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	10MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	POR, PWM, Temp Sensor, WDT
Number of I/O	14
Program Memory Size	32KB (16K x 16)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1K x 32
Voltage - Supply (Vcc/Vdd)	2.375V ~ 2.625V
Data Converters	A/D 5x24b, 8x24b; D/A 1x14b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TA)
Mounting Type	Surface Mount
Package / Case	48-LQFP
Supplier Device Package	48-LQFP (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7060bstz32

FUNCTIONAL BLOCK DIAGRAM

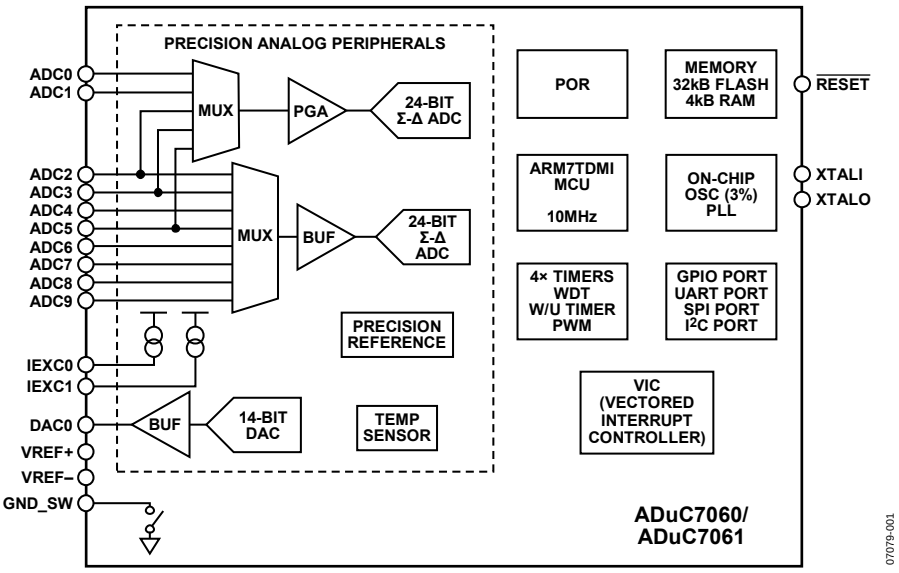

Figure 1.

TIMING SPECIFICATIONS

I²C Timing

Table 2. I²C® Timing in Standard Mode (100 kHz)

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

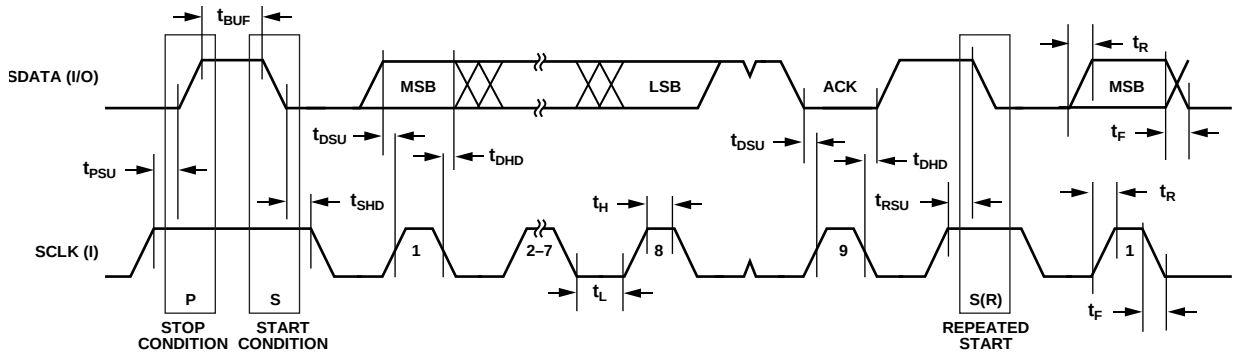

Figure 2. I²C Compatible Interface Timing

such as C, it is necessary to ensure that the stack does not overflow. This is dependent on the performance of the compiler that is used.

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 9. The FIQ mode has more registers (R8 to R12) supporting faster interrupt processing. With the increased number of noncritical registers, the interrupt can be processed without the need to save or restore these registers, thereby reducing the response time of the interrupt handling process.

More information relative to the programmer's model and the ARM7TDMI core architecture can be found in ARM7TDMI technical and ARM architecture manuals available directly from ARM Ltd.

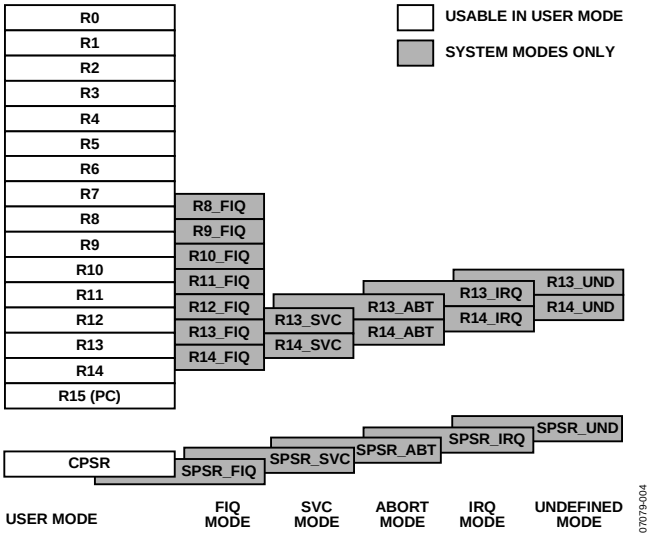

Figure 9. Register Organization

INTERRUPT LATENCY

The worst-case latency for an FIQ consists of the longest time that the request can take to pass through the synchronizer, plus the time for the longest instruction to complete (the longest instruction is an LDM) that loads all the registers including the PC, plus the time for the data abort entry, plus the time for FIQ entry. At the end of this time, the ARM7TDMI is executing the instruction at 0x1C (FIQ interrupt vector address). The maximum total time is 50 processor cycles, or just over 4.88 µs in a system using a continuous 10.24 MHz processor clock. The maximum IRQ latency calculation is similar but must allow for the FIQ having higher priority, which can delay entry into the IRQ handling routine for an arbitrary length of time. This time can be reduced to 42 cycles if the LDM command is not used; some compilers have an option to compile without using this command. Another option is to run the part in Thumb mode where this time is reduced to 22 cycles.

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

Note that the ARM7TDMI initially (first instruction) runs in ARM (32-bit) mode when an exception occurs. The user can immediately switch from ARM mode to Thumb mode if required, for example, when executing interrupt service routines.

MEMORY ORGANIZATION

The ARM7, a von Neumann architecture MCU core, sees memory as a linear array of 232-byte locations. As shown in Figure 10, the ADuC7060/ADuC7061 maps this into four distinct user areas: a memory area that can be remapped, an SRAM area, a Flash/EE area, and a memory mapped register (MMR) area.

The first 30 kB of this memory space is used as an area into which the on-chip Flash/EE or SRAM can be remapped. Any access, either reading or writing, to an area not defined in the memory map results in a data abort exception.

Memory Format

The ADuC7060/ADuC7061 memory organization is configured in little endian format: the least significant byte is located in the lowest byte address and the most significant byte in the highest byte address (see Figure 11).

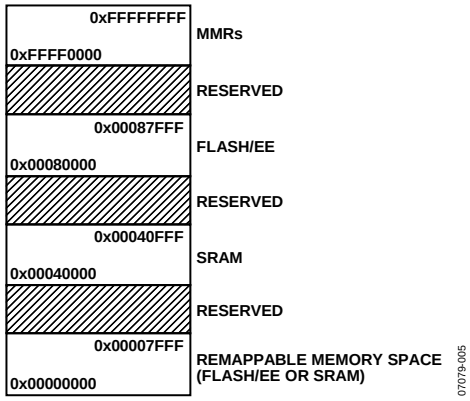

Figure 10. Memory Map

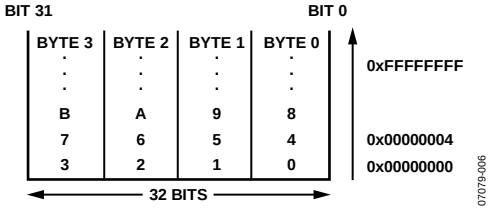

Figure 11. Little Endian Format

SRAM

The ADuC7060/ADuC7061 features 4 kB of SRAM, organized as 1024 × 32 bits, that is, 1024 words located at 0x40000. The RAM space can be used as data memory as well as volatile program space.

ARM code can run directly from SRAM at full clock speed given that the SRAM array is configured as a 32-bit wide memory array. SRAM is read/writable in 8-, 16-, and 32-bit segments.

Table 19. Timer Address Base = 0xFFFF0300

Address	Name	Bytes	Access Type	Default Value	Description
0x0320	T0LD	4	R/W	0x00000000	Timer0 load register.
0x0324	T0VAL	4	R	0xFFFFFFFF	Timer0 value register.
0x0328	T0CON	4	R/W	0x01000000	Timer0 control MMR.
0x032C	T0CLRI	1	W	N/A	Timer0 interrupt clear register.
0x0330	T0CAP	4	R	0x00000000	Timer0 capture register.
0x0340	T1LD	4	R/W	0x00000000	Timer1 load register.
0x0344	T1VAL	4	R	0xFFFFFFFF	Timer1 value register.
0x0348	T1CON	2	R/W	0x0000	Timer1 control MMR.
0x034C	T1CLRI	1	W	N/A	Timer1 interrupt clear register.
0x0360	T2LD	2	R/W	0x3BF8	Timer2 load register.
0x0364	T2VAL	2	R	0x3BF8	Timer2 value register.
0x0368	T2CON	2	R/W	0x0000	Timer2 control MMR.
0x036C	T2CLRI	1	W	N/A	Timer2 interrupt clear register.
0x0380	T3LD	2	R/W	0x0000	Timer3 load register.
0x0384	T3VAL	2	R	0xFFFF	Timer3 value register.
0x0388	T3CON	4	R/W	0x00000000	Timer3 control MMR.
0x038C	T3CLRI	1	W	N/A	Timer3 interrupt clear register.
0x0390	T3CAP	2	R	0x0000	Timer3 capture register.

Table 20. PLL Base Address = 0xFFFF0400

Address	Name	Bytes	Access Type	Default Value	Description
0x0404	POWKEY1	2	W	0xXXXX	POWCON0 prewrite key.
0x0408	POWCON0	1	R/W	0x7B	Power control and core speed control register.
0x040C	POWKEY2	2	W	0xXXXX	POWCON0 postwrite key.
0x0410	PLLKEY1	2	W	0xXXXX	PLLCON prewrite key.
0x0414	PLLCON	1	R/W	0x00	PLL clock source selection MMR.
0x0418	PLLKEY2	2	W	0xXXXX	PLLCON postwrite key.
0x0434	POWKEY3	2	W	0xXXXX	POWCON1 prewrite key.
0x0438	POWCON1	2	R/W	0x124	Power control register.
0x043C	POWKEY4	2	W	0xXXXX	POWCON1 postwrite key.
0x0464	GP0KEY1	2	W	0xXXXX	GP0CON1 prewrite key.
0x0468	GP0CON1	1	R/W	0x00	Configures P0.0, P0.1, P0.2, and P0.3 as analog inputs or digital I/Os. Also enables SPI or I^2C mode.
0x046C	GP0KEY2	2	W	0xXXXX	GP0CON1 postwrite key.

Table 21. ADC Address Base = 0xFFFF0500

Address	Name	Bytes	Access Type	Default Value	Description
0x0500	ADCSTA	2	R	0x0000	ADC status MMR.
0x0504	ADCMSKI	2	R/W	0x0000	ADC interrupt source enable MMR.
0x0508	ADCMDE	1	R/W	0x03	ADC mode register.
0x050C	ADC0CON	2	R/W	0x8000	Primary ADC control MMR.
0x0510	ADC1CON	2	R/W	0x0000	Auxiliary ADC control MMR.
0x0514	ADCFLT	2	R/W	0x0007	ADC filter control MMR.
0x0518	ADCCFG	1	R/W	0x00	ADC configuration MMR.
0x051C	ADC0DAT	4	R	0x00000000	Primary ADC result MMR.
0x0520	ADC1DAT	4	R	0x00000000	Auxiliary ADC result MMR
0x0524	ADC0OF[1]	2	R/W	0x0000, part specific, factory programmed	Primary ADC offset calibration setting.
0x0528	ADC1OF[1]	2	R/W	0x0000, part specific, factory programmed	Auxiliary ADC offset MMR.
0x052C	ADC0GN[1]	2	R/W	0x5555	Primary ADC offset MMR.
0x0530	ADC1GN[1]	2	R/W	0x5555	Auxiliary ADC offset MMR. See the ADC operation mode configuration bit (ADCLPMCFG[1:0]) in Table 42.
0x0534	ADC0RCR	2	R/W	0x0001	Primary ADC result counter/reload MMR.
0x0538	ADC0RCV	2	R	0x0000	Primary ADC result counter MMR.
0x053C	ADC0TH	2	R/W	0x0000	Primary ADC 16-bit comparator threshold MMR.
0x0540	ADC0THC	2	R/W	0x0001	Primary ADC 16-bit comparator threshold counter limit.
0x0544	ADC0THV	2	R	0x0000	ADC0 8-bit threshold exceeded counter register
0x0548	ADC0ACC	4	R	0x00000000	Primary ADC accumulator.
0x054C	ADC0ATH	4	R/W	0x00000000	Primary ADC 32-bit comparator threshold MMR.
0x0570	IEXCON	1	R/W	0x00	Excitation current sources control register.

[1] Updated by the kernel to part specific calibration value.

Table 22. DAC Control Address Base = 0xFFFF0600

Address	Name	Bytes	Access Type	Default Value	Description
0x0600	DAC0CON	2	R/W	0x0200	DAC control register.
0x0604	DAC0DAT	4	R/W	0x00000000	DAC output data register.

Table 23. UART Base Address = 0xFFFF0700

Address	Name	Bytes	Access Type	Default Value	Description
0x0700	COMTX	1	W	N/A	UART transmit register.
0x0700	COMRX	1	R	0x00	UART receive register.
0x0700	COMDIV0	1	R/W	0x00	UART Standard Baud Rate Generator Divisor Value 0.
0x0704	COMIEN0	1	R/W	0x00	UART Interrupt Enable MMR 0.
0x0704	COMDIV1	1	R/W	0x00	UART Standard Baud Rate Generator Divisor Value 1.
0x0708	COMIID0	1	R	0x01	UART Interrupt Identification 0.
0x070C	COMCON0	1	R/W	0x00	UART Control Register 0.
0x0710	COMCON1	1	R/W	0x00	UART Control Register 1.
0x0714	COMSTA0	1	R	0x60	UART Status Register 0.
0x0718	COMSTA1	1	R	0x00	UART Status Register 1.
0X072C	COMDIV2	2	R/W	0x0000	UART fractional divider MMR.

ADC Interrupt Mask Register

Name: ADCMSKI

Address: 0xFFFF0504

Default value: 0x0000

Access: Read and write

Function: This register allows the ADC interrupt sources to be enabled individually. The bit positions in this register are the same as the lower eight bits in the ADCSTA MMR. If a bit is set by user code to 1, the respective interrupt is enabled. By default, all bits are 0, meaning all ADC interrupt sources are disabled.

Table 41. ADCMSKI MMR Bit Designations

Bit	Name	Description
7		Not used. This bit is reserved for future functionality and should not be monitored by user code.
6	ADC0ATHEX_INTEN	ADC0 accumulator comparator threshold exceeded interrupt enable bit. When set to 1, this bit enables an interrupt when the ADC0ATHEX bit in the ADCSTA register is set. When this bit is cleared, this interrupt source is disabled.
5		Not used. This bit is reserved for future functionality and should not be monitored by user code.
4	ADC0THEX_INTEN	Primary channel ADC comparator threshold exceeded interrupt enable bit. When set to 1, this bit enables an interrupt when the ADC0THEX bit in the ADCSTA register is set. When this bit is cleared, this interrupt source is disabled.
3	ADC0OVR_INTEN	When set to 1, this bit enables an interrupt when the ADC0OVR bit in the ADCSTA register is set. When this bit is cleared, this interrupt source is disabled.
2		Not used. This bit is reserved for future functionality and should not be monitored by user code.
1	ADC1RDY_INTEN	Auxiliary ADC result ready bit. When set to 1, this bit enables an interrupt when the ADC1RDY bit in the ADCSTA register is set. When this bit is cleared, this interrupt source is disabled.
0	ADC0RDY_INTEN	Primary ADC result ready bit. When set to 1, this bit enables an interrupt when the ADC0RDY bit in the ADCSTA register is set. When this bit is cleared, this interrupt source is disabled.

ADC Mode Register

Name: ADCMDE

Address: 0xFFFF0508

Default value: 0x03

Access: Read and write

Function: The ADC mode MMR is an 8-bit register that configures the mode of operation of the ADC subsystem.

Table 42. ADCMDE MMR Bit Designations

Bit	Name	Description
7	ADCCLKSEL	Set this bit to 1 to enable ADCCLK = 512 kHz. This bit should be set for normal ADC operation. Clear this bit to enable ADCCLK = 131 kHz. This bit should be cleared for low power ADC operation.
6		Not used. This bit is reserved for future functionality and should not be monitored by user code.
5	ADCLPMEN	Enable low power mode. This bit has no effect if ADCMDE[4:3] = 00 (ADC is in normal mode). This bit must be set to 1 in low power mode. Clearing this bit in low power mode results in erratic ADC results.

Primary Channel ADC Data Register

Name: ADC0DAT

Address: 0xFFFF051C

Default value: 0x00000000

Access: Read only

Function: This ADC data MMR holds the 24-bit conversion result from the primary ADC. The ADC does not update this MMR if the ADC0 conversion result ready bit (ADCSTA[0]) is set. A read of this MMR by the MCU clears all asserted ready flags (ADCSTA[1:0]).

Table 49. ADC0DAT MMR Bit Designations

Bit	Description
23:0	ADC0 24-bit conversion result.

Auxiliary Channel ADC Data Register

Name: ADC1DAT

Address: 0xFFFF0520

Default value: 0x00000000

Access: Read only

Function: This ADC data MMR holds the 24-bit conversion result from the auxiliary ADC. The ADC does not update this MMR if the ADC0 conversion result ready bit (ADCSTA[1]) is set.

Table 50. ADC1DAT MMR Bit Designations

Bit	Description
23:0	ADC1 24-bit conversion result.

Primary Channel ADC Offset Calibration Register

Name: ADC0OF

Address: 0xFFFF0524

Default value: Part specific, factory programmed

Access: Read and write

Function: This ADC offset MMR holds a 16-bit offset calibration coefficient for the primary ADC. The register is configured at power-on with a factory default value. However, this register automatically overwrites if an offset calibration of the primary ADC is initiated by the user via bits in the ADCMDE MMR. User code can write to this calibration register only if the ADC is in idle mode. An ADC must be enabled and in idle mode before being written to any offset or gain register. The ADC must be in idle mode for at least 23 μs.

Table 51. ADC0OF MMR Bit Designations

Bit	Description
15:0	ADC0 16-bit offset calibration value.

Auxiliary Channel ADC Offset Calibration Register

Name: ADC1OF

Address: 0xFFFF0528

Default value: Part specific, factory programmed

Access: Read and write

Function: This offset MMR holds a 16-bit offset calibration coefficient for the auxiliary channel. The register is configured at power-on with a factory default value. However, this register is automatically overwritten if an offset calibration of the auxiliary channel is initiated by the user via bits in the ADCMDE MMR. User code can write to this calibration register only if the ADC is in idle mode. An ADC must be enabled and in idle mode before being written to any offset or gain register. The ADC must be in idle mode for at least 23 μs.

Table 52. ADC1OF MMR Bit Designations

Bit	Description
15:0	ADC1 16-bit offset calibration value.

Primary Channel ADC Gain Calibration Register

Name: ADC0GN

Address: 0xFFFF052C

Default value: Part specific, factory programmed

Access: Read and write

Function: This gain MMR holds a 16-bit gain calibration coefficient for scaling the primary ADC conversion result. The register is configured at power-on with a factory default value. However, this register is automatically overwritten if a gain calibration of the primary ADC is initiated by the user via bits in the ADCMDE MMR. User code can write to this calibration register only if the ADC is in idle mode. An ADC must be enabled and in idle mode before being written to any offset or gain register. The ADC must be in idle mode for at least 23 μs.

Table 53. ADC0GN MMR Bit Designations

Bits	Description
15:0	ADC0 16-bit calibration gain value.

Auxiliary Channel Gain Calibration Register

Name: ADC1GN

Address: 0xFFFF0530

Default value: Part specific, factory programmed

Access: Read and write

Function: This gain MMR holds a 16-bit gain calibration coefficient for scaling an auxiliary channel conversion result. The register is configured at power-on with a factory default value. However, this register is automatically overwritten if a gain calibration of the auxiliary channel is initiated by the user via bits in the ADCMDE MMR. User code can write to this calibration register only if the ADC is in idle mode. An ADC must be enabled and in idle mode before being written to any offset or gain register. The ADC must be in idle mode for at least 23 μs.

Table 54. ADC1GN MMR Bit Designations

Bits	Description
15:0	ADC1 16-bit gain calibration value.

Primary Channel ADC Result Counter Limit Register

Name: ADC0RCR

Address: 0xFFFF0534

Default value: 0x0001

Access: Read and write

Function: This 16-bit MMR sets the number of conversions required before an ADC interrupt is generated. By default, this register is set to 0x01. The ADC counter function must be enabled via the ADC result counter enable bit in the ADCCFG MMR.

Table 55. ADC0RCR MMR Bit Designations

Bits	Description
15:0	ADC0 result counter limit/reload register.

Primary Channel ADC Result Counter Register

Name: ADC0RCV

Address: 0xFFFF0538

Default value: 0x0000

Access: Read only

Function: This 16-bit, read-only MMR holds the current number of primary ADC conversion results. It is used in conjunction with ADC0RCR to mask primary channel ADC interrupts, generating a lower interrupt rate. When ADC0RCV = ADC0RCR, the value in ADC0RCV resets to 0 and recommences counting. It can also be used in conjunction with the accumulator (ADC0ACC) to allow an average calculation to be taken. The result counter is enabled via ADCCFG[0]. This MMR is also reset to 0 when the primary ADC is reconfigured, that is, when the ADC0CON or ADCMDE is written.

Table 56. ADC0RCV MMR Bit Designations

Bits	Description
15:0	ADC0 result counter register.

Primary Channel ADC Threshold Register

Name: ADC0TH

Address: 0xFFFF053C

Default value: 0x0000

Access: Read and write

Function: This 16-bit MMR sets the threshold against which the absolute value of the primary ADC conversion result is compared. In unipolar mode, ADC0TH[15:0] are compared, and in twos complement mode, ADC0TH[14:0] are compared.

Table 57. ADC0TH MMR Bit Designations

Bit	Description
15:0	ADC0 16-bit comparator threshold register.

Primary Channel ADC Threshold Counter Limit Register

Name: ADC0THC

Address: 0xFFFF0540

Default value: 0x0001

Access: Read and write

Function: This 8-bit MMR determines how many cumulative (values below the threshold decrement or reset the count to 0) primary ADC conversion result readings above ADC0TH must occur before the primary ADC comparator threshold bit is set in the ADCSTA MMR, generating an ADC interrupt. The primary ADC comparator threshold bit is asserted as soon as ADC0THV = ADC0RCR.

Table 58. ADC0THC MMR Bit Designations

Bit	Description
15:8	Reserved.
7:0	ADC0 8-bit threshold counter limit register.

Primary Channel ADC Threshold Counter Register

Name: ADC0THV

Address: 0xFFFF0544

Default value: 0x0000

Access: Read only

Function: This 8-bit MMR is incremented every time the absolute value of a primary ADC conversion result |Result| ≥ ADC0TH. This register is decremented or reset to 0 every time the absolute value of a primary ADC conversion result |Result| < ADC0TH. The configuration of this function is enabled via the primary channel ADC comparator bits in the ADCCFG MMR.

Table 59. ADC0THV MMR Bit Designations

Bit	Description
7:0	ADC0 8-bit threshold exceeded counter register.

Primary Channel ADC Accumulator Register

Name: ADC0ACC

Address: 0xFFFF0548

Default value: 0x00000000

Access: Read only

Function: This 32-bit MMR holds the primary ADC accumulator value. The primary ADC ready bit in the ADCSTA MMR should be used to determine when it is safe to read this MMR. The MMR value is reset to 0 by disabling the accumulator in the ADCCFG MMR or by reconfiguring the primary channel ADC.

Table 60. ADC0ACC MMR Bit Designations

Bit	Description
31:0	ADC0 32-bit accumulator register.

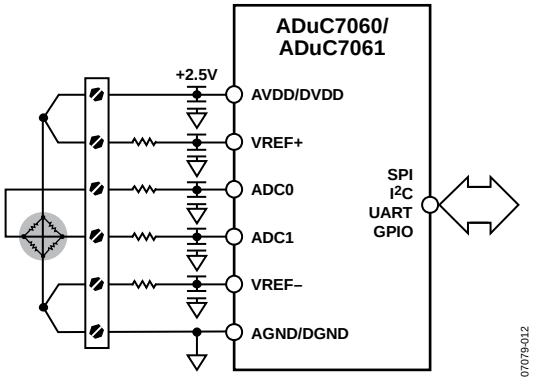

Figure 18. Bridge Interface Circuit

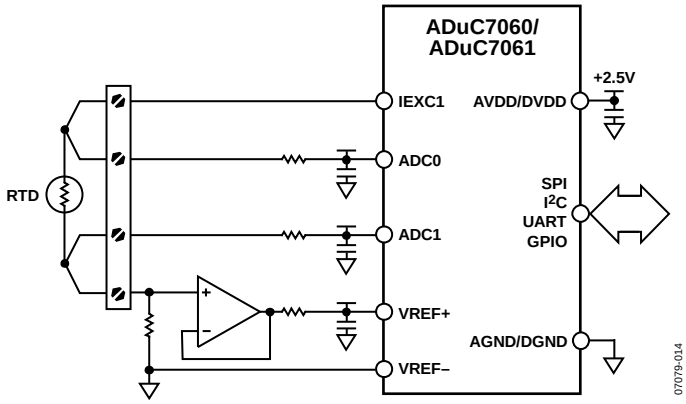

Figure 20. Example of an RTD Interface Circuit

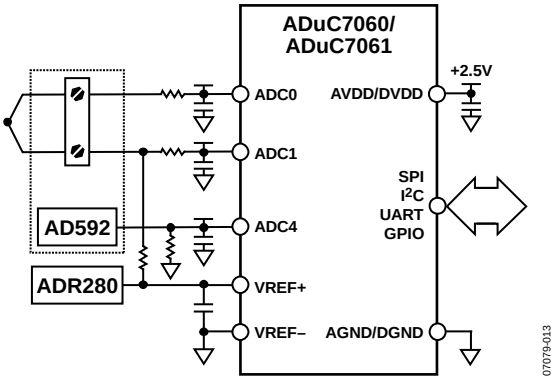

Figure 19. Example of a Thermocouple Interface Circuit

DAC0DAT Register

Name: DAC0DAT

Address: 0xFFFF0604

Default value: 0x00000000

Access: Read and write

Function: This 32-bit MMR contains the DAC output value.

Table 64. DAC0DAT MMR Bit Designations

Bit	Description
31:28	Reserved.
27:16	12-bit data for DAC0.
15:12	Extra four bits used in interpolation mode.
11:0	Reserved.

USING THE DAC

The on-chip DAC architecture consists of a resistor string DAC followed by an output buffer amplifier.

The reference source for the DAC is user selectable in software. It can be AVDD, VREF±, or ADCx/EXT_REF2IN±.

- In 0-to-AVDD mode, the DAC output transfer function spans from 0 V to the voltage at the AVDD pin.
- In VREF± and ADCx/EXT_REF2IN± modes, the DAC output transfer function spans from negative input voltage to the voltage positive input pin. Note that these voltages must never go below 0 V or above AVDD.
- In 0-to-V_{REF} mode, the DAC output transfer function spans from 0 V to the internal 1.2 V reference, V_{REF}.

The DAC can be configured in three different user modes: normal mode, DAC interpolation mode, and op amp mode.

Normal DAC Mode

In this mode of operation, the DAC is configured as a 12-bit voltage output DAC. By default, the DAC buffer is enabled, but the output buffer can be disabled. If the DAC output buffer is disabled, the DAC is capable of driving a capacitive load of only 20 pF. The DAC buffer is disabled by setting the DACBUFBYPASS bit in DAC0CON.

The DAC output buffer amplifier features a true, rail-to-rail output stage implementation. This means that when unloaded, each output is capable of swinging to within less than 5 mV of both AVDD and ground. Moreover, the linearity specification of the DAC (when driving a 5 kΩ resistive load to ground) is guaranteed through the full transfer function except for Code 0 to Code 100 and, in 0-to- AVDD mode only, Code 3995 to Code 4095. Linearity degradation near ground and AVDD is caused by saturation of the output amplifier, and a general representation of its effects (neglecting offset and gain error) is illustrated in Figure 21. The dotted line in Figure 21 indicates the ideal transfer function, and the solid line represents what the transfer function may look like with endpoint nonlinearities due to saturation of the output amplifier. Note that Figure 21 represents a transfer function in 0-to-AVDD mode only. In 0-to-V_{REF} or, VREF±, and ADCx/EXT_REF2IN± modes (with V_{REF} < AVDD or ADCx/EXT_REF2IN± < AVDD), the lower nonlinearity is similar. However, the upper portion of the transfer function follows the ideal line all the way to the end (V_{REF} in this case, not AVDD), showing no signs of endpoint linearity errors.

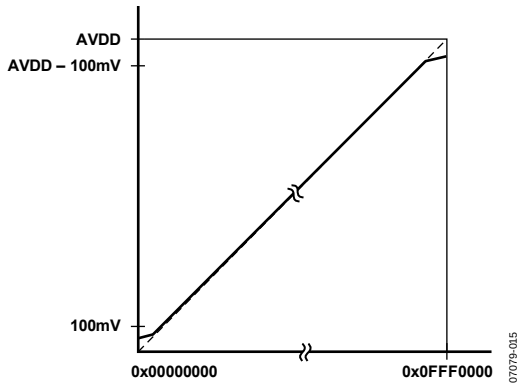

Figure 21. Endpoint Nonlinearities Due to Amplifier Saturation

The endpoint nonlinearities conceptually illustrated in Figure 21 worsen as a function of output loading. Most of the ADuC7060/ADuC7061 data sheet specifications in normal mode assume a 5 kΩ resistive load to ground at the DAC output. As the output is forced to source or sink more current, the nonlinear regions at the top or bottom (respectively) of Figure 21 become larger. With larger current demands, this can significantly limit output voltage swing.

DAC Interpolation Mode

In interpolation mode, a higher DAC output resolution of 16 bits is achieved with a longer update rate than normal mode. The update rate is controlled by the interpolation clock rate selected in the DAC0CON register. In this mode, an external RC filter is required to create a constant voltage.

Op Amp Mode

In op amp mode, the DAC output buffer is used as an op amp with the DAC itself disabled.

ADC6 is the positive input to the op amp, ADC7 is the negative input, and ADC8 is the output. In this mode, the DAC should be powered down by setting Bit 9 of DAC0CON.

NONVOLATILE FLASH/EE MEMORY

The ADuC7060/ADuC7061 incorporates Flash/EE memory technology on chip to provide the user with nonvolatile, in-circuit reprogrammable memory space.

Like EEPROM, flash memory can be programmed in-system at a byte level, although it must first be erased. The erase is performed in page blocks. As a result, flash memory is often and, more correctly, referred to as Flash/EE memory.

Overall, Flash/EE memory represents a step closer to the ideal memory device that includes nonvolatility, in-circuit programmability, high density, and low cost. Incorporated in the ADuC7060/ADuC7061, Flash/EE memory technology allows the user to update program code space in-circuit, without the need to replace one time programmable (OTP) devices at remote operating nodes.

The ADuC7060/ADuC7061 contains a 32 kB array of Flash/EE memory. The lower 30 kB are available to the user and the upper 2 kB contain permanently embedded firmware, allowing in-circuit serial download. These 2 kB of embedded firmware also contain a power-on configuration routine that downloads factory-calibrated coefficients to the various calibrated peripherals (such as ADC, temperature sensor, and band gap references). This 2 kB embedded firmware is hidden from user code.

FLASH/EE MEMORY RELIABILITY

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

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

- Initial page erase sequence
- Read/verify sequence for a single Flash/EE
- Byte program sequence memory
- Second read/verify sequence endurance cycle

In reliability qualification, every half word (16-bit wide) location of the three pages (top, middle, and bottom) in the Flash/EE memory is cycled 10,000 times from 0x0000 to 0xFFFF. The Flash/EE memory endurance qualification is carried out in accordance with JEDEC *Retention Lifetime Specification A117* over the industrial temperature range of −40°C to +125°C. The results allow the specification of a minimum endurance figure over a supply temperature of 10,000 cycles.

Retention quantifies the ability of the Flash/EE memory to retain its programmed data over time. Again, the parts are qualified in accordance with the formal *JEDEC Retention Lifetime Specification A117* at a specific junction temperature (T_J = 85°C). As part of this qualification procedure, the Flash/EE memory is cycled to its specified endurance limit, described previously, before data retention is characterized. This means that the Flash/EE memory is guaranteed to retain its data for its fully specified retention lifetime every time that the Flash/EE memory is reprogrammed. Also note that retention lifetime, based on activation energy of 0.6 eV, derates with T_J, as shown in Figure 22.

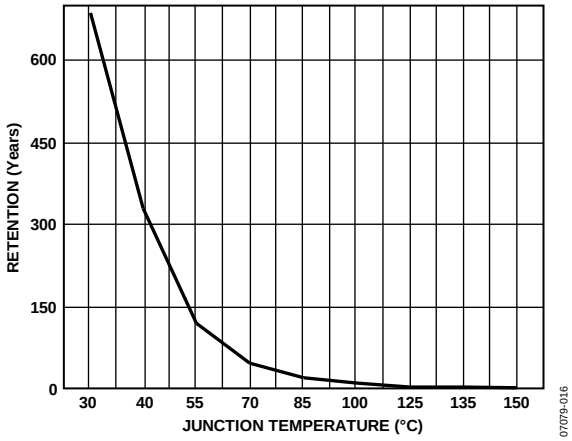

Figure 22. Flash/EE Memory Data Retention

PROGRAMMING

The 30 kB of Flash/EE memory can be programmed in-circuit, using the serial download mode or the provided JTAG mode.

Serial Downloading (In-Circuit Programming)

The ADuC7060/ADuC7061 facilitates code download via the standard UART serial port. The parts enter serial download mode after a reset or power cycle if the NTRST/$\overline{\text{BM}}$ pin is pulled low through an external 1 kΩ resistor. When in serial download mode, the user can download code to the full 30 kB of Flash/EE memory while the device is in-circuit in its target application hardware. An executable PC serial download is provided as part of the development system for serial downloading via the UART.

When the ADuC7060/ADuC7061 enters download mode, the user should be aware that the internal watchdog is enabled with a time-out period of 2 minutes. If the flash erase/write sequence is not completed in this period, a reset occurs.

JTAG Access

The JTAG protocol uses the on-chip JTAG interface to facilitate code download and debug.

TIMER0

Timer0 is a 32-bit, general-purpose timer, count down or count up, with a programmable prescaler. The prescaler source can be the low power 32.768 kHz oscillator, the core clock, or from one of two external GPIOs. This source can be scaled by a factor of 1, 16, 256, or 32,768. This gives a minimum resolution of 97.66 ns with a prescaler of 1 (ignoring the external GPIOs).

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

Timer0 has a capture register (T0CAP) that is triggered by a selected IRQ source initial assertion. When triggered, the current timer value is copied to T0CAP, and the timer continues to run. Use this feature to determine the assertion of an event with increased accuracy. Note that only peripherals that have their IRQ source enabled can be used with the timer capture feature.

The Timer0 interface consists of five MMRS: T0LD, T0VAL, T0CAP, T0CLRI, and T0CON.

- T0LD, T0VAL, and T0CAP are 32-bit registers and hold 32-bit, unsigned integers of which T0VAL and T0CAP are read only.
- T0CLRI is an 8-bit register and writing any value to this register clears the Timer0 interrupt.
- T0CON is the configuration MMR, which is described in Table 80.

Timer0 features a postscaler that allows the user to count between 1 and 256 the number of Timer0 timeouts. To activate the postscaler, the user sets Bit 18 and writes the desired number to count into Bits[24:31] of T0CON. When that number of timeouts is reached, Timer0 can generate an interrupt if T0CON[18] is set.

Note that, if the part is in a low power mode and Timer0 is clocked from the GPIO or low power oscillator source, Timer0 continues to operate.

Timer0 reloads the value from T0LD when Timer0 overflows.

Timer0 Load Registers

Name: T0LD

Address: 0xFFFF0320

Default value: 0x00000000

Access: Read and write

Function: T0LD is a 32-bit register that holds the 32-bit value that is loaded into the counter.

Timer0 Clear Register

Name: T0CLRI

Address: 0xFFFF032C

Access: Write only

Function: This 8-bit, write-only MMR is written (with any value) by user code to clear the interrupt.

Timer0 Value Register

Name: T0VAL

Address: 0xFFFF0324

Default value: 0xFFFFFFFF

Access: Read only

Function: T0VAL is a 32-bit register that holds the current value of Timer0.

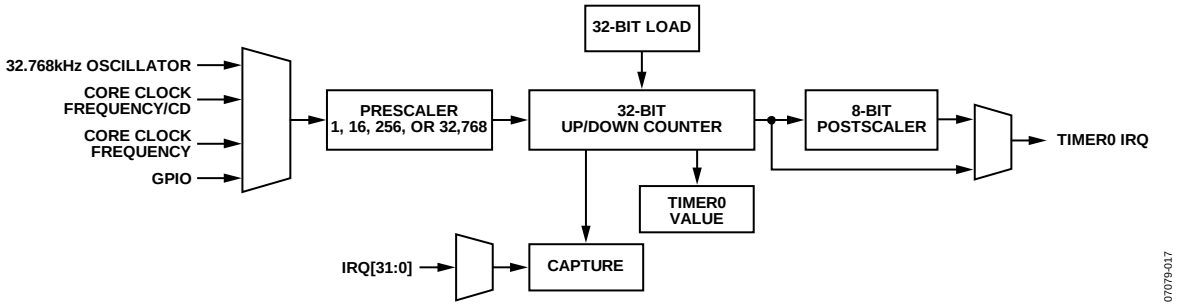

Figure 23. Timer0 Block Diagram

PWM2COM0 Compare Register

Name: PWM2COM0

Address: 0xFFFF0FA4

Default value: 0x0000

Access: Read and write

Function: PWM4 output pin goes high when the PWM timer reaches the count value stored in this register.

PWM2COM1 Compare Register

Name: PWM2COM1

Address: 0xFFFF0FA8

Default value: 0x0000

Access: Read and write

Function: PWM4 output pin goes low when the PWM timer reaches the count value stored in this register.

PWM2COM2 Compare Register

Name: PWM2COM2

Address: 0xFFFF0FAC

Default value: 0x0000

Access: Read and write

Function: PWM5 output pin goes low when the PWM timer reaches the count value stored in this register.

PWM2LEN Register

Name: PWM2LEN

Address: 0xFFFF0FB0

Default value: 0x0000

Access: Read and write

Function: PWM5 output pin goes high when the PWM timer reaches the value stored in this register.

PWMCLRI Register

Name: PWMCLRI

Address: 0xFFFF0FB8

Default value: 0x0000

Access: Write only

Function: Write any value to this register to clear a PWM interrupt source. This register must be written to before exiting a PWM interrupt service routine; otherwise, multiple interrupts occur.

Table 90. COMCON0 MMR Bit Designations

Bit	Name	Description
7	DLAB	Divisor latch access. Set by user to enable access to the COMDIV0 and COMDIV1 registers. Cleared by user to disable access to COMDIV0 and COMDIV1 and enable access to COMRX, COMTX, and COMIEN0.
6	BRK	Set break. Set by user to force transmit to 0. Cleared to operate in normal mode.
5	SP	Stick parity. Set by user to force parity to defined values. 1 if EPS = 1 and PEN = 1. 0 if EPS = 0 and PEN = 1.
4	EPS	Even parity select bit. Set for even parity. Cleared for odd parity.
3	PEN	Parity enable bit. Set by user to transmit and check the parity bit. Cleared by user for no parity transmission or checking.
2	Stop	Stop bit. Set by user to transmit 1.5 stop bits if the word length is 5 bits, or 2 stop bits if the word length is 6 bits, 7 bits, or 8 bits. The receiver checks the first stop bit only, regardless of the number of stop bits selected. Cleared by user to generate one stop bit in the transmitted data.
1:0	WLS	Word length select. [00] = 5 bits. [01] = 6 bits. [10] = 7 bits. [11] = 8 bits.

Table 97. I2CMCON MMR Bit Designations

Bit	Name	Description
15:9		Reserved. These bits are reserved and should not be written to.
8	I2CMCENI	I^2C transmission complete interrupt enable bit. Set this bit to enable an interrupt on detecting a stop condition on the I^2C bus. Clear this interrupt source.
7	I2CNACKENI	I^2C no acknowledge (NACK) received interrupt enable bit. Set this bit to enable interrupts when the I^2C master receives a no acknowledge. Clear this interrupt source.
6	I2CALENI	I^2C arbitration lost interrupt enable bit. Set this bit to enable interrupts when the I^2C master did not gain control of the I^2C bus. Clear this interrupt source.
5	I2CMTENI	I^2C transmit interrupt enable bit. Set this bit to enable interrupts when the I^2C master has transmitted a byte. Clear this interrupt source.
4	I2CMRENI	I^2C receive interrupt enable bit. Set this bit to enable interrupts when the I^2C master receives data. Cleared by user to disable interrupts when the I^2C master is receiving data.
3		Reserved. A value of 0 should be written to this bit.
2	I2CILEN	I^2C internal loopback enable. Set this bit to enable loopback test mode. In this mode, the SCL and SDA signals are connected internally to their respective input signals. Cleared by user to disable loopback mode.
1	I2CBD	I^2C master backoff disable bit. Set this bit to allow the device to compete for control of the bus even if another device is currently driving a start condition. Clear this bit to back off until the I^2C bus becomes free.
0	I2CMEN	I^2C master enable bit. Set by user to enable the I^2C master mode. Cleared to disable the I^2C master mode.

I²C Slave Status, I2CSSTA, Register

Name: I2CSSTA

Address: 0xFFFF092C

Default value: 0x0000

Access: Read and write

Function: This 16-bit MMR is the I²C status register in slave mode.

Table 105. I2CSSTA MMR Bit Designations

Bit	Name	Description
15		Reserved bit.
14	I2CSTA	This bit is set to 1 if a start condition followed by a matching address is detected, a start byte (0x01) is received, or general calls are enabled and a general call code of 0x00 is received. This bit is cleared on receiving a stop condition
13	I2CREPS	This bit is set to 1 if a repeated start condition is detected. This bit is cleared on receiving a stop condition. A read of the I2CSSTA register also clears this bit.
12:11	I2CID[1:0]	I²C address matching register. These bits indicate which I2CIDx register matches the received address. [00] = received address matches I2CID0. [01] = received address matches I2CID1. [10] = received address matches I2CID2. [11] = received address matches I2CID3.
10	I2CSS	I²C stop condition after start detected bit. This bit is set to 1 when a stop condition is detected after a previous start and matching address. When the I2CSSENI bit in I2CSCON is set, an interrupt is generated. This bit is cleared by reading this register.
9:8	I2CGCID[1:0]	I²C general call ID bits. [00] = no general call received. [01] = general call reset and program address. [10] = general program address. [11] = general call matching alternative ID. Note that these bits are not cleared by a general call reset command. Clear these bits by writing a 1 to the I2CGCCLR bit in I2CSCON.
7	I2CGC	I²C general call status bit. This bit is set to 1 if the slave receives a general call command of any type. If the command received was a reset command, then all registers return to their default states. If the command received was a hardware general call, the receive FIFO holds the second byte of the command, and this can be compared with the I2CALT register. Clear this bit by writing a 1 to the I2CGCCLR bit in I2CSCON.
6	I2CSBUSY	I²C slave busy status bit. Set to 1 when the slave receives a start condition. Cleared by hardware if the received address does not match any of the I2CIDx registers, the slave device receives a stop condition, or a repeated start address does not match any of the I2CIDx registers.
5	I2CSNA	I²C slave no acknowledge data bit. This bit is set to 1 when the slave responds to a bus address with a no acknowledge. This bit is asserted under the following conditions: if a no acknowledge was returned because there was no data in the transmit FIFO or if the I2CNACKEN bit was set in the I2CSCON register. This bit is cleared in all other conditions.
4	I2CSRxFO	Slave receive FIFO overflow. This bit is set to 1 when a byte is written to the receive FIFO when it is already full. This bit is cleared in all other conditions.
3	I2CSRXQ	I²C slave receive request bit. This bit is set to 1 when the receive FIFO of the slave is not empty. This bit causes an interrupt to occur if the I2CSRXENI bit in I2CSCON is set. The receive FIFO must be read or flushed to clear this bit.

SERIAL PERIPHERAL INTERFACE

The ADuC7060/ADuC7061 integrates a complete hardware serial peripheral interface (SPI) on chip. SPI is an industry standard, synchronous serial interface that allows eight bits of data to be synchronously transmitted and simultaneously received, that is, full duplex up to a maximum bit rate of 2.56 Mbps.

The SPI port can be configured for master or slave operation and typically consists of four pins: MISO, MOSI, SCLK, and $\overline{SS}$.

MISO (MASTER IN, SLAVE OUT) PIN

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

MOSI (MASTER OUT, SLAVE IN) PIN

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

SCLK (SERIAL CLOCK I/O) PIN

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

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

$$f_{SERIAL\ CLOCK} = \frac{f_{UCLK}}{2 \times (1 + SPIDIV)}$$

The maximum speed of the SPI clock is independent of the clock divider bits.

In slave mode, the SPICON register must be configured with the phase and polarity of the expected input clock. The slave accepts data from an external master up to 5.12 Mbps.

In both master and slave modes, data transmit on one edge of the SCLK signal and sample on the other. Therefore, it is important that the polarity and phase be configured the same for the master and slave devices.

SLAVE SELECT (P0.0/$\overline{SS}$) INPUT PIN

In SPI slave mode, a transfer is initiated by the assertion of $\overline{SS}$ on the P0.0/$\overline{SS}$ pin, which is an active low input signal. The SPI port then transmits and receives 8-bit data until the transfer is concluded by deassertion of $\overline{SS}$. In slave mode, $\overline{SS}$ is always an input.

In SPI master mode, $\overline{SS}$ is an active low output signal. It asserts itself automatically at the beginning of a transfer and deasserts itself upon completion.

CONFIGURING EXTERNAL PINS FOR SPI FUNCTIONALITY

The SPI pins of the ADuC7060/ADuC7061 device are represented by the P0[0:3] function of the following pins:

- P0.0/$\overline{SS}$ is the slave chip select pin. In slave mode, this pin is an input and must be driven low by the master. In master mode, this pin is an output and goes low at the beginning of a transfer and high at the end of a transfer.
- P0.1/SCLK/SCL is the SCLK pin.
- P0.2/MISO is the master in, slave out (MISO) pin.
- P0.3/MOSI/SDA is the master out, slave in (MOSI) pin.

To configure P0.0 to P0.3 for SPI mode, Bit 0, Bit 4, Bit 8, and Bit 12 of the GP0CON0 register must be set to 1. Bit 1 of the GP0CON1 must be set to 1. Note that to write to GP0CON1, the GP0KEY1 register must be set to 0x7 immediately before writing to GP0CON1. Also, the GP0KEY2 register must be set to 0x13 immediately after writing to GP0CON1. The following code example shows this in detail:

```
GP0CON0 = BIT0 + BIT4 + BIT8 + BIT12;   //Select SPI/I²C alternative function for P0[0...3]
GP0KEY1 = 0x7;                          //Write to GP0KEY1
GP0CON1 &=~ BIT1;                       //Select SPI functionality for P0.0 to P0.3
GP0KEY2 = 0x13;                         //Write to GP0KEY2
```

GENERAL-PURPOSE I/O

The ADuC7060/ADuC7061 features up to 16 general-purpose bidirectional input/output (GPIO) pins. In general, many of the GPIO pins have multiple functions that are configurable by user code. By default, the GPIO pins are configured in GPIO mode. All GPIO pins have an internal pull-up resistor with a drive capability of 1.6 mA.

All I/O pins are 3.3 V tolerant, meaning that the GPIOs support an input voltage of 3.3 V.

When the ADuC7060/ADuC7061 enters power-saving mode, the GPIO pins retain their state.

The GPIO pins are grouped into three port buses.

Table 110 lists all the GPIO pins and their alternative functions. A GPIO pin alternative function can be selected by writing to the correct bits of the GPxCON register.

Table 110. GPIO Multifunction Pin Descriptions

	Configuration via GPxCON Including GP0CON0		
Port	**Pin Mnemonic**	**00**	**01**
0	P0.0/$\overline{SS}$	GPIO	$\overline{SS}$ (SPI slave select).
	P0.1/SCLK/SCL	GPIO	SCLK/SCL (serial clock/SPI clock).
	P0.2/MISO	GPIO	MISO (SPI—master in/slave out).
	P0.3/MOSI/SDA	GPIO	MOSI (SPI—master out/slave in).
	P0.4/IRQ0/PWM1	GPIO/IRQ0	PWM1 (PWM Output 1).
	P0.5/CTS	GPIO	CTS. UART clear to send pin.
	P0.6/RTS	GPIO	RTS. UART request to send pin.
1	P1.0/IRQ1/SIN/T0	GPIO/IRQ1	SIN (serial input).
	P1.1/SOUT	GPIO	SOUT (serial output).
	P1.2/SYNC	GPIO	PWM sync (PWM sync input pin).
	P1.3/TRIP	GPIO	PWM trip (PWM trip input pin).
	P1.4/PWM2	GPIO	PWM2 (PWM Output 2).
	P1.5/PWM3	GPIO	PWM3 (PWM Output 3).
	P1.6/PWM4	GPIO	PWM4 (PWM Output 4).
2	P2.0/IRQ2/PWM0/EXTCLK	GPIO/IRQ2/EXTCLK	PWM0 (PWM Output 0).
	P2.1/IRQ3/PWM5	GPIO/IRQ3	PWM5 (PWM Output 5).

GPxCON REGISTERS

GPxCON are the Port x (where x is 0, 1, or 2) control registers, which select the function of each pin of Port x as described in Table 112.

Table 111. GPxCON Registers

Name	Address	Default Value	Access
GP0CON0	0xFFFF0D00	0x00000000	R/W
GP1CON	0xFFFF0D04	0x00000000	R/W
GP2CON	0xFFFF0D08	0x00000000	R/W

Table 120. GPxPAR MMR Bit Designations

Bit	Name	Description
31:15		Reserved.
23:16	GPL[7:0]	General I/O port pin functionality lock registers. GPL[7:0] = 0, normal operation. GPL[7:0] = 1, for each GPIO pin, if this bit is set, writing to the corresponding bit in GPxCON or GPxDAT register bit has no effect.
15:8	GPDS[7:0]	Drive strength configuration. This bit is configurable. GPDS[x] = 0, maximum source current is 2 mA. GPDS[x] = 1, maximum source current is 4 mA.
7:0	GPPD[7:0]	Pull-Up Disable Port x[7:0]. GPPD[x] = 0, pull-up resistor is active. GPPD[x] = 1, pull-up resistor is disabled.

GP0CON1 Control Registers

The GP0CON1 write values are as follows: GP0KEY1 = 0x7, GP0CON1 = user value, and GP0KEY2 = 0x13.

Name: GP0CON1

Address: 0xFFFF0468

Default value: 0x00

Access: Read and write

Function: This register controls the P0.0, P0.1, P0.2, and P0.3 functionality of the multifunction GPIO pins.

Table 121. GP0CON1 Write Sequence

Name	Value
GP0KEY1	0x7
GP0CON1	User value
GP0KEY2	0x13

Table 122. GP0CON1 MMR Bit Designations

Bit	Name	Description
7:2	Reserved	These bits must always be set to 0.
1	SPII2CSEL	This bit configures the P0.0 to P0.3 functions in I^2C or SPI mode. Note that Bit 0 of GP0CON1 must be set to 0 for this bit to work. To select the P0.0, P0.1, P0.2, and P0.3 functions in SPI mode, clear this bit to 0. To select the P0.0, P0.1, P0.2, and P0.3 functions in I^2C mode, set this bit to 1. This bit is cleared by default.
0	ADCSEL	This bit configures the P0.0 to P0.3 functions as GPIO pins or as ADC input pins. To enable P0.0, P0.1, P0.2 and P0.3 functions as ADC inputs, set this bit to 1. To enable P0.0, P0.1, P0.2, and P0.3 functions as digital I/O, clear this bit to 0. This bit is cleared by default.

Name GP0KEY1

Address: 0xFFFF0464

Default value: 0xXXXX

Access: Write only

Function: When writing to GP0CON1, the value of 0x07 must be written to this register in the instruction immediately before writing to GP0CON1.

Name: GP0KEY2

Address: 0xFFFF046C

Default value: 0xXXXX

Access: Write only

Function: When writing to GP0CON1, the value of 0x13 must be written to this register in the instruction immediately after writing to GP0CON1.