Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Data Sheet: JN5148-001

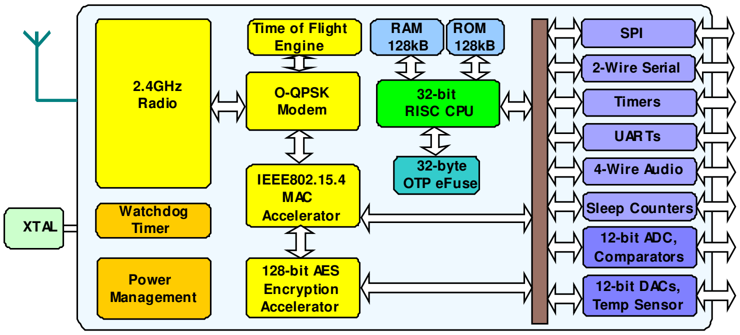
IEEE802.15.4 Wireless Microcontroller

Overview

The JN5148-001 is an ultra low power, high performance wireless microcontroller targeted at JenNet and ZigBee PRO networking applications. The device features an enhanced 32-bit RISC processor offering high coding efficiency through variable width instructions, a multi-stage instruction pipeline and low power operation with programmable clock speeds. It also includes a 2.4GHz IEEE802.15.4 compliant transceiver, 128kB of ROM, 128kB of RAM, and a rich mix of analogue and digital peripherals. The large memory footprint allows the device to run both a network stack (e.g. ZigBee PRO) and an embedded application or in a co-processor mode. The operating current is below 18mA, allowing operation direct from a coin cell.

Enhanced peripherals include low power pulse counters running in sleep mode designed for pulse counting in AMR applications and a unique Time of Flight ranging engine, allowing accurate location services to be implemented on wireless sensor networks. It also includes a 4-wire I^{2}S audio interface, to interface directly to mainstream audio CODECs, as well as conventional MCU peripherals.

Block Diagram



Benefits

- Single chip integrates transceiver and microcontroller for wireless sensor networks
- Large memory footprint to run ZigBee PRO or JenNet together with an application
- Very low current solution for long battery life
- Highly featured 32-bit RISC CPU for high performance and low power
- System BOM is low in component count and cost
- Extensive user peripherals

Applications

- Robust and secure low power wireless applications
- ZigBee PRO and JenNet networks
- Smart metering (e.g. AMR)
- Home and commercial building automation
- Location Aware services – e.g. Asset Tracking
- Industrial systems
- Telemetry
- Remote Control
- Toys and gaming peripherals

Features: Transceiver

- 2.4GHz IEEE802.15.4 compliant
- Time of Flight ranging engine
- 128-bit AES security processor
- MAC accelerator with packet formatting, CRCs, address check, auto-acks, timers
- 500 & 667kbps data rate modes
- Integrated sleep oscillator for low power
- On chip power regulation for 2.0V to 3.6V battery operation
- Deep sleep current 100nA
- Sleep current with active sleep timer 1.25µA
- <$0.50 external component cost
- Rx current 17.5mA
- Tx current 15.0mA
- Receiver sensitivity -95dBm
- Transmit power 2.5dBm

Features: Microcontroller

- Low power 32-bit RISC CPU, 4 to 32MHz clock speed
- Variable instruction width for high coding efficiency
- Multi-stage instruction pipeline
- 128kB ROM and 128kB RAM for bootloaded program code & data
- JTAG debug interface
- 4-input 12-bit ADC, 2 12-bit DACs, 2 comparators
- 3 application timer/counters,
- 2 UARTs
- SPI port with 5 selects
- 2-wire serial interface
- 4-wire digital audio interface
- Watchdog timer
- Low power pulse counters
- Up to 21 DIO

Industrial temp (-40°C to +85°C)

8x8mm 56-lead Punched QFN

Lead-free and RoHS compliant

Contents

1 Introduction

The JN5148-001 is an IEEE802.15.4 wireless microcontroller that provides a fully integrated solution for applications using the IEEE802.15.4 standard in the 2.4 - 2.5GHz ISM frequency band [1], including JenNet and ZigBee PRO. It includes all of the functionality required to meet the IEEE802.15.4, JenNet and ZigBee PRO specifications and has additional processor capability to run a wide range of applications including, but not limited to Smart Energy, Automatic Meter Reading, Remote Control, Home and Building Automation, Toys and Gaming.

Applications that transfer data wirelessly tend to be more complex than wired ones. Wireless protocols make stringent demands on frequencies, data formats, timing of data transfers, security and other issues. Application development must consider the requirements of the wireless network in addition to the product functionality and user interfaces. To minimise this complexity, NXP provides a series of software libraries and interfaces that control the transceiver and peripherals of the JN5148. These libraries and interfaces remove the need for the developer to understand wireless protocols and greatly simplifies the programming complexities of power modes, interrupts and hardware functionality.

In view of the above, the register details of the JN5148 are not provided in the datasheet.

The device includes a Wireless Transceiver, RISC CPU, on chip memory and an extensive range of peripherals.

Hereafter, the JN5148-001 will be referred to as JN5148.

1.1 Wireless Transceiver

The Wireless Transceiver comprises a 2.45GHz radio, a modem, a baseband controller and a security coprocessor. In addition, the radio also provides an output to control transmit-receive switching of external devices such as power amplifiers allowing applications that require increased transmit power to be realised very easily. Appendix B.4, describes a complete reference design including Printed Circuit Board (PCB) design and Bill Of Materials (BOM).

The security coprocessor provides hardware-based 128-bit AES-CCM* modes as specified by the IEEE802.15.4 2006 standard. Specifically this includes encryption and authentication covered by the MIC –32/ -64/ -128, ENC and ENC-MIC –32/ -64/ -128 modes of operation.

The transceiver elements (radio, modem and baseband) work together to provide IEEE802.15.4 Medium Access Control (MAC) under the control of a protocol stack. Applications incorporating IEEE802.15.4 functionality can be rapidly developed by combining user-developed application software with a protocol stack library.

1.2 RISC CPU and Memory

A 32-bit RISC CPU allows software to be run on chip, its processing power being shared between the IEEE802.15.4 MAC protocol, other higher layer protocols and the user application. The JN5148 has a unified memory architecture, code memory, data memory, peripheral devices and I/O ports are organised within the same linear address space. The device contains 128kbytes of ROM, 128kbytes of RAM and a 32-byte One Time Programmable (OTP) eFuse memory.

1.3 Peripherals

The following peripherals are available on chip:

- Master SPI port with five select outputs
- Two UARTs with support for hardware or software flow control
- Three programmable Timer/Counters – all three support Pulse Width Modulation (PWM) capability, two have capture/compare facility
- Two programmable Sleep Timers and a Tick Timer
- Two-wire serial interface (compatible with SMbus and I^{2}C) supporting master and slave operation
- Four-wire digital audio interface (compatible with I^{2}S)
- Slave SPI port for Intelligent peripheral mode (shared with digital I/O)
- Twenty-one digital I/O lines (multiplexed with peripherals such as timers and UARTs)
- Four channel, 12-bit, Analogue to Digital converter
- Two 12-bit Digital to Analogue converters
- Two programmable analogue comparators
- Internal temperature sensor and battery monitor
- Time Of Flight ranging engine
- Two low power pulse counters
- Random number generator
- Watchdog Timer and Voltage Brown-out
- Sample FIFO for digital audio interface or ADC/DAC
- JTAG hardware debug port

User applications access the peripherals using the Integrated Peripherals API. This allows applications to use a tested and easily understood view of the peripherals allowing rapid system development.

1.4 Block Diagram

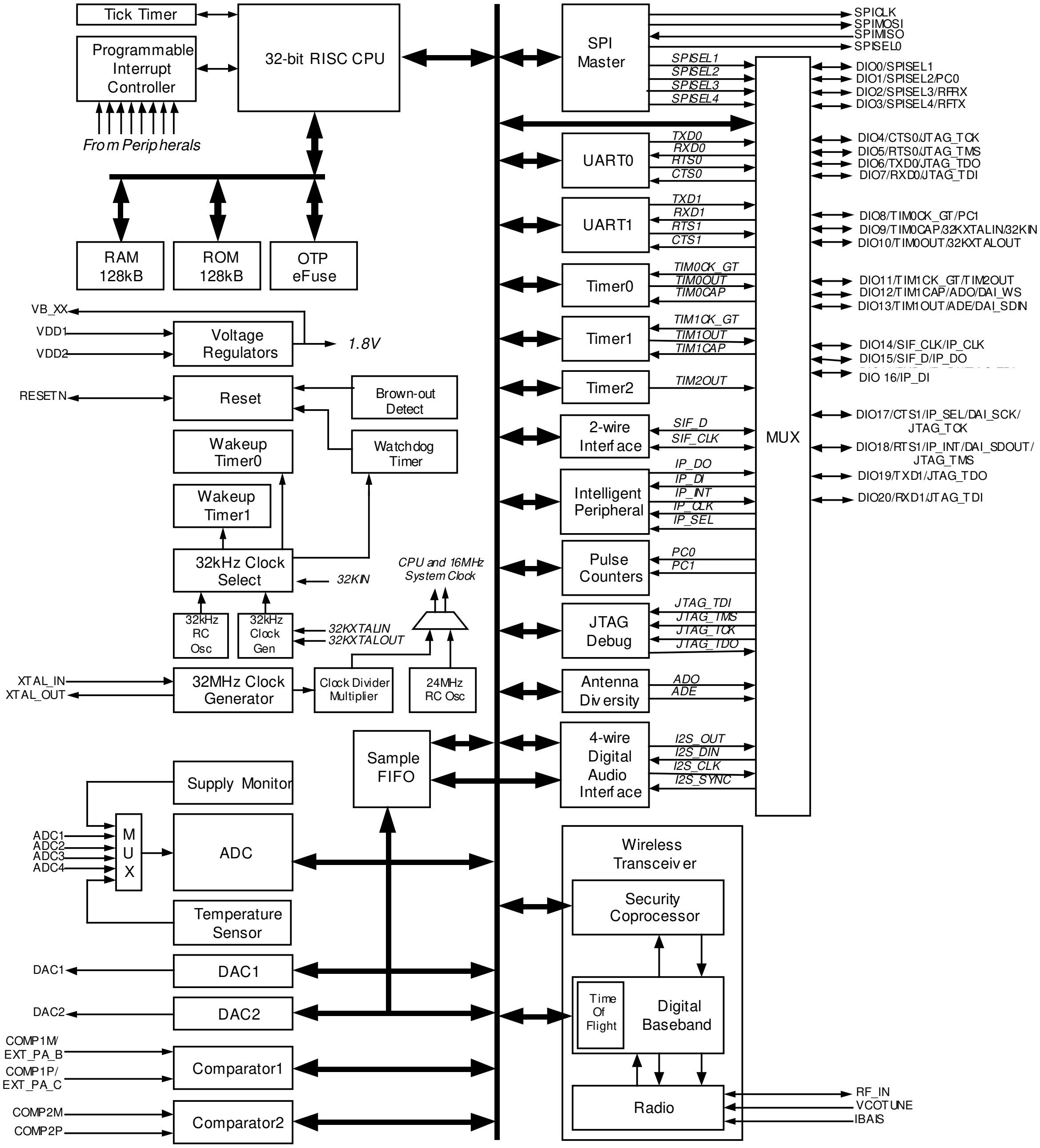


Figure 1: JN5148 Block Diagram

2 Pin Configurations

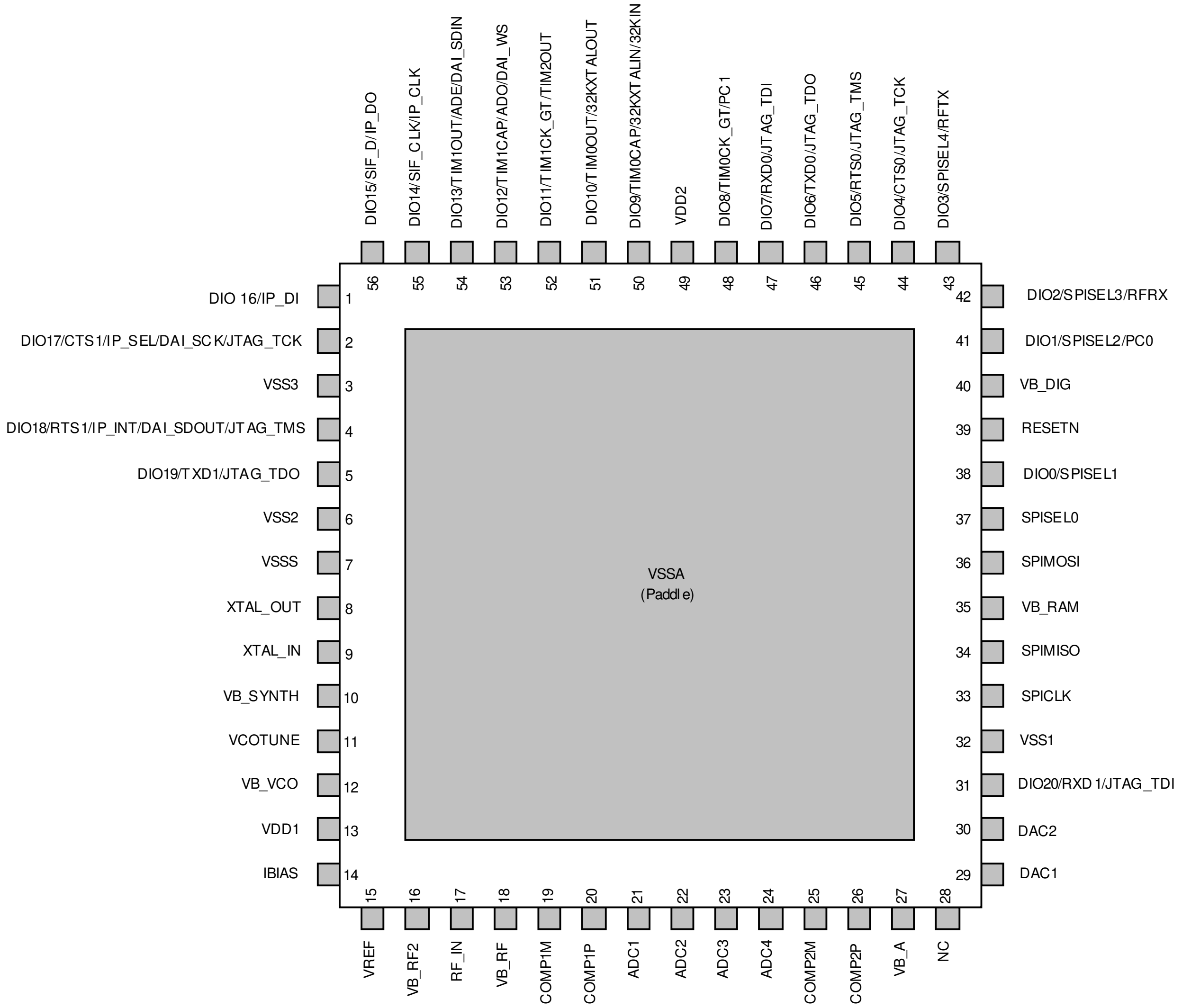


Figure 2: 56-pin QFN Configuration (top view)

Note: Please refer to Appendix B.4 JN5148 Module Reference Design for important applications information regarding the connection of the PADDLE to the PCB.

2.1 Pin Assignment

Pin No	Power supplies				Signal Type	Description
10, 12, 16, 18, 27, 35, 40	VB_SYNTH, VB_VCO, VB_RF2, VB_RF, VB_A, VB_RAM, VB_DIG				1.8V	Regulated supply voltage
13, 49	VDD1, VDD2				3.3V	Supplies: VDD1 for analogue, VDD2 for digital
32, 6, 3, 7, Paddle	VSS1, VSS2, VSS3, VSSS, VSSA				0V	Grounds (see appendix A.2 for paddle details)
28	NC					No connect
	General					
39	RESETN				CMOS	Reset input
8, 9	XTAL_OUT, XTAL_IN				1.8V	System crystal oscillator
	Radio					
11	VCOTUNE				1.8V	VCO tuning RC network
14	IBIAS				1.8V	Bias current control
17	RF_IN				1.8V	RF antenna
	Analogue Peripheral I/O					
21, 22, 23, 24	ADC1, ADC2, ADC3, ADC4				3.3V	ADC inputs
15	VREF				1.8V	Analogue peripheral reference voltage
29, 30	DAC1, DAC2				3.3V	DAC outputs
19, 20	COMP1M/EXT_PA_B, COMP1P/EXT_PA_C				3.3V	Comparator 1 inputs and external PA control
25, 26	COMP2M, COMP2P				3.3V	Comparator 2 inputs
	Digital Peripheral I/O					
	Primary	**Alternate Functions**				
33	SPICLK				CMOS	SPI Clock Output
36	SPIMOSI				CMOS	SPI Master Out Slave In Output
34	SPIMISO				CMOS	SPI Master In Slave Out Input
37	SPISEL0				CMOS	SPI Slave Select Output 0
38	DIO0	SPISEL1			CMOS	DIO0 or SPI Slave Select Output 1
41	DIO1	SPISEL2	PC0		CMOS	DIO1, SPI Slave Select Output 2 or Pulse Counter0 Input
42	DIO2	SPISEL3	RFRX		CMOS	DIO2, SPI Slave Select Output 3 or Radio Receive Control Output
43	DIO3	SPISEL4	RFTX		CMOS	DIO3, SPI Slave Select Output 4 or Radio Transmit Control Output
44	DIO4	CTS0	JTAG_TCK		CMOS	DIO4, UART 0 Clear To Send Input or JTAG CLK
45	DIO5	RTS0	JTAG_TMS		CMOS	DIO5, UART 0 Request To Send Output or JTAG Mode Select
46	DIO6	TXD0	JTAG_TDO		CMOS	DIO6, UART 0 Transmit Data Output or JTAG Data Output
47	DIO7	RXD0	JTAG_TDI		CMOS	DIO7, UART 0 Receive Data Input or JTAG Data Input
48	DIO8	TIM0CK_GT	PC1		CMOS	DIO8, Timer0 Clock/Gate Input or Pulse Counter1 Input
50	DIO9	TIM0CAP	32KXTALIN	32KIN	CMOS	DIO9, Timer0 Capture Input, 32K External Crystal Input or 32K Clock Input

Pin No	Digital Peripheral I/O					Signal Type	Description
	Primary	Alternate Functions					
51	DIO10	TIM0OUT	32KXTALOUT			CMOS	DIO10, Timer0 PWM Output or 32K External Crystal Output
52	DIO11	TIM1CK_GT	TIM2OUT			CMOS	DIO11, Timer1 Clock/Gate Input or Timer2 PWM Output
53	DIO12	TIM1CAP	ADO	DAI_WS		CMOS	DIO12, Timer1 Capture Input, Antenna Diversity or Digital Audio Word Select
54	DIO13	TIM1OUT	ADE	DAI_SDIN		CMOS	DIO13, Timer1 PWM Output, Antenna Diversity or Digital Audio Data Input
55	DIO14	SIF_CLK	IP_CLK			CMOS	DIO14, Serial Interface Clock or Intelligent Peripheral Clock Input
56	DIO15	SIF_D	IP_DO			CMOS	DIO15, Serial Interface Data or Intelligent Peripheral Data Out
1	DIO16	IP_DI				CMOS	DIO16 or Intelligent Peripheral Data In
2	DIO17	CTS1	IP_SEL	DAI_SCK	JTAG_TCK	CMOS	DIO17, UART 1 Clear To Send Input, Intelligent Peripheral Device Select Input or Digital Audio Clock or JTAG CLK
4	DIO18	RTS1	IP_INT	DAI_SDOUT	JTAG_TMS	CMOS	DIO18, UART 1 Request To Send Output, Intelligent Peripheral Interrupt Output or Digital Audio Data Output or JTAG Mode Select
5	DIO19	TXD1			JTAG_TDO	CMOS	DIO19 or UART 1 Transmit Data Output or JTAG Data Out
31	DIO 20	RXD1			JTAG_TDI	CMOS	DIO 20, UART 1 Receive Data Input or JTAG data In

The PCB schematic and layout rules detailed in Appendix B.4 must be followed. Failure to do so will likely result in the JN5148 failing to meet the performance specification detailed herein and worst case may result in device not functioning in the end application.

2.2 Pin Descriptions

2.2.1 Power Supplies

The device is powered from the VDD1 and VDD2 pins, each being decoupled with a 100nF ceramic capacitor. VDD1 is the power supply to the analogue circuitry; it should be decoupled to ground. VDD2 is the power supply for the digital circuitry; and should also be decoupled to ground. A 10uF tantalum capacitor is required. Decoupling pins for the internal 1.8V regulators are provided which require a 100nF capacitor located as close to the device as practical. VB_RF, VB_A and VB_SYNTH should be decoupled with an additional 47pF capacitor, while VB_RAM and VB_DIG require only 100nF. VB_RF and VB_RF2 should be connected together as close to the device as practical, and only require one 100nF capacitor and one 47pF capacitor. The pin VB_VCO requires a 10nF capacitor in parallel with a 47pF capacitor. Refer to B.4.1 for schematic diagram.

VSSA, VSSS, VSS1, VSS2, VSS3 are the ground pins.

Users are strongly discouraged from connecting their own circuits to the 1.8v regulated supply pins, as the regulators have been optimised to supply only enough current for the internal circuits.

2.2.2 Reset

RESETN is a bi-directional active low reset pin that is connected to a 40kΩ internal pull-up resistor. It may be pulled low by an external circuit, or can be driven low by the JN5148 if an internal reset is generated. Typically, it will be used to provide a system reset signal. Refer to section 6.2, External Reset, for more details.

2.2.3 32MHz Oscillator

A crystal is connected between XTALIN and XTALOUT to form the reference oscillator, which drives the system clock. A capacitor to analogue ground is required on each of these pins. Refer to section 5.1 16MHz System Clock for more details. The 32MHz reference frequency is divided down to 16MHz and this is used as the system clock throughout the device.

2.2.4 Radio

The radio is a single ended design, requiring a capacitor and just two inductors to match to 50Ω microstrip line to the RF_IN pin.

An external resistor (43kΩ) is required between IBIAS and analogue ground to set various bias currents and references within the radio.

2.2.5 Analogue Peripherals

Several of the analogue peripherals require a reference voltage to use as part of their operations. They can use either an internal reference voltage or an external reference connected to VREF. This voltage is referenced to analogue ground and the performance of the analogue peripherals is dependant on the quality of this reference.

There are four ADC inputs, two pairs of comparator inputs and two DAC outputs. The analogue I/O pins on the JN5148 can have signals applied up to 0.3v higher than VDD1. A schematic view of the analogue I/O cell is shown in Figure 3: Analogue I/O Cell

In reset and deep sleep, the analogue peripherals are all off and the DAC outputs are in a high impedance state.

In sleep, the ADC and DACs are off, with the DAC outputs in high impedance state. The comparators may optionally be used as a wakeup source.

Unused ADC and comparator inputs should be left unconnected.

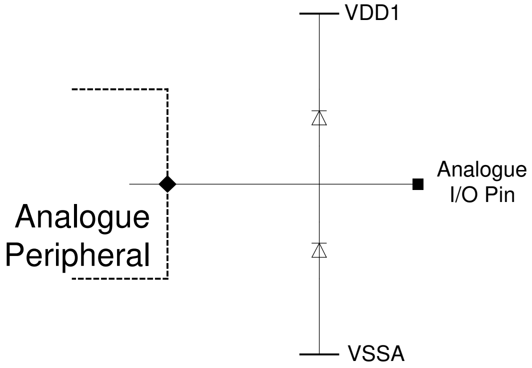


Figure 3: Analogue I/O Cell

2.2.6 Digital Input/Output

Digital I/O pins on the JN5148 can have signals applied up to 2V higher than VDD2 (with the exception of pins DIO9 and DIO10 that are 3V tolerant) and are therefore TTL-compatible with VDD2 > 3V. For other DC properties of these pins see section 22.2.3 I/O Characteristics.

When used in their primary function all Digital Input/Output pins are bi-directional and are connected to weak internal pull up resistors (40kΩ nominal) that can be disabled. When used in their secondary function (selected when the appropriate peripheral block is enabled through software library calls) then their direction is fixed by the function. The pull up resistor is enabled or disabled independently of the function and direction; the default state from reset is enabled.

A schematic view of the digital I/O cell is in Figure 4: DIO Pin Equivalent Schematic.

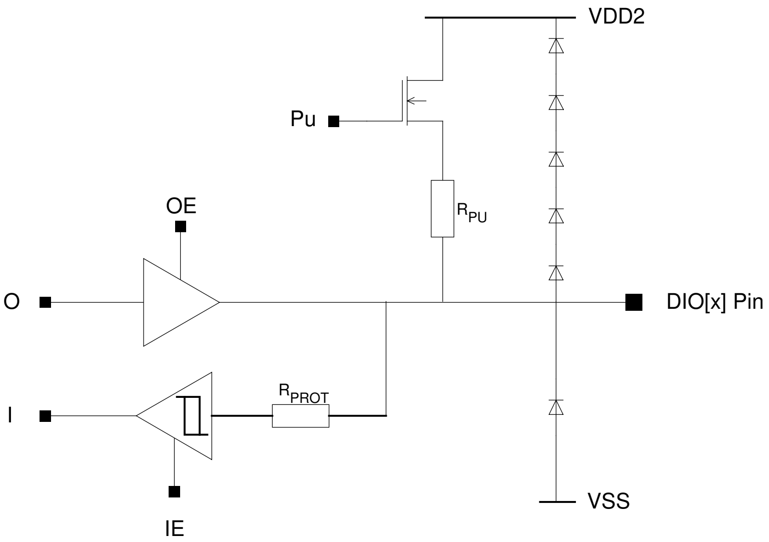


Figure 4: DIO Pin Equivalent Schematic

In reset, the digital peripherals are all off and the DIO pins are set as high-impedance inputs. During sleep and deep sleep, the DIO pins retain both their input/output state and output level that was set as sleep commences. If the DIO pins were enabled as inputs and the interrupts were enabled then these pins may be used to wake up the JN5148 from sleep.

3 CPU

The CPU of the JN5148 is a 32-bit load and store RISC processor. It has been architected for three key requirements:

- Low power consumption for battery powered applications
- High performance to implement a wireless protocol at the same time as complex applications
- Efficient coding of high-level languages such as C provided with the NXP Software Developer's Kit

It features a linear 32-bit logical address space with unified memory architecture, accessing both code and data in the same address space. Registers for peripheral units, such as the timers, UARTs and the baseband processor are also mapped into this space.

The CPU has access to a block of 15 32-bit General-Purpose (GP) registers together with a small number of special purpose registers which are used to store processor state and control interrupt handling. The contents of any GP register can be loaded from or stored to memory, while arithmetic and logical operations, shift and rotate operations, and signed and unsigned comparisons can be performed either between two registers and stored in a third, or between registers and a constant carried in the instruction. Operations between general or special-purpose registers execute in one cycle while those that access memory require a further cycle to allow the memory to respond.

The instruction set manipulates 8, 16 and 32-bit data; this means that programs can use objects of these sizes very efficiently. Manipulation of 32-bit quantities is particularly useful for protocols and high-end applications allowing algorithms to be implemented in fewer instructions than on smaller word-size processors, and to execute in fewer clock cycles. In addition, the CPU supports hardware Multiply that can be used to efficiently implement algorithms needed by Digital Signal Processing applications.

The instruction set is designed for the efficient implementation of high-level languages such as C. Access to fields in complex data structures is very efficient due to the provision of several addressing modes, together with the ability to be able to use any of the GP registers to contain the address of objects. Subroutine parameter passing is also made more efficient by using GP registers rather than pushing objects onto the stack. The recommended programming method for the JN5148 is by using C, which is supported by a software developer kit comprising a C compiler, linker and debugger.

The CPU architecture also contains features that make the processor suitable for embedded, real-time applications. In some applications, it may be necessary to use a real-time operating system to allow multiple tasks to run on the processor. To provide protection for device-wide resources being altered by one task and affecting another, the processor can run in either supervisor or user mode, the former allowing access to all processor registers, while the latter only allows the GP registers to be manipulated. Supervisor mode is entered on reset or interrupt; tasks starting up would normally run in user mode in a RTOS environment.

Embedded applications require efficient handling of external hardware events. When using JenOS, prioritised interrupts are supported, with 15 priority levels, and can be configured as required by the application.

To improve power consumption a number of power-saving modes are implemented in the JN5148, described more fully in section 21 - Power Management and Sleep Modes. One of these modes is the CPU doze mode; under software control, the processor can be shut down and on an interrupt it will wake up to service the request. Additionally, it is possible under software control, to set the speed of the CPU to 4, 8, 16 or 32MHz. This feature can be used to trade-off processing power against current consumption.

4 Memory Organisation

This section describes the different memories found within the JN5148. The device contains ROM, RAM, OTP eFuse memory, the wireless transceiver and peripherals all within the same linear address space.

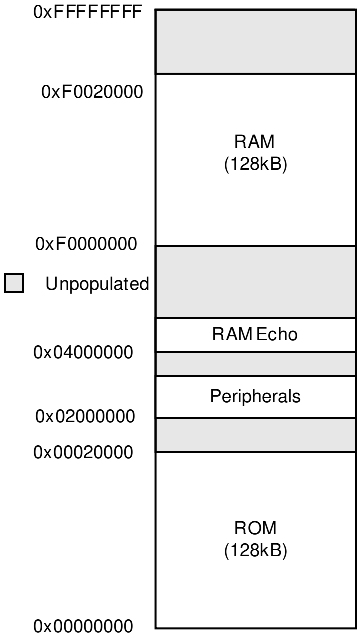


Figure 5: JN5148 Memory Map

4.1 ROM

The ROM is 128k bytes in size, and can be accessed by the processor in a single CPU clock cycle. The ROM contents include bootloader to allow external Flash memory contents to be bootloaded into RAM at runtime, a default interrupt vector table, an interrupt manager, IEEE802.15.4 MAC and APIs for interfacing on-chip peripherals. The operation of the boot loader is described in detail in Application Note [7]. The interrupt manager routes interrupt calls to the application's soft interrupt vector table contained within RAM. Section 7 contains further information regarding the handling of interrupts. ROM contents are shown in Figure 6.

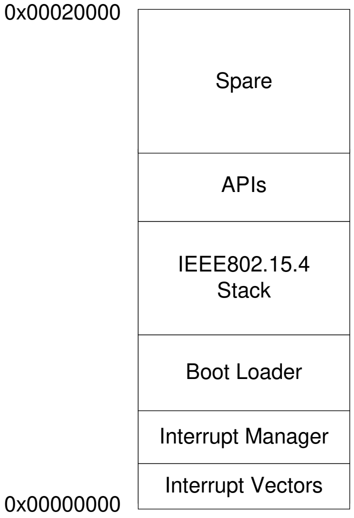


Figure 6: Typical ROM contents

4.2 RAM

The JN5148 contains 128kBytes of high speed RAM. It can be used for both code and data storage and is accessed by the CPU in a single clock cycle. At reset, a boot loader controls the loading of segments of code and data from an external memory connected to the SPI port, into RAM. Software can control the power supply to the RAM allowing the contents to be maintained during a sleep period when other parts of the device are un-powered. Typical RAM contents are shown in Figure 7.

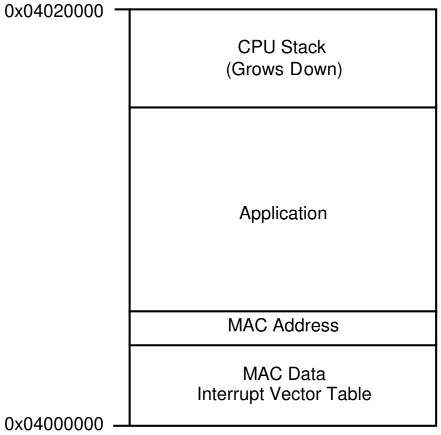


Figure 7: Typical RAM Contents

4.3 OTP eFuse Memory

The JN5148 contains a total of 32bytes of eFuse memory; this is a One Time Programmable (OTP) memory that can be used to support on chip 64-bit MAC ID and a 128-bit AES security key. A limited number of bits are available for customer use for storage of configuration information; configuration of these is made through use of software APIs.

For further information on how to program and use the eFuse memory, please contact technical support via the on-line tech-support system.

Alternatively, NXP can provide an eFuse programming service for customers that wish to use the eFuse but do not wish to undertake this for themselves. For further details of this service, please contact your local NXP sales office.

4.4 External Memory

An external memory with an SPI interface may be used to provide storage for program code and data for the device when external power is removed. The memory is connected to the SPI interface using select line SPISEL0; this select line is dedicated to the external memory interface and is not available for use with other external devices. See Figure 8 for connection details.

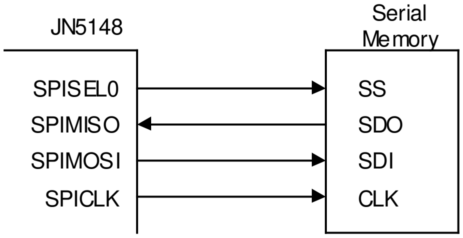


Figure 8: Connecting External Serial Memory

At reset, the contents of this memory are copied into RAM by the software boot loader. The Flash memory devices that are supported as standard through the JN5148 bootloader are given in Table 1. NXP recommends that where possible one of these devices should be selected.

Manufacturer	Device Number
SST (Silicon Storage Technology)	25VF010A (1Mbit device)
Numonyx	M25P10-A (1Mbit device), M25P40 (4Mbit device)

Table 1: Supported Flash Memories

Applications wishing to use an alternate Flash memory device should refer to application note [2] JN-AN-1038 Programming Flash devices not supported by the JN51xx ROM-based bootloader. This application note provides guidance on developing an interface to an alternate device.

4.4.1 External Memory Encryption

The contents of the external serial memory may be encrypted. The AES security processor combined with a user programmable 128-bit encryption key is used to encrypt the contents of the external memory. The encryption key is stored in eFuse.

When bootloading program code from external serial memory, the JN5148 automatically accesses the encryption key to execute the decryption process. User program code does not need to handle any of the decryption process; it is transparent.

With encryption enabled, the time taken to boot code from external flash is increased.

4.5 Peripherals

All peripherals have their registers mapped into the memory space. Access to these registers requires 3 clock cycles. Applications have access to the peripherals through the software libraries that present a high-level view of the peripheral’s functions through a series of dedicated software routines. These routines provide both a tested method for using the peripherals and allow bug-free application code to be developed more rapidly. For details, see the JN51xx Integrated Peripherals API User Guide (JN-UG-3066)[5].

4.6 Unused Memory Addresses

Any attempt to access an unpopulated memory area will result in a bus error exception (interrupt) being generated.

5 System Clocks

Two system clocks are used to provide timing references into the on-chip subsystems of the JN5148. A 16MHz clock, generated by a crystal-controlled 32MHz oscillator, is used by the transceiver, processor, memory and digital and analogue peripherals. A 32kHz clock is used by the sleep timer and during the startup phase of the chip.

5.1 16MHz System Clock

The 16MHz system clock is used by the digital and analogue peripherals and the transceiver. A scaled version (4,8,16 or 32MHz) of this clock is also used by the processor and memories. For most operations it is necessary to source this clock from the 32MHz oscillator.

Crystal oscillators are generally slow to start. Hence to provide a faster start-up following a sleep cycle a fast RC oscillator is provided that can be used as the source for the 16MHz system clock. The oscillator starts very quickly and is typically 24MHz causing the system clock to run at 12MHz. Using a clock of this speed scales down the speed of the processor and any peripherals in use. For the SPI interface this causes no functional issues as the generated SPI clock is slightly slower and is used to clock the external SPI slave. Use of the radio is not possible when using the 24MHz RC oscillator. Additionally, timers and UARTs should not be used as the exact frequency will not be known.

The JN5148 device can be configured to wake up from sleep using the fast RC oscillator and automatically switch over to use the 32MHz xtal as the clock source, when it has started up. This could allow application code to be downloaded from the flash before the xtal is ready, typically improving start-up time by 550usec. Alternatively, the switch over can be controlled by software, or the system could always use the 32MHz oscillator as the clock source.

5.1.1 32MHz Oscillator

The JN5148 contains the necessary on chip components to build a 32MHz reference oscillator with the addition of an external crystal resonator and two tuning capacitors. The schematic of these components are shown in Figure 9. The two capacitors, C1 and C2, should typically be 15pF and use a COG dielectric. Due to the small size of these capacitors, it is important to keep the traces to the external components as short as possible. The on chip transconductance amplifier is compensated for temperature variation, and is self-biasing by means of the internal resistor R1. The electrical specification of the oscillator can be found in section 22.3.13. Please refer to Appendix B for development support with the crystal oscillator circuit.

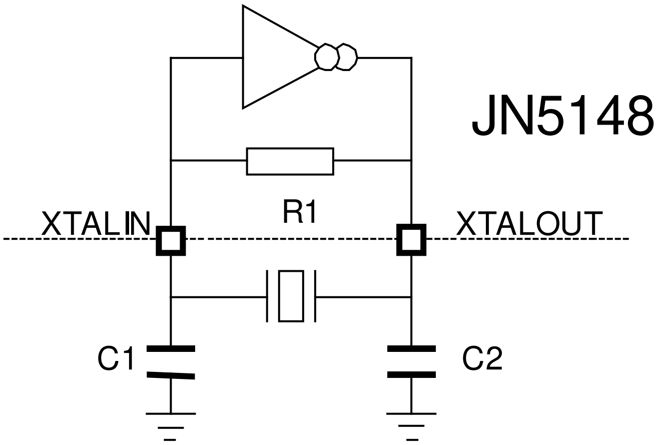


Figure 9: 32MHz Crystal Oscillator Connections

5.1.2 24MHz RC Oscillator

An on-chip 24MHz RC oscillator is provided. No external components are required for this oscillator. The electrical specification of the oscillator can be found in section 22.3.14.

5.2 32kHz System Clock

The 32kHz system clock is used for timing the length of a sleep period (see section 21 Power Management and Sleep Modes) and also to generate the system clock used internally during reset. The clock can be selected from one of three sources through the application software:

- 32kHz RC Oscillator
- 32kHz Crystal Oscillator
- 32kHz External Clock

Upon a chip reset or power-up the JN5148 defaults to using the internal 32kHz RC Oscillator. If another clock source is selected then it will remain in use for all 32kHz timing until a chip reset is performed.

5.2.1 32kHz RC Oscillator

The internal 32kHz RC oscillator requires no external components. The internal timing components of the oscillator have a wide tolerance due to manufacturing process variation and so the oscillator runs nominally at 32kHz ±30%. To make this useful as a timing source for accurate wakeup from sleep, a frequency calibration factor derived from the more accurate 16MHz clock may be applied. The calibration factor is derived through software, details can be found in section 11.3.1. For detailed electrical specifications, see section 22.3.11.

5.2.2 32kHz Crystal Oscillator

In order to obtain more accurate sleep periods, the JN5148 contains the necessary on-chip components to build a 32kHz oscillator with the addition of an external 32.768kHz crystal and two tuning capacitors. The crystal should be connected between 32KXTALIN and 32KXTALOUT (DIO9 and DIO10), with two equal capacitors to ground, one on each pin. Due to the small size of the capacitors, it is important to keep the traces to the external components as short as possible.

The electrical specification of the oscillator can be found in section 22.3.12. The oscillator cell is flexible and can operate with a range of commonly available 32.768kHz crystals with load capacitances from 6 to 12.5pF. However, the maximum ESR of the crystal and the supply current are both functions of the actual crystal used, see appendix B.1 for more details.

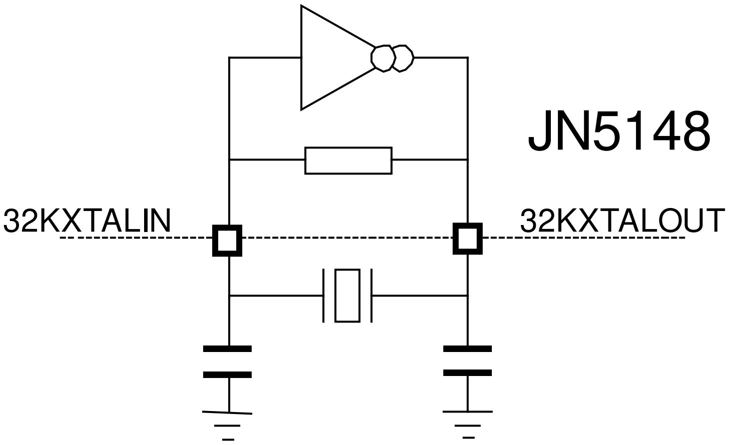


Figure 10: 32kHz crystal oscillator connections

5.2.3 32kHz External Clock

An externally supplied 32kHz reference clock on the 32KIN input (DIO9) may be provided to the JN5148. This would allow the 32kHz system clock to be sourced from a very stable external oscillator module, allowing more accurate sleep cycle timings compared to the internal RC oscillator. (See section 22.2.3 I/O Characteristics, DIO9 is a 3V tolerant input)

6 Reset

A system reset initialises the device to a pre-defined state and forces the CPU to start program execution from the reset vector. The reset process that the JN5148 goes through is as follows.

When power is applied, the 32kHz RC oscillator starts up and stabilises, which takes approximately 100µsec. At this point, the 32MHz crystal oscillator is enabled and power is applied to the processor and peripheral logic. The logic blocks are held in reset until the 32MHz crystal oscillator stabilises, typically this takes 0.75ms. Then the internal reset is removed from the CPU and peripheral logic and the CPU starts to run code beginning at the reset vector, consisting of initialisation code and the resident boot loader. [7] Section 22.3.1 provides detailed electrical data and timing.

The JN5148 has five sources of reset:

- Internal Power-on Reset
- External Reset
- Software Reset
- Watchdog timer
- Brown-out detect

Note: When the device exits a reset condition, device operating parameters (voltage, frequency, temperature, etc.) must be met to ensure operation. If these conditions are not met, then the device must be held in reset until the operating conditions are met. (See section 22.3)

6.1 Internal Power-on Reset

For the majority of applications the internal power-on reset is capable of generating the required reset signal. When power is applied to the device, the power-on reset circuit monitors the rise of the VDD supply. When the VDD reaches the specified threshold, the reset signal is generated and can be observed as a rising edge on the RESETN pin. This signal is held internally until the power supply and oscillator stabilisation time has elapsed, when the internal reset signal is then removed and the CPU is allowed to run.

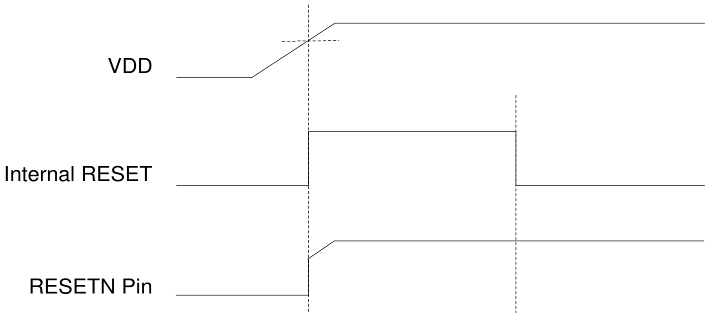


Figure 11: Internal Power-on Reset

When the supply drops below the power on reset ‘falling’ threshold, it will re-trigger the reset. Use of the external reset circuit show in Figure 12 is suggested.

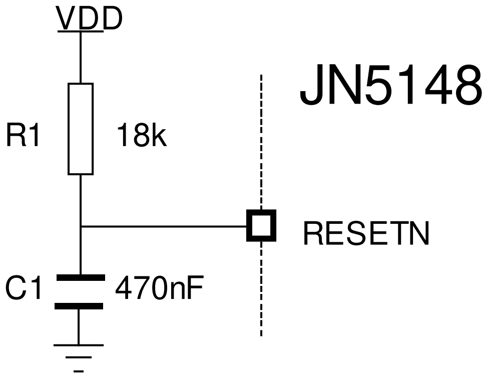


Figure 12: External Reset Generation

The external resistor and capacitor provide a simple reset operation when connected to the RESETN pin.

6.2 External Reset

An external reset is generated by a low level on the RESETN pin. Reset pulses longer than the minimum pulse width will generate a reset during active or sleep modes. Shorter pulses are not guaranteed to generate a reset. The JN5148 is held in reset while the RESETN pin is low. When the applied signal reaches the Reset Threshold Voltage (V_{RST}) on its positive edge, the internal reset process starts.

Multiple devices may connect to the RESETN pin in an open-collector mode. The JN5148 has an internal pull-up resistor connect to the RESETN pin. The pin is an input for an external reset, an output during the power-on reset and may optionally be an output during a software reset. No devices should drive the RESETN pin high.

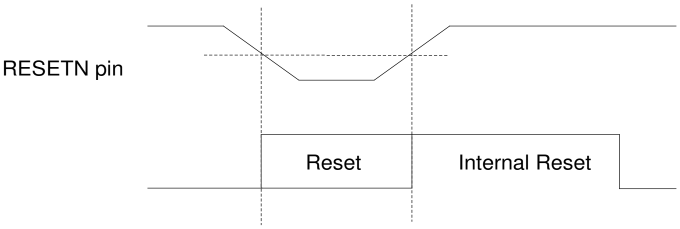


Figure 13: External Reset

6.3 Software Reset

A system reset can be triggered at any time through software control, causing a full chip reset and invalidating the RAM contents. For example this can be executed within a user's application upon detection of a system failure. When performing the reset, the RESETN pin is driven low for 1µsec; depending on the external components this may or may not be visible on the pin.

In addition, the RESETN line can be driven low by the JN5148 to provide a reset to other devices in the system (e.g. external sensors) without resetting itself. When the RESETN line is not driven it will pull back high through either the internal pull-up resistor or any external circuitry. It is essential to ensure that the RESETN line pulls back high within 100µsec after the JN5148 stops driving the line; otherwise a system reset will occur. Due to this, careful consideration should be taken of any capacitance on this line. For instance, the RC values recommended in section 6.1 may need to be replaced with a suitable reset IC

6.4 Brown-out Detect

An internal brown-out detect module is used to monitor the supply voltage to the JN5148; this can be used whilst the device is awake or is in CPU doze mode. Dips in the supply voltage below a variable threshold can be detected and can be used to cause the JN5148 to perform a chip reset. Equally, dips in the supply voltage can be detected and used to cause an interrupt to the processor, when the voltage either drops below the threshold or rises above it.

The brown-out detect is enabled by default from power-up and can extend the reset during power-up. This will keep the CPU in reset until the voltage exceeds the brown-out threshold voltage. The threshold voltage is configurable to 2.0V, 2.3V, 2.7V and 3.0V and is controllable by software. From power-up the threshold is set by eFuse settings and the default chip configuration is for the 2.3V threshold. It is recommended that the threshold is set so that, as a minimum, the chip is held in reset until the voltage reaches the level required by the external memory device on the SPI interface.

6.5 Watchdog Timer

A watchdog timer is provided to guard against software lockups. It operates by counting cycles of the 32kHz system clock. A pre-scaler is provided to allow the expiry period to be set between typically 8ms and 16.4 seconds. Failure to restart the watchdog timer within the pre-configured timer period will cause a chip reset to be performed. A status bit is set if the watchdog was triggered so that the software can differentiate watchdog initiated resets from other resets, and can perform any required recovery once it restarts. If the source of the 32kHz system clock is the 32kHz RC oscillator then the watchdog expiry periods are subject to the variation in period of the RC oscillator.

After power up, reset, start from deep sleep or start from sleep, the watchdog is always enabled with the largest timeout period and will commence counting as if it had just been restarted. Under software control the watchdog can be disabled. If it is enabled, the user must regularly restart the watchdog timer to stop it from expiring and causing a reset. The watchdog runs continuously, even during doze, however the watchdog does not operate during sleep or deep sleep, or when the hardware debugger has taken control of the CPU. It will recommence automatically if enabled once the debugger un-stalls the CPU.

7 Interrupt System

The interrupt system on the JN5148 is a hardware-vectored interrupt system. The JN5148 provides several interrupt sources, some associated with CPU operations (CPU exceptions) and others which are used by hardware in the device. When an interrupt occurs, the CPU stops executing the current program and loads its program counter with a fixed hardware address specific to that interrupt. The interrupt handler or interrupt service routine is stored at this location and is run on the next CPU cycle. Execution of interrupt service routines is always performed in supervisor mode. Interrupt sources and their vector locations are listed in Table 2 below:

Interrupt Source	Vector Location	Interrupt Definition
Bus error	0x08	Typically cause by an attempt to access an invalid address or a disabled peripheral
Tick timer	0x0e	Tick timer interrupt asserted
Alignment error	0x14	Load/store address to non-naturally-aligned location
Illegal instruction	0x1a	Attempt to execute an unrecognised instruction
Hardware interrupt	0x20	interrupt asserted
System call	0x26	System call initiated by b.sys instruction
Trap	0x2c	caused by the b.trap instruction or the debug unit
Reset	0x38	Caused by software or hardware reset.
Stack Overflow	0x3e	Stack overflow

Table 2: Interrupt Vectors

7.1 System Calls

The b.trap and b.sys instructions allow processor exceptions to be generated by software.

A system call exception will be generated when the b.sys instruction is executed. This exception can, for example, be used to enable a task to switch the processor into supervisor mode when a real time operating system is in use. (See section 3 for further details.)

The b.trap instruction is commonly used for trapping errors and for debugging.

7.2 Processor Exceptions

7.2.1 Bus Error

A bus error exception is generated when software attempts to access a memory address that does not exist, or is not populated with memory or peripheral registers or when writing to ROM.

7.2.2 Alignment

Alignment exceptions are generated when software attempts to access objects that are not aligned to natural word boundaries. 16-bit objects must be stored on even byte boundaries, while 32-bit objects must be stored on quad byte boundaries. For instance, attempting to read a 16-bit object from address 0xFFF1 will trigger an alignment exception as will a read of a 32-bit object from 0xFFF1, 0xFFF2 or 0xFFF3. Examples of legal 32-bit object addresses are 0xFFF0, 0xFFF4, 0xFFF8 etc.

7.2.3 Illegal Instruction

If the CPU reads an unrecognised instruction from memory as part of its instruction fetch, it will cause an illegal instruction exception.

7.2.4 Stack Overflow

When enabled, a stack overflow exception occurs if the stack pointer reaches a programmable location.

7.3 Hardware Interrupts

Hardware interrupts generated from the transceiver, analogue or digital peripherals and DIO pins are individually masked using the Programmable Interrupt Controller (PIC). Management of interrupts is provided in the peripherals library [5]. For details of the interrupts generated from each peripheral see the respective section in this datasheet.

Interrupts can be used to wake the JN5148 from sleep. The peripherals, baseband controller, security coprocessor and PIC are powered down during sleep but the DIO interrupts and optionally the pulse counters, wake-up timers and analogue comparator interrupts remain powered to bring the JN5148 out of sleep.

Prioritised external interrupt handling (i.e., interrupts from hardware peripherals) is provided to enable an application to control an events priority to provide for deterministic program execution.

The priority Interrupt controller provides 15 levels of prioritised interrupts. The priority level of all interrupts can be set, with value 0 being used to indicate that the source can never produce an external interrupt, 1 for the lowest priority source(s) and 15 for the highest priority source(s). Note that multiple interrupt sources can be assigned the same priority level if desired.

If while processing an interrupt, a new event occurs at the same or lower priority level, a new external interrupt will not be triggered. However, if a new higher priority event occurs, the external interrupt will again be asserted, interrupting the current interrupt service routine.

Once the interrupt service routine is complete, lower priority events can be serviced.