**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 | 41.78MHz |
| Connectivity | I²C, SPI, UART/USART |
| Peripherals | PLA, POR, PWM, PSM, Temp Sensor, WDT |
| Number of I/O | 28 |
| Program Memory Size | 126KB (63K x 16) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 2K x 32 |
| Voltage - Supply (Vcc/Vdd) | 3V ~ 3.6V |
| Data Converters | A/D 10x12b; D/A 1x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 125°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 64-LQFP |
| Supplier Device Package | 64-LQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/analog-devices/aduc7128bstz126-rl |

| Parameter | Min | Typ | Max | Unit | Test Conditions/Comments |
|---|---|---|---|---|---|
| Relative Accuracy | | ±2 | | LSB | |
| Differential Nonlinearity, +VE | | 0.35 | | LSB | |
| Differential Nonlinearity, −VE | | −0.15 | | LSB | |
| Offset Error | | | −190 | mV | |
| Gain Error | | | +150 | mV | |
| Voltage Output Settling Time to 0.1% | | | 5 | µs | |
| Line Driver Output | | | | | As measured into a range of specified loads (see Figure 2) at LD1TX and LD2TX, unless otherwise noted |
| Total Harmonic Distortion | | −52 | | dB | PLM operating at 691.2 kHz |
| Output Voltage Swing | | ±1.768 | | V rms | |
| COMMON MODE | | | | | |
| AC Mode | | 1.65 | | V | Each output has a common mode of 0.5 V × $AV_{DD}$ and swings 0.5 V × $V_{REF}$ above and below this; $V_{REF}$ is the internal 2.5 V reference |
| DC Mode | | 1.5 | | V | Each output has a common mode of 0.5 V × $V_{REF}$ and swings 0.6 V × $V_{REF}$ above and below this; $V_{REF}$ is the internal 2.5 V reference |
| DIFFERENTIAL INPUT IMPEDANCE | 11 | 13 | | kΩ | Line driver buffer disabled |
| Leakage Current LD1TX, LD2TX | | | 7 | µA | Line driver buffer disabled |
| Short-Circuit Current | | ±50 | | mA | No protection diodes, max allowable current |
| Line Driver Tx Power-Up Time | | | 20 | µs | |
| 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[3, 5] | 2 | | 15 | mV | Hysteresis can be turned on or off via the CMPHYST bit in the CMPCON register |
| Response Time | | 1 | | µs | Response time can be modified via the CMPRES bits in the CMPCON register |
| TEMPERATURE SENSOR | | | | | |
| Voltage Output at 25°C | | 780 | | mV | |
| Voltage Temperature Coefficient | | −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 |
| GLITCH IMMUNITY ON $\overline{RST}$ PIN[3] | | 50 | | µs | |
| WATCHDOG TIMER (WDT) | | | | | |
| Timeout Period | 0 | | | ms | |
| | | | 512 | sec | |
| FLASH/EE MEMORY[7, 8] | | | | | |
| Endurance | 10,000 | | | Cycles | |
| Data Retention | 20 | | | Years | $T_J$ = 85°C |
| DIGITAL INPUTS | | | | | All digital inputs, including XCLKI and XCLKO |
| Logic 1 Input Current (Leakage Current) | | ±0.2 | ±1 | µA | $V_{INH} = V_{DD}$ or $V_{INH}$ = 5 V |
| Logic 0 Input Current (Leakage Current) | | −40 | −65 | µA | $V_{INL}$ = 0 V, except TDI |
| | | −80 | +125 | µA | $V_{INL}$ = 0 V, TDI Only |
| Input Capacitance | | 15 | | pF | |

## TIMING SPECIFICATIONS

**Table 2. External Memory Write Cycle**

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

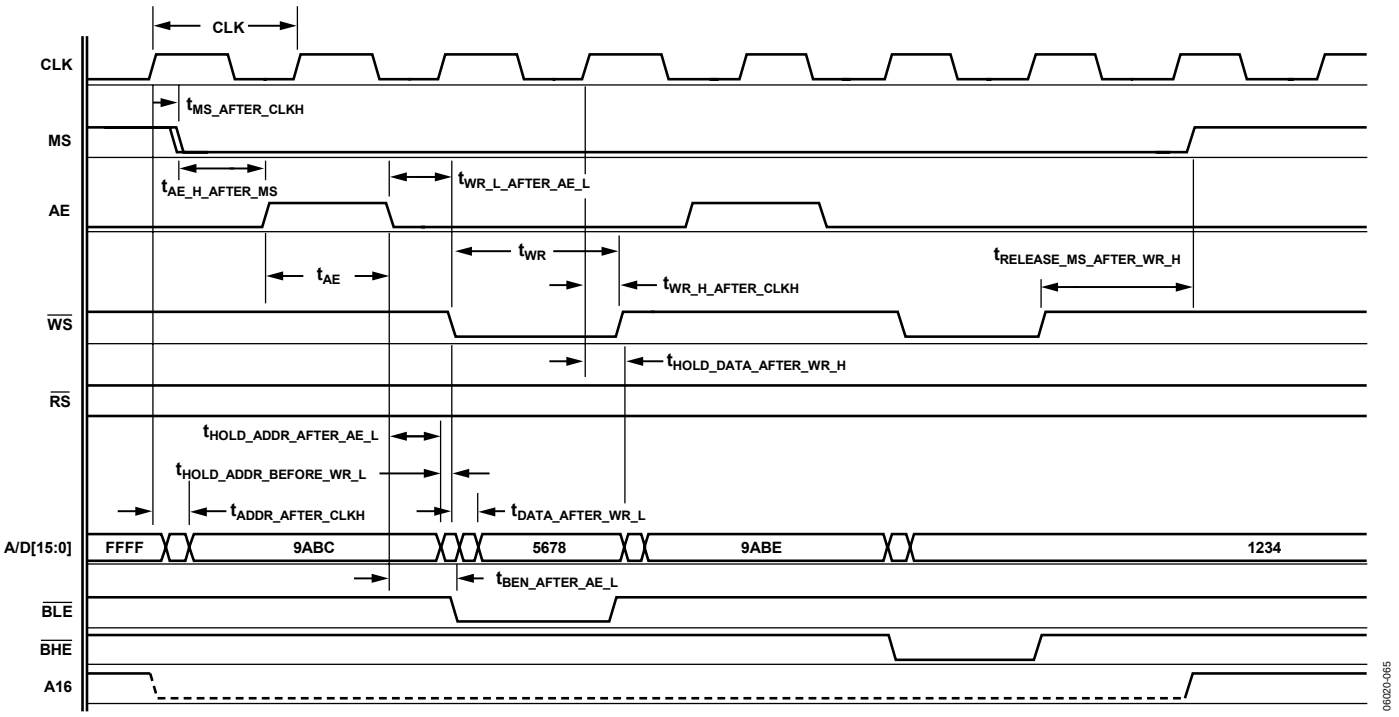



*Figure 3. External Memory Write Cycle*

### *SPI Timing Specifications*

**Table 5. SPI Master Mode Timing (PHASE Mode = 1)**

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

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

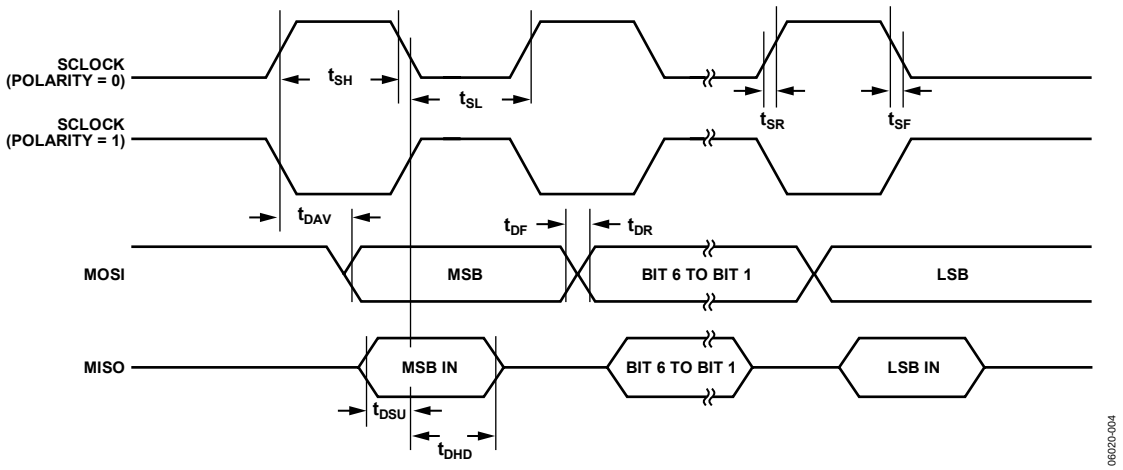



*Figure 6. SPI Master Mode Timing (PHASE Mode = 1)*

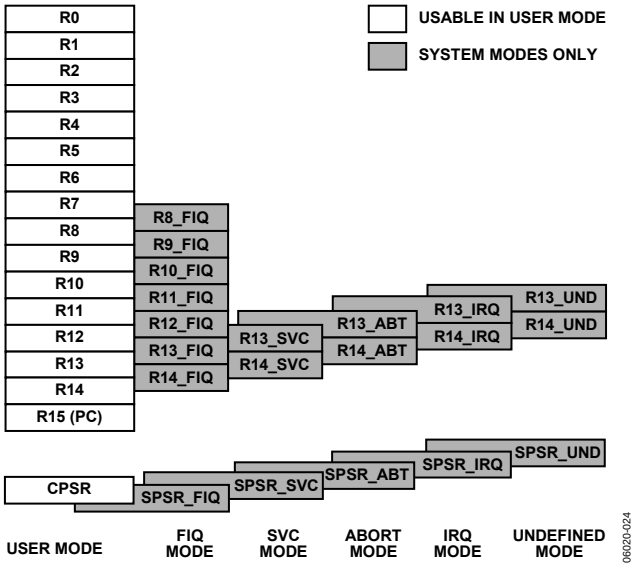



*Figure 28. Register Organization*

## INTERRUPT LATENCY

The worst case latency for an FIQ consists of the following:

- The longest time the request can take to pass through the synchronizer
- The time for the longest instruction to complete (the longest instruction is an LDM) that loads all the registers, including the PC
- The time for the data abort entry
- The time for FIQ entry

At the end of this time, the ARM7TDMI executes the instruction at Address 0x1C (FIQ interrupt vector address). The maximum total time is 50 processor cycles, which is just under 1.2 μs in a system using a continuous 41.78 MHz processor clock.

The maximum IRQ latency calculation is similar, but it must allow for the fact that FIQ has higher priority and could delay entry into the IRQ handling routine for an arbitrary length of time. This time can be reduced to 42 cycles if the LDM command is not used; some compilers have an option to compile without using this command. Another option is to run the part in Thumb mode, where the time is reduced to 22 cycles.

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

Note that the ARM7TDMI always runs in ARM (32-bit) mode when in privileged modes, that is, when executing interrupt service routines.

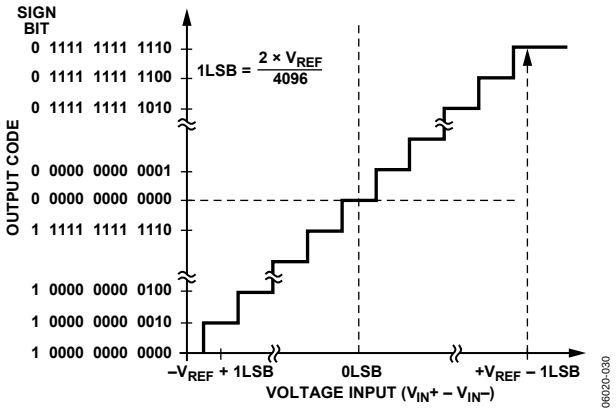



*Figure 34. ADC Transfer Function in Differential Mode*

## TYPICAL OPERATION

Once configured via the ADC control and channel selection registers, the ADC converts the analog input and provides an 11-bit result in the ADC data register.

The top four bits are the sign bits, and the 12-bit result is placed from Bit 16 to Bit 27, as shown in Figure 35. For fully differential mode, the result is ±11 bits. Again, it should be noted that in fully differential mode, the result is represented in twos complement format shifted one bit to the right, and in pseudo differential and single-ended mode, the result is represented in straight binary format.

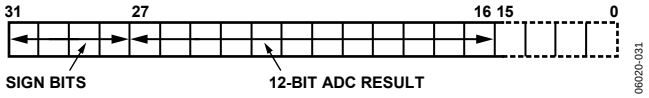



*Figure 35. ADC Result Format*

### *Current Consumption*

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

### *Timing*

Figure 36 gives details of the ADC timing. Users control the ADC clock speed and the number of acquisition clock in the ADCCON MMR. By default, the acquisition time is eight clocks and the clock divider is two. The number of extra clocks (such as bit trial or write) is set to 19, giving a sampling rate of 774 kSPS. For conversion on the temperature sensor, the ADC acquisition time is automatically set to 16 clocks and the ADC clock divider is set to 32. When using multiple channels, including the temperature sensor, the timing settings revert back to the user-defined settings after reading the temperature sensor channel.

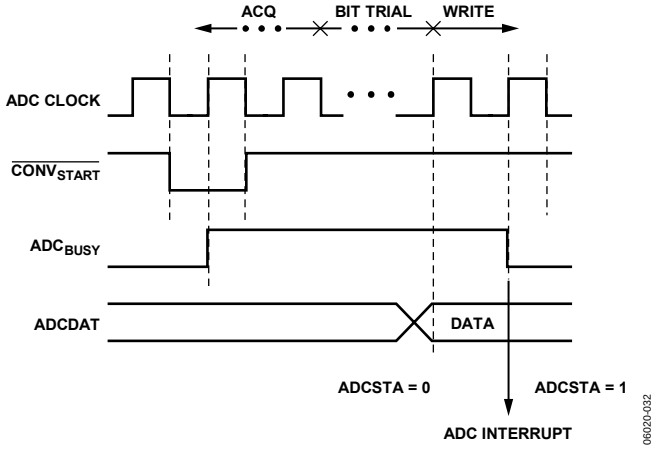



*Figure 36. ADC Timing*

### *ADC MMRs Interface*

The ADC is controlled and configured via a number of MMRs (see Table 32) that are described in detail in the following pages.

**Table 32. ADC MMRs**

| Name | Description |
|---|---|
| ADCCON | ADC Control Register. Allows the programmer to enable the ADC peripheral, to select the mode of operation of the ADC (either single-ended, pseudo differential, or fully differential mode), and to select the conversion type (see Table 33). |
| ADCCP | ADC Positive Channel Selection Register. |
| ADCCN | ADC Negative Channel Selection Register. |
| ADCSTA | ADC Status Register. Indicates when an ADC conversion result is ready. The ADCSTA register contains only one bit, ADCREADY (Bit 0), representing the status of the ADC. This bit is set at the end of an ADC conversion generating an ADC interrupt. It is cleared automatically by reading the ADCDAT MMR. When the ADC is performing a conversion, the status of the ADC can be read externally via the $ADC_{Busy}$ pin. This pin is high during a conversion. When the conversion is finished, $ADC_{Busy}$ goes back low. This information can be available on P0.5 (see the General-Purpose I/O section) if enabled in the GP0CON register. |
| ADCDAT | ADC Data Result Register. Holds the 12-bit ADC result, as shown in Table 35. |
| ADCRST | ADC Reset Register. Resets all the ADC registers to their default values. |

# OTHER ANALOG PERIPHERALS

## DAC

The ADuC7128/ADuC7129 feature a 10-bit current DAC that can be used to generate user-defined waveforms or sine waves generated by the DDS. The DAC consists of a 10-bit IDAC followed by a current-to-voltage conversion.

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

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

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

**Table 49. DACCON MMR Bit Designations**

| Bit | Value | Description |
|---|---|---|
| 10:9 | | Reserved. These bits should be written to 0 by the user. |
| 8 | | Reserved. This bit should be written to 0 by the user. |
| 7 | | Reserved. This bit should be written to 0 by the user. |
| 6 | | Reserved. This bit should be written to 0 by the user. |
| 5 | | Output Enable. This bit operates in all modes. In Line Driver mode, this bit should be set.<br>Set by user to enable the line driver output.<br>Cleared by user to disable the line driver output. In this mode the line driver output is high impedance. |
| 4 | | Single-Ended or Differential Output Control.<br>Set by user to operate in differential mode, the output is the differential voltage between LD1TX and LD2TX. The voltage output range is $V_{REF}/2 \pm V_{REF}/2$.<br>Cleared by user to reference the LD1TX output to AGND. The voltage output range is $AV_{DD}/2 \pm V_{REF}/2$. |
| 3 | | Reserved. This bit should be set to 0 by the user. |
| 2:1 | | Operation Mode Control. This bit selects the mode of operation of the DAC. |
| | 00 | Power-Down. |
| | 01 | Reserved. |
| | 10 | Reserved. |
| | 11 | DDS and DAC Mode. Selected by DACEN. |
| 0 | | DAC Update Rate Control. This bit has no effect when in DDS mode.<br>Set by user to update the DAC on the negative edge of Timer1. This allows the user to use any one of the core CLK, OSC CLK, baud CLK, or user CLK and divide these down by 1, 16, 256, or 32,768. A user can do waveform generation by writing to the DAC data register from RAM and updating the DAC at regular intervals via Timer1.<br>Cleared by user to update the DAC on the negative edge of HCLK. |

### *DDSFRQ Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| DDSFRQ | 0xFFFF0694 | 0x00000000 | R/W |

**Table 54. DDSFRQ MMR Bit Designations**

| Bit | Description |
|---|---|
| 31:0 | Frequency select word (FSW) |

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

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

### *DDSPHS Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| DDSPHS | 0xFFFF0698 | 0x00000000 | R/W |

**Table 55. DDSPHS MMR Bit Designations**

| Bit | Description |
|---|---|
| 31:12 | Reserved |
| 11:0 | Phase |

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

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

## POWER SUPPLY MONITOR

The power supply monitor on the ADuC7128/ADuC7129 indicates when the $IOV_{DD}$ supply pin drops below one of two supply trip points. The monitor function is controlled via the PSMCON register (see Table 56). If enabled in the IRQEN or FIQEN register, the monitor interrupts the core using the PSMI bit in the PSMCON MMR. This bit is cleared immediately once CMP goes high. Note that if the interrupt generated is exited before CMP goes high ($IOV_{DD}$ is above the trip point), no further interrupts are generated until CMP returns high. The user should ensure that code execution remains within the ISR until CMP returns high.

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

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

## COMPARATOR

The ADuC7128/ADuC7129 integrate an uncommitted voltage comparator. The positive input is multiplexed with ADC2, and the negative input has two options: ADC3 or the internal reference. The output of the comparator can be configured to generate a system interrupt, can be routed directly to the programmable logic array, can start an ADC conversion, or can be on an external pin, $CMP_{OUT}$.

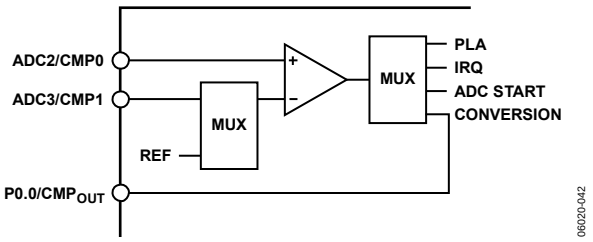



*Figure 46. Comparator*

### *Hysteresis*

Figure 47 shows how the input offset voltage and hysteresis terms are defined. Input offset voltage ($V_{OS}$) is the difference between the center of the hysteresis range and the ground level. This can either be positive or negative. The hysteresis voltage ($V_H$) is ½ the width of the hysteresis range.

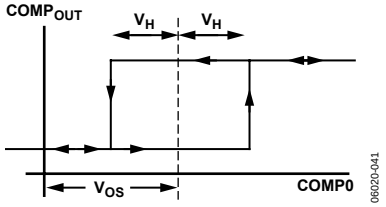



*Figure 47. Comparator Hysteresis Transfer Function*

**Table 56. PSMCON MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 3 | CMP | Comparator Bit. This is a read-only bit that directly reflects the state of the comparator.<br>Read 1 indicates the $IOV_{DD}$ supply is above its selected trip point or the PSM is in power-down mode.<br>Read 0 indicates the $IOV_{DD}$ supply is below its selected trip point. This bit should be set before leaving the interrupt service routine. |
| 2 | TP | Trip Point Selection Bit.<br>0 = 2.79 V<br>1 = 3.07 V |
| 1 | PSMEN | Power Supply Monitor Enable Bit.<br>Set to 1 by the user to enable the power supply monitor circuit.<br>Cleared to 0 by the user to disable the power supply monitor circuit. |
| 0 | PSMI | Power Supply Monitor Interrupt Bit. This bit is set high by the MicroConverter if CMP is low, indicating low I/O supply. The PSMI bit can be used to interrupt the processor. Once CMP returns high, the PSMI bit can be cleared by writing a 1 to this location. A write of 0 has no effect. There is no timeout delay. PSMI can be cleared immediately once CMP goes high. |

**Table 68. QENCON MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 15:11 | RSVD | Reserved. |
| 10 | FILTEN | Set to 1 by the user to enable filtering on the S1 pin.<br>Cleared by user to disable filtering on the S1 pin. |
| 9 | RSVD | Reserved. This bit should be set to 0 by the user. |
| 8 | S2INV | Set to 1 by the user to invert the S2 input.<br>Cleared by user to use the S2 input as normal.<br>If the DIRCON bit is set, then S2INV controls the direction of the counter.<br>In this case, set to 1 by the user to operate the counter in increment mode.<br>Cleared by user to operate the counter in decrement mode. |
| 7 | S1INV | Set to 1 by the user to invert the S1 input.<br>Cleared by user to use the S1 input as normal. |
| 6 | DIRCON | Direction Control.<br>Set to 1 by the user to enable S1 as the input to the counter clock. The direction of the counter is controlled via the S2INV bit.<br>Cleared by user to operate in normal mode. |
| 5 | S1IRQEN | Set to 1 by the user to generate an IRQ when a low-to-high transition is detected on S1.<br>Cleared by the user to disable the interrupt. |
| 4 | RSVD | This bit should be set to 0 by the user. |
| 3 | UIRQEN | Underflow IRQ Enable.<br>Set to 1 by the user to generate an interrupt if QENVAL underflows.<br>Cleared by the user to disable the interrupt. |
| 2 | OIREQEN | Overflow IRQ Enable.<br>Set to 1 by the user to generate an interrupt if QENVAL overflows.<br>Cleared by user to disable the interrupt. |
| 1 | RSVD | This bit should be set to 0 by the user. |
| 0 | ENQEN | Quadrature Encoder Enable.<br>Set to 1 by the user to enable the quadrature encoder.<br>Cleared by user to disable the quadrature encoder. |

**Table 69. QENSTA MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 7:5 | RSVD | Reserved. |
| 4 | S1EDGE | S1 Rising Edge.<br>This bit is set automatically on a rising edge of S1.<br>Cleared by reading QENSTA. |
| 3 | RSVD | Reserved. |
| 2 | UNDER | Underflow Flag.<br>This bit is set automatically if an underflow occurs.<br>Cleared by reading QENSTA. |
| 1 | OVER | This bit is set automatically if an overflow has occurred.<br>Cleared by reading QENSTA. |
| 0 | DIR | Direction of the Counter.<br>Set to 1 by hardware to indicate that the counter is incrementing.<br>Set to 0 by hardware to indicate that the counter is decrementing. |

### *QENDAT Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| QENDAT | 0xFFFF0F08 | 0Xffff | R/W |

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

### *QENVAL Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| QENVAL | 0xFFFF0F0C | 0x0000 | R/W |

The QENVAL register contains the current value of the quadrature encoder counter.

***GPxDAT Register***

| Name | Address | Default Value | Access |
|---|---|---|---|
| GP0DAT | 0xFFFF0D20 | 0x000000XX | R/W |
| GP1DAT | 0xFFFF0D30 | 0x000000XX | R/W |
| GP2DAT | 0xFFFF0D40 | 0x000000XX | R/W |
| GP3DAT | 0xFFFF0D50 | 0x000000XX | R/W |
| GP4DAT | 0xFFFF0D60 | 0x000000XX | R/W |

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

**Table 73. GPxDAT MMR Bit Designations**

| Bit | Description |
|---|---|
| 31:24 | Direction of the Data.<br>Set to 1 by user to configure the GPIO pins as outputs.<br>Cleared to 0 by user to configure the GPIO pins as inputs. |
| 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). |

***GPxSET Register***

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

GPxSET is a data set Port x register.

**Table 74. GPxSET MMR Bit Designations**

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

***GPxCLR Register***

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

GPxCLR is a data clear Port x register.

**Table 75. GPxCLR MMR Bit Designations**

| Bit | Description |
|---|---|
| 31:24 | Reserved. |
| 23:16 | Data Port x Clear Bit.<br>Set to 1 by user to clear bit on Port x; also clears the corresponding bit in the GPxDAT MMR.<br>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 (two $I^2C$s, an SPI, and two UARTs) and the programmable logic array (PLA) to a set of 10 GPIO pins. Each pin must be configured to its specific I/O function as described in Table 76.

**Table 76. SPM Configuration**

| Pin | GPIO (00) | UART (01) | UART/I2C/SPI (10) | PLA (11) |
|---|---|---|---|---|
| SPM0 | P1.0 | SIN0 | I2C0SCL | PLAI[0] |
| SPM1 | P1.1 | SOUT0 | I2C0SDA | PLAI[1] |
| SPM2 | P1.2 | RTS0 | I2C1SCL | PLAI[2] |
| SPM3 | P1.3 | CTS0 | I2C1SDA | PLAI[3] |
| SPM4 | P1.4 | RI0 | SPICLK | PLAI[4] |
| SPM5 | P1.5 | DCD0 | SPIMISO | PLAI[5] |
| SPM6 | P1.6 | DSR0 | SPIMOSI | PLAI[6] |
| SPM7 | P1.7 | DTR0 | SPICSL | PLAO[0] |
| SPM8 | P0.7 | ECLK | SIN0 | PLAO[4] |
| SPM9 | P2.0[1] | PWMSYNC | SOUT0 | PLAO[5] |
| SPM10 | P2.2[1] | RTS1 | RS | PLAO[7] |
| SPM11 | P2.3[1] | CTS1 | AE | |
| SPM12 | P2.4[1] | RI1 | MS0 | |
| SPM13 | P2.5[1] | DCD1 | MS1 | |
| SPM14 | P2.6[1] | DSR1 | MS2 | |
| SPM15 | P2.7[1] | DTR1 | MS3 | |
| SPM16 | P4.6 | SIN1 | AD14 | PLAO[14] |
| SPM17 | P4.7 | SOUT1 | AD15 | PLAO[15] |

[1] Available only on the 80-lead ADuC7129.

Table 76 details the mode for each of the SPMUX GPIO pins. This configuration has to be performed via the GP0CON, GP1CON and GP2CON MMRs. By default these pins are configured as GPIOs.

## UART SERIAL INTERFACE

The ADuC7128/ADuC7129 contain two identical UART blocks. Although only UART0 is described here, UART1 functions in exactly the same way.

The UART peripheral is a full-duplex universal asynchronous receiver/transmitter, fully compatible with the 16450 serial port standard.

The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a modem, and parallel-to-serial conversion on data characters received from the CPU. The UART includes a fractional divider for baud rate generation and has a network-addressable mode. The UART function is made available on 10 pins of the ADuC7128/ADuC7129 (see Table 77).

**Table 82. COMxIEN0 MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 7:4 | RSVD | Reserved. |
| 3 | EDSSI | Modem Status Interrupt Enable Bit.<br>Set by user to enable generation of an interrupt if any of COMxSTA1[3:0] are set.<br>Cleared by user. |
| 2 | ELSI | RX Status Interrupt Enable Bit.<br>Set by user to enable generation of an interrupt if any of COMxSTA0[3:1] are set.<br>Cleared by user. |
| 1 | ETBEI | Enable Transmit Buffer Empty Interrupt.<br>Set by user to enable interrupt when buffer is empty during a transmission.<br>Cleared by user. |
| 0 | ERBFI | Enable Receive Buffer Full Interrupt.<br>Set by user to enable interrupt when buffer is full during a reception.<br>Cleared by user. |

**Table 83. COMxIID0 MMR Bit Designations**

| Bit 2:1 Status Bits | Bit 0 NINT | Priority | Definition | Clearing Operation |
|---|---|---|---|---|
| 00 | 1 | | No Interrupt. | |
| 11 | 0 | 1 | Receive Line Status Interrupt. | Read COMxSTA0. |
| 10 | 0 | 2 | Receive Buffer Full Interrupt. | Read COMxRX. |
| 01 | 0 | 3 | Transmit Buffer Empty Interrupt. | Write data to COMxTX or read COMxIID0. |
| 00 | 0 | 4 | Modem Status Interrupt. | Read COMxSTA1. |

**Table 84. COMxCON1 MMR Bit Designations**

| Bit | Name | Description |
|---|---|---|
| 7:5 | RSVD | Reserved. |
| 4 | LOOPBACK | Loop Back.<br>Set by user to enable loop-back mode. In loop-back mode, the SOUT is forced high. In addition, the modem signals are directly connected to the status inputs (RTS to CTS, DTR to DSR, OUT1 to RI, and OUT2 to DCD). |
| 3 | | Reserved. |
| 2 | | Reserved. |
| 1 | RTS | Request to Send.<br>Set by user to force the RTS output to 0.<br>Cleared by user to force the RTS output to 1. |
| 0 | DTR | Data Terminal Ready.<br>Set by user to force the DTR output to 0.<br>Cleared by user to force the DTR output to 1. |

***SPIRX Register***

| Name | Address | Default Value | Access |
|---|---|---|---|
| SPIRX | 0xFFFF0A04 | 0x00 | R |

SPIRX is an 8-bit read-only receive register.

***SPITX Register***

| Name | Address | Default Value | Access |
|---|---|---|---|
| SPITX | 0xFFFF0A08 | 0x00 | W |

SPITX is an 8-bit write-only transmit register.

***SPIDIV Register***

| Name | Address | Default Value | Access |
|---|---|---|---|
| SPIDIV | 0xFFFF0A0C | 0x1B | R/W |

SPIDIV is an 8-bit serial clock divider register.

***SPICON Register***

| Name | Address | Default Value | Access |
|---|---|---|---|
| SPICON | 0xFFFF0A10 | 0x0000 | R/W |

SPICON is a 16-bit control register.

**Table 91. SPICON MMR Bit Designations**

| Bit | Description |
|---|---|
| 15:13 | Reserved. |
| 12 | Continuous Transfer Enable.<br>Set by user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the TX register. $\overline{CS}$ is asserted and remains asserted for the duration of each 8-bit serial transfer until TX is empty.<br>Cleared by user to disable continuous transfer. Each transfer consists of a single 8-bit serial transfer. If valid data exists in the SPITX register, then a new transfer is initiated after a stall period. |
| 11 | Loopback Enable.<br>Set by user to connect MISO to MOSI and test software.<br>Cleared by user to be in normal mode. |
| 10 | Slave Output Enable.<br>Set by user to enable the slave output.<br>Cleared by user to disable slave output. |
| 9 | Slave Select Input Enable.<br>Set by user in master mode to enable the output. |
| 8 | SPIRX Overflow Overwrite Enable.<br>Set by user, the valid data in the RX register is overwritten by the new serial byte received.<br>Cleared by user, the new serial byte received is discarded. |
| 7 | SPITX Underflow Mode.<br>Set by user to transmit 0.<br>Cleared by user to transmit the previous data. |
| 6 | Transfer and Interrupt Mode (Master Mode).<br>Set by user to initiate transfer with a write to the SPITX register. Interrupt occurs when TX is empty.<br>Cleared by user to initiate transfer with a read of the SPIRX register. Interrupt occurs when RX is full. |
| 5 | LSB First Transfer Enable Bit.<br>Set by user, the LSB is transmitted first.<br>Cleared by user, the MSB is transmitted first. |
| 4 | Reserved. Should be set to 0. |
| 3 | Serial Clock Polarity Mode Bit.<br>Set by user, the serial clock idles high.<br>Cleared by user, the serial clock idles low. |
| 2 | Serial Clock Phase Mode Bit.<br>Set by user, the serial clock pulses at the beginning of each serial bit transfer.<br>Cleared by user, the serial clock pulses at the end of each serial bit transfer. |
| 1 | Master Mode Enable Bit.<br>Set by user to enable master mode.<br>Cleared by user to enable slave mode. |
| 0 | SPI Enable Bit.<br>Set by user to enable the SPI.<br>Cleared to disable the SPI. |

# PROCESSOR REFERENCE PERIPHERALS

## INTERRUPT SYSTEM

There are 30 interrupt sources on the ADuC7128/ADuC7129 controlled by the interrupt controller. Most interrupts are generated from the on-chip peripherals, such as ADC and UART. Two additional interrupt sources are generated from external interrupt request pins, XIRQ0 and XIRQ1. The ARM7TDMI CPU core only recognizes interrupts as one of two types: a normal interrupt request (IRQ) or a fast interrupt request (FIQ). All the interrupts can be masked separately.

The control and configuration of the interrupt system are managed through nine interrupt-related registers, four dedicated to IRQ, four dedicated to FIQ, and an additional MMR that is used to select the programmed interrupt source. The bits in each IRQ and FIQ register represent the same interrupt source as described in Table 104.

**Table 104. IRQ/FIQ MMRs Bit Designations**

| Bit | Description |
|---|---|
| 0 | FIQ Source. |
| 1 | SWI. Not used in IRQEN/CLR and FIQEN/CLR. |
| 2 | Timer0. |
| 3 | Timer1. |
| 4 | Wake-Up Timer—Timer2. |
| 5 | Watchdog Timer—Timer3. |
| 6 | Timer4. |
| 7 | Flash Controller 0. |
| 8 | Flash Controller 1. |
| 9 | ADC. |
| 10 | Quadrature Encoder. |
| 11 | I2C0 Slave. |
| 12 | I2C1 Slave. |
| 13 | I2C0 Master. |
| 14 | I2C1 Master. |
| 15 | SPI Slave. |
| 16 | SPI Master. |
| 17 | UART0. |
| 18 | UART1. |
| 19 | External IRQ0. |
| 20 | Comparator. |
| 21 | PSM. |
| 22 | External IRQ1. |
| 23 | PLA IRQ0. |
| 24 | PLA IRQ1. |
| 25 | External IRQ2. |
| 26 | External IRQ3. |
| 27 | PWM Trip. |
| 28 | PLL Lock. |
| 29 | Reserved. |
| 30 | Reserved. |

### *IRQ*

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

The four 32-bit registers dedicated to IRQ are listed in Table 105.

**Table 105. IRQ Interface MMRs**

| Register | Description |
|---|---|
| IRQSIG | Reflects the status of the different IRQ sources. If a peripheral generates an IRQ signal, the corresponding bit in the IRQSIG is set; otherwise, it is cleared. The IRQSIG bits are cleared when the interrupt in the particular peripheral is cleared. All IRQ sources can be masked in the IRQEN MMR. IRQSIG is read only. |
| IRQEN | Provides the value of the current enable mask. When set to 1, the source request is enabled to create an IRQ exception. When set to 0, the source request is disabled or masked but does not create an IRQ exception. To clear a bit in IRQEN, use the IRQCLR MMR. |
| IRQCLR | Write-only register allows clearing the IRQEN register to mask an interrupt source. Each bit set to 1 clears the corresponding bit in the IRQEN register without affecting the remaining bits. The pair of registers, IRQEN and IRQCLR, allows independent manipulation of the enable mask without requiring an automatic read-modify-write. |
| IRQSTA | Read-only register provides the current enabled IRQ source status. When set to 1, that source should generate an active IRQ request to the ARM7TDMI core. There is no priority encoder or interrupt vector generation. This function is implemented in software in a common interrupt handler routine. All 32 bits are logically OR'ed to create the IRQ signal to the ARM7TDMI core. |

### *FIQ*

The 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.

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

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

### *Programmed Interrupts*

As the programmed interrupts are nonmaskable, they are controlled by the SWICFG register that writes into both the IRQSTA and IRQSIG registers and/or FIQSTA and FIQSIG registers at the same time. The 32-bit register dedicated to software interrupt is SWICFG described in Table 106. This MMR allows the control of programmed source interrupt.

**Table 106. SWICFG MMR Bit Designations**

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

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

## TIMERS

The ADuC7128/ADuC7129 have five general purpose timers/counters.

- Timer0
- Timer1
- Timer2 or wake-up timer
- Timer3 or watchdog timer
- Timer4

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

In free-running mode, the counter decrements or increments from the maximum or minimum value until zero scale or full scale and starts again at the maximum or minimum value.

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

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

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

**Table 107. Event Selection Numbers**

| ES | Interrupt Number | Name |
|---|---|---|
| 00000 | 2 | RTOS Timer (Timer0) |
| 00001 | 3 | GP Timer0 (Timer1) |
| 00010 | 4 | Wake-Up Timer (Timer2) |
| 00011 | 5 | Watchdog Timer (Timer3) |
| 00100 | 6 | GP Timer1 (Timer4) |
| 00101 | 7 | Flash Control 0 |
| 00110 | 8 | Flash Control 1 |
| 00111 | 9 | ADC Channel |
| 01000 | 10 | Quadrature Encoder |
| 01001 | 11 | I2C Slave0 |
| 01010 | 12 | I2C Slave1 |
| 01011 | 13 | I2C Master0 |
| 01100 | 14 | I2C Master1 |
| 01101 | 15 | SPI Slave |
| 01110 | 16 | SPI Master |
| 01111 | 17 | UART0 |
| 10000 | 18 | UART1 |
| 10001 | 19 | External IRQ0 |

## TIMER0—LIFETIME TIMER

Timer0 is a general-purpose, 48-bit count up, or a 16-bit count up/down timer with a programmable prescaler. Timer0 is clocked from the core clock, with a prescaler of 1, 16, 256, or 32,768. This gives a minimum resolution of 22 ns when the core is operating at 41.78 MHz and with a prescaler of 1.

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

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

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

The Timer0 interface consists of six MMRs, shown in Table 108.

**Table 108. Timer0 Interface MMRs**

| Name | Description |
|---|---|
| T0LD | A 16-bit register that holds the 16-bit value loaded into the counter. Available only in 16-bit mode. |
| T0CAP | A 16-bit register that holds the 16-bit value captured by an enabled IRQ event. Available only in 16-bit mode. |
| T0VAL0/ T0VAL1 | TOVAL0 is a 16 bit register that holds the 16 least significant bits (LSBs).<br>T0VAL1 is a 32-bit register that holds the 32 most significant bits (MSBs).<br>T0VAL0 and T0VAL1 are read only. In 16-bit mode, 16-bit T0VAL0 is used. In 48-bit mode, both 16-bit T0VAL0 and 32-bit T0VAL1 are used. |
| T0ICLR | An 8-bit register. Writing any value to this register clears the interrupt. Available only in 16-bit mode. |
| T0CON | The configuration MMR (see Table 109). |

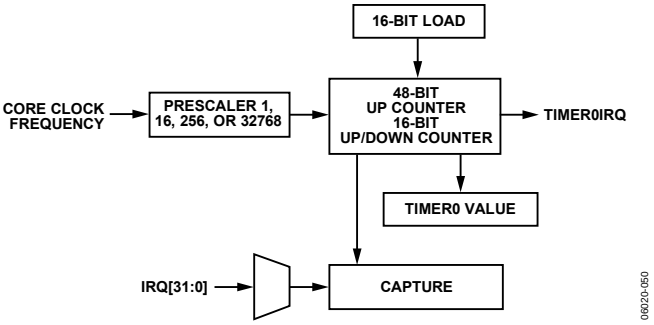



*Figure 55. Timer0 Block Diagram*

### *Timer0 Value Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T0VAL0 | 0xFFFF0304 | 0x00 | R |
| T0VAL1 | 0xFFFF0308 | 0x00 | R |

T0VAL0 and T0VAL1 are 16-bit and 32-bit registers that hold the 16 least significant bits and 32 most significant bits, respectively. T0VAL0 and T0VAL1 are read-only. In 16-bit mode, 16-bit T0VAL0 is used. In 48-bit mode, both 16-bit T0VAL0 and 32-bit T0VAL1 are used.

### *Timer0 Capture Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T0CAP | 0xFFFF0314 | 0x00 | R |

This is a 16-bit register that holds the 16-bit value captured by an enabled IRQ event; available only in 16-bit mode.

### *Timer0 Control Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T0CON | 0xFFFF030C | 0x00 | R/W |

The 17-bit MMR configures the mode of operation of Timer0.

**Table 109. T0CON MMR Bit Designations**

| Bit | Value | Description |
|---|---|---|
| 31:18 | | Reserved. |
| 17 | | Event Select Bit.<br>Set by user to enable time capture of an event.<br>Cleared by user to disable time capture of an event. |
| 16:12 | | Event Select Range, 0 to 31. The events are as described in the Timers section. |
| 11 | | Reserved. |
| 10:9 | | Reserved. |
| 8 | | Count Up. Available only in 16-bit mode.<br>Set by user for timer 0 to count up.<br>Cleared by user for timer 0 to count down (default). |
| 7 | | Timer0 Enable Bit.<br>Set by user to enable Timer0.<br>Cleared by user to disable Timer0 (default). |
| 6 | | Timer0 Mode.<br>Set by user to operate in periodic mode.<br>Cleared by user to operate in free-running mode (default). |
| 5 | | Reserved. |
| 4 | | Timer0 Mode of Operation. |
| | 0 | 16-bit operation (default). |
| | 1 | 48-bit operation. |
| 3:0 | | Prescaler. |
| | 0000 | Source clock/1 (default). |
| | 0100 | Source clock/16. |
| | 1000 | Source clock/256. |
| | 1111 | Source clock/32,768. |

### *Timer0 Load Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T0LD | 0xFFFF0300 | 0x00 | R/W |

T0LD is a 16-bit register that holds the 16-bit value that is loaded into the counter; available only in 16-bit mode.

### *Timer0 Clear Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T0ICLR | 0xFFFF0310 | 0x00 | W |

This 8-bit, write-only MMR is written (with any value) by user code to refresh (reload) Timer0.

## TIMER3—WATCHDOG TIMER

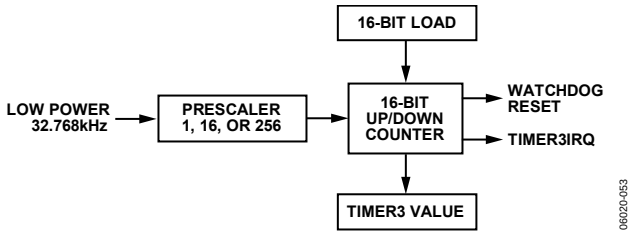



*Figure 58. Timer3 Block Diagram*

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

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

### *Normal Mode*

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

### *Watchdog Mode*

Watchdog mode is entered by setting T3CON[5]. Timer3 decrements from the timeout value present in the T3LD register to 0. The maximum timeout is 512 seconds, using the maximum prescalar/256 and full scale in T3LD.

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

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

Once watchdog mode is entered, T3LD and T3CON are write protected. These two registers cannot be modified until a power-on reset event resets the watchdog timer. After any other reset event, the watchdog timer continues to count. The watchdog timer should be configured in the initial lines of user code to avoid an infinite loop of watchdog resets.

Timer3 is automatically halted during JTAG debug access and only recommences counting once JTAG has relinquished control of the ARM7 core. By default, Timer3 continues to count during power-down. This can be disabled by setting Bit 0 in T3CON. It is recommended that the default value is used, that is, the watchdog timer continues to count during power-down.

### *Timer3 Interface*

The Timer3 interface consists of four MMRs, as shown in Table 114.

**Table 114. Timer3 Interface MMRs**

| Name | Description |
|---|---|
| T3CON | The configuration MMR (see Table 115). |
| T3LD | A 16-bit register (Bit 0 to Bit15). Holds 16-bit unsigned integers. |
| T3VAL | A 16-bit register (Bit 0 to Bit 15). Holds 16-bit unsigned integers. This register is read only. |
| T3ICLR | An 8-bit register. Writing any value to this register clears the Timer3 interrupt in normal mode or resets a new timeout period in watchdog mode. |

#### *Timer3 Load Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T3LD | 0xFFFF0360 | 0x03D7 | R/W |

This 16-bit MMR holds the Timer3 reload value.

#### *Timer3 Value Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T3VAL | 0xFFFF0364 | 0x03D7 | R |

This 16-bit, read-only MMR holds the current Timer3 count value.

#### *Timer3 Clear Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T3ICLR | 0xFFFF036C | 0x00 | W |

This 8-bit, write-only MMR is written (with any value) by user code to refresh (reload) Timer3 in watchdog mode to prevent a watchdog timer reset event.

#### *Timer3 Control Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T3CON | 0xFFFF0368 | 0x00 | R/W once only |

The 16-bit MMR configures the mode of operation of Timer3. as described in detail in Table 115.

**Table 115. T3CON MMR Bit Designations**

| Bit | Value | Description |
|---|---|---|
| 16:9 | | These bits are reserved and should be written as 0s by user code. |
| 8 | | Count Up/Down Enable.<br>Set by user code to configure Timer3 to count up.<br>Cleared by user code to configure Timer3 to count down. |
| 7 | | Timer3 Enable.<br>Set by user code to enable Timer3.<br>Cleared by user code to disable Timer3. |
| 6 | | Timer3 Operating Mode.<br>Set by user code to configure Timer3 to operate in periodic mode.<br>Cleared by user to configure Timer3 to operate in free-running mode. |
| 5 | | Watchdog Timer Mode Enable.<br>Set by user code to enable watchdog mode.<br>Cleared by user code to disable watchdog mode. |
| 4 | | Secure Clear Bit.<br>Set by user to use the secure clear option.<br>Cleared by user to disable the secure clear option by default. |
| 3:2 | | Timer3 Clock (32.768 kHz) Prescaler. |
| | 00 | Source Clock/1 (Default). |
| | 01 | Reserved. |
| | 10 | Reserved. |
| | 11 | Reserved. |
| 1 | | Watchdog Timer IRQ Enable.<br>Set by user code to produce an IRQ instead of a reset when the watchdog reaches 0.<br>Cleared by user code to disable the IRQ option. |
| 0 | | PD_OFF.<br>Set by user code to stop Timer3 when the peripherals are powered down via Bit 4 in the POWCON MMR.<br>Cleared by user code to enable Timer3 when the peripherals are powered down via Bit 4 in the POWCON MMR. |

### *Secure Clear Bit (Watchdog Mode Only)*

The secure clear bit is provided for a higher level of protection. When set, a specific sequential value must be written to T3ICLR to avoid a watchdog reset. The value is a sequence generated by the 8-bit linear feedback shift register (LFSR) polynomial equal to X8 + X6 + X5 + X + 1, as shown in Figure 59.

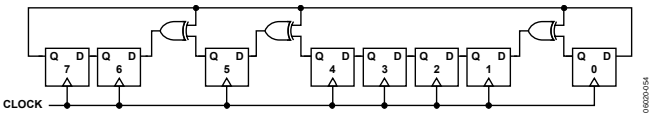



*Figure 59. 8-Bit LFSR*

The initial value or seed is written to T3ICLR before entering watchdog mode. After entering watchdog mode, a write to T3ICLR must match this expected value. If it matches, the LFSR is advanced to the next state when the counter reload happens. If it fails to match the expected state, reset is immediately generated, even if the count has not yet expired.

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

The following is an example of a sequence:

1. Enter initial seed, 0 xAA, in T3ICLR before starting Timer3 in watchdog mode.
2. Enter 0 xAA in T3ICLR; Timer3 is reloaded.
3. Enter 0x37 in T3ICLR; Timer3 is reloaded.
4. Enter 0x6E in T3ICLR; Timer3 is reloaded.
5. Enter 0x66. 0xDC was expected; the watchdog resets the chip.

## TIMER4—GENERAL-PURPOSE TIMER

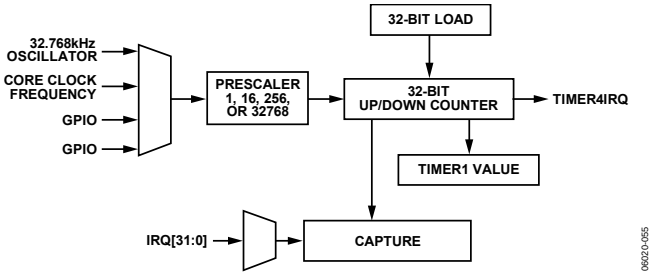



*Figure 60. Timer4 Block Diagram*

Timer4 is a 32-bit, general-purpose count down or count up timer with a programmable prescalar. The prescalar source can be the 32 kHz oscillator, the core clock, or 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 42 ns when operating at CD zero, the core is operating at 41.78 MHz, and with a prescalar of 1 (ignoring external GPIO).

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

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

The Timer4 interface consists of five MMRs.

**Table 116. Timer4 Interface MMRs**

| Name | Description |
|---|---|
| T4LD | A 32-bit register. Holds 32-bit unsigned integers. |
| T4VAL | A 32-bit register. Holds 32-bit unsigned integers. This register is read only. |
| T4CAP | A 32-bit register. Holds 32-bit unsigned integers. This register is read only. |
| T4ICLR | An 8-bit register. Writing any value to this register clears the Timer1 interrupt. |
| T4CON | The configuration MMR (see Table 117). |

Note that if the part is in a low power mode and Timer4 is clocked from the GPIO or oscillator source, Timer4 continues to operate.

Timer4 reloads the value from T4LD either when Timer 4 overflows, or immediately when T4ICLR is written.

### *Timer4 Load Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T4LD | 0xFFFF0380 | 0x00000 | R/W |

T4LD is a 32-bit register that holds the 32-bit value that is loaded into the counter.

### *Timer4 Clear Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T4ICLR | 0xFFFF038C | 0x00 | W |

This 8-bit, write only MMR is written (with any value) by user code to refresh (reload) Timer4.

### *Timer4 Value Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T4VAL | 0xFFFF0384 | 0x0000 | R |

T4VAL is a 32-bit register that holds the current value of Timer4.

### *Timer4 Capture Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T4CAP | 0xFFFF0390 | 0x00 | R |

This is a 32-bit register that holds the 32-bit value captured by an enabled IRQ event.

### *Timer4 Control Register*

| Name | Address | Default Value | Access |
|---|---|---|---|
| T4CON | 0xFFFF0388 | 0x0000 | R/W |

This 32-bit MMR configures the mode of operation of Timer4.

## POWER-ON RESET OPERATION

An internal power-on reset (POR) is implemented on the ADuC7128/ADuC7129. For $LV_{DD}$ below 2.45 V, the internal POR holds the ADuC7128/ADuC7129 in reset. As $LV_{DD}$ rises above 2.45 V, an internal timer times out for typically 64 ms before the part is released from reset. The user must ensure that the power supply, $IOV_{DD}$, has reached a stable 3.0 V minimum level by this time. On power-down, the internal POR holds the ADuC7128/ADuC7129 in reset until $LV_{DD}$ has dropped below 2.45 V. Figure 72 illustrates the operation of the internal POR in detail.

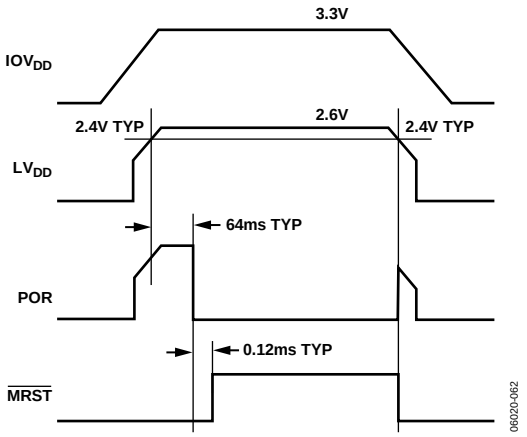



*Figure 72. Internal Power-On Reset Operation*

# OUTLINE DIMENSIONS

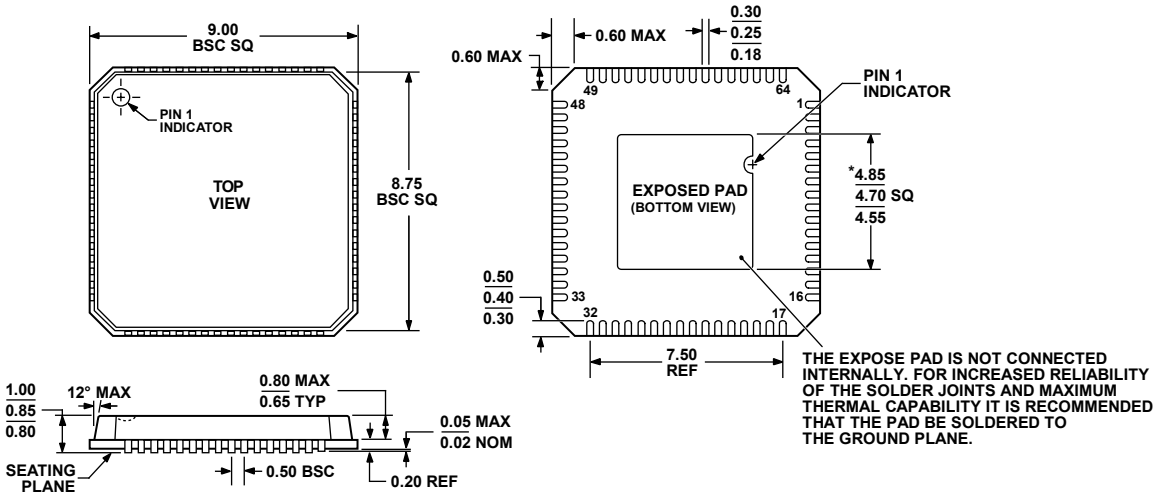



*Figure 73. 64-Lead Lead Frame Chip Scale Package [LFCSP_VQ]*
*9 mm × 9 mm Body, Very Thin Quad*
*(CP-64-1)*
*Dimensions shown in millimeters*

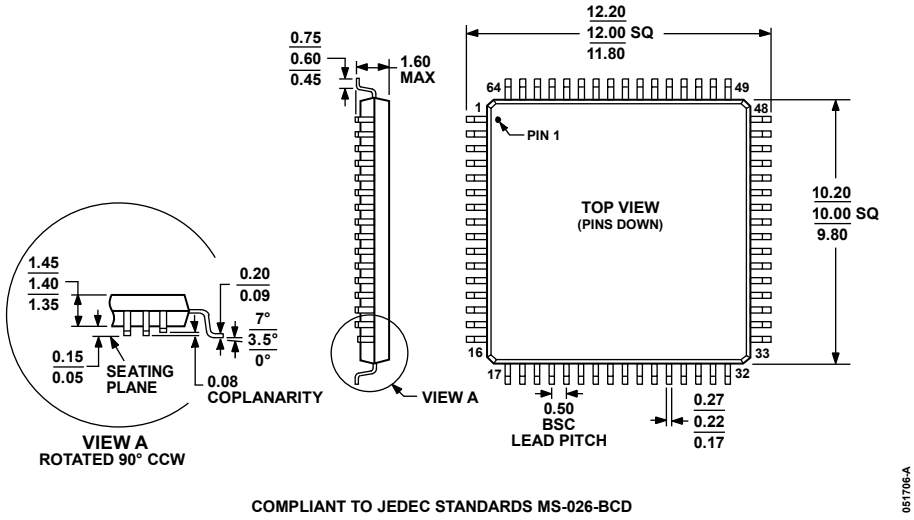
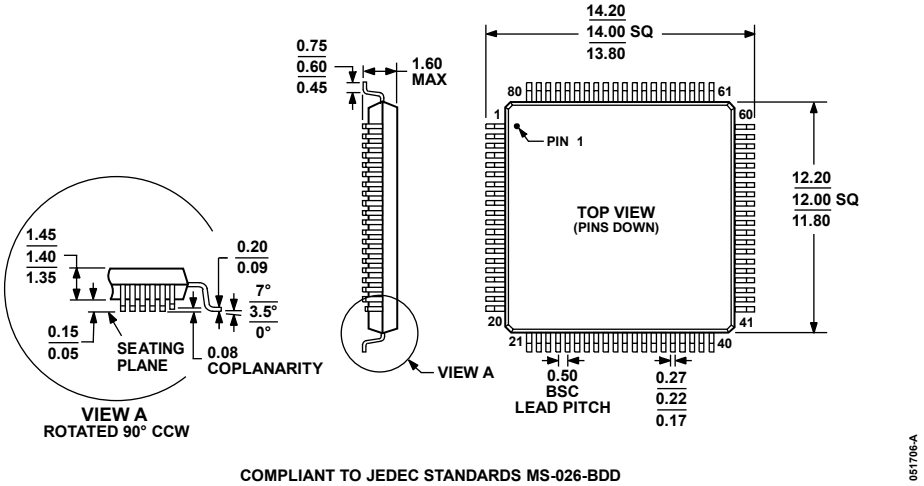



*Figure 74. 64-Lead Low Profile Quad Flat Package [LQFP]*
*(ST-64-2)*
*Dimensions shown in millimeters*




**COMPLIANT TO JEDEC STANDARDS MS-026-BDD**

*Figure 75. 80-Lead Low Profile Quad Flat Package [LQFP]*
*(ST-80-1)*
*Dimensions shown in millimeters*

## ORDERING GUIDE

| Model[1] | Temperature Range | Package Description | Package Option |
|---|---|---|---|
| ADUC7128BCPZ126[2] | −40°C to +125°C | 64-Lead Lead Frame Chip Scale Package [LFCSP_VQ] | CP-64-1 |
| ADUC7128BCPZ126-RL[2] | −40°C to +125°C | 64-Lead Lead Frame Chip Scale Package [LFCSP_VQ] | CP-64-1 |
| ADUC7128BSTZ126[2] | −40°C to +125°C | 64-Lead LQFP | ST-64-2 |
| ADUC7128BSTZ126-RL[2] | −40°C to +125°C | 64-Lead LQFP | ST-64-2 |
| ADUC7129BSTZ126[2] | −40°C to +125°C | 80-Lead LQFP | ST-80-1 |
| ADUC7129BSTZ126-RL[2] | −40°C to +125°C | 80-Lead LQFP | ST-80-1 |
| EVAL-ADUC7128QSPZ[2] | | Evaluation Board | |

[1] Reel quantities are 2,500 for the LFCSP and 1,000 for the LQFP.
[2] Z = RoHS Compliant Part.

Purchase of licensed I²C components of Analog Devices or one of its sublicensed Associated Companies conveys a license for the purchaser under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.


**D06020-0-4/07(0)**