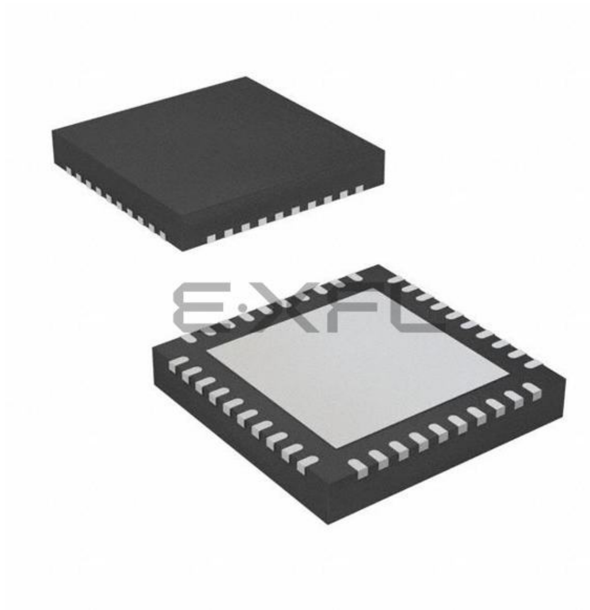


Welcome to E-XFL.COM

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	13
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 8x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	40-VFQFN Exposed Pad, CSP
Supplier Device Package	40-LFCSP-VQ (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7021bcpz62i

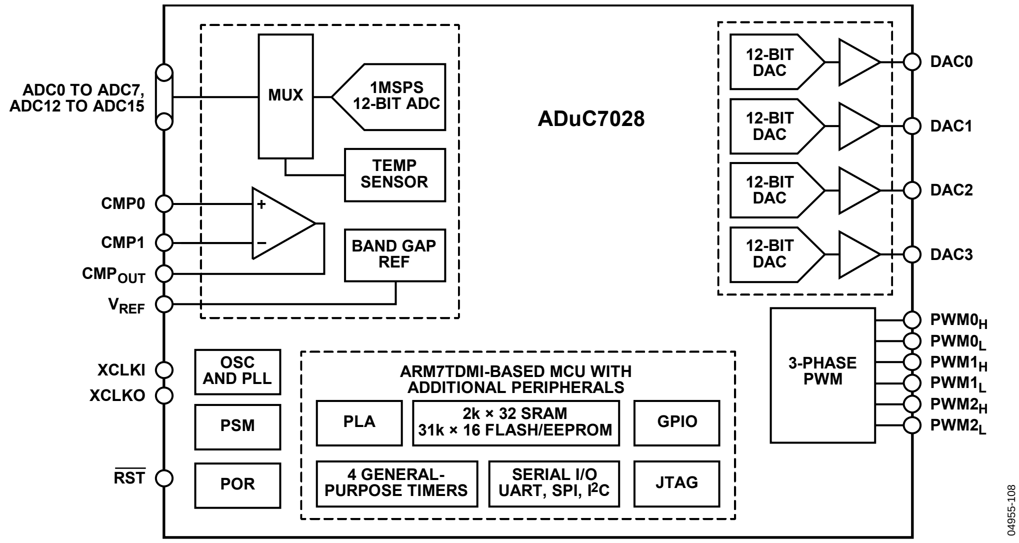


Figure 9.

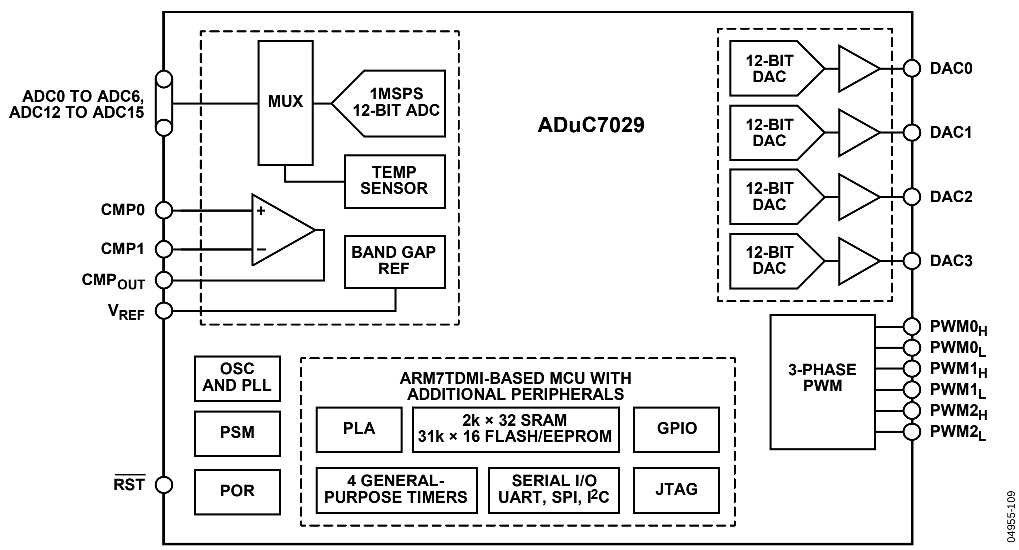


Figure 10.

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
DAC AC CHARACTERISTICS					
Voltage Output Settling Time		10		µs	
Digital-to-Analog Glitch Energy		±20		nV-sec	1 LSB change at major carry (where maximum number of bits simultaneously changes in the DACxDAT register)
COMPARATOR					
Input Offset Voltage		±15		mV	
Input Bias Current		1		µA	
Input Voltage Range	AGND		AV_{DD} − 1.2	V	
Input Capacitance		7		pF	
Hysteresis[4, 6]	2		15	mV	Hysteresis turned on or off via the CMPHYST bit in the CMPCON register
Response Time		3		µs	100 mV overdrive and configured with CMPRES = 11
TEMPERATURE SENSOR					
Voltage Output at 25°C		780		mV	
Voltage TC		−1.3		mV/°C	
Accuracy		±3		°C	
POWER SUPPLY MONITOR (PSM)					
IOV_{DD} Trip Point Selection		2.79		V	Two selectable trip points
		3.07		V	
Power Supply Trip Point Accuracy		±2.5		%	Of the selected nominal trip point voltage
POWER-ON-RESET		2.36		V	
GLITCH IMMUNITY ON RESET PIN[4]		50		µs	
WATCHDOG TIMER (WDT)					
Timeout Period	0		512	sec	
FLASH/EE MEMORY					
Endurance[9]	10,000			Cycles	
Data Retention[10]	20			Years	T_J = 85°C
DIGITAL INPUTS					All digital inputs excluding XCLKI and XCLKO
Logic 1 Input Current		±0.2	±1	µA	V_{IH} = IOV_{DD} or V_{IH} = 5 V
Logic 0 Input Current		−40	−60	µA	V_{IL} = 0 V; except TDI on ADuC7019/20/21/22/24/25/29
		−80	−120	µA	V_{IL} = 0 V; TDI on ADuC7019/20/21/22/24/25/29
Input Capacitance		10		pF	
LOGIC INPUTS[3]					All logic inputs excluding XCLKI
V_{INL}, Input Low Voltage			0.8	V	
V_{INH}, Input High Voltage	2.0			V	
LOGIC OUTPUTS					All digital outputs excluding XCLKO
V_{OH}, Output High Voltage	2.4			V	I_{SOURCE} = 1.6 mA
V_{OL}, Output Low Voltage[11]			0.4	V	I_{SINK} = 1.6 mA
CRYSTAL INPUTS XCLKI and XCLKO					
Logic Inputs, XCLKI Only					
V_{INL}, Input Low Voltage		1.1		V	
V_{INH}, Input High Voltage		1.7		V	
XCLKI Input Capacitance		20		pF	
XCLKO Output Capacitance		20		pF	
INTERNAL OSCILLATOR		32.768		kHz	
			±3	%	
			±2[4]	%	T_A = 0°C to 85°C range

TIMING SPECIFICATIONS

Table 2. External Memory Write Cycle

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

[1] See Table 78.

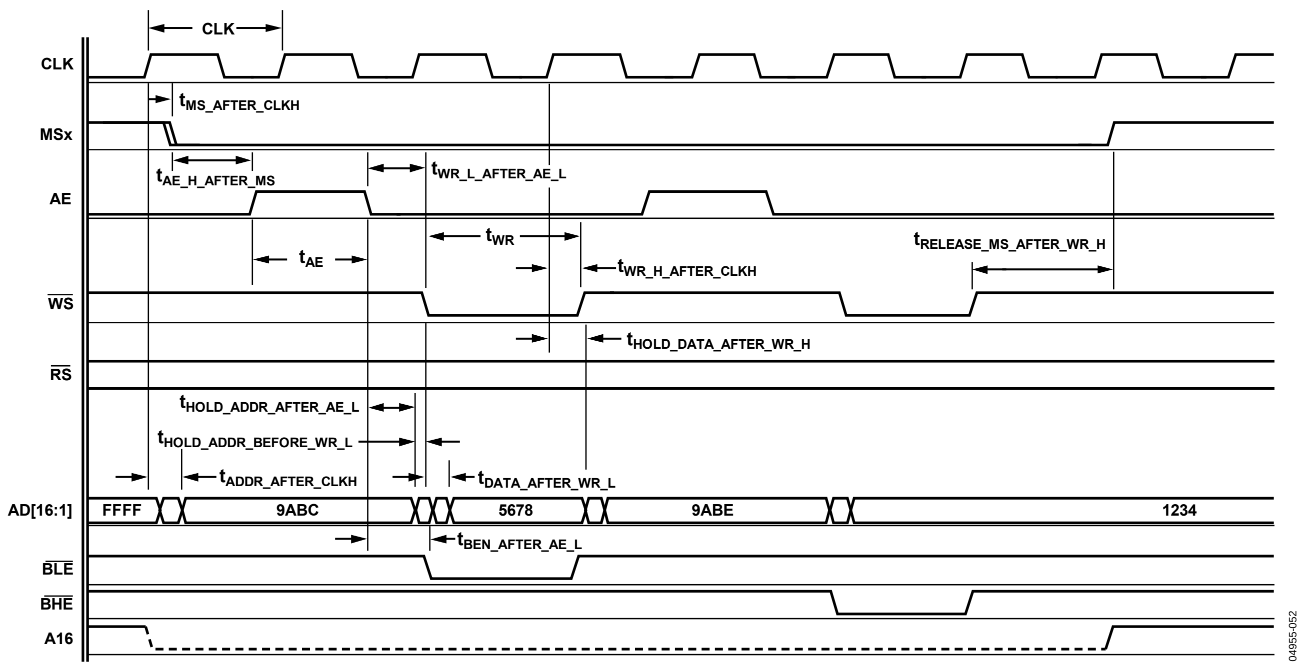


Figure 12. External Memory Write Cycle (See Table 78)

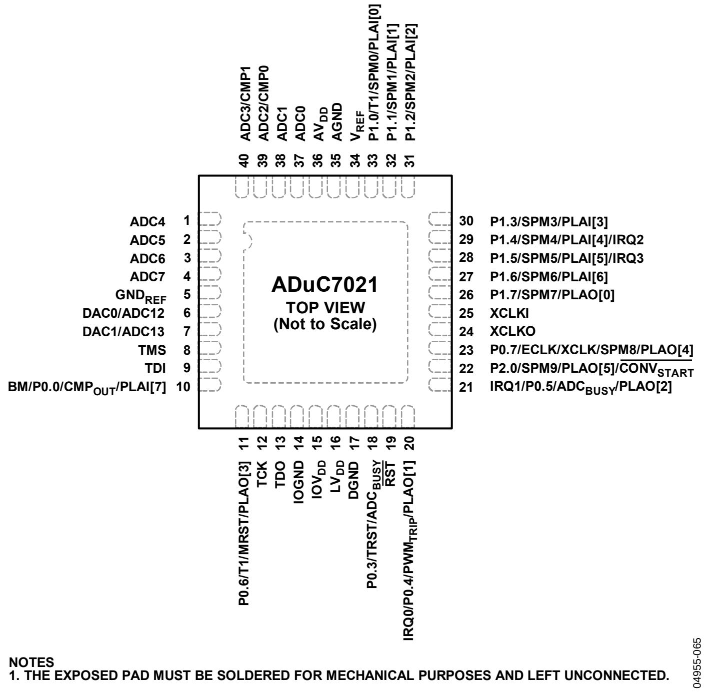


Figure 21. 40-Lead LFCSP_WQ Pin Configuration (ADuC7021)

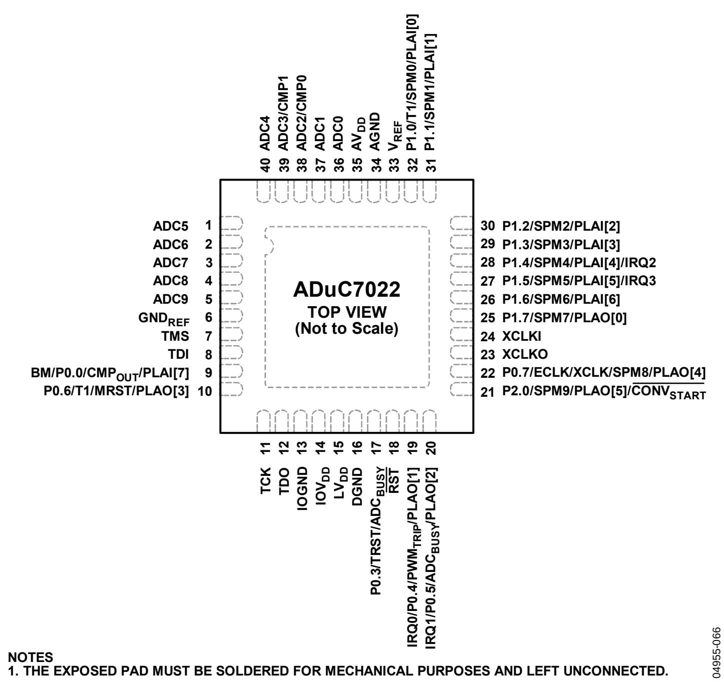


Figure 22. 40-Lead LFCSP_WQ Pin Configuration (ADuC7022)

ADUC7028

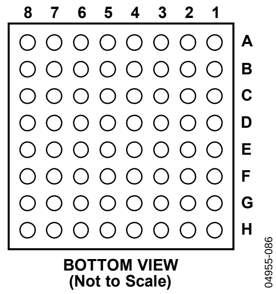


Figure 26. 64-Ball CSP_BGA Pin Configuration (ADuC7028)

Table 14. Pin Function Descriptions (ADuC7028)

Pin No.	Mnemonic	Description
A1	ADC3/CMP1	Single-Ended or Differential Analog Input 3/Comparator Negative Input.
A2	DACV$_{DD}$	3.3 V Power Supply for the DACs. Must be connected to AV$_{DD}$.
A3	AV$_{DD}$	3.3 V Analog Power.
A4	AGND	Analog Ground. Ground reference point for the analog circuitry.
A5	DACGND	Ground for the DAC. Typically connected to AGND.
A6	P4.2/PLAO[10]	General-Purpose Input and Output Port 4.2/Programmable Logic Array Output Element 10.
A7	P1.1/SPM1/PLAI[1]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.1/UART, I2C0/Programmable Logic Array Input Element 1.
A8	P1.2/SPM2/PLAI[2]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.2/UART, I2C1/Programmable Logic Array Input Element 2.
B1	ADC4	Single-Ended or Differential Analog Input 4.
B2	ADC2/CMP0	Single-Ended or Differential Analog Input 2/Comparator Positive Input.
B3	ADC1	Single-Ended or Differential Analog Input 1.
B4	DAC$_{REF}$	External Voltage Reference for the DACs. Range: DACGND to DACV$_{DD}$.
B5	V$_{REF}$	2.5 V Internal Voltage Reference. Must be connected to a 0.47 µF capacitor when using the internal reference.
B6	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.
B7	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.
B8	P1.3/SPM3/PLAI[3]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.3/UART, I2C1/Programmable Logic Array Input Element 3.
C1	ADC6	Single-Ended or Differential Analog Input 6.
C2	ADC5	Single-Ended or Differential Analog Input 5.
C3	ADC0	Single-Ended or Differential Analog Input 0.
C4	P4.5/PLAO[13]	General-Purpose Input and Output Port 4.5/Programmable Logic Array Output Element 13.
C5	P4.3/PLAO[11]	General-Purpose Input and Output Port 4.3/Programmable Logic Array Output Element 11.
C6	P4.0/PLAO[8]	General-Purpose Input and Output Port 4.0/Programmable Logic Array Output Element 8.
C7	P4.1/PLAO[9]	General-Purpose Input and Output Port 4.1/Programmable Logic Array Output Element 9.
C8	IOGND	Ground for GPIO (see Table 78). Typically connected to DGND.
D1	ADCNEG	Bias Point or Negative Analog Input of the ADC in Pseudo Differential Mode. Must be connected to the ground of the signal to convert. This bias point must be between 0 V and 1 V.
D2	GND$_{REF}$	Ground Voltage Reference for the ADC. For optimal performance, the analog power supply should be separated from IOGND and DGND.
D3	ADC7	Single-Ended or Differential Analog Input 7.
D4	P4.4/PLAO[12]	General-Purpose Input and Output Port 4.4/Programmable Logic Array Output Element 12.
D5	P3.6/PWM$_{TRIP}$/PLAI[14]	General-Purpose Input and Output Port 3.6/PWM Safety Cutoff/Programmable Logic Array Input Element 14.
D6	P1.7/SPM7/PLAO[0]	Serial Port Multiplexed. General-Purpose Input and Output Port 1.7/UART, SPI/Programmable Logic Array Output Element 0.

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.

MEMORY ORGANIZATION

The ADuC7019/20/21/22/24/25/26/27/28/29 incorporate two separate blocks of memory: 8 kB of SRAM and 64 kB of on-chip Flash/EE memory. The 62 kB of on-chip Flash/EE memory is available to the user, and the remaining 2 kB are reserved for the factory-configured boot page. These two blocks are mapped as shown in Figure 45.

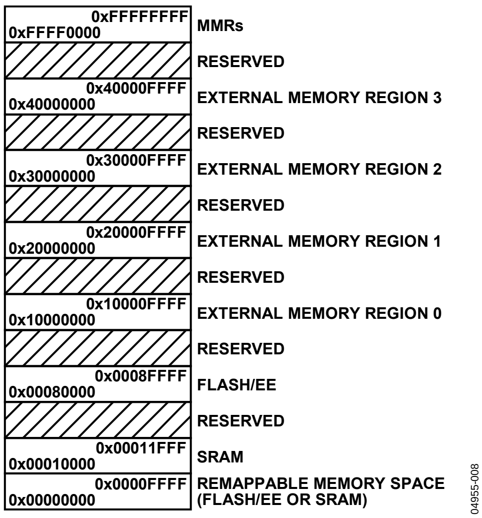


Figure 45. Physical Memory Map

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

MEMORY ACCESS

The ARM7 core sees memory as a linear array of a 2^{32} byte location where the different blocks of memory are mapped as outlined in Figure 45.

The ADuC7019/20/21/22/24/25/26/27/28/29 memory organizations are configured in little endian format, which means that the least significant byte is located in the lowest byte address, and the most significant byte is in the highest byte address.

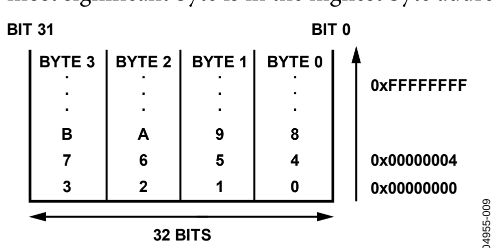


Figure 46. Little Endian Format

FLASH/EE MEMORY

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

Sixty-two kilobytes of Flash/EE memory are available to the user as code and nonvolatile data memory. There is no distinction between data and program because ARM code shares the same space. The real width of the Flash/EE memory is 16 bits, which means that in ARM mode (32-bit instruction), two accesses to the Flash/EE are necessary for each instruction fetch. It is therefore recommended to use thumb mode when executing from Flash/EE memory for optimum access speed. The maximum access speed for the Flash/EE memory is 41.78 MHz in thumb mode and 20.89 MHz in full ARM mode. More details about Flash/EE access time are outlined in the Execution Time from SRAM and Flash/EE section.

SRAM

Eight kilobytes of SRAM are available to the user, organized as 2 k × 32 bits, that is, two words. ARM code can run directly from SRAM at 41.78 MHz, given that the SRAM array is configured as a 32-bit wide memory array. More details about SRAM access time are outlined in the Execution Time from SRAM and Flash/EE section.

MEMORY MAPPED REGISTERS

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

The MMR space provides an interface between the CPU and all on-chip peripherals. All registers, except the core registers, reside in the MMR area. All shaded locations shown in Figure 47 are unoccupied or reserved locations and should not be accessed by user software. Table 16 shows the full MMR memory map.

The access time for reading from or writing to an MMR depends on the advanced microcontroller bus architecture (AMBA) bus used to access the peripheral. The processor has two AMBA buses: the advanced high performance bus (AHB) used for system modules and the advanced peripheral bus (APB) used for lower performance peripheral. Access to the AHB is one cycle, and access to the APB is two cycles. All peripherals on the ADuC7019/20/21/22/24/25/26/27/28/29 are on the APB except the Flash/EE memory, the GPIOs (see Table 78), and the PWM.

TYPICAL OPERATION

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

The top four bits are the sign bits. The 12-bit result is placed from Bit 16 to Bit 27, as shown in Figure 51. Again, it should be noted that, in fully differential mode, the result is represented in twos complement format. In pseudo differential and single-ended modes, the result is represented in straight binary format.

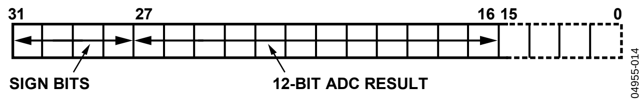


Figure 51. ADC Result Format

The same format is used in DACxDAT, simplifying the software.

Current Consumption

The ADC in standby mode, that is, powered up but not converting, typically consumes 640 µA. The internal reference adds 140 µA. During conversion, the extra current is 0.3 µA multiplied by the sampling frequency (in kilohertz (kHz)). Figure 43 shows the current consumption vs. the sampling frequency of the ADC.

Timing

Figure 52 gives details of the ADC timing. Users control the ADC clock speed and the number of acquisition clocks in the ADCCON MMR. By default, the acquisition time is eight clocks and the clock divider is 2. The number of extra clocks (such as bit trial or write) is set to 19, which gives a sampling rate of 774 kSPS. For conversion on the temperature sensor, the ADC acquisition time is automatically set to 16 clocks, and the ADC clock divider is set to 32. When using multiple channels, including the temperature sensor, the timing settings revert to the user-defined settings after reading the temperature sensor channel.

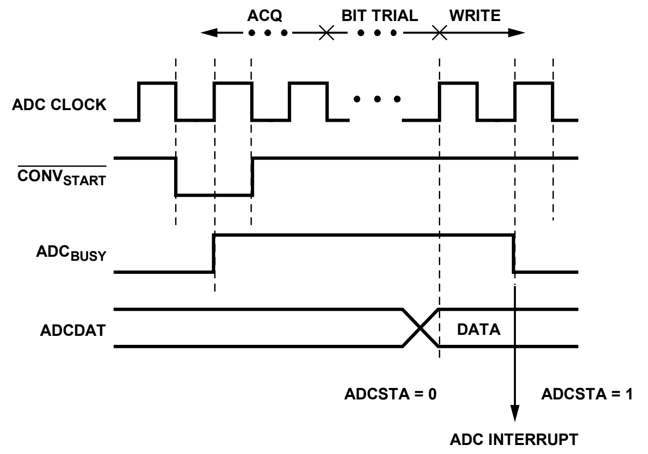


Figure 52. ADC Timing

ADuC7019

The ADuC7019 is identical to the ADuC7020 except for one buffered ADC channel, ADC3, and it has only three DACs. The output buffer of the fourth DAC is internally connected to the ADC3 channel as shown in Figure 53.

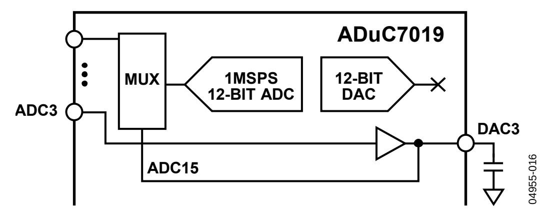


Figure 53. ADC3 Buffered Input

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

MMRS INTERFACE

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

Table 17. ADCCON Register

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

ADCCON is an ADC control register that allows the programmer to enable the ADC peripheral, select the mode of operation of the ADC (in single-ended mode, pseudo differential mode, or fully differential mode), and select the conversion type. This MMR is described in Table 18.

Pseudo Differential Mode

In pseudo differential mode, Channel− is linked to the V_{IN-} pin of the ADuC7019/20/21/22/24/25/26/27/28/29. SW2 switches between A (Channel−) and B (V_{REF}). The V_{IN-} pin must be connected to ground or a low voltage. The input signal on V_{IN+} can then vary from V_{IN-} to $V_{REF} + V_{IN-}$. Note that V_{IN-} must be chosen so that $V_{REF} + V_{IN-}$ does not exceed AV_{DD}.

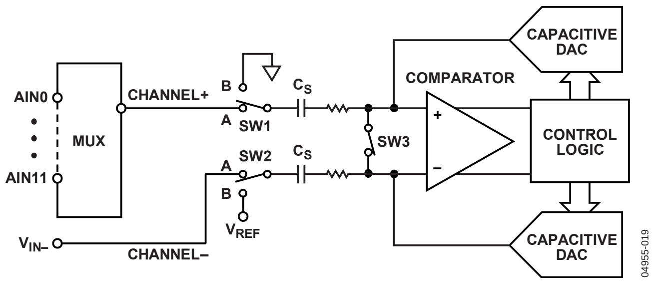


Figure 56. ADC in Pseudo Differential Mode

Single-Ended Mode

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

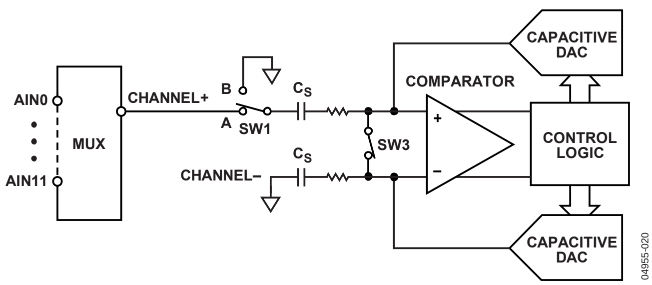


Figure 57. ADC in Single-Ended Mode

Analog Input Structure

Figure 58 shows the equivalent circuit of the analog input structure of the ADC. The four diodes provide ESD protection for the analog inputs. Care must be taken to ensure that the analog input signals never exceed the supply rails by more than 300 mV; exceeding 300 mV causes these diodes to become forward-biased and start conducting into the substrate. These diodes can conduct up to 10 mA without causing irreversible damage to the part.

The C1 capacitors in Figure 58 are typically 4 pF and can be primarily attributed to pin capacitance. The resistors are lumped components made up of the on resistance of the switches. The value of these resistors is typically about 100 Ω. The C2 capacitors are the ADC's sampling capacitors and typically have a capacitance of 16 pF.

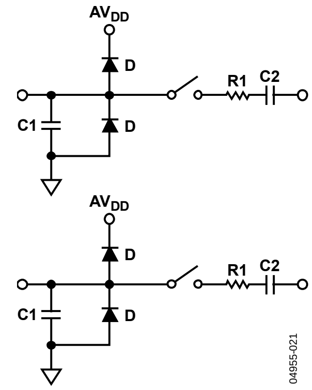


Figure 58. Equivalent Analog Input Circuit Conversion Phase: Switches Open, Track Phase: Switches Closed

For ac applications, removing high frequency components from the analog input signal is recommended by using an RC low-pass filter on the relevant analog input pins. In applications where harmonic distortion and signal-to-noise ratio are critical, the analog input should be driven from a low impedance source. Large source impedances significantly affect the ac performance of the ADC. This can necessitate the use of an input buffer amplifier. The choice of the op amp is a function of the particular application. Figure 59 and Figure 60 give an example of an ADC front end.

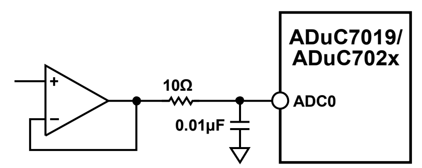


Figure 59. Buffering Single-Ended/Pseudo Differential Input

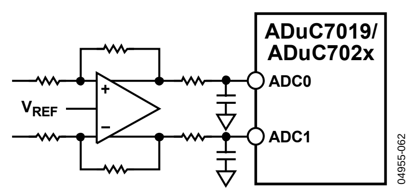


Figure 60. Buffering Differential Inputs

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

DRIVING THE ANALOG INPUTS

Internal or external references can be used for the ADC. In the differential mode of operation, there are restrictions on the common-mode input signal (V_{CM}), which is dependent upon the reference value and supply voltage used to ensure that the signal remains within the supply rails. Table 28 gives some calculated V_{CM} minimum and V_{CM} maximum values.

SECURITY

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

Bit 31 of the FEEPRO/FEEHIDE MMR (see Table 42) protects the 62 kB from being read through JTAG programming mode. The other 31 bits of this register protect writing to the flash memory. Each bit protects four pages, that is, 2 kB. Write protection is activated for all types of access.

Three Levels of Protection

- Protection can be set and removed by writing directly into FEEHIDE MMR. This protection does not remain after reset.
- Protection can be set by writing into the FEEPRO MMR. It takes effect only after a save protection command (0x0C) and a reset. The FEEPRO MMR is protected by a key to avoid direct access. The key is saved once and must be entered again to modify FEEPRO. A mass erase sets the key back to 0xFFFF but also erases all the user code.
- Flash can be permanently protected by using the FEEPRO MMR and a particular value of key: 0xDEADDEAD. Entering the key again to modify the FEEPRO register is not allowed.

Sequence to Write the Key

1. Write the bit in FEEPRO corresponding to the page to be protected.
2. Enable key protection by setting Bit 6 of FEEMOD (Bit 5 must equal 0).
3. Write a 32-bit key in FEEADR and FEEDAT.
4. Run the write key command 0x0C in FEECON; wait for the read to be successful by monitoring FEESTA.
5. Reset the part.

To remove or modify the protection, the same sequence is used with a modified value of FEEPRO. If the key chosen is the value 0xDEAD, 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 Page 4 to Page 7 of the Flash):

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

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

FLASH/EE CONTROL INTERFACE

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

Table 31. FEESTA Register

Name	Address	Default Value	Access
FEESTA	0xFFFFF800	0x20	R

FEESTA is a read-only register that reflects the status of the flash control interface as described in Table 32.

Table 32. FEESTA MMR Bit Designations

Bit	Description
15:6	Reserved.
5	Reserved.
4	Reserved.
3	Flash interrupt status bit. Set automatically when an interrupt occurs, that is, when a command is complete and the Flash/EE interrupt enable bit in the FEEMOD register is set. Cleared when reading the FEESTA register.
2	Flash/EE controller busy. Set automatically when the controller is busy. Cleared automatically when the controller is not busy.
1	Command fail. Set automatically when a command completes unsuccessfully. Cleared automatically when reading the FEESTA register.
0	Command pass. Set by the MicroConverter when a command completes successfully. Cleared automatically when reading the FEESTA register.

Table 33. FEEMOD Register

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

FEEMOD sets the operating mode of the flash control interface. Table 34 shows FEEMOD MMR bit designations.

Table 34. FEEMOD MMR Bit Designations

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

Table 35. FEECON Register

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

FEECON is an 8-bit command register. The commands are described in Table 36.

Table 36. Command Codes in FEECON

Code	Command	Description
0x00[1]	Null	Idle state.
0x01[1]	Single read	Load FEEDAT with the 16-bit data. Indexed by FEEADR.
0x02[1]	Single write	Write FEEDAT at the address pointed to by FEEADR. This operation takes 50 µs.
0x03[1]	Erase/write	Erase the page indexed by FEEADR and write FEEDAT at the location pointed by FEEADR. This operation takes approximately 24 ms.
0x04[1]	Single verify	Compare the contents of the location pointed by FEEADR to the data in FEEDAT. The result of the comparison is returned in FEESTA, Bit 1.
0x05[1]	Single erase	Erase the page indexed by FEEADR.
0x06[1]	Mass erase	Erase 62 kB of user space. The 2 kB of kernel are protected. This operation takes 2.48 sec. To prevent accidental execution, a command sequence is required to execute this instruction. See the Command Sequence for Executing a Mass Erase section.
0x07	Reserved	Reserved.
0x08	Reserved	Reserved.
0x09	Reserved	Reserved.
0x0A	Reserved	Reserved.
0x0B	Signature	Give a signature of the 64 kB of Flash/EE in the 24-bit FEESIGN MMR. This operation takes 32,778 clock cycles.
0x0C	Protect	This command can run only once. The value of FEEPRO is saved and removed only with a mass erase (0x06) of the key.
0x0D	Reserved	Reserved.
0x0E	Reserved	Reserved.
0x0F	Ping	No operation; interrupt generated.

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

Table 37. FEEDAT Register

Name	Address	Default Value	Access
FEEDAT	0xFFFFF80C	0xXXXX[1]	R/W

[1] X = 0, 1, 2, or 3.

FEEDAT is a 16-bit data register.

Table 38. FEEADR Register

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

FEEADR is another 16-bit address register.

Table 39. FEESIGN Register

Name	Address	Default Value	Access
FEESIGN	0xFFFFF818	0xFFFFFF	R

FEESIGN is a 24-bit code signature.

Table 40. FEEPRO Register

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

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

Table 41. FEEHIDE Register

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

FEEHIDE MMR provides immediate protection. It does not require any software key. Note that the protection settings in FEEHIDE are cleared by a reset (see Table 42).

Table 42. FEEPRO and FEEHIDE MMR Bit Designations

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

Command Sequence for Executing a Mass Erase

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

Example source code

```
    t2val_old= T2VAL;
     T2LD = 5;
     TCON = 0x480;

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

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

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

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

External Clock Selection

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

Example source code

```
    t2val_old= T2VAL;
     T2LD = 5;
     TCON = 0x480;

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

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

Power Control System

A choice of operating modes is available on the ADuC7019/20/21/22/24/25/26/27/28/29. Table 57 describes what part is powered on in the different modes and indicates the power-up time.

Table 58 gives some typical values of the total current consumption (analog + digital supply currents) in the different modes, depending on the clock divider bits. The ADC is turned off. Note that these values also include current consumption of the regulator and other parts on the test board where these values are measured.

Table 57. Operating Modes[1]

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

[1] X indicates that the part is powered on.

Table 58. Typical Current Consumption at 25°C in Milliamperes

PC[2:0]	Mode	CD = 0	CD = 1	CD = 2	CD = 3	CD = 4	CD = 5	CD = 6	CD = 7
000	Active	33.1	21.2	13.8	10	8.1	7.2	6.7	6.45
001	Pause	22.7	13.3	8.5	6.1	4.9	4.3	4	3.85
010	Nap	3.8	3.8	3.8	3.8	3.8	3.8	3.8	3.8
011	Sleep	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4
100	Stop	0.4	0.4	0.4	0.4	0.4	0.4	0.4	0.4

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.

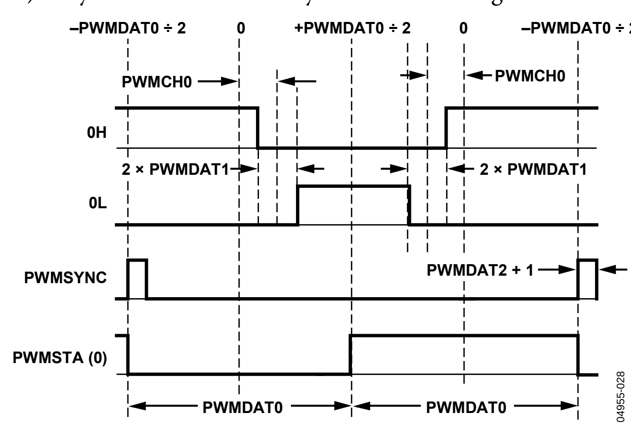


Figure 69. Typical PWM Outputs of the 3-Phase Timing Unit (Single Update Mode)

Output Control Unit

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

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

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

This situation is illustrated in Figure 71, where it can be seen that both the 0H and 1L signals are identical because PWMCH0 = PWMCH1 and the crossover bit for Phase B is set.

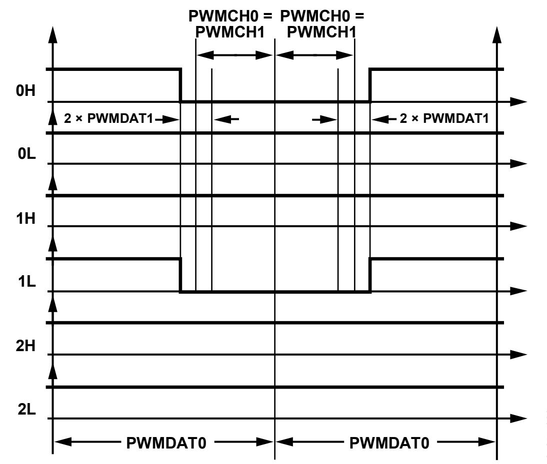


Figure 71. Active Low PWM Signals Suitable for ECM Control, PWMCH0 = PWMCH1, Crossover 1H/1L Pair and Disable 0L, 1H, 2H, and 2L Outputs in Single Update Mode.

In addition, the other four signals (0L, 1H, 2H, and 2L) have been disabled by setting the appropriate enable/disable bits of the PWMEN register. In Figure 71, the appropriate value for the PWMEN register is 0x00A7. In normal ECM operation, each inverter leg is disabled for certain periods of time to change the PWMEN register based on the position of the rotor shaft (motor commutation).

Gate Drive Unit

The gate drive unit of the PWM controller adds features that simplify the design of isolated gate-drive circuits for PWM inverters. If a transformer-coupled, power device, gate-drive amplifier is used, the active PWM signal must be chopped at a high frequency. The 16-bit read/write PWMCFG register programs this high frequency chopping mode. The chopped active PWM signals can be required for the high-side drivers only, the low-side drivers only, or both the high-side and low-side switches. Therefore, independent control of this mode for both high-side and low-side switches is included with two separate control bits in the PWMCFG register.

Typical PWM output signals with high frequency chopping enabled on both high-side and low-side signals are shown in Figure 72. Chopping of the high-side PWM outputs (0H, 1H, and 2H) is enabled by setting Bit 8 of the PWMCFG register. Chopping of the low-side PWM outputs (0L, 1L, and 2L) is enabled by setting Bit 9 of the PWMCFG register. The high chopping frequency is controlled by the 8-bit word (GDCLK) placed in Bit 0 to Bit 7 of the PWMCFG register. The period of this high frequency carrier is

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

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

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

The 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).

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

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.

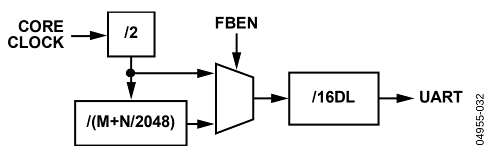


Figure 75. Baud Rate Generation Options

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

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

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

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 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.

Table 140. I2CxDIV Registers

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

I2CxDIV are the clock divider registers.

Table 141. I2CxIDx Registers

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

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

Table 142. I2CxCCNT Registers

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

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

Table 143. I2CxFSTA Registers

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

I2CxFSTA are FIFO status registers.

Table 144. I2C0FSTA MMR Bit Descriptions

Bit	Access Type	Value	Description
15:10			Reserved.
9	R/W		Master transmit FIFO flush. Set by the user to flush the master Tx FIFO. Cleared automatically when the master Tx FIFO is flushed. This bit also flushes the slave receive FIFO.
8	R/W		Slave transmit FIFO flush. Set by the user to flush the slave Tx FIFO. Cleared automatically after the slave Tx FIFO is flushed.
7:6	R		Master Rx FIFO status bits.
		00	FIFO empty.
		01	Byte written to FIFO.
		10	One byte in FIFO.
		11	FIFO full.
5:4	R		Master Tx FIFO status bits.
		00	FIFO empty.
		01	Byte written to FIFO.
		10	One byte in FIFO.
		11	FIFO full.
3:2	R		Slave Rx FIFO status bits.
		00	FIFO empty.
		01	Byte written to FIFO.
		10	One byte in FIFO.
		11	FIFO full.
1:0	R		Slave Tx FIFO status bits.
		00	FIFO empty.
		01	Byte written to FIFO.
		10	One byte in FIFO.
		11	FIFO full.

Table 191. T3CON MMR Bit Descriptions

Bit	Value	Description
15:9		Reserved.
8		Count up. Set by user for Timer3 to count up. Cleared by user for Timer3 to count down by default.
7		Timer3 enable bit. Set by user to enable Timer3. Cleared by user to disable Timer3 by default.
6		Timer3 mode. Set by user to operate in periodic mode. Cleared by user to operate in free-running mode. Default mode.
5		Watchdog mode enable bit. Set by user to enable watchdog mode. Cleared by user to disable watchdog mode by default.
4		Secure clear bit. Set by user to use the secure clear option. Cleared by user to disable the secure clear option by default.
3:2		Prescale.
	00	Source Clock/1 by default.
	01	Source Clock/16.
	10	Source Clock/256.
	11	Undefined. Equivalent to 00.
1		Watchdog IRQ option bit. Set by user to produce an IRQ instead of a reset when the watchdog reaches 0. Cleared by user to disable the IRQ option.
0		Reserved.

Table 192. T3CLRI Register

Name	Address	Default Value	Access
T3CLRI	0xFFFF036C	0x00	W

T3CLRI is an 8-bit register. Writing any value to this register on successive occassions clears the Timer3 interrupt in normal mode or resets a new timeout period in watchdog mode.

Note that the user must perform successive writes to this register to ensure resetting the timeout period.

Secure Clear Bit (Watchdog Mode Only)

The secure clear bit is provided for a higher level of protection. When set, a specific sequential value must be written to T3CLRI to avoid a watchdog reset. The value is a sequence generated by the 8-bit linear feedback shift register (LFSR) polynomial = X8 + X6 + X5 + X + 1, as shown in Figure 81.

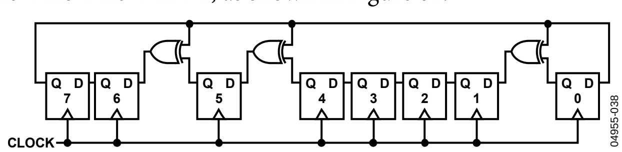


Figure 81. 8-Bit LFSR

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

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

The following is an example of a sequence:

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

EXTERNAL MEMORY INTERFACING

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

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

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

The pins required for interfacing to an external memory are shown in Table 193.

Table 193. External Memory Interfacing Pins

Pin	Function
AD[16:1]	Address/data bus
A16	Extended addressing for 8-bit memory only
MS[3:0]	Memory select
$\overline{\text{WS}}$	Write strobe
$\overline{\text{RS}}$	Read strobe
AE	Address latch enable
$\overline{\text{BHE}}$, $\overline{\text{BLE}}$	Byte write capability

There are four external memory regions available, as described in Table 194. Associated with each region are the MS[3:0] pins. These signals allow access to the particular region of external memory. The size of each memory region can be 128 kB maximum, 64 k × 16 or 128 k × 8. To access 128 k with an 8-bit memory, an extra address line (A16) is provided (see the example in Figure 82). The four regions are configured independently.

Table 194. Memory Regions

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

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

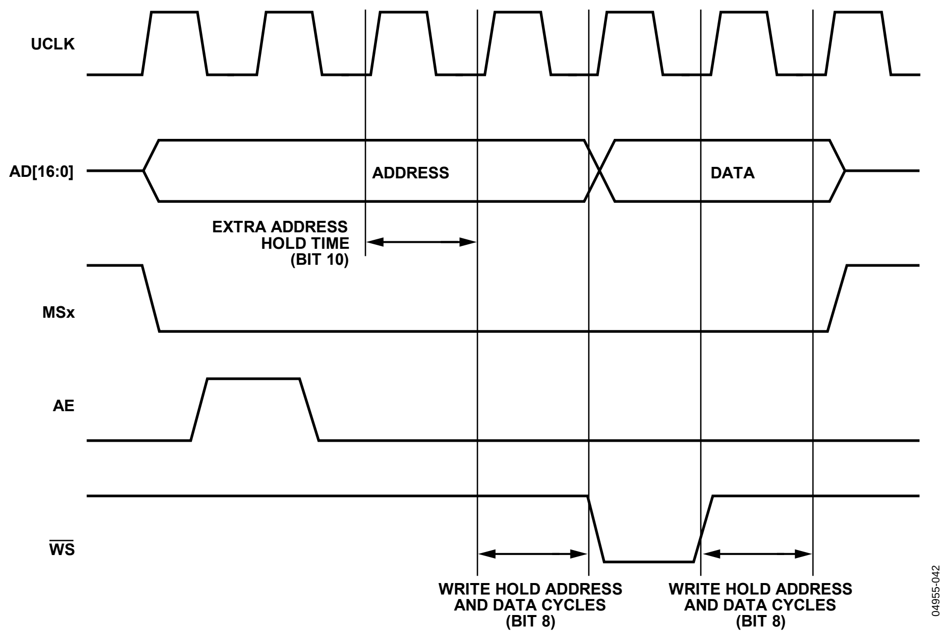


Figure 85. External Memory Write Cycle with Address and Write Hold Cycles

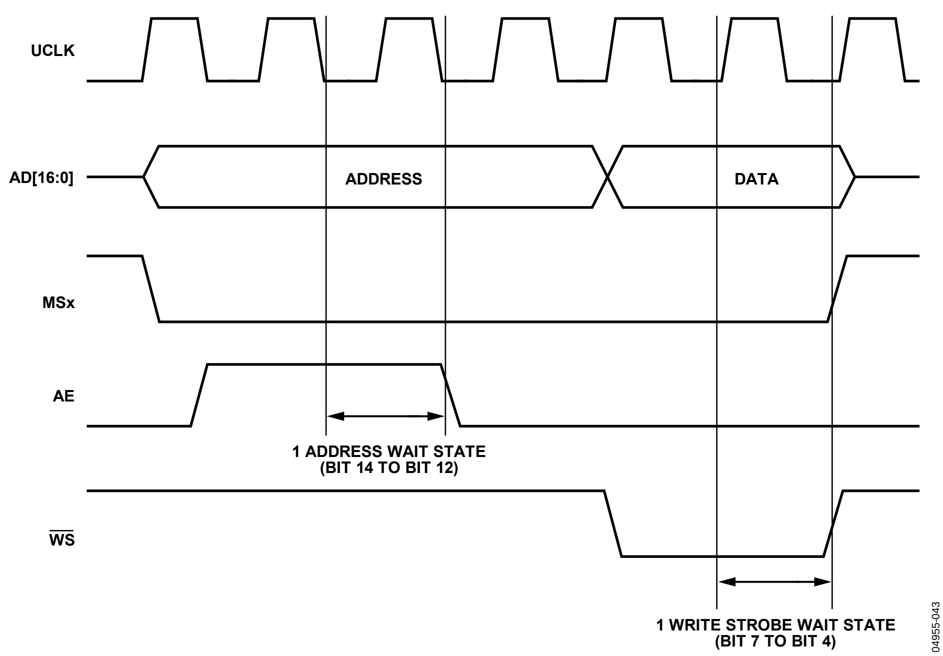
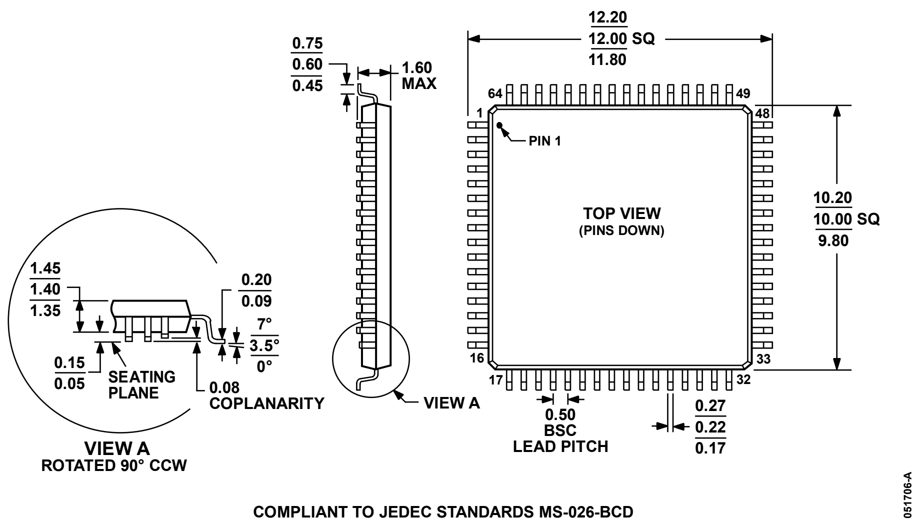
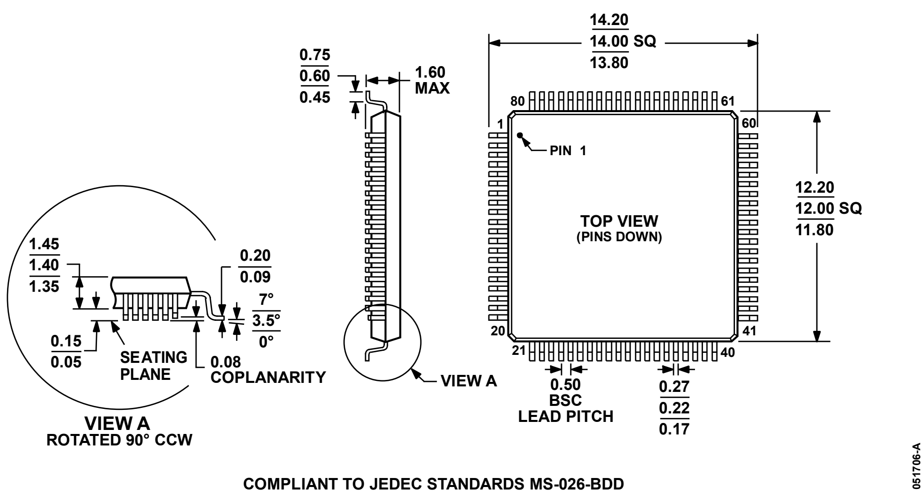


Figure 86. External Memory Write Cycle with Wait States



COMPLIANT TO JEDEC STANDARDS MS-026-BCD

Figure 98. 64-Lead Low Profile Quad Flat Package [LQFP] (ST-64-2) Dimensions shown in millimeters



COMPLIANT TO JEDEC STANDARDS MS-026-BDD

Figure 99. 80-Lead Low Profile Quad Flat Package [LQFP] (ST-80-1) Dimensions shown in millimeters