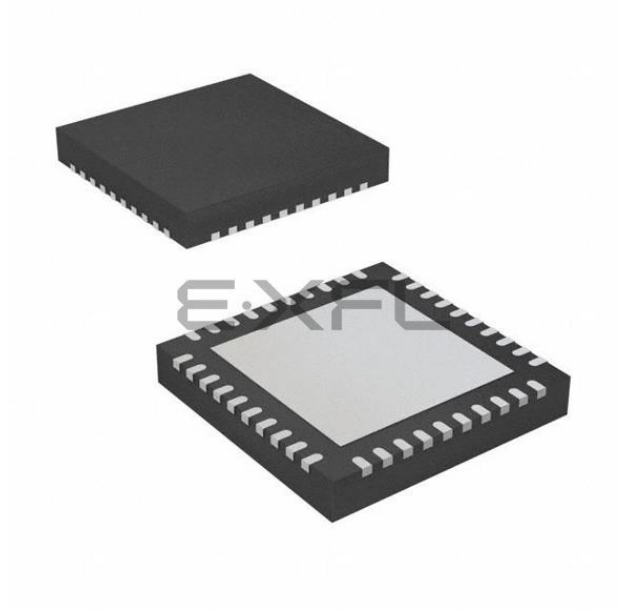


Welcome to E-XFL.COM

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	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 32
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 10x12b
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/aduc7022bcpz32-rl

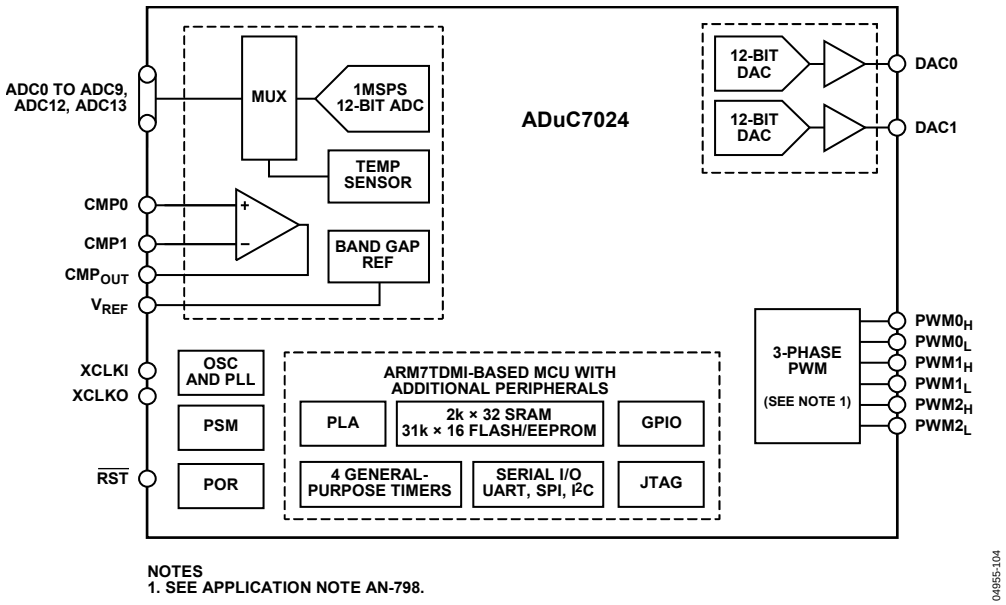


Figure 5.

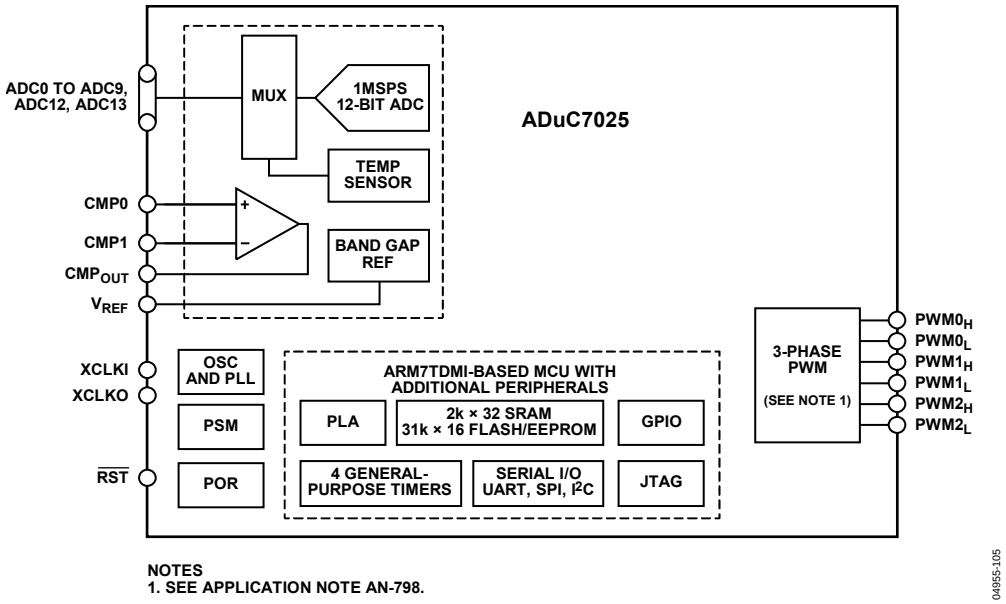


Figure 6.

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

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.

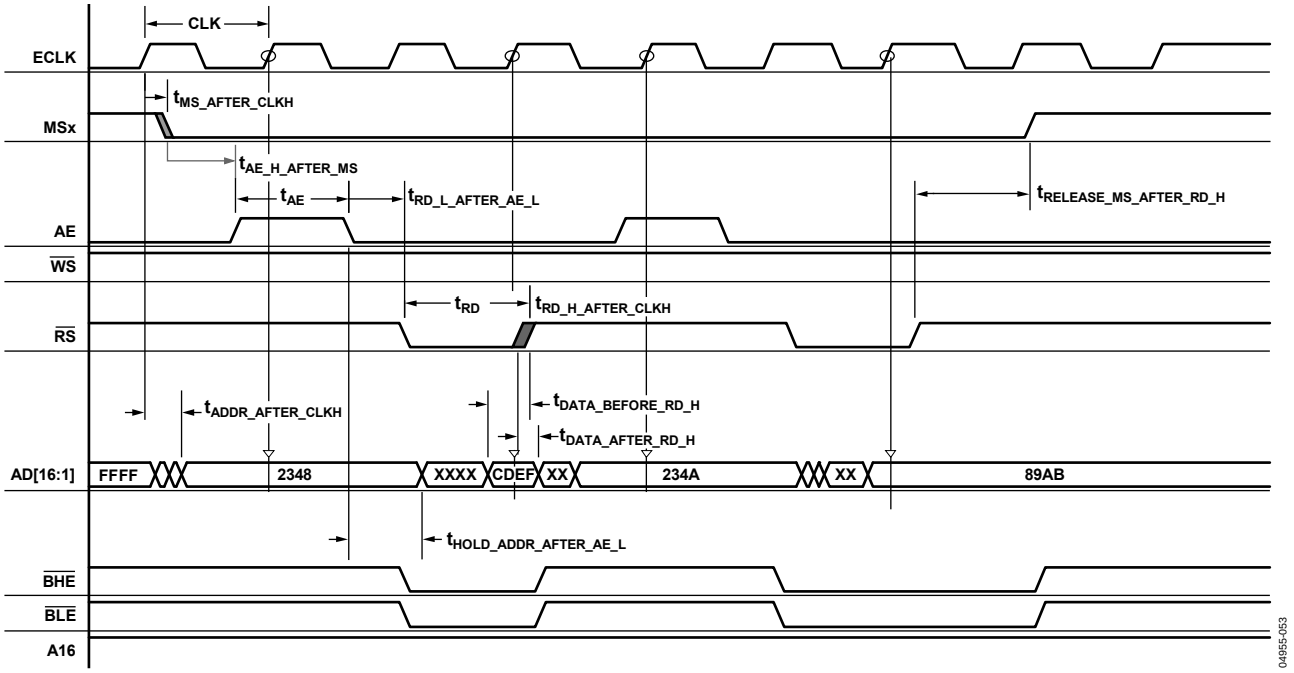


Figure 13. External Memory Read Cycle (See Table 78)

Table 7. SPI Master Mode Timing (Phase Mode = 0)

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

[1] t_{HCLK} depends on the clock divider or CD bits in the POWCONMMR. $t_{HCLK} = t_{UCLK}/2^{CD}$; see Figure 67.
[2] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider; see Figure 67.

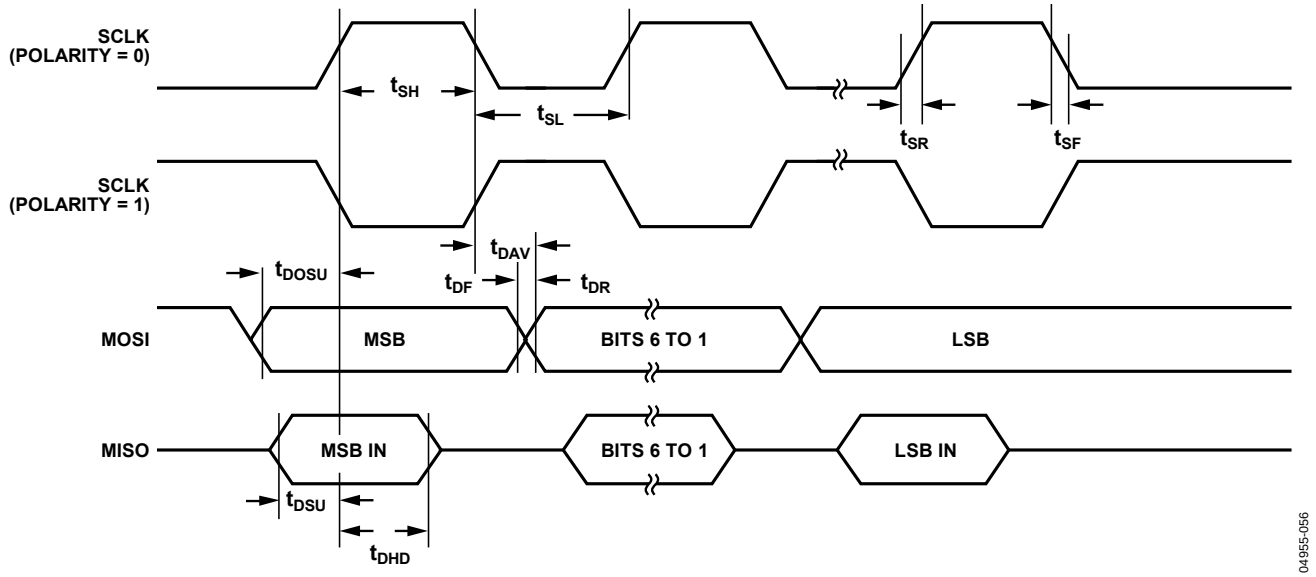


Figure 16. SPI Master Mode Timing (Phase Mode = 0)

Table 8. SPI Slave Mode Timing (Phsae Mode = 1)

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

[1] t_{UCLK} = 23.9 ns. It corresponds to the 41.78 MHz internal clock from the PLL before the clock divider; see Figure 67.
[2] t_{HCLK} depends on the clock divider or CD bits in the POWCONMMR. $t_{HCLK} = t_{UCLK}/2^{CD}$; see Figure 67.

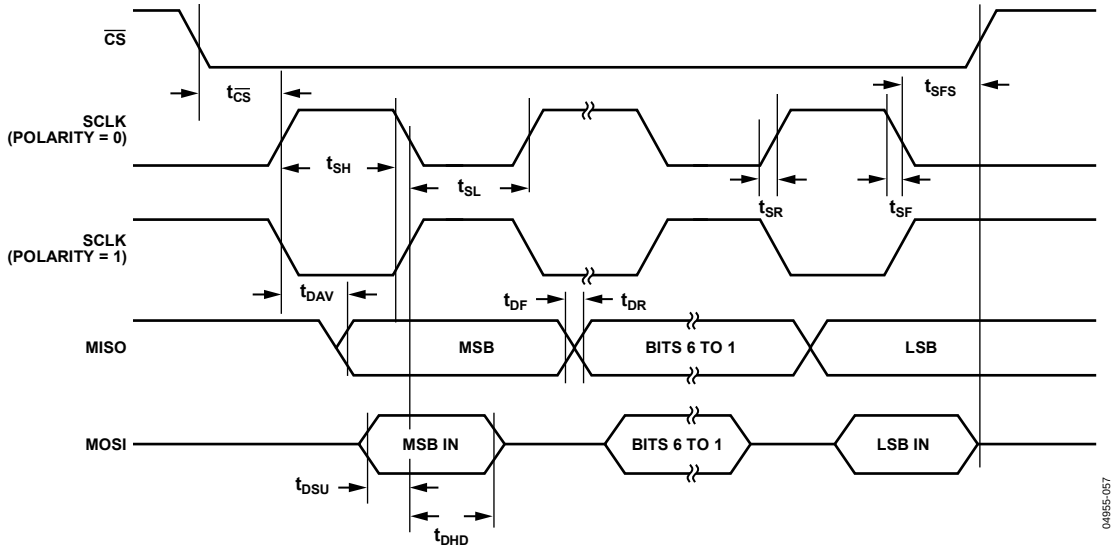


Figure 17. SPI Slave Mode Timing (Phase Mode = 1)

Table 11. Pin Function Descriptions (ADuC7019/ADuC7020/ADuC7021/ADuC7022)

Pin No.				
7019/7020	7021	7022	Mnemonic	Description
38	37	36	ADC0	Single-Ended or Differential Analog Input 0.
39	38	37	ADC1	Single-Ended or Differential Analog Input 1.
40	39	38	ADC2/CMP0	Single-Ended or Differential Analog Input 2/Comparator Positive Input.
1	40	39	ADC3/CMP1	Single-Ended or Differential Analog Input 3 (Buffered Input on ADuC7019)/ Comparator Negative Input.
2	1	40	ADC4	Single-Ended or Differential Analog Input 4.
–	2	1	ADC5	Single-Ended or Differential Analog Input 5.
–	3	2	ADC6	Single-Ended or Differential Analog Input 6.
–	4	3	ADC7	Single-Ended or Differential Analog Input 7.
–	–	4	ADC8	Single-Ended or Differential Analog Input 8.
–	–	5	ADC9	Single-Ended or Differential Analog Input 9.
3	5	6	GND_{REF}	Ground Voltage Reference for the ADC. For optimal performance, the analog power supply should be separated from IOGND and DGND.
4	6	–	DAC0/ADC12	DAC0 Voltage Output/Single-Ended or Differential Analog Input 12.
5	7	–	DAC1/ADC13	DAC1 Voltage Output/Single-Ended or Differential Analog Input 13.
6	–	–	DAC2/ADC14	DAC2 Voltage Output/Single-Ended or Differential Analog Input 14.
7	–	–	DAC3/ADC15	DAC3 Voltage Output on ADuC7020. On the ADuC7019, a 10 nF capacitor must be connected between this pin and AGND/Single-Ended or Differential Analog Input 15 (see Figure 53).
8	8	7	TMS	Test Mode Select, JTAG Test Port Input. Debug and download access. This pin has an internal pull-up resistor to IOV_{DD}. In some cases, an external pull-up resistor (~100K) is also required to ensure that the part does not enter an erroneous state.
9	9	8	TDI	Test Data In, JTAG Test Port Input. Debug and download access.
10	10	9	BM/P0.0/CMP_{OUT}/PLAI[7]	Multifunction I/O Pin. Boot Mode (BM). The ADuC7019/20/21/22 enter serial download mode if BM is low at reset and execute code if BM is pulled high at reset through a 1 kΩ resistor/General-Purpose Input and Output Port 0.0/Voltage Comparator Output/Programmable Logic Array Input Element 7.
11	11	10	P0.6/T1/MRST/PLAO[3]	Multifunction Pin. Driven low after reset. General-Purpose Output Port 0.6/ Timer1 Input/Power-On Reset Output/Programmable Logic Array Output Element 3.
12	12	11	TCK	Test Clock, JTAG Test Port Input. Debug and download access. This pin has an internal pull-up resistor to IOV_{DD}. In some cases an external pull-up resistor (~100K) is also required to ensure that the part does not enter an erroneous state.
13	13	12	TDO	Test Data Out, JTAG Test Port Output. Debug and download access.
14	14	13	IOGND	Ground for GPIO (see Table 78). Typically connected to DGND.
15	15	14	IOV_{DD}	3.3 V Supply for GPIO (see Table 78) and Input of the On-Chip Voltage Regulator.
16	16	15	LV_{DD}	2.6 V Output of the On-Chip Voltage Regulator. This output must be connected to a 0.47 µF capacitor to DGND only.
17	17	16	DGND	Ground for Core Logic.
18	18	17	P0.3/TRST/ADC_{BUSY}	General-Purpose Input and Output Port 0.3/Test Reset, JTAG Test Port Input/ ADC_{BUSY} Signal Output.
19	19	18	$\overline{RST}$	Reset Input, Active Low.
20	20	19	IRQ0/P0.4/PWM_{TRIP}/PLAO[1]	Multifunction I/O Pin. External Interrupt Request 0, Active High/General-Purpose Input and Output Port 0.4/PWM Trip External Input/Programmable Logic Array Output Element 1.
21	21	20	IRQ1/P0.5/ADC_{BUSY}/PLAO[2]	Multifunction I/O Pin. External Interrupt Request 1, Active High/General-Purpose Input and Output Port 0.5/ADC_{BUSY} Signal Output/Programmable Logic Array Output Element 2.

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.

ADuC7026/ADuC7027

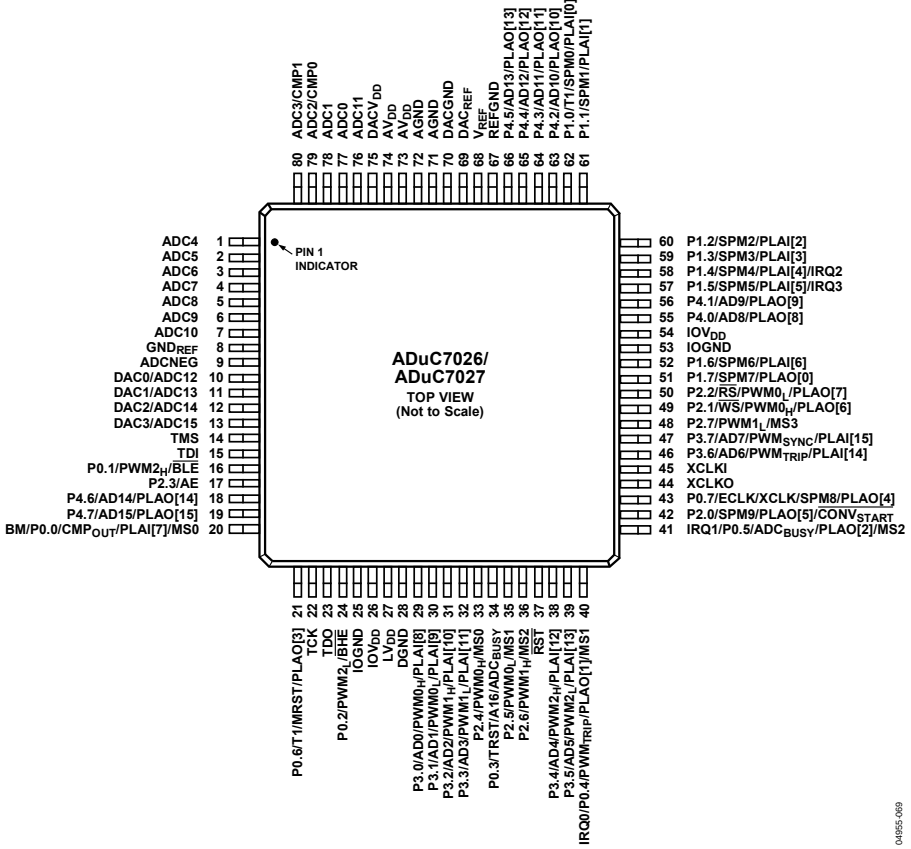


Figure 25. 80-Lead LQFP Pin Configuration (ADuC7026/ADuC7027)

Table 13. Pin Function Descriptions (ADuC7026/ADuC7027)

Pin No.	Mnemonic	Description
1	ADC4	Single-Ended or Differential Analog Input 4.
2	ADC5	Single-Ended or Differential Analog Input 5.
3	ADC6	Single-Ended or Differential Analog Input 6.
4	ADC7	Single-Ended or Differential Analog Input 7.
5	ADC8	Single-Ended or Differential Analog Input 8.
6	ADC9	Single-Ended or Differential Analog Input 9.
7	ADC10	Single-Ended or Differential Analog Input 10.
8	GND_{REF}	Ground Voltage Reference for the ADC. For optimal performance, the analog power supply should be separated from IOGND and DGND.
9	ADCNEG	Bias Point or Negative Analog Input of the ADC in Pseudo Differential Mode. Must be connected to the ground of the signal to convert. This bias point must be between 0 V and 1 V.
10	DAC0/ADC12	DAC0 Voltage Output/Single-Ended or Differential Analog Input 12. DAC outputs are not present on the ADuC7027.
11	DAC1/ADC13	DAC1 Voltage Output/Single-Ended or Differential Analog Input 13. DAC outputs are not present on the ADuC7027.
12	DAC2/ADC14	DAC2 Voltage Output/Single-Ended or Differential Analog Input 14. DAC outputs are not present on the ADuC7027.
13	DAC3/ADC15	DAC3 Voltage Output/Single-Ended or Differential Analog Input 15. DAC outputs are not present on the ADuC7027.
14	TMS	JTAG Test Port Input, Test Mode Select. Debug and download access.
15	TDI	JTAG Test Port Input, Test Data In. Debug and download access.
16	$P0.1/PWM2_H/\overline{BLE}$	General-Purpose Input and Output Port 0.1/PWM Phase 2 High-Side Output/External Memory Byte Low Enable.
17	P2.3/AE	General-Purpose Input and Output Port 2.3/External Memory Access Enable.

Pin No.	Mnemonic	Description
18	P4.6/AD14/PLAO[14]	General-Purpose Input and Output Port 4.6/External Memory Interface/Programmable Logic Array Output Element 14.
19	P4.7/AD15/PLAO[15]	General-Purpose Input and Output Port 4.7/External Memory Interface/Programmable Logic Array Output Element 15.
20	BM/P0.0/CMP_{OUT}/PLAI[7]/MS0	Multifunction I/O Pin. Boot Mode. The ADuC7026/ADuC7027 enter UART download mode if BM is low at reset and execute code if BM is pulled high at reset through a 1 kΩ resistor/General-Purpose Input and Output Port 0.0/Voltage Comparator Output/Programmable Logic Array Input Element 7/External Memory Select 0.
21	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.
22	TCK	JTAG Test Port Input, Test Clock. Debug and download access.
23	TDO	JTAG Test Port Output, Test Data Out. Debug and download access.
24	P0.2/$PWM2_L$/$\overline{BHE}$	General-Purpose Input and Output Port 0.2/PWM Phase 2 Low-Side Output/External Memory Byte High Enable.
25	IOGND	Ground for GPIO (see Table 78). Typically connected to DGND.
26	IOV_{DD}	3.3 V Supply for GPIO (see Table 78) and Input of the On-Chip Voltage Regulator.
27	LV_{DD}	2.6 V Output of the On-Chip Voltage Regulator. This output must be connected to a 0.47 µF capacitor to DGND only.
28	DGND	Ground for Core Logic.
29	P3.0/AD0/$PWM0_H$/PLAI[8]	General-Purpose Input and Output Port 3.0/External Memory Interface/PWM Phase 0 High-Side Output/Programmable Logic Array Input Element 8.
30	P3.1/AD1/$PWM0_L$/PLAI[9]	General-Purpose Input and Output Port 3.1/External Memory Interface/PWM Phase 0 Low-Side Output/Programmable Logic Array Input Element 9.
31	P3.2/AD2/$PWM1_H$/PLAI[10]	General-Purpose Input and Output Port 3.2/External Memory Interface/PWM Phase 1 High-Side Output/Programmable Logic Array Input Element 10.
32	P3.3/AD3/$PWM1_L$/PLAI[11]	General-Purpose Input and Output Port 3.3/External Memory Interface/PWM Phase 1 Low-Side Output/Programmable Logic Array Input Element 11.
33	P2.4/$PWM0_H$/MS0	General-Purpose Input and Output Port 2.4/PWM Phase 0 High-Side Output/External Memory Select 0.
34	P0.3/TRST/A16/ADC_{BUSY}	General-Purpose Input and Output Port 0.3/JTAG Test Port Input, Test Reset/ADC_{BUSY} Signal Output.
35	P2.5/$PWM0_L$/MS1	General-Purpose Input and Output Port 2.5/PWM Phase 0 Low-Side Output/External Memory Select 1.
36	P2.6/$PWM1_H$/MS2	General-Purpose Input and Output Port 2.6/PWM Phase 1 High-Side Output/External Memory Select 2.
37	$\overline{RST}$	Reset Input, Active Low.
38	P3.4/AD4/$PWM2_H$/PLAI[12]	General-Purpose Input and Output Port 3.4/External Memory Interface/PWM Phase 2 High-Side Output/Programmable Logic Array Input 12.
39	P3.5/AD5/$PWM2_L$/PLAI[13]	General-Purpose Input and Output Port 3.5/External Memory Interface/PWM Phase 2 Low-Side Output/Programmable Logic Array Input Element 13.
40	IRQ0/P0.4/PWM_{TRIP}/PLAO[1]/MS1	Multifunction I/O Pin. External Interrupt Request 0, Active High/General-Purpose Input and Output Port 0.4/PWM Trip External Input/Programmable Logic Array Output Element 1/External Memory Select 1.
41	IRQ1/P0.5/ADC_{BUSY}/PLAO[2]/MS2	Multifunction I/O Pin. External Interrupt Request 1, Active High/General-Purpose Input and Output Port 0.5/ADC_{BUSY} Signal Output/Programmable Logic Array Output Element 2/External Memory Select 2.
42	P2.0/SPM9/PLAO[5]/$\overline{CONV_{START}}$	Serial Port Multiplexed. General-Purpose Input and Output Port 2.0/UART/Programmable Logic Array Output Element 5/Start Conversion Input Signal for ADC.
43	P0.7/ECLK/XCLK/SPM8/PLAO[4]	Serial Port Multiplexed. General-Purpose Input and Output Port 0.7/Output for External Clock Signal/Input to the Internal Clock Generator Circuits/UART/Programmable Logic Array Output Element 4.
44	XCLKO	Output from the Crystal Oscillator Inverter.
45	XCLKI	Input to the Crystal Oscillator Inverter and Input to the Internal Clock Generator Circuits.

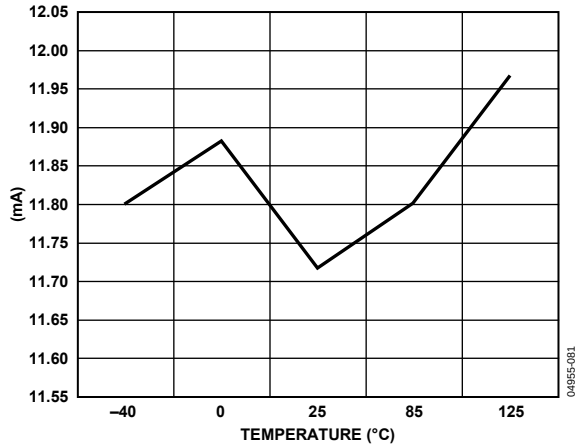


Figure 40. Current Consumption vs. Temperature @ CD = 3

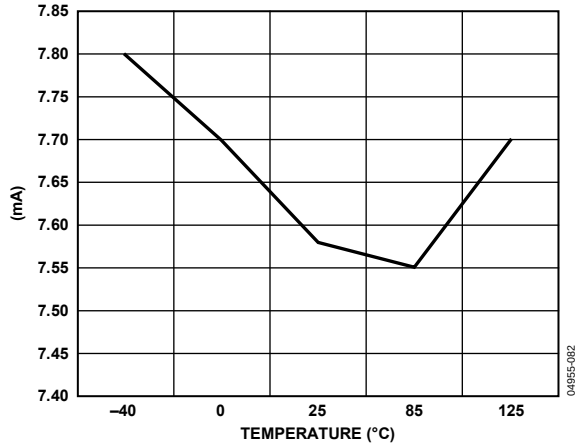


Figure 41. Current Consumption vs. Temperature @ CD = 7

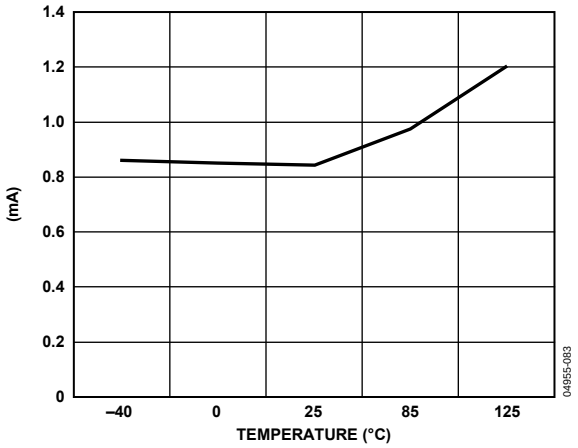


Figure 42. Current Consumption vs. Temperature in Sleep Mode

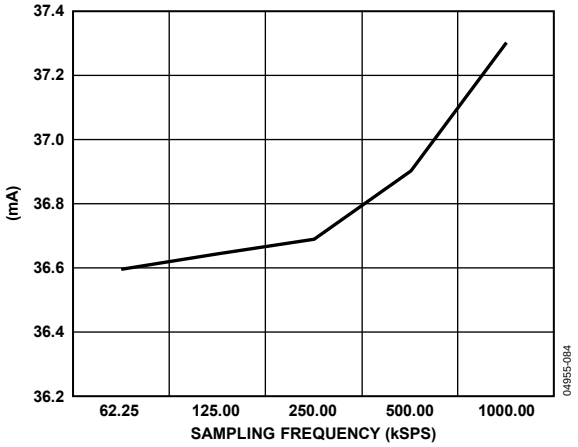


Figure 43. Current Consumption vs. Sampling Frequency

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.

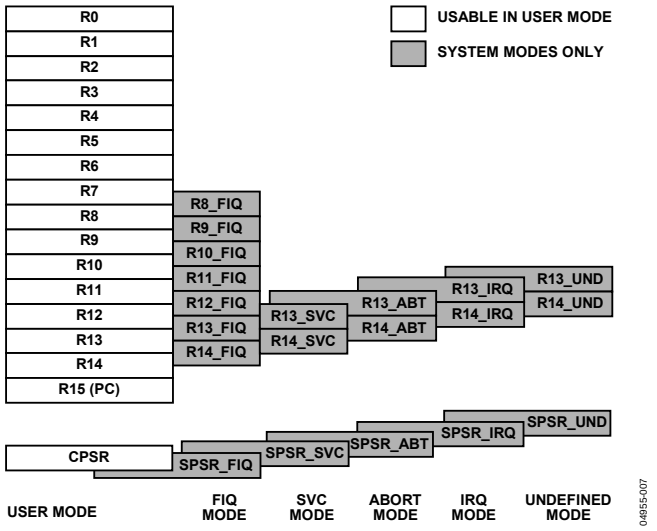


Figure 44. Register Organization

Address	Name	Byte	Access Type	Default Value	Page
Reference Address Base = 0xFFFF0480					
0x048C	REFCON	1	R/W	0x00	50
ADC Address Base = 0xFFFF0500					
0x0500	ADCCON	2	R/W	0x0600	46
0x0504	ADCCP	1	R/W	0x00	47
0x0508	ADCCN	1	R/W	0x01	47
0x050C	ADCSTA	1	R	0x00	48
0x0510	ADCDAT	4	R	0x00000000	48
0x0514	ADCRST	1	R/W	0x00	48
0x0530	ADCGN	2	R/W	0x0200	48
0x0534	ADCOF	2	R/W	0x0200	48
DAC Address Base = 0xFFFF0600					
0x0600	DAC0CON	1	R/W	0x00	56
0x0604	DAC0DAT	4	R/W	0x00000000	56
0x0608	DAC1CON	1	R/W	0x00	56
0x060C	DAC1DAT	4	R/W	0x00000000	56
0x0610	DAC2CON	1	R/W	0x00	56
0x0614	DAC2DAT	4	R/W	0x00000000	56
0x0618	DAC3CON	1	R/W	0x00	56
0x061C	DAC3DAT	4	R/W	0x00000000	56
UART Base Address = 0xFFFF0700					
0x0700	COMTX	1	R/W	0x00	71
	COMRX	1	R	0x00	71
	COMDIV0	1	R/W	0x00	71
0x0704	COMIEN0	1	R/W	0x00	71
	COMDIV1	1	R/W	0x00	72
0x0708	COMIID0	1	R	0x01	72
0x070C	COMCON0	1	R/W	0x00	72
0x0710	COMCON1	1	R/W	0x00	72
0x0714	COMSTA0	1	R	0x60	72
0x0718	COMSTA1	1	R	0x00	73
0x071C	COMSCR	1	R/W	0x00	73
0x0720	COMIEN1	1	R/W	0x04	73
0x0724	COMIID1	1	R	0x01	73
0x0728	COMADR	1	R/W	0xAA	74
0x072C	COMDIV2	2	R/W	0x0000	73

Address	Name	Byte	Access Type	Default Value	Page
I2C0 Base Address = 0xFFFF0800					
0x0800	I2C0MSTA	1	R/W	0x00	76
0x0804	I2C0SSTA	1	R	0x01	76
0x0808	I2C0SRX	1	R	0x00	77
0x080C	I2C0STX	1	W	0x00	77
0x0810	I2C0MRX	1	R	0x00	77
0x0814	I2C0MTX	1	W	0x00	77
0x0818	I2C0CNT	1	R/W	0x00	77
0x081C	I2C0ADR	1	R/W	0x00	77
0x0824	I2C0BYTE	1	R/W	0x00	77
0x0828	I2C0ALT	1	R/W	0x00	78
0x082C	I2C0CFG	1	R/W	0x00	78
0x0830	I2C0DIV	2	R/W	0x1F1F	79
0x0838	I2C0ID0	1	R/W	0x00	79
0x083C	I2C0ID1	1	R/W	0x00	79
0x0840	I2C0ID2	1	R/W	0x00	79
0x0844	I2C0ID3	1	R/W	0x00	79
0x0848	I2C0CCNT	1	R/W	0x01	79
0x084C	I2C0FSTA	2	R/W	0x0000	79
I2C1 Base Address = 0xFFFF0900					
0x0900	I2C1MSTA	1	R/W	0x00	76
0x0904	I2C1SSTA	1	R	0x01	76
0x0908	I2C1SRX	1	R	0x00	77
0x090C	I2C1STX	1	W	0x00	77
0x0910	I2C1MRX	1	R	0x00	77
0x0914	I2C1MTX	1	W	0x00	77
0x0918	I2C1CNT	1	R/W	0x00	77
0x091C	I2C1ADR	1	R/W	0x00	77
0x0924	I2C1BYTE	1	R/W	0x00	77
0x0928	I2C1ALT	1	R/W	0x00	78
0x092C	I2C1CFG	1	R/W	0x00	78
0x0930	I2C1DIV	2	R/W	0x1F1F	79
0x0938	I2C1ID0	1	R/W	0x00	79
0x093C	I2C1ID1	1	R/W	0x00	79
0x0940	I2C1ID2	1	R/W	0x00	79
0x0944	I2C1ID3	1	R/W	0x00	79
0x0948	I2C1CCNT	1	R/W	0x01	79
0x094C	I2C1FSTA	2	R/W	0x0000	79
SPI Base Address = 0xFFFF0A00					
0x0A00	SPISTA	1	R	0x00	75
0x0A04	SPIRX	1	R	0x00	75
0x0A08	SPITX	1	W	0x00	75
0x0A0C	SPIDIV	1	R/W	0x1B	75
0x0A10	SPICON	2	R/W	0x0000	75

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

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

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

On the high side

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

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

and the corresponding duty cycles (*d*)

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

and on the low side

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

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

and the corresponding duty cycles (*d*)

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

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

Figure 70 shows the output signals from the timing unit for operation in double update mode. It illustrates a general case where the switching frequency, dead time, and duty cycle are all changed in the second half of the PWM period. The same value for any or all of these quantities can be used in both halves of the PWM cycle. However, there is no guarantee that symmetrical PWM signals are produced by the timing unit in double update mode. Figure 70 also shows that the dead time insertions into the PWM signals are done in the same way as in single update mode.

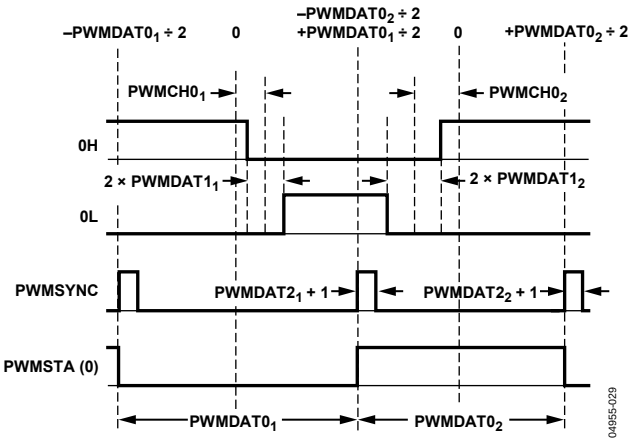


Figure 70. Typical PWM Outputs of the 3-Phase Timing Unit (Double Update Mode)

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

On the high side

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

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

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

The corresponding duty cycles (*d*) are

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

On the low side

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

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

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

The corresponding duty cycles (d) are

$$d_{0L} = t_{0LH}/t_S = (\text{PWMDAT0}_1/2 + PWMDAT0_2/2 + PWMCH0_1 + PWMCH0_2 + PWMDAT1_1 + PWMDAT1_2)/(PWMDAT0_1 + PWMDAT0_2)$$

For the completely general case in double update mode (see Figure 70), the switching period is given by

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

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

PWM signals similar to those illustrated in Figure 69 and Figure 70 can be produced on the 1H, 1L, 2H, and 2L outputs by programming the PWMCH1 and PWMCH2 registers in a manner identical to that described for PWMCH0. The PWM controller does not produce any PWM outputs until all of the PWMDAT0, PWMCH0, PWMCH1, and PWMCH2 registers have been written to at least once. When these registers are written, internal counting of the timers in the 3-phase timing unit is enabled.

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

Output Control Unit

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

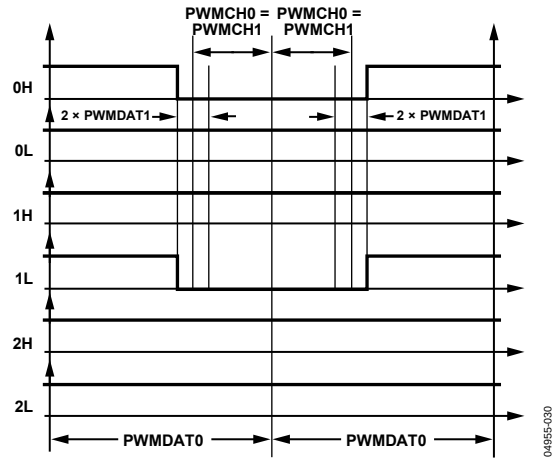


Figure 71. Active Low PWM Signals Suitable for ECM Control, PWMCH0 = PWMCH1, Crossover 1H/1L Pair and Disable 0L, 1H, 2H, and 2L Outputs in Single Update Mode.

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

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

Table 85. GPxDAT Registers

Name	Address	Default Value[1]	Access
GP0DAT	0xFFFFF420	0x000000XX	R/W
GP1DAT	0xFFFFF430	0x000000XX	R/W
GP2DAT	0xFFFFF440	0x000000XX	R/W
GP3DAT	0xFFFFF450	0x000000XX	R/W
GP4DAT	0xFFFFF460	0x000000XX	R/W

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

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

Table 86. GPxDAT MMR Bit Descriptions

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

Table 87. GPxSET Registers

Name	Address	Default Value[1]	Access
GP0SET	0xFFFFF424	0x000000XX	W
GP1SET	0xFFFFF434	0x000000XX	W
GP2SET	0xFFFFF444	0x000000XX	W
GP3SET	0xFFFFF454	0x000000XX	W
GP4SET	0xFFFFF464	0x000000XX	W

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

GPxSET are data set Port x registers.

Table 88. GPxSET MMR Bit Descriptions

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

Table 89. GPxCLR Registers

Name	Address	Default Value[1]	Access
GP0CLR	0xFFFFF428	0x000000XX	W
GP1CLR	0xFFFFF438	0x000000XX	W
GP2CLR	0xFFFFF448	0x000000XX	W
GP3CLR	0xFFFFF458	0x000000XX	W
GP4CLR	0xFFFFF468	0x000000XX	W

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

GPxCLR are data clear Port x registers.

Table 90. GPxCLR MMR Bit Descriptions

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

SERIAL PORT MUX

The serial port mux multiplexes the serial port peripherals (an SPI, UART, and two I^2Cs) and the programmable logic array (PLA) to a set of 10 GPIO pins. Each pin must be configured to one of its specific I/O functions as described in Table 91.

Table 91. SPM Configuration

SPMMUX	GPIO (00)	UART (01)	UART/I^2C/SPI (10)	PLA (11)
SPM0	P1.0	SIN	I2C0SCL	PLAI[0]
SPM1	P1.1	SOUT	I2C0SDA	PLAI[1]
SPM2	P1.2	RTS	I2C1SCL	PLAI[2]
SPM3	P1.3	CTS	I2C1SDA	PLAI[3]
SPM4	P1.4	RI	SCLK	PLAI[4]
SPM5	P1.5	DCD	MISO	PLAI[5]
SPM6	P1.6	DSR	MOSI	PLAI[6]
SPM7	P1.7	DTR	$\overline{CS}$	PLAO[0]
SPM8	P0.7	ECLK/XCLK	SIN	PLAO[4]
SPM9	P2.0	CONV	SOUT	PLAO[5]

Table 91 also details the mode for each of the SPMMUX pins. This configuration must be done via the GP0CON, GP1CON, and GP2CON MMRs. By default, these 10 pins are configured as GPIOs.

UART SERIAL INTERFACE

The UART peripheral is a full-duplex, universal, asynchronous receiver/transmitter. It is fully compatible with the 16,450 serial port standard. The UART performs serial-to-parallel conversions on data characters received from a peripheral device or modem, and parallel-to-serial conversions on data characters received from the CPU. The UART includes a fractional divider for baud rate generation and has a network addressable mode. The UART function is made available on the 10 pins of the ADuC7019/20/21/22/24/25/26/27/28/29 (see Table 92).

Table 92. UART Signal Description

Pin	Signal	Description
SPM0 (Mode 1)	SIN	Serial receive data.
SPM1 (Mode 1)	SOUT	Serial transmit data.
SPM2 (Mode 1)	RTS	Request to send.
SPM3 (Mode 1)	CTS	Clear to send.
SPM4 (Mode 1)	RI	Ring indicator.
SPM5 (Mode 1)	DCD	Data carrier detect.
SPM6 (Mode 1)	DSR	Data set ready.
SPM7 (Mode 1)	DTR	Data terminal ready.
SPM8 (Mode 2)	SIN	Serial receive data.
SPM9 (Mode 2)	SOUT	Serial transmit data.

I²C-COMPATIBLE INTERFACES

The ADuC7019/20/21/22/24/25/26/27/28/29 support two licensed I²C interfaces. The I²C interfaces are both implemented as a hard-ware master and a full slave interface. Because the two I²C inter-faces are identical, this data sheet describes only I2C0 in detail. Note that the two masters and one of the slaves have individual interrupts (see the Interrupt System section).

Note that when configured as an I²C master device, the ADuC7019/20/21/22/24/25/26/27/28/29 cannot generate a repeated start condition.

The two GPIO pins used for data transfer, SDAx and SCLx, are configured in a wired-AND format that allows arbitration in a multimaster system. These pins require external pull-up resistors. Typical pull-up values are 10 kΩ.

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

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

The I²C peripheral can be configured only as a master or slave at any given time. The same I²C channel cannot simultaneously support master and slave modes.

Serial Clock Generation

The I²C master in the system generates the serial clock for a transfer. The master channel can be configured to operate in fast mode (400 kHz) or standard mode (100 kHz).

The bit rate is defined in the I2C0DIV MMR as follows:

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

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

Thus, for 100 kHz operation,

$DIVH = DIVL$ = 0xCF

and for 400 kHz,

$DIVH$ = 0x28, $DIVL$ = 0x3C

The I2CxDIV registers correspond to DIVH:DIVL.

Slave Addresses

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

I²C Registers

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

Table 126. I2CxMSTA Registers

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

I2CxMSTA are status registers for the master channel.

Table 127. I2C0MSTA MMR Bit Descriptions

Bit	Access Type	Description
7	R/W	Master transmit FIFO flush. Set by user to flush the master Tx FIFO. Cleared automatically after the master Tx FIFO is flushed. This bit also flushes the slave receive FIFO.
6	R	Master busy. Set automatically if the master is busy. Cleared automatically.
5	R	Arbitration loss. Set in multimaster mode if another master has the bus. Cleared when the bus becomes available.
4	R	No ACK. Set automatically if there is no acknowledge of the address by the slave device. Cleared automatically by reading the I2C0MSTA register.
3	R	Master receive IRQ. Set after receiving data. Cleared automatically by reading the I2C0MRX register.
2	R	Master transmit IRQ. Set at the end of a transmission. Cleared automatically by writing to the I2C0MTX register.
1	R	Master transmit FIFO underflow. Set automatically if the master transmit FIFO is underflowing. Cleared automatically by writing to the I2C0MTX register.
0	R	Master TX FIFO not full. Set automatically if the slave transmit FIFO is not full. Cleared automatically by writing twice to the I2C0STX register.

Table 128. I2CxSSTA Registers

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

I2CxSSTA are status registers for the slave channel.

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).

HARDWARE DESIGN CONSIDERATIONS

POWER SUPPLIES

The ADuC7019/20/21/22/24/25/26/27/28/29 operational power supply voltage range is 2.7 V to 3.6 V. Separate analog and digital power supply pins (AV_{DD} and IOV_{DD}, respectively) allow AV_{DD} to be kept relatively free of noisy digital signals often present on the system IOV_{DD} line. In this mode, the part can also operate with split supplies; that is, it can use different voltage levels for each supply. For example, the system can be designed to operate with an IOV_{DD} voltage level of 3.3 V whereas the AV_{DD} level can be at 3 V or vice versa. A typical split supply configuration is shown in Figure 87.

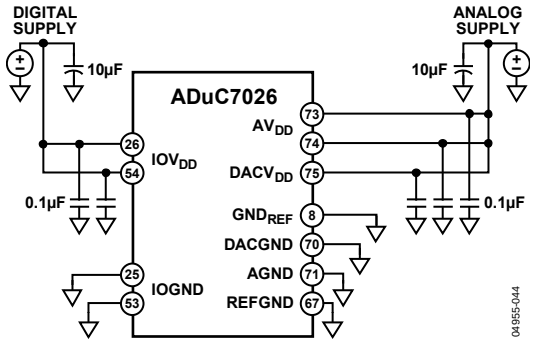


Figure 87. External Dual Supply Connections

As an alternative to providing two separate power supplies, the user can reduce noise on AV_{DD} by placing a small series resistor and/or ferrite bead between AV_{DD} and IOV_{DD} and then decoupling AV_{DD} separately to ground. An example of this configuration is shown in Figure 88. With this configuration, other analog circuitry (such as op amps and voltage reference) can be powered from the AV_{DD} supply line as well.

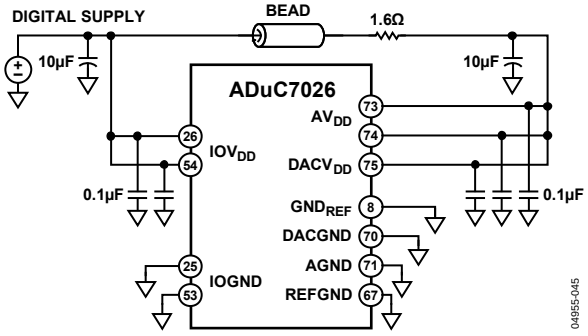


Figure 88. External Single Supply Connections

Note that in both Figure 87 and Figure 88, a large value (10 µF) reservoir capacitor sits on IOV_{DD}, and a separate 10 µF capacitor sits on AV_{DD}. In addition, local small-value (0.1 µF) capacitors are located at each AV_{DD} and IOV_{DD} pin of the chip. As per standard design practice, be sure to include all of these capacitors and ensure that the smaller capacitors are close to each AV_{DD} pin with trace lengths as short as possible. Connect the ground terminal of each of these capacitors directly to the underlying ground plane.

Finally, note that the analog and digital ground pins on the ADuC7019/20/21/22/24/25/26/27/28/29 must be referenced to the same system ground reference point at all times.

IOV_{DD} Supply Sensitivity

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

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

If decoupling values recommended in the Power Supplies section do not sufficiently dampen all noise sources below 50 mV on IOV_{DD}, a filter such as the one shown in Figure 89 is recommended.

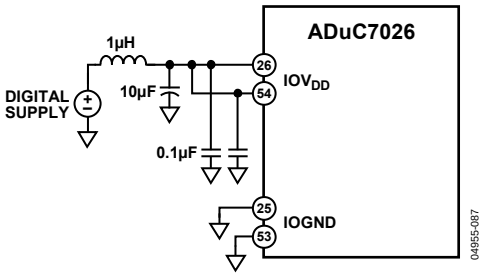


Figure 89. Recommended IOV_{DD} Supply Filter

Linear Voltage Regulator

Each ADuC7019/20/21/22/24/25/26/27/28/29 requires a single 3.3 V supply, but the core logic requires a 2.6 V supply. An on-chip linear regulator generates the 2.6 V from IOV_{DD} for the core logic. The LV_{DD} pin is the 2.6 V supply for the core logic. An external compensation capacitor of 0.47 µF must be connected between LV_{DD} and DGND (as close as possible to these pins) to act as a tank of charge as shown in Figure 90.

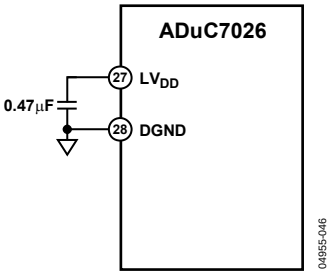


Figure 90. Voltage Regulator Connections

The LV_{DD} pin should not be used for any other chip. It is also recommended to use excellent power supply decoupling on IOV_{DD} to help improve line regulation performance of the on-chip voltage regulator.

POWER-ON RESET OPERATION

An internal power-on reset (POR) is implemented on the ADuC7019/20/21/22/24/25/26/27/28/29. For LV_{DD} below 2.35 V typical, the internal POR holds the part in reset. As LV_{DD} rises above 2.35 V, an internal timer times out for, typically, 128 ms before the part is released from reset. The user must ensure that the power supply IOV_{DD} reaches a stable 2.7 V minimum level by this time. Likewise, on power-down, the internal POR holds the part in reset until LV_{DD} drops below 2.35 V.

Figure 94 illustrates the operation of the internal POR in detail.

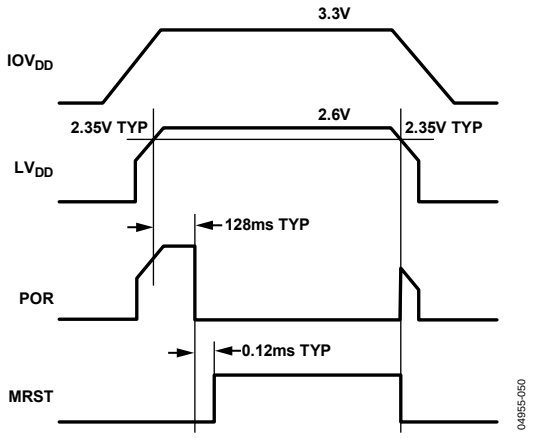


Figure 94. Internal Power-On Reset Operation

TYPICAL SYSTEM CONFIGURATION

A typical ADuC7020 configuration is shown in Figure 95. It summarizes some of the hardware considerations discussed in the previous sections. The bottom of the CSP package has an exposed pad that must be soldered to a metal plate on the board for mechanical reasons. The metal plate of the board can be connected to ground.

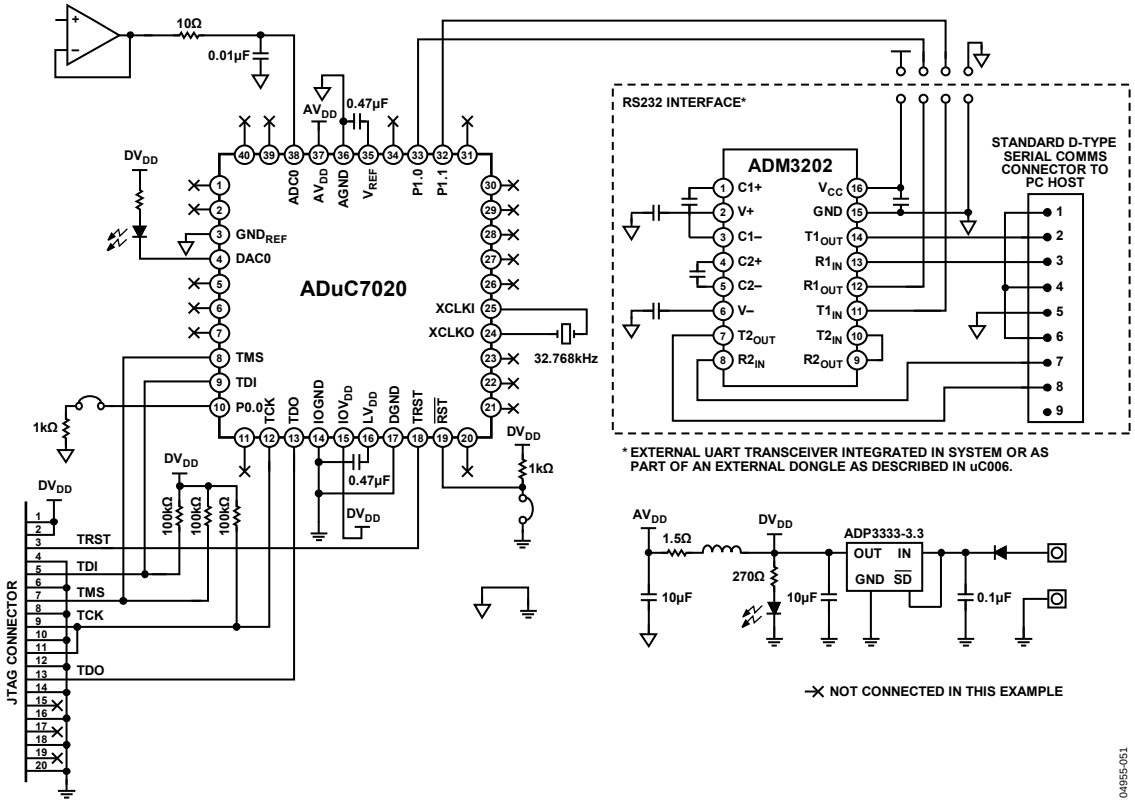
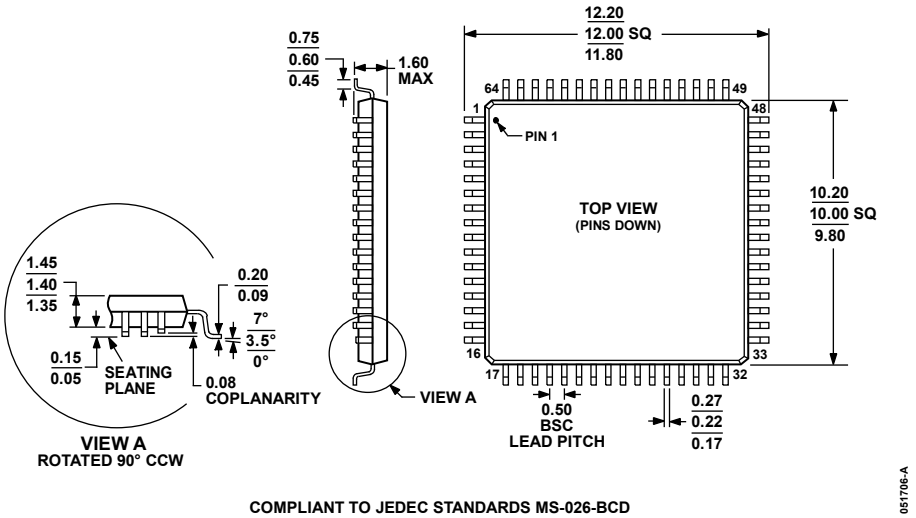
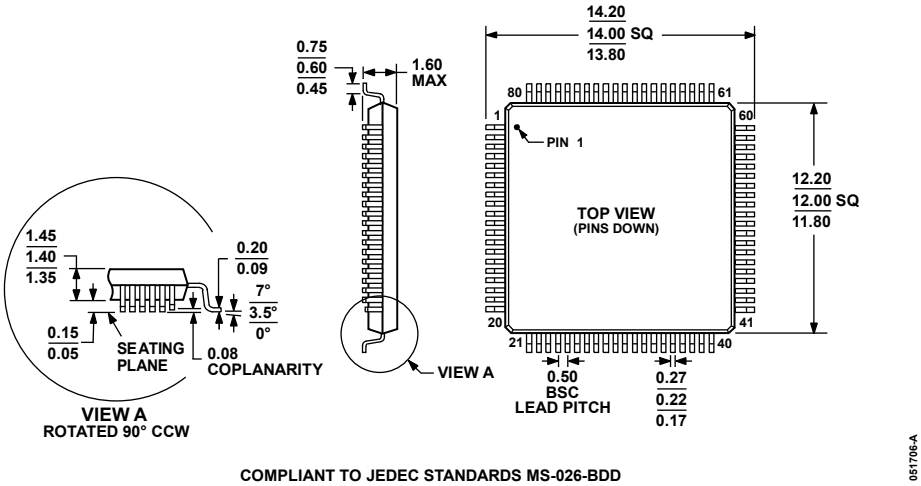


Figure 95. Typical System Configuration



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