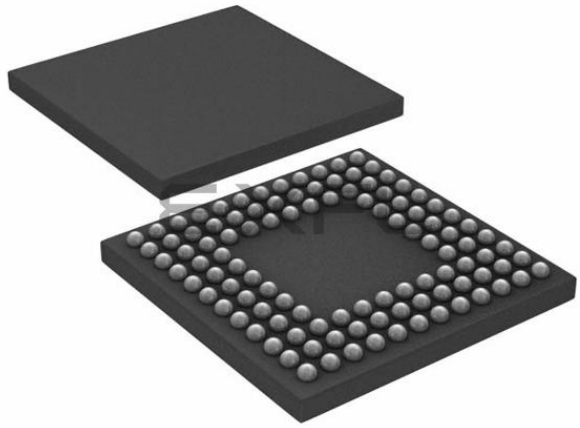

Welcome to E-XFL.COM

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	POR, PWM, WDT
Number of I/O	32
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 13x12b; D/A 12x12b
Oscillator Type	Internal
Operating Temperature	-10°C ~ 95°C (TA)
Mounting Type	Surface Mount
Package / Case	108-LFBGA, CSPBGA
Supplier Device Package	108-CSPBGA (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/analog-devices/aduc7122bbcz-rl

SPECIFICATIONS

AV_{DD} = IOV_{DD} = 3.0 V to 3.6 V, V_{REF} = 2.5 V internal reference, f_{CORE} = 41.78 MHz, T_A = −10°C to +95°C, unless otherwise noted.

Table 1.

Parameter	Min	Typ	Max	Unit	Test Conditions/Comments
ADC CHANNEL SPECIFICATIONS					Eight acquisition clocks and $f_{ADC}/2$
ADC Power-Up Time		5		µs	
DC Accuracy[1, 2]					
Resolution	12			Bits	
Integral Nonlinearity		±0.6	±2	LSB	2.5 V internal reference, not production tested for PADC0/PADC1 channels
Differential Nonlinearity[3, 4]		±0.5	+1.4/−0.99	LSB	2.5 V internal reference, gauranteed monotonic
DC Code Distribution		1		LSB	ADC input is a dc voltage
ENDPOINT ERRORS[5]					Internally unbuffered channels
Offset Error		±2	±5	LSB	
Offset Error Match		±1		LSB	
Gain Error		±2	±5	LSB	
Gain Error Match		±1		LSB	
DYNAMIC PERFORMANCE					f_{IN} = 10 kHz sine wave, f_{SAMPLE} = 1 MSPS internally unbuffered channels
Signal-to-Noise Ratio (SNR)		69		dB	Includes distortion and noise components
Total Harmonic Distortion (THD)		−78		dB	
Peak Harmonic or Spurious Noise		−75		dB	
Channel-to-Channel Crosstalk		−80		dB	Measured on adjacent channels
ANALOG INPUT					
Input Voltage Ranges					
Differential Mode			V_{CM}[6] ± $V_{REF}/2$	V	See Table 35 and Table 36
Single-Ended Mode			0 to V_{REF}	V	Buffer bypassed
Single-Ended Mode	0.15		AV_{DD} − 1.5	V	Buffer enabled
Leakage Current		±0.2		µA	
Input Capacitance		20		pF	During ADC acquisition buffer bypassed
Input Capacitance		20		pF	During ADC acquisition buffer enabled
PADC0 INPUT					28.3 kΩ resistor, PGA gain = 3; acquisition time = 6 µs, pseudo differential mode
Full Scale Input Range	20		1000	µA	
Input Leakage at PADC0P[4]		0.15	2	nA	
Resolution	11			Bits	0.1% accuracy, 5 ppm external resistor for I to V
Gain Error[4]			1	%	
Gain Drift[4]			50	ppm/°C	
Offset[4]		3	6	nA	PGA offset not included
Offset Drift[4]		30	60	pA/°C	
PADC0P Compliant Range	0.1		AV_{DD} − 1.2	V	
PADC1 INPUT					53.5 kΩ resistor, PGA gain = 3; acquisition time = 6 µs, pseudo differential mode
Full Scale Input Range	10.6		700	µA	
Input Leakage at PADC1P[4]		0.15	2	nA	
Resolution	11			Bits	0.1% accuracy, 5 ppm external resistor for I to V
Gain Error[4]			1	%	
Gain Drift[4]			50	ppm/°C	
Offset[4]		3	6	nA	PGA offset not included
Offset Drift[4]		30	60	pA/°C	
PADC1P Compliant Range	0.1		AV_{DD} − 1.2	V	

TIMING SPECIFICATIONS

Table 2. I²C Timing in Fast Mode (400 kHz)

		Slave			Master			
Parameter	**Description**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
t_L	SCLx low pulse width	200				1360		ns
t_H	SCLx high pulse width	100				1140		ns
t_{SHD}	Start condition hold time	300						ns
t_{DSU}	Data setup time	100				740		ns
t_{DHD}	Data hold time	0				400		ns
t_{RSU}	Setup time for repeated start	100						ns
t_{PSU}	Stop condition setup time	100				800		ns
t_{BUF}	Bus-free time between a stop condition and a start condition	1.3						µs
t_R	Rise time for both SCLx and SDAx			300		200		ns
t_F	Fall time for both SCLx and SDAx			300				ns

Table 3. I²C Timing in Standard Mode (100 kHz)

		Slave		
Parameter	**Description**	**Min**	**Typ**	**Max**	**Unit**
t_L	SCLx low pulse width	4.7			µs
t_H	SCLx 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 SCLx and SDAx			1	µs
t_F	Fall time for both SCLx and SDAx			300	ns

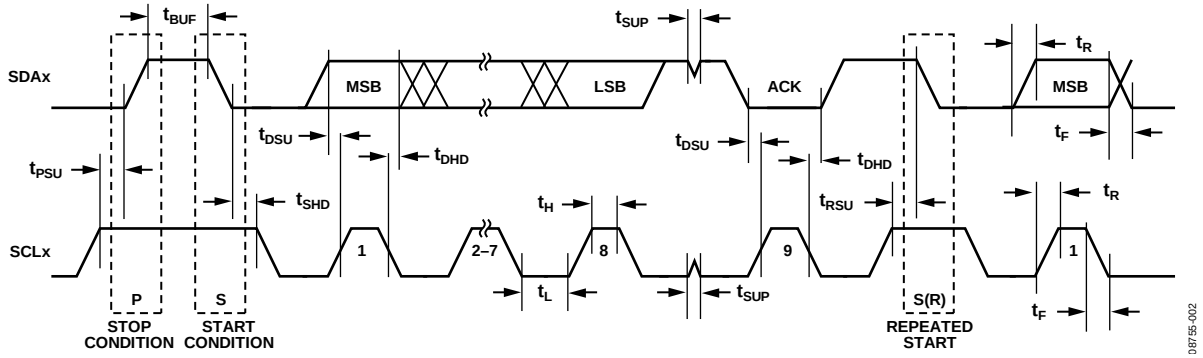

Figure 2. I²C-Compatible Interface Timing

ABSOLUTE MAXIMUM RATINGS

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

Table 8.

Parameter	Rating
AV_{DD} to IOV_{DD}	−0.3 V to +0.3 V
AGND to DGND	−0.3 V to +0.3 V
IOV_{DD} to IOGND, AV_{DD} to AGND	−0.3 V to +6 V
Digital Input Voltage to IOGND	−0.3 V to +5.3 V
Digital Output Voltage to IOGND	−0.3 V to IOV_{DD} + 0.3 V
V_{REF}_2.5 and V_{REF}_1.2 to AGND	−0.3 V to AV_{DD} + 0.3 V
Analog Inputs to AGND	−0.3 V to AV_{DD} + 0.3 V
Analog Outputs to AGND	−0.3 V to AV_{DD} + 0.3 V
Operating Temperature Range	−10°C to +95°C
Storage Temperature Range	−65°C to +150°C
Junction Temperature	150°C
θ_{JA} Thermal Impedance	
108-Ball CSP_BGA	40°C/W
Peak Solder Reflow Temperature	
SnPb Assemblies (10 sec to 30 sec)	240°C
RoHS-Compliant Assemblies (20 sec to 40 sec)	260°C

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

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

ESD CAUTION

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

TERMINOLOGY

ADC SPECIFICATIONS

Integral Nonlinearity (INL)
The maximum deviation of any code from a straight line passing through the endpoints of the ADC transfer function. The endpoints of the transfer function are zero scale, a point ½ LSB below the first code transition, and full scale, a point ½ LSB above the last code transition.

Differential Nonlinearity (DNL)
The difference between the measured and the ideal 1 LSB change between any two adjacent codes in the ADC.

Offset Error
The deviation of the first code transition (0000…000) to (0000…001) from the ideal, that is, ½ LSB.

Gain Error
The deviation of the last code transition from the ideal AIN voltage (full scale − 1.5 LSB) after the offset error has been adjusted out.

Signal-to-(Noise + Distortion) Ratio
The measured ratio of signal to (noise + distortion) at the output of the ADC. The signal is the rms amplitude of the fundamental. Noise is the rms sum of all nonfundamental signals up to half the sampling frequency ($f_S/2$), excluding dc. The ratio is dependent upon the number of quantization levels in the digitization process; the more levels there are, the smaller the quantization noise becomes.

The theoretical signal-to-(noise + distortion) ratio for an ideal N-bit converter with a sine wave input is given by

$$Signal\ to\ (Noise + Distortion) = (6.02\ N + 1.76)\ dB$$

Thus, for a 12-bit converter, this is 74 dB.

Total Harmonic Distortion
The ratio of the rms sum of the harmonics to the fundamental.

DAC SPECIFICATIONS

Relative Accuracy
Otherwise known as endpoint linearity, relative accuracy is a measure of the maximum deviation from a straight line passing through the endpoints of the DAC transfer function. It is measured after adjusting for zero error and full-scale error.

Voltage Output Settling Time
The amount of time it takes the output to settle to within a 1 LSB level for a full-scale input change.

OVERVIEW OF THE ARM7TDMI CORE

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

The ARM7TDMI is an ARM7 core with four additional features:

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

THUMB MODE (T)

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

However, the thumb mode has two limitations:

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

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

LONG MULTIPLY (M)

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

EmbeddedICE (I)

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

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

EXCEPTIONS

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

- Normal interrupt or IRQ. This is provided to service general-purpose interrupt handling of internal and external events.
- Fast interrupt or FIQ. This is provided to service data transfers or communication channels with low latency. FIQ has priority over IRQ.
- Memory abort.
- Attempted execution of an undefined instruction.
- Software interrupt instruction (SWI). This can be used to make a call to an operating system.

Typically, the programmer defines an interrupt as IRQ, but for a higher priority interrupt, that is, faster response time, the programmer can define the interrupt as FIQ.

ARM REGISTERS

ARM7TDMI has a total of 37 registers: 31 general-purpose registers and six status registers. Each operating mode has dedicated banked registers.

When writing user-level programs, 15 general-purpose, 32-bit registers (R0 to R14), the program counter (R15) and the current program status register (CPSR) are usable. The remaining registers are only used for system-level programming and exception handling.

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 8. The fast interrupt mode has more registers (R8 to R12) for fast interrupt processing. This means the interrupt processing can begin without the need to save or restore these registers, and thus save critical time in the interrupt handling process.

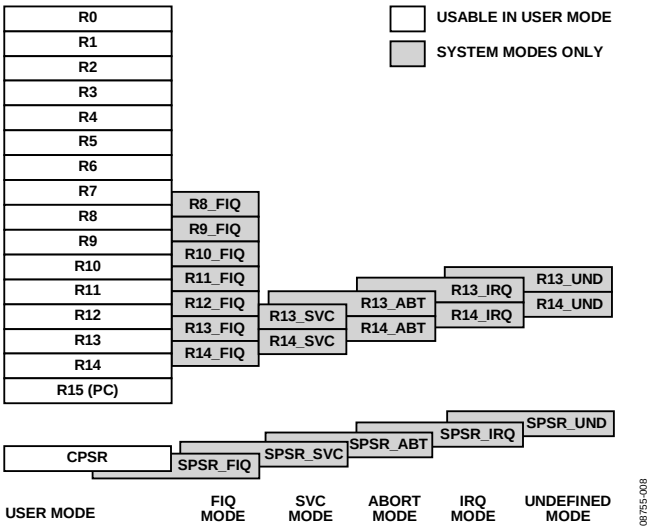

Figure 8. Register Organization

MEMORY ORGANIZATION

The ADuC7122 incorporates three separate blocks of memory: 8 kB of SRAM and two 64 kB of on-chip Flash/EE memory. There are 126 kB of on-chip Flash/EE memory available to the user, and the remaining 2 kB are reserved for the factory-configured boot page. These two blocks are mapped as shown in Figure 9.

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

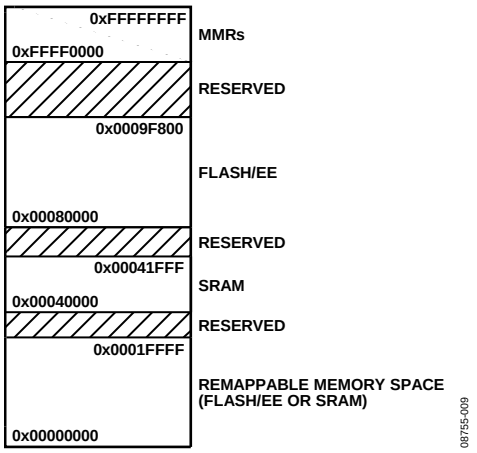

Figure 9. Physical Memory Map

Memory Access

The ARM7 core sees memory as a linear array of 2^{32} byte locations, where the different blocks of memory are mapped as outlined in Figure 9.

The ADuC7122 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.

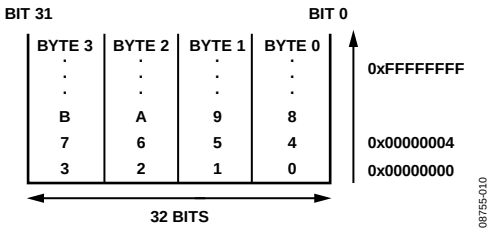

Figure 10. Little Endian Format

FLASH/EE MEMORY

The 128 kB of Flash/EE are organized as two banks of 32k × 16 bits. Block 0 starts at Address 0x90000 and finishes at Address 0x9F700. In this block, 31k × 16 bits is user space and 1k × 16 bits are reserved for the factory-configured boot page. The page size of this Flash/EE memory is 512 bytes.

Block 1 starts at Address 0x80000 and finishes at Address 0x90000. In this block 64 kB block is arranged in 32k × 16 bits, all of which is available as user space.

The 126 kB of Flash/EE are available to the user as code and nonvolatile data memory. There is no distinction between data and program because ARM code shares the same space. The real width of the Flash/EE memory is 16 bits, meaning that in ARM mode (32-bit instruction), two accesses to the Flash/EE are necessary for each instruction fetch. Therefore, it is recommended that Thumb mode be used when executing from Flash/EE memory for optimum access speed. The maximum access speed for the Flash/EE memory is 41.78 MHz in Thumb mode and 20.89 MHz in full ARM mode (see the Execution Time from SRAM and FLASH/EE section).

SRAM

The 8 kB of SRAM are available to the user, organized as 2k × 32 bits, that is, 2k words. ARM code can run directly from SRAM at 41.78 MHz, given that the SRAM array is configured as a 32-bit wide memory array (see the Execution Time from SRAM and FLASH/EE section).

MEMORY MAPPED REGISTERS

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

The MMR space provides an interface between the CPU and all on-chip peripherals. All registers except the core registers reside in the MMR area. All shaded locations shown in Figure 11 are unoccupied or reserved locations and should not be accessed by user software. Table 10 to Table 26 show a full MMR memory map.

The access time reading or writing a MMR depends on the advanced microcontroller bus architecture (AMBA) bus used to access the peripheral. The processor has two AMBA buses: advanced high performance bus (AHB) used for system modules and advanced peripheral bus (APB) used for lower performance peripheral. Access to the AHB is one cycle, and access to the APB is two cycles. All peripherals on the ADuC7122 are on the APB except the Flash/EE memory and the GPIOs.

Table 57. FEE0PRO and FEE0HID MMR Bit Designations

Bit	Description
31	Read protection. Cleared by the user to protect Block 0. Set by the user to allow reading of Block 0.
30:0	Write protection for Page 123 to Page 0. Each bit protects a group of 4 pages. Cleared by the user to protect the pages when writing to flash. Thus preventing an accidental write to specific pages in flash Set by the user to allow writing the pages.

Table 58. FEE1PRO and FEE1HID MMR Bit Designations

Bit	Description
31	Read protection. Cleared by the user to protect Block 1. Set by the user to allow reading of Block 1.
30	Write protection for Page 127 to Page 120. Cleared by the user to protect the pages when writing to flash. Thus preventing an accidental write to specific pages in flash. Set by the user to allow writing the pages.
29:0	Write protection for Page 119 to Page 0. Each bit protects a group of 4 pages. Cleared by the user to protect the pages when writing to flash. Thus preventing an accidental write to specific pages in flash Set by the user to allow writing the pages.

EXECUTION TIME FROM SRAM AND FLASH/EE

This section describes SRAM and Flash/EE access times during execution of applications where execution time is critical.

Execution from SRAM

Fetching instructions from SRAM takes one clock cycle because the access time of the SRAM is 2 ns and a clock cycle is 22 ns minimum. However, if the instruction involves reading or writing data to memory, one extra cycle must be added if the data is in SRAM (or three cycles if the data is in Flash/EE), one cycle to execute the instruction and two cycles to obtain the 32-bit data from Flash/EE. A control flow instruction, such as a branch instruction, takes one cycle to fetch, but it also takes two cycles to fill the pipeline with the new instructions.

Execution from Flash/EE

Because the Flash/EE width is 16 bits and access time for 16-bit words is 23 ns, execution from Flash/EE cannot be completed in one cycle (contrary to a SRAM fetch, which can be completed in a single cycle when CD bits = 0). Dependent on the instruction, some dead times may be required before accessing data for any value of CD bits.

In ARM mode, where instructions are 32 bits, two cycles are needed to fetch any instruction when CD = 0. In Thumb mode, where instructions are 16 bits, one cycle is needed to fetch any instruction.

Timing is identical in both modes when executing instructions that involve using Flash/EE for data memory. If the instruction to be executed is a control flow instruction, an extra cycle is needed to decode the new address of the program counter and then four cycles are needed to fill the pipeline. A data processing instruction involving only core registers does not require any extra clock cycles, but if it involves data in Flash/EE, an extra clock cycle is needed to decode the address of the data and two cycles to obtain the 32-bit data from Flash/EE. An extra cycle must also be added before fetching another instruction. Data transfer instructions are more complex and are summarized in Table 59.

Table 59. Execution Cycles in ARM/Thumb Mode

Instructions	Fetch Cycles	Dead Time	Data Access	Dead Time
LD	2/1	1	2	1
LDH	2/1	1	1	1
LDM/PUSH	2/1	N	2 × N	N
STR	2/1	1	2 × 20 µs	1
STRH	2/1	1	20 µs	1
STRM/POP	2/1	N	2 × N × 20 µs	N

With 1 < N ≤ 16, N is the number of bytes of data to load or store in the multiple load/store instruction. The SWAP instruction combines an LD and STR instruction with only one fetch, giving a total of eight cycles plus 40 µs.

POWER CONTROL SYSTEM

A choice of operating modes is available on the ADuC7122. Table 68 describes which blocks of the ADuC7122 are powered on in the different modes and indicates the power-up time. Table 69 gives some typical values of the total current consumption (analog and digital supply currents) in the different modes, depending on the clock divider bits when the ADC is turned off. Note that these values also include current consumption of the regulator and other parts on the test board on which these values were measured.

Table 68. Operating Modes

Mode	Core	Peripherals	PLL	XTAL/Timer2/Timer3	XIRQ	Start-Up/Power-On Time
Active	On	On	On	On	On	130 ms at CD = 0
Pause		On	On	On	On	24 ns at CD = 0; 3.06 µs at CD = 7
Nap			On	On	On	24 ns at CD = 0; 3.06 µs at CD = 7
Sleep				On	On	1.58 ms
Stop					On	1.7 ms

Table 69. Typical Current Consumption at 25°C

PC[2:0]	Mode	CD = 0	CD = 1	CD = 2	CD = 3	CD = 4	CD = 5	CD = 6	CD = 7
000	Active	30	21.2	13.8	11	8.1	7.2	6.7	6.45
001	Pause	22.7	13.3	8.5	6.1	4.9	4.3	4	3.85
010	Nap	3.8	3.8	3.8	3.8	3.8	3.8	3.8	3.8
011	Sleep	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
100	Stop	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25

Bit	Name	Description
8:6	PWMCP[2:0]	PWM clock prescaler bits. Sets the UCLK divider. 000 = UCLK/2. 001 = UCLK/4. 010 = UCLK/8. 011 = UCLK/16. 100 = UCLK/32. 101 = UCLK/64. 110 = UCLK/128. 111 = UCLK/256.
5	POINV	Set to 1 by the user to invert all PWM outputs. Cleared by the user to use PWM outputs as normal.
4	HOFF	High-side off. Set to 1 by the user to force PWM1 and PWM3 outputs high. This also forces PWM2 and PWM4 low. Cleared by the user to use the PWM outputs as normal.
3	LCOMP	Load compare registers. Set to 1 by the user to load the internal compare registers with the values in PWMxCOMx on the next transition of the PWM timer from 0x00 to 0x01. Cleared by the user to use the values previously stored in the internal compare registers.
2	DIR	Direction control. Set to 1 by the user to enable PWM1 and PWM2 as the output signals while PWM3 and PWM4 are held low. Cleared by the user to enable PWM3 and PWM4 as the output signals while PWM1 and PWM2 are held low.
1	HMODE	Enables H-bridge mode. Set to 1 by the user to enable H-Bridge mode and Bit 1 to Bit 5 of PWMCON1. Cleared by the user to operate the PWMs in standard mode.
0	PWMEN	Set to 1 by the user to enable all PWM outputs. Cleared by the user to disable all PWM outputs.

UART Status Register 0

Name: COMSTA0

Address: 0xFFFF0814

Default Value: 0x60

Access: Read only

Function: This 8-bit read-only register reflects the current status on the UART.

Table 97. COMSTA0 MMR Bit Designations

Bit	Name	Description
7		Reserved.
6	TEMT	COMTX and shift register empty status bit. Set automatically if COMTX and the shift register are empty. This bit indicates that the data has been transmitted, that is, no more data is present in the shift register. Cleared automatically when writing to COMTX.
5	THRE	COMTX empty status bit. Set automatically if COMTX is empty. COMTX can be written as soon as this bit is set, the previous data might not have been transmitted yet and can still be present in the shift register. Cleared automatically when writing to COMTX.
4	BI	Break indicator. Set when SIN is held low for more than the maximum word length. Cleared automatically.
3	FE	Framing error. Set when the stop bit is invalid. Cleared automatically.
2	PE	Parity error. Set when a parity error occurs. Cleared automatically.
1	OE	Overrun error. Set automatically if data are overwritten before being read. Cleared automatically.
0	DR	Data ready. Set automatically when COMRX is full. Cleared by reading COMRX.

SERIAL PERIPHERAL INTERFACE

The ADuC7122 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 20 Mb.

The SPI port can be configured for master or slave operation and typically consists of four pins: SPIMISO, SPIMOSI, SPICLK, and SPI$\overline{CS}$.

SPIMISO (MASTER IN, SLAVE OUT) PIN

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

SPIMOSI (MASTER OUT, SLAVE IN) PIN

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

SPICLK (SERIAL CLOCK I/O) PIN

The master serial clock (SPICLK) synchronizes the data being transmitted and received through the MOSI SPICLK period. Therefore, a byte is transmitted/received after eight SPICLK periods. The SPICLK 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 on 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 10 Mb.

In both master and slave modes, data is transmitted on one edge of the SPICLK signal and sampled on the other. Therefore, it is important that the polarity and phase are configured the same for the master and slave devices.

SPI CHIP SELECT (SPI$\overline{CS}$ INPUT) PIN

In SPI slave mode, a transfer is initiated by the assertion of SPI$\overline{CS}$, 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 SPI$\overline{CS}$. In slave mode, SPI$\overline{CS}$ is always an input.

In SPI master mode, SPI$\overline{CS}$ 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 ADuC7122 device are P0.2 to P0.5.

P0.5 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.

P0.2 is the SPICLK pin.

P0.3 is the master in, slave out (SPIMISO) pin.

P0.4 is the master out, slave in (SPIMOSI) pin.

To configure P0.2 to P0.5 for SPI mode, see the General-Purpose I/O section.

SPIRX Register

Name: SPIRX

Address: 0xFFFF0A04

Default Value: 0x00

Access: Read

Function: This 8-bit MMR is the SPI receive register.

SPITX Register

Name: SPITX

Address: 0xFFFF0A08

Default Value: 0x00

Access: Write

Function: This 8-bit MMR is the SPI transmit register.

SPIDIV Register

Name: SPIDIV

Address: 0xFFFF0A0C

Default Value: 0x1B

Access: Read/write

Function: This 8-bit MMR is the SPI baud rate selection register.

SPI Control Register

Name: SPICON

Address: 0xFFFF0A10

Default Value: 0x0000

Access: Read/write

Function: This 16-bit MMR configures the SPI peripheral in both master and slave modes.

Table 112. SPICON MMR Bit Designations

Bit	Name	Description
15:14	SPIMDE	SPI IRQ mode bits. These bits configure when the Tx/Rx interrupts occur in a transfer. 00 = Tx interrupt occurs when one byte has been transferred. Rx interrupt occurs when one or more bytes have been received by the FIFO. 01 = Tx interrupt occurs when two bytes have been transferred. Rx interrupt occurs when two or more bytes have been received by the FIFO. 10 = Tx interrupt occurs when three bytes have been transferred. Rx interrupt occurs when three or more bytes have been received by the FIFO. 11 = Tx interrupt occurs when four bytes have been transferred. Rx interrupt occurs when the Rx FIFO is full, or four bytes present.
13	SPITFLH	SPI Tx FIFO flush enable bit. Set this bit to flush the Tx FIFO. This bit does not clear itself and should be toggled if a single flush is required. If this bit is left high, then either the last transmitted value or 0x00 is transmitted depending on the SPIZEN bit. When the flush enable bit is set, the FIFO is cleared within a single microprocessor cycle. Any writes to the Tx FIFO are ignored while this bit is set. Clear this bit to disable Tx FIFO flushing.
12	SPIRFLH	SPI Rx FIFO flush enable bit. Set this bit to flush the Rx FIFO. This bit does not clear itself and should be toggled if a single flush is required. When the flush enable bit is set, the FIFO is cleared within a single microprocessor cycle. If this bit is set, all incoming data is ignored and no interrupts are generated. If set and SPITMDE = 0, a read of the Rx FIFO initiates a transfer. Clear this bit to disable Rx FIFO flushing.
11	SPICONT	Continuous transfer enable. Set by the user to enable continuous transfer. In master mode, the transfer continues until no valid data is available in the Tx register. SPI$\overline{CS}$ is asserted and remains asserted for the duration of each 8-bit serial transfer until Tx is empty. Cleared by the 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 of one serial clock cycle.
10	SPILP	Loop back enable bit. Set by the user to connect MISO to MOSI and test software. Cleared by the user to place in normal mode.
9	SPIOEN	Slave MISO output enable bit. Set this bit for SPIMISO to operate as normal. Clear this bit to disable the output driver on the SPIMISO pin. The SPIMISO pin is open-drain when this bit is clear.

PROGRAMMABLE LOGIC ARRAY (PLA)

The ADuC7122 integrates a fully programmable logic array (PLA) that consists of two, independent but interconnected PLA blocks. Each block consists of eight PLA elements, giving each part a total of 16 PLA elements.

Each PLA element contains a two-input look-up table that can be configured to generate any logic output function based on two inputs and a flip-flop. This is represented in Figure 35.

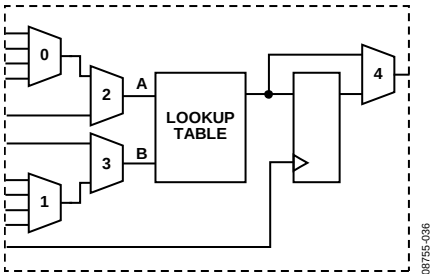

Figure 35. PLA Element

In total, 32 GPIO pins are available on each ADuC7122 for the PLA. These include 16 input pins and 16 output pins that need to be configured in the GPxCON register as PLA pins before using the PLA. Note that the comparator output is also included as one of the 16 input pins.

The PLA is configured via a set of user MMRs. The output(s) of the PLA can be routed to the internal interrupt system, to the $\overline{\text{CONVST}}$ signal of the ADC, to an MMR, or to any of the 16 PLA output pins.

The two blocks can be interconnected as follows:

- Output of Element 15 (Block 1) can be fed back to Input 0 of Mux 0 of Element 0 (Block 0)
- Output of Element 7 (Block 0) can be fed back to the Input 0 of Mux 0 of Element 8 (Block 1)

Table 113. Element Input/Output

PLA Block 0			PLA Block 1		
Element	**Input**	**Output**	**Element**	**Input**	**Output**
0	P2.7	P3.0	8	P1.4	P3.4
1	P2.2	P3.1	9	P1.5	P3.5
2	P0.6	P3.2	10	P0.5	P3.6
3	P0.7	P3.3	11	P0.4	P3.7
4	P0.1	P1.7	12	P2.1	P0.3
5	P0.0	P1.6	13	P2.0	P0.2
6	P1.1	P2.5	14	P2.3	P1.3
7	P1.0	P2.4	15	P2.6	P1.2

PLA MMRs Interface

The PLA peripheral interface consists of the 21 MMRs described in Table 114 to Table 128.

Table 114. PLAELMx Registers

Name	Address	Default Value	Access
PLAELM0	0xFFFF0B00	0x0000	R/W
PLAELM1	0xFFFF0B04	0x0000	R/W
PLAELM2	0xFFFF0B08	0x0000	R/W
PLAELM3	0xFFFF0B0C	0x0000	R/W
PLAELM4	0xFFFF0B10	0x0000	R/W
PLAELM5	0xFFFF0B14	0x0000	R/W
PLAELM6	0xFFFF0B18	0x0000	R/W
PLAELM7	0xFFFF0B1C	0x0000	R/W
PLAELM8	0xFFFF0B20	0x0000	R/W
PLAELM9	0xFFFF0B24	0x0000	R/W
PLAELM10	0xFFFF0B28	0x0000	R/W
PLAELM11	0xFFFF0B2C	0x0000	R/W
PLAELM12	0xFFFF0B30	0x0000	R/W
PLAELM13	0xFFFF0B34	0x0000	R/W
PLAELM14	0xFFFF0B38	0x0000	R/W
PLAELM15	0xFFFF0B3C	0x0000	R/W

PLAELMx are Element 0 to Element 15 control registers. They configure the input and output mux of each element, select the function in the look-up table, and bypass/use the flip-flop. See Table 115 and Table 118.

Table 119. PLAIRQ Register

Name	Address	Default Value	Access
PLAIRQ	0xFFFF0B44	0x00000000	R/W

PLAIRQ enables IRQ0 and/or IRQ1 and selects the source of the IRQ.

Table 120. PLAIRQ MMR Bit Descriptions

Bit	Value	Description
15:13		Reserved.
12		PLA IRQ1 enable bit.
	1	Set by the user to enable the IRQ1 output from PLA.
	0	Cleared by the user to disable IRQ1 output from PLA.
11:8		PLA IRQ1 source.
	0000	PLA Element 0.
	0001	PLA Element 1.
	1111	PLA Element 15.
7:5		Reserved.
4		PLA IRQ0 enable bit. Set by the user to enable IRQ0 output from PLA. Cleared by the user to disable IRQ0 output from PLA.
3:0		PLA IRQ0 source.
	0000	PLA Element 0.
	0001	PLA Element 1.
	1111	PLA Element 15.

Table 121. PLAADC Register

Name	Address	Default Value	Access
PLAADC	0xFFFF0B48	0x00000000	R/W

PLAADC is the PLA source for the ADC start conversion signal.

Table 122. PLAADC MMR Bit Descriptions

Bit	Value	Description
31:5		Reserved.
4		ADC start conversion enable bit.
	1	Set by the user to enable ADC start conversion from PLA.
	0	Cleared by the user to disable ADC start conversion from PLA.
3:0		ADC start conversion source.
	0000	PLA Element 0.
	0001	PLA Element 1.
	1111	PLA Element 15.

Table 123. PLADIN Register

Name	Address	Default Value	Access
PLADIN	0xFFFF0B4C	0x00000000	R/W

Table 124. PLADIN MMR Bit Descriptions

Bit	Description
31:16	Reserved.
15:0	Input bit to Element 15 to Element 0.

PLADIN is a data input MMR for PLA.

Table 125. PLADOUT Register

Name	Address	Default Value	Access
PLADOUT	0xFFFF0B50	0x00000000	R

PLADOUT is a data output MMR for PLA. This register is always updated.

Table 126. PLADOUT MMR Bit Descriptions

Bit	Description
31:16	Reserved.
15:0	Output bit from Element 15 to Element 0.

Table 127. PLACLK Register

Name	Address	Default Value	Access
PLACLK	0xFFFF0B40	0x00	W

PLACLK is a PLA lock option. Bit 0 is written only once. When set, it does not allow modification of any of the PLA MMRs, except PLADIN. A PLA tool is provided in the development system to easily configure the PLA.

INTERRUPT SYSTEM

There are 27 interrupt sources on the ADuC7122 that are controlled by the interrupt controller. All interrupts are generated from the on-chip peripherals, except for the software interrupt (SWI), which is programmable by the user. The ARM7TDMI CPU core only recognizes interrupts as one of two types: a normal interrupt request (IRQ) and a fast interrupt request (FIQ). All the interrupts can be masked separately.

The control and configuration of the interrupt system is managed through a number of interrupt-related registers. The bits in each IRQ and FIQ register represent the same interrupt source, as described in Table 128.

The ADuC7122 contains a vectored interrupt controller (VIC) that supports nested interrupts up to eight levels. The VIC also allows the programmer to assign priority levels to all interrupt sources. Interrupt nesting needs to be enabled by setting the ENIRQN bit in the IRQCONN register. A number of extra MMRs are used when the full vectored interrupt controller is enabled.

IRQSTA/FIQSTA should be saved immediately upon entering the interrupt service routine (ISR) to ensure that all valid interrupt sources are serviced.

Table 128. IRQ/FIQ[1] MMRs Bit Designations

Bit	Description	Comments
0	All interrupts OR'ed (FIQ only)	This bit is set if any FIQ is active
1	Software interrupt	User programmable interrupt source
2	Timer0	General-Purpose Timer0
3	Timer1	General-Purpose Timer1
4	Timer2 or wake-up timer	General-Purpose Timer2 or wake-up timer
5	Timer3 or watchdog timer	General-Purpose Timer3 or watchdog timer
6	Timer4	General-Purpose Timer4
7	Reserved	Reserved
8	PSM	Power supply monitor
9	Undefined	This bit is not used
10	Flash Control 0	Flash controller for Block 0 interrupt
11	Flash Control 1	Flash controller for Block 1 interrupt
12	ADC	ADC interrupt source bit
13	UART	UART interrupt source bit
14	SPI	SPI interrupt source bit
15	I2C0 master IRQ	I^2C master interrupt source bit
16	I2C0 slave IRQ	I^2C slave interrupt source bit
17	I2C1 master IRQ	I^2C master interrupt source bit
18	I2C1 slave IRQ	I^2C slave interrupt source bit
19	XIRQ0 (GPIO IRQ0)	External Interrupt 0
20	XIRQ1 (GPIO IRQ1)	External Interrupt 1
21	XIRQ2 (GPIO IRQ2)	External Interrupt 2
22	XIRQ3 (GPIO IRQ3)	External Interrupt 3
23	PWM	PWM trip interrupt source bit
24	XIRQ4 (GPIO IRQ4)	External Interrupt 4
25	XIRQ5 (GPIO IRQ5)	External Interrupt 5
26	PLA IRQ0	PLA Block 0 IRQ bit
27	PLA IRQ1	PLA Block 1 IRQ bit

[1] Applies to IRQEN, FIQEN, IRQCLR, FIQCLR, IRQSTA, and FIQSTA registers.

IRQ

The IRQ is the exception signal to enter the IRQ mode of the processor. It services general-purpose interrupt handling of internal and external events.

All 32 bits are logically OR'ed to create a single IRQ signal to the ARM7TDMI core. The four 32-bit registers dedicated to IRQ are IRQSIG, IRQEN, IRQCLR, and IRQSTA.

IRQSIG

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 clear when the interrupt in the particular peripheral is cleared. All IRQ sources can be masked in the IRQEN MMR. IRQSIG is read only.

IRQSIG Register

Name: IRQSIG

Address: 0xFFFF0004

Default Value: 0x00000000

Access: Read only

IRQEN

IRQEN provides the value of the current enable mask. When a bit is set to 1, the corresponding source request is enabled to create an IRQ exception. When a bit is set to 0, the corresponding source request is disabled or masked, which does not create an IRQ exception. The IRQEN register cannot be used to disable an interrupt.

IRQEN Register

Name: IRQEN

Address: 0xFFFF0008

Default Value: 0x00000000

Access: Read/write

IRQCLR

IRQCLR is a write-only register that allows the IRQEN register to clear to mask an interrupt source. Each bit that is 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 atomic read-modify-write.

IRQCLR Register

Name: IRQCLR

Address: 0xFFFF000C

Default Value: 0x00000000

Access: Write only

IRQSTA

IRQSTA is a read-only register that provides the current enabled IRQ source status (effectively a logic AND of the IRQSIG and IRQEN bits). When set to 1, that source generates 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.

IRQSTA Register

Name: IRQSTA

Address: 0xFFFF0000

Default Value: 0x00000000

Access: Read only

FAST INTERRUPT REQUEST (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 and provides 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 to 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 clears, as a side effect, the same bit in IRQEN. Likewise, a bit set to 1 in IRQEN clears, as a side effect, the same bit in FIQEN. An interrupt source can be disabled in both IRQEN and FIQEN masks.

FIQSIG

FIQSIG reflects the status of the different FIQ sources. If a peripheral generates an FIQ signal, the corresponding bit in the FIQSIG is set; otherwise, it is cleared. The FIQSIG bits are cleared when the interrupt in the particular peripheral is cleared. All FIQ sources can be masked in the FIQEN MMR. FIQSIG is read only.

FIQSIG Register

Name: FIQSIG

Address: 0xFFFF0104

Default Value: 0x00000000

Access: Read only

IRQCLRE Register

Name: IRQCLRE

Address: 0xFFFF0038

Default Value: 0x00000000

Access: Write only

Table 141. IRQCLRE MMR Bit Designations

Bit	Name	Description
31:26	Reserved	These bits are reserved and should not be written to.
25	IRQ5CLRI	A 1 must be written to this bit in the IRQ5 interrupt service routine to clear an edge.
24	IRQ4CLRI	A 1 must be written to this bit in the IRQ4 interrupt service routine to clear an edge.
23	Reserved	This bit is reserved and should not be written to.
22	IRQ3CLRI	A 1 must be written to this bit in the IRQ3 interrupt service routine to clear an edge triggered IRQ3 interrupt.
21	IRQ2CLRI	A 1 must be written to this bit in the IRQ2 interrupt service routine to clear an edge triggered IRQ2 interrupt.
20	IRQ1CLRI	A 1 must be written to this bit in the IRQ1 interrupt service routine to clear an edge triggered IRQ1 interrupt.
19	IRQ0CLRI	A 1 must be written to this bit in the IRQO interrupt service routine to clear an edge triggered IRQ0 interrupt.
18:0	Reserved	These bits are reserved and should not be written to.

TIMERS

The ADuC7122 has 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/increments from the maximum/minimum value until zero scale/full scale and starts again at the maximum/minimum value.

In periodic mode, the counter decrements/increments from the value in the load register (TxLD MMR) until zero scale/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 the clear register of the particular timer (TxCLRI).

The event selection feature allows flexible interrupt generation based on Timer0 and Timer1. T0CON and T1CON can be used to configure the interrupt sources, as shown in Table 142. When either Timer0 or Timer1 expires, an interrupt occurs based on the event selection in T0CON and T1CON MMRs.

Table 142. Event Selection Numbers

Event Selection (TxCON[16:12])	Interrupt Number	Name
00000	2	Timer0
00001	3	Timer1
00010	4	Wake-up timer (Timer2)
00011	5	Watchdog timer (Timer3)
00100	6	Timer4
00101	7	Reserved
00110	8	Power supply monitor
00111	9	Undefined
01000	10	Flash Block 0
01001	11	Flash Block 1
01010	12	ADC
01011	13	UART
01100	14	SPI
01101	15	I2C0 master
01110	16	I2C0 slave
01111	17	I2C1 master
10000	18	I2C1 slave
10001	19	External IRQ0

HOUR:MINUTE:SECOND:1/128 FORMAT

To use the timer in hour:minute:second:hundredths format, select the 32,768 kHz clock and prescaler of 256. The hundredths field does not represent milliseconds but 1/128 of a second (256/32,768). The bits representing the hour, minute, and second are not consecutive in the register. This arrangement applies to TxLD and TxVAL when using the hour:minute:second: hundredths format as set in TxCON[5:4]. See Table 143 for additional details.

Table 143. Hour:Minnute:Second:Hundredths Format

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

TIMER0—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. Timer0 can also be clocked from the undivided core clock, internal 32 kHz oscillator, or external 32 kHz crystal.

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 that 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's source initial assertion. When 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 T0CLRI is written.

The Timer0 interface consists of six MMRs, shown in Table 144.

Table 144. Timer0 Interface MMRs

Name	Description
T0LD	16-bit register that holds the 16-bit value loaded into the counter. Available only in 16-bit mode.
T0CAP	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). T0VAL1 is a 32-bit register that holds the 32 most significant bits (MSBs).
T0CLRI	8-bit register. Writing any value to this register clears the interrupt. Available only in 16-bit mode.
T0CON	Configuration MMR.

Table 145. 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.

Table 146. 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; it is only available in 16-bit mode.

Table 147. 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 148. T0CON MMR Bit Designations

Bit	Value	Description
31:18		Reserved.
17		Event select bit. Set by the user to enable time capture of an event. Cleared by the user to disable time capture of an event.
16:12		Event select (ES) range, 0 to 17. The events are as described in the Timers section.
11		Reserved.
10:9		Clock select.
	00	Internal 32 kHz oscillator.
	01	UCLK.
	10	External 32 kHz crystal.
	11	HCLK.
8		Count up. Available only in 16-bit mode. Set by the user for Timer0 to count up. Cleared by the user for Timer0 to count down (default).
7		Timer0 enable bit. Set by the user to enable Timer0. Cleared by the user to disable Timer0 (default).
6		Timer0 mode. Set by the user to operate in periodic mode. Cleared by the 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.

Table 149. Timer0 Load Registers

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; it is available only in 16-bit mode.

Table 150. Timer0 Clear Register

Name	Address	Default Value	Access
T0CLRI	0xFFFF0310	0x00	W

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

TIMER2—WAKE-UP TIMER

Timer2 is a 32-bit wake-up timer, count down or count up, with a programmable prescaler. The prescaler is clocked directly from 1 of 4 clock sources, including the core clock (default selection), the internal 32.768 kHz oscillator, the external 32.768 kHz watch crystal, or the PLL undivided clock. The selected clock source can be scaled by a factor of 1, 16, 256, or 32,768. The wake-up timer continues to run when the core clock is disabled. This gives a minimum resolution of 22 ns when the core is operating at 41.78 MHz and with a prescaler of 1. Capture of the current timer value is enabled if the Timer2 interrupt is enabled via IRQEN[4] (see Table 128).

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

Timer2 reloads the value from T2LD either when Timer2 overflows or immediately when T2ICLR is written.

The Timer2 interface consists of four MMRs, shown in Table 158.

Table 158. Timer2 Interface Registers

Register	Description
T2LD	32-bit register. Holds 32-bit unsigned integers.
T2VAL	32-bit register. Holds 32-bit unsigned integers. This register is read only.
T2CLRI	8-bit register. Writing any value to this register clears the Timer2 interrupt.
T2CON	Configuration MMR.

Table 159. Timer2 Load Registers

Name	Address	Default Value	Access
T2LD	0xFFFF0340	0x00000	R/W

T2LD is a 32-bit register, which holds the 32-bit value that is loaded into the counter.

Table 160. Timer2 Clear Register

Name	Address	Default Value	Access
T2CLRI	0xFFFF034C	0x00	W

This 8-bit write-only MMR is written (with any value) by user code to refresh (reload) Timer2.

Table 161. Timer2 Value Register

Name	Address	Default Value	Access
T2VAL	0xFFFF0344	0x0000	R

T2VAL is a 32-bit register that holds the current value of Timer2.

Table 162. Timer2 Control Register

Name	Address	Default Value	Access
T2CON	0xFFFF0348	0x0000	R/W

This 32-bit MMR configures the mode of operation for Timer2.

Table 163. T2CON MMR Bit Designations

Bit	Value	Description
31:11		Reserved.
10:9		Clock source select.
	00	Internal 32.768 kHz oscillator (default).
	01	Core clock.
	10	External 32.768 kHz watch crystal.
	11	UCLK.
8		Count up.
	1	Set by the user for Timer2 to count up.
	0	Cleared by the user for Timer2 to count down (default).
7		Timer2 enable bit.
	1	Set by the user to enable Timer2.
	0	Cleared by the user to disable Timer2 (default).
6		Timer2 mode.
	1	Set by the user to operate in periodic mode.
	0	Cleared by the user to operate in free-running mode (default).
5:4		Format.
	00	Binary (default).
	01	Reserved.
	10	Hr:min:sec:hundredths. 23 hours to 0 hours.
	11	Hr:min:sec:hundredths. 255 hours to 0 hours.
3:0		Prescaler.
	0000	Source clock/1 (default).
	0100	Source clock/16.
	1000	Source clock/256 (this setting should be used in conjunction with Timer2 Format 10 and Format 11).
	1111	Source clock/32,768.