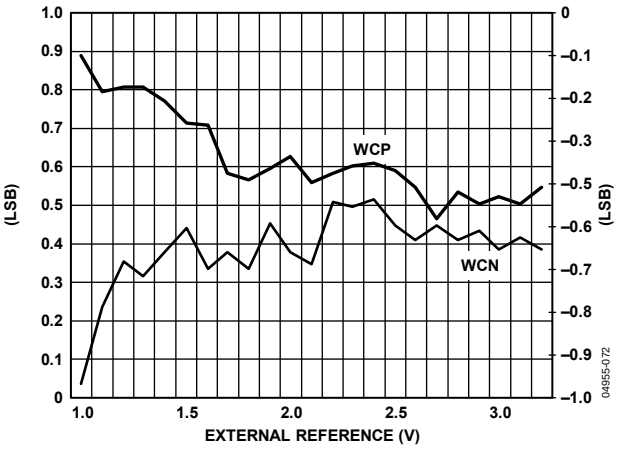


Figure 30. Typical Worst-Case (Positive (WCP) and Negative (WCN)) INL Error vs. V_{REF}, f_S = 774 kSPS

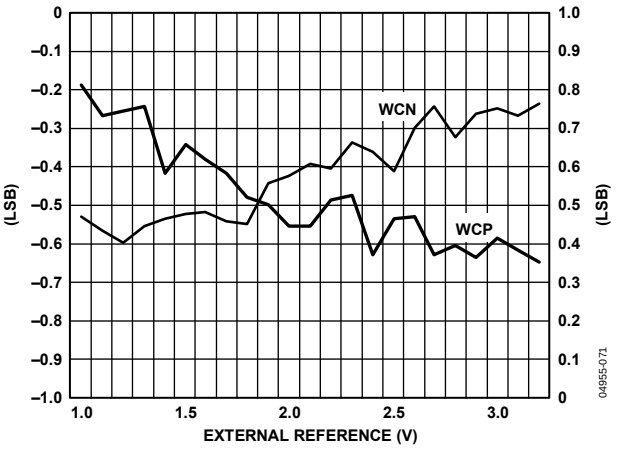


Figure 33. Typical Worst-Case (Positive (WCP)and Negative (WCN)) DNL Error vs. V_{REF}, f_S = 774 kSPS

OVERVIEW OF THE ARM7TDMI CORE

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

The ARM7TDMI is an ARM7 core with four additional features.

- T support for the thumb (16-bit) instruction set.
- D support for debug.
- M support for long multiplications.
- I includes the EmbeddedICE module to support embedded system debugging.

THUMB MODE (T)

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

However, the thumb mode has two limitations.

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

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

LONG MULTIPLY (M)

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

EmbeddedICE (I)

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

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

EXCEPTIONS

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

- Normal interrupt or IRQ, which is provided to service general-purpose interrupt handling of internal and external events.
- Fast interrupt or FIQ, which is provided to service data transfers or communication channels with low latency. FIQ has priority over IRQ.
- Memory abort.
- Attempted execution of an undefined instruction.
- Software interrupt instruction (SWI), which 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 used for system-level programming and exception handling only.

When an exception occurs, some of the standard registers are replaced with registers specific to the exception mode. All exception modes have replacement banked registers for the stack pointer (R13) and the link register (R14), as represented in Figure 44. The fast interrupt mode has more registers (R8 to R12) for fast interrupt processing. This means that interrupt processing can begin without the need to save or restore these registers and, thus, save critical time in the interrupt handling process.

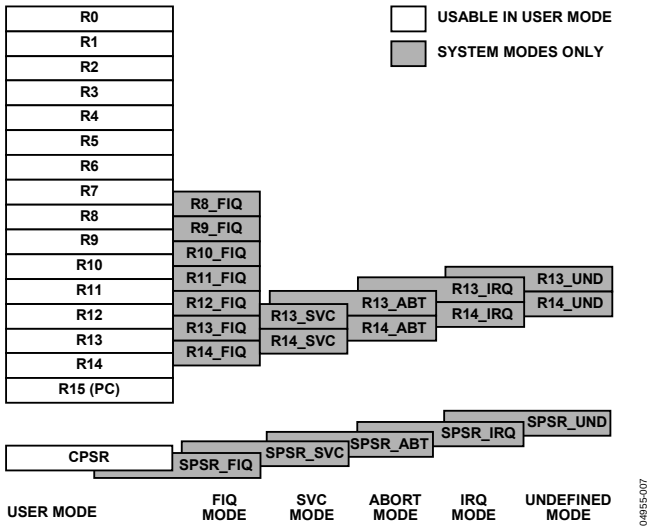


Figure 44. Register Organization

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

$$t_{D(max)} = 1023 \times 2 \times t_{CORE} = 1023 \times 2 \times 24 \times 10^{-9} = 48.97\ \mu s$$

for a core clock of 41.78 MHz.

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

PWM Operating Mode (PWMCON and PWMSTA MMRs)

As discussed in the 3-Phase PWM section, the PWM controller of the ADuC7019/20/21/22/24/25/26/27/28/29 can operate in two distinct modes: single update mode and double update mode. The operating mode of the PWM controller is determined by the state of Bit 2 of the PWMCON register. If this bit is cleared, the PWM operates in the single update mode. Setting Bit 2 places the PWM in the double update mode. The default operating mode is single update mode.

In single update mode, a single PWMSYNC pulse is produced in each PWM period. The rising edge of this signal marks the start of a new PWM cycle and is used to latch new values from the PWM configuration registers (PWMDAT0 and PWMDAT1) and the PWM duty cycle registers (PWMCH0, PWMCH1, and PWMCH2) into the 3-phase timing unit. In addition, the PWMEN register is latched into the output control unit on the rising edge of the PWMSYNC pulse. In effect, this means that the characteristics and resulting duty cycles of the PWM signals can be updated only once per PWM period at the start of each cycle. The result is symmetrical PWM patterns about the midpoint of the switching period.

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

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

PWM Duty Cycles (PWMCH0, PWMCH1, and PWMCH2 MMRs)

The duty cycles of the six PWM output signals on Pin $PWM0_H$ to Pin $PWM2_L$ are controlled by the three 16-bit read/write duty cycle registers, PWMCH0, PWMCH1, and PWMCH2. The duty cycle registers are programmed in integer counts of the fundamental time unit, t_{CORE}. They define the desired on time of the high-side PWM signal produced by the 3-phase timing unit over half the PWM period. The switching signals produced by the 3-phase timing unit are also adjusted to incorporate the programmed dead time value in the PWMDAT1 register. The 3-phase timing unit produces active high signals so that a high level corresponds to a command to turn on the associated power device.

Figure 69 shows a typical pair of PWM outputs (in this case, 0H and 0L) from the timing unit in single update mode. All illustrated time values indicate the integer value in the associated register and can be converted to time by simply multiplying by the fundamental time increment, t_{CORE}. Note that the switching patterns are perfectly symmetrical about the midpoint of the switching period in this mode because the same values of PWMCH0, PWMDAT0, and PWMDAT1 are used to define the signals in both half cycles of the period.

Figure 69 also demonstrates how the programmed duty cycles are adjusted to incorporate the desired dead time into the resulting pair of PWM signals. The dead time is incorporated by moving the switching instants of both PWM signals (0H and 0L) away from the instant set by the PWMCH0 register.

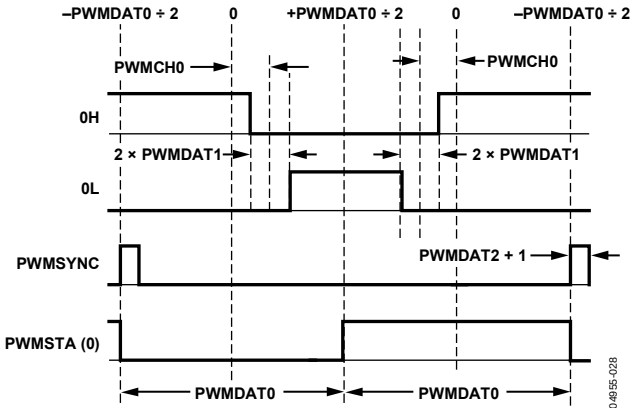


Figure 69. Typical PWM Outputs of the 3-Phase Timing Unit (Single Update Mode)

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

Baud Rate Generation

There are two ways of generating the UART baud rate, normal 450 UART baud rate generation and the fractional divider.

Normal 450 UART Baud Rate Generation

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

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

Table 93 gives some common baud rate values.

Table 93. Baud Rate Using the Normal Baud Rate Generator

Baud Rate	CD	DL	Actual Baud Rate	% Error
9600	0	0x88	9600	0
19,200	0	0x44	19,200	0
115,200	0	0x0B	118,691	3
9600	3	0x11	9600	0
19,200	3	0x08	20,400	6.25
115,200	3	0x01	163,200	41.67

Fractional Divider

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

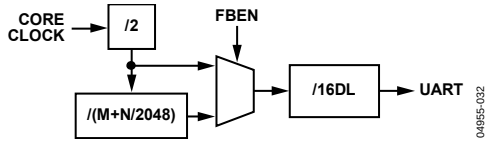


Figure 75. Baud Rate Generation Options

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

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

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

For example, generation of 19,200 baud with CD bits = 3 (Table 93 gives DL = 0x08) is

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

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

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

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

where:
Baud Rate = 19,200 bps

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

UART Register Definitions

The UART interface consists of 12 registers: COMTX, COMRX, COMDIV0, COMIEN0, COMDIV1, COMIID0, COMCON0, COMCON1, COMSTA0, COMSTA1, COMSCR, and COMDIV2.

Table 94. COMTX Register

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

COMTX is an 8-bit transmit register.

Table 95. COMRX Register

Name	Address	Default Value	Access
COMRX	0xFFFF0700	0x00	R

COMRX is an 8-bit receive register.

Table 96. COMDIV0 Register

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

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

Table 97. COMIEN0 Register

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

COMIEN0 is the interrupt enable register.

Table 98. COMIEN0 MMR Bit Descriptions

Bit	Name	Description
7:4	N/A	Reserved.
3	EDSSI	Modem status interrupt enable bit. Set by user to enable generation of an interrupt if any of COMSTA1[3:1] is set. Cleared by user.
2	ELSI	Rx status interrupt enable bit. Set by user to enable generation of an interrupt if any of COMSTA0[4:1] is 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 99. COMDIV1 Register

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

COMDIV1 is a divisor latch (high byte) register.

Table 100. COMIID0 Register

Name	Address	Default Value	Access
COMIID0	0xFFFF0708	0x01	R

COMIID0 is the interrupt identification register.

Table 101. COMIID0 MMR Bit Descriptions

Bit 2:1 Status Bits	Bit 0 NINT	Priority	Definition	Clearing Operation
00	1	N/A	No interrupt	N/A
11	0	1 (Highest)	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 (Lowest)	Modem status interrupt	Read COMSTA1

Table 102. COMCON0 Register

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

COMCON0 is the line control register.

Table 103. COMCON0 MMR Bit Descriptions

Bit	Name	Description
7	DLAB	Divisor latch access. Set by user to enable access to the COMDIV0 and COMDIV1 registers. Cleared by user to disable access to COMDIV0 and COMDIV1 and enable access to COMRX and COMTX.
6	BRK	Set break. Set by user to force SOUT to 0. Cleared to operate in normal mode.
5	SP	Stick parity. Set by user to force parity to defined values: 1 if EPS = 1 and PEN = 1, 0 if EPS = 0 and PEN = 1.
4	EPS	Even parity select bit. Set for even parity. Cleared for odd parity.
3	PEN	Parity enable bit. Set by user to transmit and check the parity bit. Cleared by user for no parity transmission or checking.
2	STOP	Stop bit. Set by user to transmit 1.5 stop bits if the word length is five bits or 2 stop bits if the word length is six bits, seven bits, or eight 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 = five bits, 01 = six bits, 10 = seven bits, 11 = eight bits.

Table 104. COMCON1 Register

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

COMCON1 is the modem control register.

Table 105. COMCON1 MMR Bit Descriptions

Bit	Name	Description
7:5		Reserved.
4	LOOPBACK	Loopback. Set by user to enable loopback mode. In loopback mode, SOUT (see Table 78) is forced high. The modem signals are also directly connected to the status inputs (RTS to CTS and DTR to DSR). Cleared by user to be in normal mode.
3	PEN	Parity enable bit. Set by user to transmit and check the parity bit. Cleared by user for no parity transmission or checking.
2	STOP	Stop bit. Set by user to transmit 1.5 stop bits if the word length is five bits, or 2 stop bits if the word length is six bits, seven bits, or eight bits. The receiver checks the first stop bit only, regardless of the number of stop bits selected. Cleared by user to generate 1 stop bit in the transmitted data.
1	RTS	Request to send. Set by user to force the RTS output to 0. Cleared by user to force the RTS output to 1.
0	DTR	Data terminal ready. Set by user to force the DTR output to 0. Cleared by user to force the DTR output to 1.

Table 106. COMSTA0 Register

Name	Address	Default Value	Access
COMSTA0	0xFFFF0714	0x60	R

COMSTA0 is the line status register.

Table 107. COMSTA0 MMR Bit Descriptions

Bit	Name	Description
7		Reserved.
6	TEMT	COMTX and shift register empty status bit. Set automatically if COMTX and shift register are empty. Cleared automatically when writing to COMTX.
5	THRE	COMTX empty. Set automatically if COMTX is empty. Cleared automatically when writing to COMTX.
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 an invalid stop bit occurs. Cleared automatically.
2	PE	Parity error. Set when a parity error occurs. Cleared automatically.
1	OE	Overrun error. Set automatically if data is over-written before being read. Cleared automatically.
0	DR	Data ready. Set automatically when COMRX is full. Cleared by reading COMRX.

Table 129. I2C0SSTA MMR Bit Descriptions

Bit	Value	Description
31:15		Reserved. These bits should be written as 0.
14		Start decode bit. Set by hardware if the device receives a valid start plus 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 and 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 I2C0SSTA register.
9:8		General call ID.
	00	No general call.
	01	General call reset and program address.
	10	General call program address.
	11	General call matching alternative ID.
7		General call interrupt. Set if the slave device receives a general call of any type. Cleared by setting Bit 8 of the I2CxCFG register. If it is a general call reset, all registers are at their default values. If it is a hardware general call, the Rx FIFO holds the second byte of the general call. This is similar to the I2C0ALT register (unless it is a general call to reprogram the device address). For more details, see the I²C bus specification, Version 2.1, January 2000.
6		Slave busy. Set automatically if the slave is busy. Cleared automatically.
5		No ACK. Set if master asking for data and no data is available. Cleared automatically by reading the I2C0SSTA register.
4		Slave receive FIFO overflow. Set automatically if the slave receive FIFO is overflowing. Cleared automatically by reading the I2C0SSTA register.
3		Slave receive IRQ. Set after receiving data. Cleared automatically by reading the I2C0SRX register or flushing the FIFO.
2		Slave transmit IRQ. Set at the end of a transmission. Cleared automatically by writing to the I2C0STX register.
1		Slave transmit FIFO underflow. Set automatically if the slave transmit FIFO is underflowing. Cleared automatically by writing to the I2C0SSTA register.
0		Slave transmit FIFO not full. Set automatically if the slave transmit FIFO is not full. Cleared automatically by writing twice to the I2C0STX register.

Table 130. I2CxSRX Registers

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

I2CxSRX are receive registers for the slave channel.

Table 131. I2CxSTX Registers

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

I2CxSTX are transmit registers for the slave channel.

Table 132. I2CxMRX Registers

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

I2CxMRX are receive registers for the master channel.

Table 133. I2CxMTX Registers

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

I2CxMTX are transmit registers for the master channel.

Table 134. I2CxCNT Registers

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

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

Table 135. I2CxADR Registers

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

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

Table 136. I2CxBYTE Registers

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

I2CxBYTE are broadcast byte registers. Data written to these registers does not go through the TxFIFO. This data is transmitted at the start of a transfer sequence before the address. After the byte is transmitted and acknowledged, the I²C expects another byte written in I2CxBYTE or an address written to the address register.

Table 137. I2CxALT Registers

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

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

Table 138. I2CxCFG Registers

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

I2CxCFG are configuration registers.

Table 139. I2C0CFG MMR Bit Descriptions

Bit	Description
31:5	Reserved. These bits should be written by the user as 0.
14	Enable stop interrupt. Set by the user to generate an interrupt upon receiving a stop condition and after receiving a valid start condition and matching address. Cleared by the user to disable the generation of an interrupt upon receiving a stop condition.
13	Reserved.
12	Reserved.
11	Enable stretch SCL (holds SCL low). Set by the user to stretch the SCL line. Cleared by the user to disable stretching of the SCL line.
10	Reserved.
9	Slave Tx FIFO request interrupt enable. Set by the user to disable the slave Tx FIFO request interrupt. Cleared by the user to generate an interrupt request just after the negative edge of the clock for the R/W bit. This allows the user to input data into the slave Tx FIFO if it is empty. At 400 ksps and the core clock running at 41.78 MHz, the user has 45 clock cycles to take appropriate action, taking interrupt latency into account.
8	General call status bit clear. Set by the user to clear the general call status bits. Cleared automatically by hardware after the general call status bits are 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	Loopback 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 backoff disable bit. Set by user in multimaster mode. If losing arbitration, the master immediately tries to retransmit. Cleared by user to enable start backoff. After losing arbitration, the master waits before trying to retransmit.
4	Hardware general call enable. When this bit and Bit 3 are set and have received a general call (Address 0x00) and a data byte, the device checks the contents of I2C0ALT against the receive register. If the contents match, the device has received a hardware general call. This is used if a device needs urgent attention from a master device without knowing which master it needs to turn to. This is a "to whom it may concern" call. The ADuC7019/20/21/22/24/25/26/27/28/29 watch for these addresses. The device that requires attention embeds its own address into the message. All masters listen, and the one that can handle the device contacts its slave and acts appropriately. The LSB of the I2C0ALT register should always be written to 1, as indicated in *The I²C-Bus Specification*, January 2000, from NXP.
3	General call enable bit. This bit is set by the user to enable the slave device to acknowledge (ACK) an I²C general call, Address 0x00 (write). The device then recognizes a data bit. If it receives a 0x06 (reset and write programmable part of slave address by hardware) as the data byte, the I²C interface resets as as indicated in *The I²C-Bus Specification,* January 2000, from NXP. This command can be used to reset an entire I²C system. The general call interrupt status bit sets on any general call. The user must take corrective action by setting up the I²C interface after a reset. If it receives a 0x04 (write programmable part of slave address by hardware) as the data byte, the general call interrupt status bit sets on any general call. The user must take corrective action by reprogramming the device address.
2	Reserved.
1	Master enable bit. Set by user to enable the master I²C channel. Cleared by user to disable the master I²C channel.
0	Slave enable bit. Set by user to enable the slave I²C channel. A slave transfer sequence is monitored for the device address in I2C0ID0, I2C0ID1, I2C0ID2, and I2C0ID3. At 400 kSPs, the core clock should run at 41.78 MHz because the interrupt latency could be up to 45 clock cycles alone. After the I²C read bit, the user has 0.5 of an I²C clock cycle to load the Tx FIFO. AT 400 kSPS, this is 1.26 µs, the interrupt latency.

PROCESSOR REFERENCE PERIPHERALS

INTERRUPT SYSTEM

There are 23 interrupt sources on the ADuC7019/20/21/22/24/25/26/27/28/29 that are controlled by the interrupt controller. Most interrupts are generated from the on-chip peripherals, such as ADC and UART. Four additional interrupt sources are generated from external interrupt request pins, IRQ0, IRQ1, IRQ2, and IRQ3. The ARM7TDMI CPU core only recognizes interrupts as one of two types: a normal interrupt request IRQ or a fast interrupt request FIQ. All the interrupts can be masked separately.

The control and configuration of the interrupt system are managed through nine interrupt-related registers, four dedicated to IRQ, and four dedicated to FIQ. An additional MMR is used to select the programmed interrupt source. The bits in each IRQ and FIQ register (except for Bit 23) represent the same interrupt source as described in Table 160.

Table 160. IRQ/FIQ MMRs Bit Description

Bit	Description
0	All interrupts OR'ed (FIQ only)
1	SWI
2	Timer0
3	Timer1
4	Wake-up timer (Timer2)
5	Watchdog timer (Timer3)
6	Flash control
7	ADC channel
8	PLL lock
9	I2C0 slave
10	I2C0 master
11	I2C1 master
12	SPI slave
13	SPI master
14	UART
15	External IRQ0
16	Comparator
17	PSM
18	External IRQ1
19	PLA IRQ0
20	PLA IRQ1
21	External IRQ2
22	External IRQ3
23	PWM trip (IRQ only)/PWM sync (FIQ only)

IRQ

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

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

Table 161. IRQSTA Register

Name	Address	Default Value	Access
IRQSTA	0xFFFF0000	0x00000000	R

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

Table 162. IRQSIG Register

Name	Address	Default Value	Access
IRQSIG	0xFFFF0004	0x00XXX000[1]	R

[1] X indicates an undefined value.

IRQSIG reflects the status of the different IRQ sources. If a peripheral generates an IRQ signal, the corresponding bit in the IRQSIG is set; otherwise, it is cleared. The IRQSIG bits 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.

Table 163. IRQEN Register

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

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

Note that to clear an already enabled interrupt source, the user must set the appropriate bit in the IRQCLR register. Clearing an interrupt's IRQEN bit does not disable the interrupt.

Table 164. IRQCLR Register

Name	Address	Default Value	Access
IRQCLR	0xFFFF000C	0x00000000	W

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

FIQ

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

Table 165. FIQSTA Register

Name	Address	Default Value	Access
FIQSTA	0xFFFF0100	0x00000000	R

Table 166. FIQSIG Register

Name	Address	Default Value	Access
FIQSIG	0xFFFF0104	0x00XXX000[1]	R

[1] X indicates an undefined value.

Table 167. FIQEN Register

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

Table 168. FIQCLR Register

Name	Address	Default Value	Access
FIQCLR	0xFFFF010C	0x00000000	W

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

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

Note that to clear an already enabled FIQ source, the user must set the appropriate bit in the FIQCLR register. Clearing an interrupt's FIQEN bit does not disable the interrupt.

Programmed Interrupts

Because the programmed interrupts are nonmaskable, they are controlled by another register, SWICFG, which simultaneously writes into the IRQSTA and IRQSIG registers and/or the FIQSTA and FIQSIG registers. The 32-bit SWICFG register is dedicated to software interrupts(see Table 170). This MMR allows the control of a programmed source interrupt.

Table 169. SWICFG Register

Name	Address	Default Value	Access
SWICFG	0xFFFF0010	0x00000000	W

Table 170. SWICFG MMR Bit Descriptions

Bit	Description
31:3	Reserved.
2	Programmed interrupt (FIQ). Setting/clearing this bit corresponds with setting/clearing Bit 1 of FIQSTA and FIQSIG.
1	Programmed interrupt (IRQ). Setting/clearing this bit corresponds with setting/clearing Bit 1 of IRQSTA and IRQSIG.
0	Reserved.

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

TIMERS

The ADuC7019/20/21/22/24/25/26/27/28/29 have four general-purpose timer/counters.

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

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

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

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

The timer interval is calculated as follows:

If the timer is set to count down then

$$Interval = \frac{(TxLD) \times Prescaler}{Source\,Clock}$$

If the timer is set to count up, then

$$Interval = \frac{(Fs - TxLD) \times Prescaler}{Source\,Clock}$$

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

Timers are started by writing in the control register of the corresponding timer (TxCON).

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

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

Hour:Minute:Second:1/128 Format

To use the timer in hour:minute:second:hundredths format, select the 32,768 kHz clock and prescaler of 256. The hundredths field does not represent milliseconds but 1/128 of a second (256/32,768). The bits representing the hour, minute, and second are not consecutive in the register. This arrangement applies to TxLD and TxVAL when using the hour:minute:second:hundredths format as set in TxCON[5:4]. See Table 171 for additional details.

Table 171. Hour:Minnute:Second:Hundredths Format

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

Timer0 (RTOS Timer)

Timer0 is a general-purpose, 16-bit timer (count down) with a programmable prescaler (see Figure 77). The prescaler source is the core clock frequency (HCLK) and can be scaled by factors of 1, 16, or 256.

Timer0 can be used to start ADC conversions as shown in the block diagram in Figure 77.

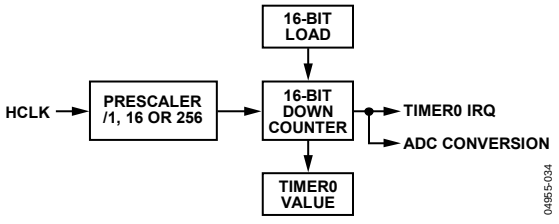


Figure 77. Timer0 Block Diagram

The Timer0 interface consists of four MMRs: T0LD, T0VAL, T0CON, and T0CLRI.

Table 172. T0LD Register

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

T0LD is a 16-bit load register.

Table 173. T0VAL Register

Name	Address	Default Value	Access
T0VAL	0xFFFF0304	0xFFFF	R

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

Table 174. T0CON Register

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

T0CON is the configuration MMR described in Table 175.

Table 175. T0CON MMR Bit Descriptions

Bit	Value	Description
15:8		Reserved.
7		Timer0 enable bit. Set by user to enable Timer0. Cleared by user to disable Timer0 by default.
6		Timer0 mode. Set by user to operate in periodic mode. Cleared by user to operate in free-running mode. Default mode.
5:4		Reserved.
3:2		Prescale.
	00	Core Clock/1. Default value.
	01	Core Clock/16.
	10	Core Clock/256.
	11	Undefined. Equivalent to 00.
1:0		Reserved.

Table 176. T0CLRI Register

Name	Address	Default Value	Access
T0CLRI	0xFFFF030C	0xFF	W

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

Timer1 (General-Purpose Timer)

Timer1 is a general-purpose, 32-bit timer (count down or count up) with a programmable prescaler. The source can be the 32 kHz external crystal, the core clock frequency, or an external GPIO (P1.0 or P0.6). The maximum frequency of the clock input is 44 Mhz). This source can be scaled by a factor of 1, 16, 256, or 32,768.

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

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

Timer1 can be used to start ADC conversions as shown in the block diagram in Figure 78.

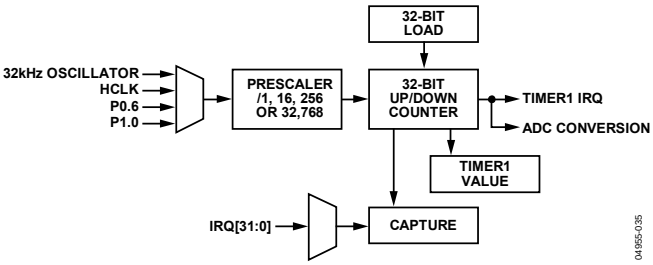


Figure 78. Timer1 Block Diagram

The Timer1 interface consists of five MMRs: T1LD, T1VAL, T1CON, T1CLRI, and T1CAP.

Table 177. T1LD Register

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

T1LD is a 32-bit load register.

Table 178. T1VAL Register

Name	Address	Default Value	Access
T1VAL	0xFFFF0324	0xFFFFFFFF	R

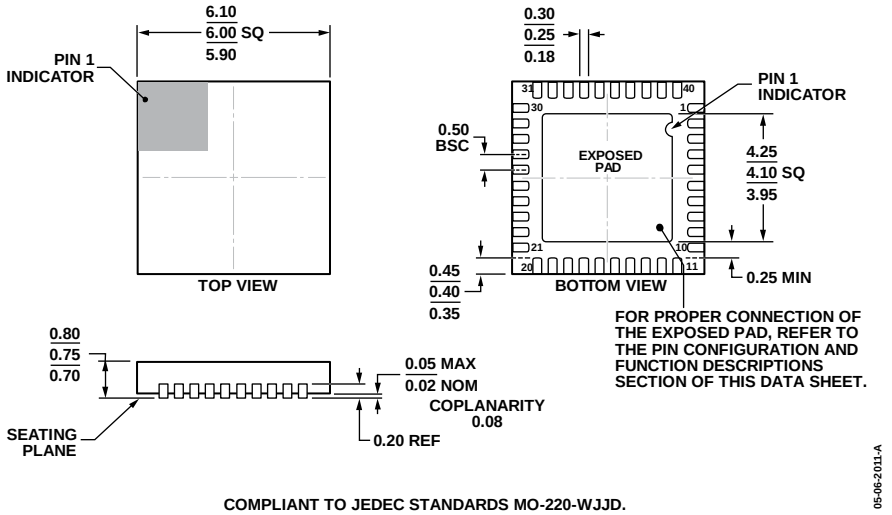
T1VAL is a 32-bit read-only register that represents the current state of the counter.

Table 179. T1CON Register

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

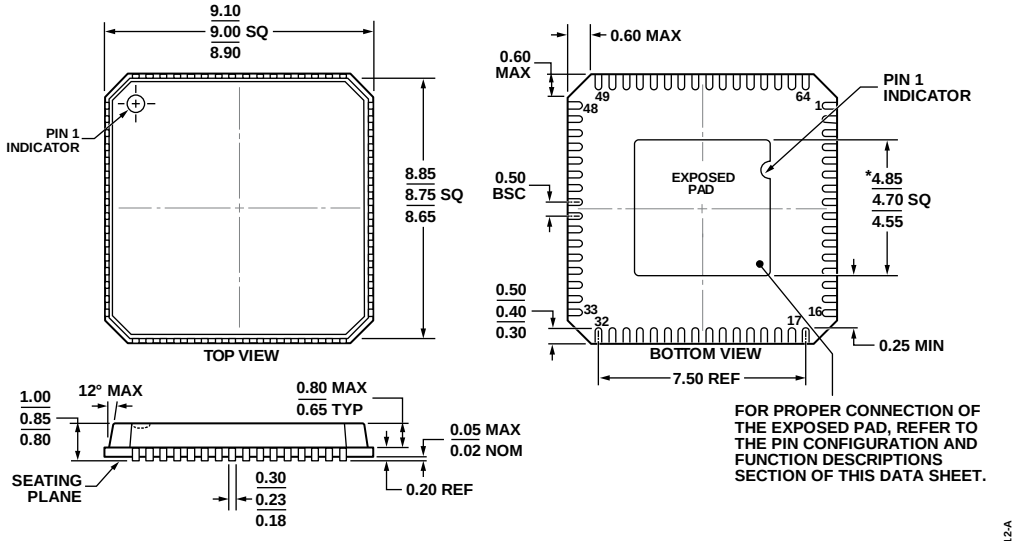
T1CON is the configuration MMR described in Table 180.

OUTLINE DIMENSIONS



COMPLIANT TO JEDEC STANDARDS MO-220-WJJD.

Figure 96. 40-Lead Lead Frame Chip Scale Package [LFCSP_WQ]
6 × 6 mm Body, Very Very Thin Quad
(CP-40-9)
Dimensions shown in millimeters



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

Figure 97. 64-Lead Lead Frame Chip Scale Package [LFCSP_VQ]
9 mm × 9 mm Body, Very Thin Quad
(CP-64-1)
Dimensions shown in millimeters