iND80230, “Herzog”

indie low power, low current 400MHz Transceiver

8/31/17 **Preliminary Data Sheet**

1.0 Revision History

Table 1 – Revision History			
Rev #	**Date**	**Action**	**By**
0.1	Sept 22nd 2014	Initial version (based on Krankl Product Specification)	GKo
0.2	July 8th 2015	Updated Pinout and Feature sets	DDK
0.3	Aug 7th, 2015	indie version and formatting	CR
1.0	Jan 15th, 2016	Clean up before release	CR

2.0 Table of Contents

3.0 List of Figures

4.0 List of Tables

5.0 General Description

This ASIC has an integrated super-heterodyne ISM band 300MHz to 415MHz ASK transceiver. The oscillator frequency for the RF is generated by fractional-N type frequency synthesizers from a 30MHz or 25MHz crystal oscillator.

It has the following features:

CPU Architecture:

- ARM Cortex-M0 processor running at crystal freuquency or 10 MHz (Internal RC)
- System Tick Timer (SysTick – 24 bits, interruptible) and 3 additional 32-bit timers
- Built-in Nested Vectored Interrupt Controller (NVIC)
- Serial Wire Debugger
- Programmable Watch-Dog Timer

Memory:

- 160kByte of Flash Program Memory, 8kByte of SRAM

Peripherals

- Super-heterodyne ISM band 300MHz to 415MHz, ASK/OOK receiver, with up to -110dBm of sensitivity
- 300MHz to 415MHz ISM band transmitter supporting ASK/OOK modulation mode, with a maximum output power of +13dBm
- Integrated fractional-N phase locked loop referenced to 30/25MHz crystal oscillator
- One (1) general purpose ADC (8-bit), total of 19 channels with selectable input references.
- One (1) 12bit pulse width modulator with 8 channels (PWM)
- 19 low voltage (3.3V nominal) general purpose I/O ports muxed to support:
    - Reduced media independent PHY interface
    - PIR support (6 pins)
    - Two (2) LED interfaces
    - UART / SPI interface, dedicated non-overlapping pins
- Power Management
- Wake-up Timer
- Herzog contains different oscillators which may be used to generate a time base
    - One (1) 30MHz high accuracy crystal oscillator used as a reference for the RF
    - One (1) 10MHz, 5% accurate RC oscillator, for current saving operation
    - One (1) 32.768kHz real time clock (RTC)
    - One (1) 10kHz, low power oscillator permanently activated for house keeping

6.0 Pin Description and Package

6.1 Package Overview

Herzog package is a QFN 5x5, 32 pins.

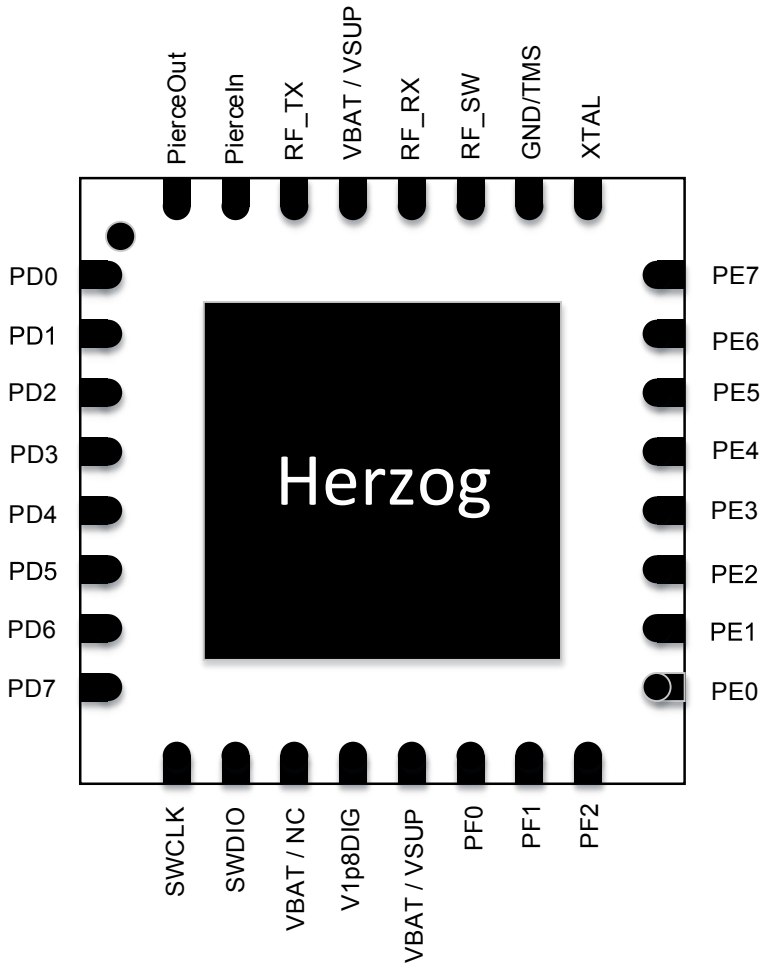
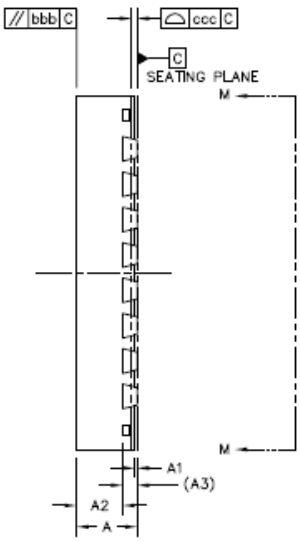


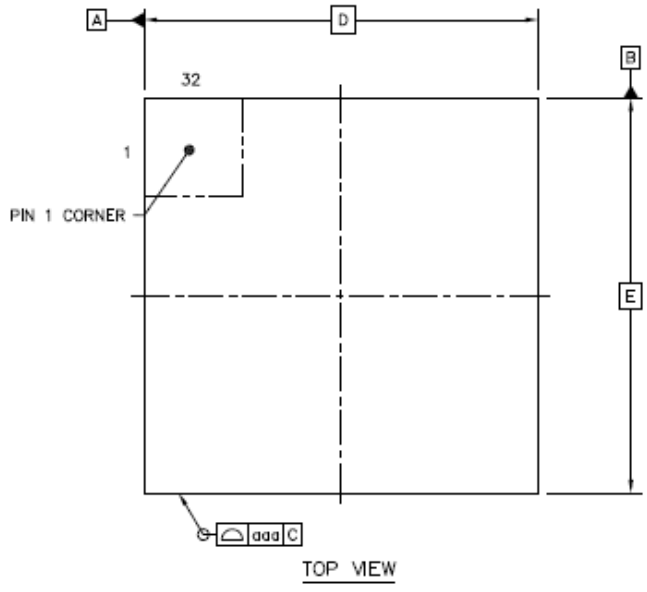
Figure 1 – Herzog Pin Out Diagram (Top View)

6.2 Package Dimensions

The dimensions of the package are defined in the following table and drawings



		SYMBOL	MIN	NOM	MAX
TOTAL THICKNESS		A	0.8	0.85	0.9
STAND OFF		A1	0	0.035	0.05
MOLD THICKNESS		A2	---	0.65	0.67
L/F THICKNESS		A3		0.203 REF	
LEAD WIDTH		b	0.2	0.25	0.3
BODY SIZE	X	D		5 BSC	
	Y	E		5 BSC	
LEAD PITCH		e		0.5 BSC	
EP SIZE	X	J	3	3.1	3.2
	Y	K	3	3.1	3.2
LEAD LENGTH		L	0.35	0.4	0.45
PACKAGE EDGE TOLERANCE		aaa		0.1	
MOLD FLATNESS		bbb		0.1	
COPLANARITY		ccc		0.08	
LEAD OFFSET		ddd		0.1	
EXPOSED PAD OFFSET		eee		0.1	



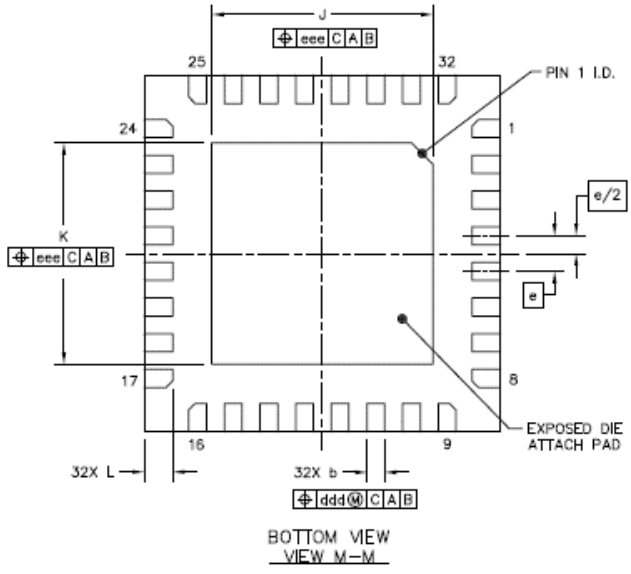


Figure 2 – QFN (5mm x 5mm) 32-pin Package Dimensions

6.3 Pin Description

Table 2 – Absolute Maximum Ratings

Pin #	Name	Type	Voltage	Description
1	PD0	Digital I/O	3.3V	General Purpose I/O
2	PD1	Digital I/O	3.3V	tpd
3	PD2	Digital I/O	3.3V	tpd
4	PD3	Digital I/O	3.3V	tpd
5	PD4	Digital I/O	3.3V	tpd
6	PD5	Digital I/O	3.3V	tpd
9	PD6	Digital I/O	3.3V	tpd
10	PD7	Digital I/O	3.3V	tpd
9	SWCLK	Digital I/O	3.3V	Input ARM controller interface Clock
10	SWDIO	Digital I/O	3.3V	Input ARM controller interface Data
11	VBAT / NC	Supply	2.6V	FLASH Supply Voltage
12	V1p8DIG	Supply	1.8V	Voltage regulator output MCU Supply Voltage
13	VBAT / VSUP	Supply	3.3V	ASIC Supply Voltage
14	PF0	Digital I/O	3.3V	tpd
15	PF1	Digital I/O	3.3V	tpd
16	PF2	Digital I/O	3.3V	tpd
17	PE0	Digital I/O	3.3V	tpd
18	PE1	Digital I/O	3.3V	tpd
19	PE2	Digital I/O	3.3V	tpd
20	PE3	Digital I/O	3.3V	tpd

Table 2 – Absolute Maximum Ratings				
Pin #	**Name**	**Type**	**Voltage**	**Description**
21	PE4	Digital I/O	3.3V	tpd
22	PE5	Digital I/O	3.3V	tpd
23	PE6	Digital I/O	3.3V	tpd
24	PE7	Digital I/O	3.3V	tpd
30	XTAL	Analog I/O	3.3V	I/O Crystal Oscillator
26	GND/TMS	Digital I/O	3.3V	Input test mode state
27	RF-SW	Analog I/O	3.3V	TRx filter switch
28	RF-RX	Analog I/O	3.3V	I/O Rx path 433MHz
29	VBAT / VSUP	Supply	3.3V	ASIC Supply Voltage
30	RF-TX	Analog I/O	3.3V	I/O Tx path 433MHz
31	PierceIn	Analog I/O	3.3V	Pierce Oscillator In
32	PierceOut	Analog I/O	3.3V	Pierce Oscillator Out

7.0 Absolut Maximum Ratings

Table 3 – Absolute Maximum Ratings					
Rating	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
Supply voltage range	Short duration, no long term damage	-0.3		3.6	V
3.3V GPIO	configured as input, no damage	-0.3		V3p3DIG+0.3	V
3V Analog IO	XTAL and RF, no damage	-0.3		V3p3AN+0.3	V
Storage Temp.		-55		+125	°C

8.0 ESD/Transient Robustness

Table 4 – ESD Ratings				
Rating	**Conditions**	**Min.**	**Max.**	**Unit**
HBM (all pins)	any pin to any other pin or ground, loose part	-2	2	kV
CDM (all pins)	configured as input, no damage	-200	200	V
MM (all pins)	configured as input, no damage	-150	150	V

9.0 ESD/Transient Robustness

Table 5 – Recommended Operating Conditions					
Name	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
Vsupply voltage		2.5	3.3	3.6	V
Operating Temp.		-20	30	65	°C

10.0 Register Convention

Several registers will be defined and explained throughout this document. The general format of the description of the registers is as follows:

REGNAME: Name of Register							
REGNAME		0x00000000			0x00		
R/W	R/W	R/W	R/W	R/W	Reserved	Reserved	Reserved
Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name	Bit_Name
MSB							LSB
Bit7 **Name**: Description 0 = Do function 1 = Do NOT do function							

Where R/W is the read and write permissions of the specific bit. An example:

RF DCDTIME: RF Decoder Time							
RF_DCDTIME		0x50011004			0x3614		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MAX_TE1	MAX_TE0	MIN_GB5	MIN_GB4	MIN_GB3	MIN_GB2	MIN_GB1	MIN_GB0
MIN_TE3	MIN_TE2	MIN_TE1	MIN_TE0	MAX_TE5	MAX_TE4	MAX_TE3	MAX_TE2
MSB							LSB

The name of this register is RF_DCDTIME (RF Decoder Time). It is a 16-bit register, located at address 0x50011004 and 0x50011005. The first row of data (MAX_TE[1:0], MIN_GB[5:0]) corresponds to address 0x50011004 with default value of 0x14 and the second row of data (MIN_TE[3:0], MAX_TE[5:2]) corresponds to address 0x50011005 with default value of 0x36.

11.0 Device Overview

Figure 3 depicts a high-level block diagram of the Herzog device. The application subsystems can be grouped into several types: RF receiver, RF transmitter, power management, general purpose I/O ports, clock generation, power on reset, brown out reset and micro controller unit.

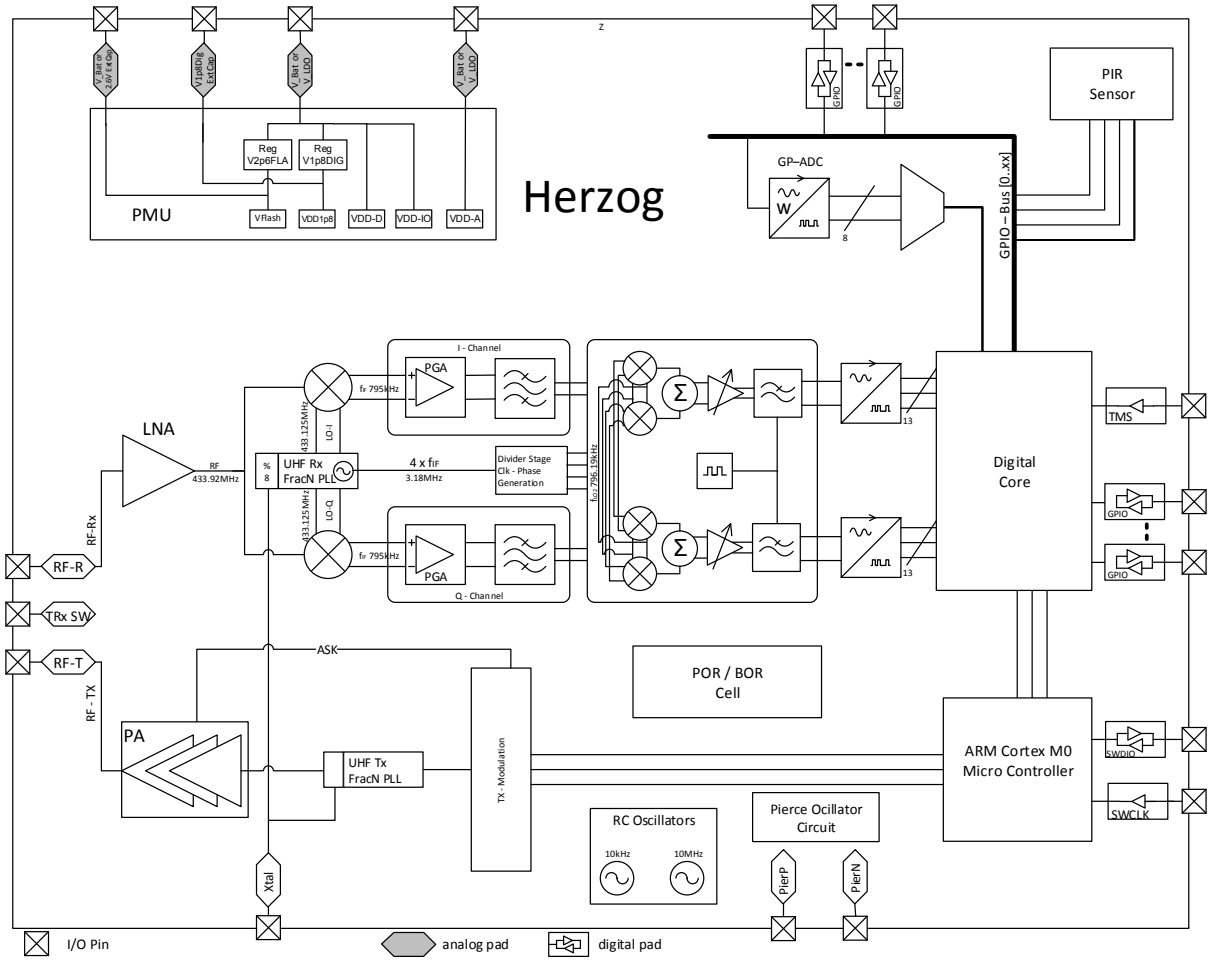


Figure 3 – Functional Block Diagram

The microcontroller serial interface consists of clock (SCK), data (SDIO) and interrupt (IOC) signals. The interface operates at a typical 3.3V CMOS level and is intended to handle communication between a microcontroller and Herzog.

The data is bidirectional, and communication can start either from the microcontroller or Herzog. Communication from the micro starts with the clock driven by the micro for either read or write. Communication from Herzog starts with an interrupt on the IOC pin followed by the micro clocking and sending a read instruction on the data (SDIO) pin.

The clock generation subsystem consists of a crystal oscillator. The crystal oscillator will be configured to operate with the crystal connected between a single pin and ground. This pin can be used alternately as an external clock source just by driving it with digital level.

The power management portion consists of a bandgap voltage reference and two voltage regulators. The voltage reference is used to provide the reference voltage for the voltage regulators, as well as for other circuits internal to Herzog, such as, for example, the A/D converters.

Herzog supports two different supply concepts, the IC can be connected either to a battery with a maximum output voltage of 3.3V or Herzog connects to an external voltage regulator ranging from 2.6V to 3.6V. In both modes the analog supply voltage (VDD-A) and the digital supply voltage (VDD-D) as well as all IOs can be directly connected the regulator (V_LDO) or the battery (V_Bat).

Integrated regulators (LDOs) provide power for the micro controller (V1p8Dig) and the FLASH memory (V2p6Dig). Both voltages are available at a pin to allow placement of bypass capacitors. For battery operation at a maximum voltage of 3.3V the FLASH memory supply can be directly connected (regulator is bypassed) to the battery.

Herzog supports a deep sleep power mode, which is essential for battery operation in order to minimize current consumption and extent battery life time. Wake-up is imitated by pin interrupt.

The RF receiver is configured to receive and decode ASK (OOK) modulated messages with nominal carrier frequency of 433.92MHz ±100KHz. Messages are fully decoded and buffered without microcontroller intervention. The transmitter is capable to transmit ASK coded signal at 433.92MHz and 13dBm.

All analog and digital IO pins as well as the microcontroller serial interface are 3.3V rated.

12.0 Micro Controller Subsystem

Herzog device includes an embedded microcontroller subsystem, which is based on the ARM Cortex M0 core. It includes a program flash memory of 160kBytes, and an SRAM of 8kBytes. Three 32-bit timers are embedded, plus a dedicated watchdog timer. Additionally, it includes a **N**ested **V**ector **I**nterrupt **C**ontroller (NVIC) to scheduled hardware interrupts, and a **W**akeup **I**nterrupt **C**ontroller (WIC), which enable the control of the various power modes. Further information can be obtained in the AyDeeKay document *<<AyDeeKay_Core_160_8.pdf>>.*

12.1 Memory Map

Table 6 – System Memory Map

Address	Memory	Description	Reference
0x00000000 - 0x00027FFF	Flash	160kByte Flash Memory	N/A
0x00028000 - 0x0003FFFF	N/A	Reserved	N/A
0x00040000 - 0x000400FF	Flash	306Byte 1st NVR Sector	N/A
0x00040100 - 0x000401FF	Flash	306Byte 2nd NVR Sector	N/A
0x00040200 - 0x1FFFFFFF	N/A	Reserved	N/A
0x20000000 - 0x20001FFF	SRAM	8kByte SRAM	N/A
0x20002000 - 0x4FFFFFFF	N/A	Reserved	N/A
0x50000000 - 0x5000007F	Peripheral	128Byte peripheral fast access	N/A
0x50000080 - 0x50000085	Peripheral	6Byte Block Transfer control	N/A
0x50000086 - 0x5000FFFF	N/A	Reserved	N/A
0x50010000 - 0x5001FFFF	Peripheral	64kByte peripheral slow access	N/A
0x50020000 - 0x5002001F	Peripheral	32Byte timer control	N/A
0x50020020 - 0x50020047	Flash	40Byte Flash program/erase control	N/A
0x50020048 - 0xDFFFFFFF	N/A	Reserved	N/A
0xE0000000 - 0xE00FFFFF	Private peripheral bus	ARM peripherals	N/A

Table 6 – System Memory Map			
0xE0100000 - 0xEFFFFFFF	N/A	Reserved	N/A
0xF0000000 - 0xF0001FFF	System ROM tables	ARM core IDs	N/A
0xF0002000 - 0xFFFFFFFF	N/A	Reserved	N/A

Table 7 – Peripheral Fast Access Memory Map			
Address	Peripheral	Description	Reference
0x50000000 - 0x50000005	PMU	Power management unit control	
0x50000008 - 0x5000000F	ADC	ADC control	
0x50000010 - 0x50000017	UART	UART control	
0x5000001C - 0x5000001F	SPI	SPI control	
0x50000020 – 0x50000031	RF RX	RF Receiver control	
0x50000032 – 0x5000003F	RF TX	RF Transmitter control	
0x50000040 - 0x5000004F	PWM	Pulse width modulator control	
0x50000050 - 0x5000005F	DTT	Digital Triac Timer control	
0x50000060 - 0x5000007F	GPIO	GPIO Byte control	

Table 8 – Peripheral Slow Access Memory Map			
Address	Peripheral	Description	Reference
0x50010000 - 0x500104FF	GPIO	GPIO bit control	
0x50000008 - 0x5000000C	ADC I	ADC and miscellaneous control	
0x50010004 - 0x50010007	TX PLL	RF TX Synthesizer Setting	
0x50018004 - 0x50018007	RX PLL	RF RX Synthesizer Setting	
0x50018008 - 0x5001800B		NOT allocated for Herzog (used to be	

Table 8 – Peripheral Slow Access Memory Map			
		ADC II)	
0x50000000	CLK Source	RC Oscillator and Xtal Setting	
0x50000060 - 0x5000006A		Port Control	
		PIR Interface	

13.0 Timers (0,1, and 2)

Herzog implements three identical timers: Timer0, Timer1 and Timer2. These timers use the system clock as clock source and once activated count up continuously. They start from the value initially loaded into the counting register (32-bit) and, if enabled, generate an interrupt upon rolling over (0xFFFFFFFF → 0x00000000).

13.1 Timer registers

TMR0REG: 32-bit Timer initial value register							
TMR0REG		0x50020000			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0
T15	T14	T13	T12	T11	T10	T9	T8
T23	T22	T21	T20	T19	T18	T17	T16
T31	T30	T29	T28	T27	T26	T30	T24
MSB							LSB
Bit31-0 **T[31:0]**: Timer Register initial value register.							

TMR0CTRL: Timer Control							
TMR0CTRL		0x50020004			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TSTART
MSB							LSB
Bit0 **TSTART**: Timer enable bit. 0 = Timer not running 1 = Timer running							

TMR1REG: 32-bit Timer initial value register							
TMR1REG		0x50020008			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0
T15	T14	T13	T12	T11	T10	T9	T8
T23	T22	T21	T20	T19	T18	T17	T16
T31	T30	T29	T28	T27	T26	T30	T24
MSB							LSB
Bit31-0 **T[31:0]**: Timer Register initial value register.							

TMR1CTRL: Timer Control							
TMR1CTRL		0x5002000C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TSTART
MSB							LSB
Bit0 **TSTART**: Timer enable bit. 0 = Timer not running 1 = Timer running							

TMR2REG: 32-bit Timer initial value register							
TMR2REG		0x50020010			0x00000000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
T7	T6	T5	T4	T3	T2	T1	T0
T15	T14	T13	T12	T11	T10	T9	T8

TMR2REG: 32-bit Timer initial value register							
T23	T22	T21	T20	T19	T18	T17	T16
T31	T30	T29	T28	T27	T26	T30	T24
MSB							LSB
Bit31-0 **T[31:0]**: Timer Register initial value register.							

TMR2CTRL: Timer Control							
TMR2CTRL		0x50020014			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TSTART
MSB							LSB
Bit0 **TSTART**: Timer enable bit. 0 = Timer not running 1 = Timer running							

13.1.1 Timer Operation

The operation of the timers is quite straightforward. Load the initial counter register, enable the timer and either check (polling mode) the current value of the counter register or enable the interrupt and process it inside the interrupt service routine.

Note: Inside the interrupt the application code must reload the timer counting register.

Code Example1: Enable Timer1 to count from 0xFFFF0000 and to generate interrupt:

```
TMR_Config( 1, TIMERON, 0xFFFF0000);  //Enable timer1 to count up from 0xFFFF0000
NVIC_ClearPendingIRQ( TIMER1_IRQn );        //Clear pending interrupt
NVIC_EnableIRQ( TIMER1_IRQn );        //Enable Timer1 interrupt


void Timer1_Handler( void )
{
  *TMR1REG = 0xFFFF0000;             //Reload Register
  //**** From this point application code inside ISR****
}
```

13.2 Watch Dog Timer

Herzog implements a WDT (**W**atch **D**og **T**imer) that can operate in one of two basic ways:

1.) Interrupt Mode: In the event of a WDT rollover an interrupt will be generated.

2.) Reset Mode: In the event of a WDT rollover the microcontroller will reset.

13.2.1 WDT Registers

The Watch Dog Timer implements two 32-bit registers:

WDTCTRL: WDT (Watch Dog Timer) Control Register. (32-bit)							
WDTCTRL		0x50020018			0x0000000x		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	WDTPRES1	WDTPRES0	RSTFLAG	RESETEN	WDTEN
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-
MSB							LSB

Bit4-3 **WDTPRES1: WDTPRES0**: WDT Prescaler:

00 = 2^{13}/SystemClock

01 = 2^{19}/SystemClock

10 = 2^{22}/SystemClock

11 = 2^{32}/SystemClock

Bit2 **RSTFLAG**: Reset Flag. This flag is set by the system at the initialization if the initialization was caused by a reset triggered by the WDT. The bit can be de-asserted by the application.

Bit1 **RESETEN**: Reset enable. If enabled a WDT time-out will force the microcontroller to reset. This bit can be asserted but it cannot be de-asserted.

Bit0 **WDTEN**: WDT enable. This bit can be asserted but it cannot be de-asserted. It means that once the WDT is enabled it cannot be turned off until a Reset or Power-On Reset occurs.

For instance, a system running from a 30MHz Crystal with WDTPRES[1…0] = 10 will trigger the WDT after approximately 0.14seconds if not cleared properly and in time by the application.

WDTCLR: WDT Clear Register. (32-bit)							
WDTCLR		0x5002001C			0x0000000x		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
WCLR7	WCLR6	WCLR5	WCLR4	WCLR3	WCLR2	WCLR1	WCLR0
WCLR15	WCLR14	WCLR13	WCLR12	WCLR11	WCLR10	WCLR9	WCLR8
WCLR23	WCLR22	WCLR21	WCLR20	WCLR19	WCLR18	WCLR17	WCLR16
WCLR31	WCLR30	WCLR29	WCLR28	WCLR27	WCLR26	WCLR30	WCLR24
MSB							LSB

Bit31-0 **WCLR[31:0]**: Clear Register. To clear the WDT counting the following words must be written in this order and without any other instruction between then:

0x3C570001

0x007F4AD6

Warning: Programming WDTCLR with other values or in the wrong order will cause the watchdog to throw an interrupt or reset the system.

Example Code: Setting and clearing the WDT. (Interrupt mode with a time of 2^22)

```
WDT_Config(WDT_INT, WDT22);   //Enable WDT in interrupt mode (2^22 system clock cycles)
WDT_Clear();                  //Clear WDT
```

13.3 Interrupt Vectors

Herzog implements an interrupt vector defined in the following table:

Table 9 – Interrupt Vector Table			
Cortex M0 Specific Exceptions			
Name	**Number**	**Comments**	**Required Interrupt Handler (Function)**
HardFault_IRQn	-13	HardFault handler*	HardFault_Handler (void)
SVCall_IRQn	-5	Supervisory call*	
PendSV_IRQn	-2	Interrupt-driven request for system level service*	
SysTick_IRQn	-1	SysTick Timer interrupt	void SysTick_Handler(void)
Cortex M0 Specific Exceptions			
Name	**Number**	**Comments**	**Required Interrupt Handler (Function)**
BrownOut_IRQn	0	Brownout detection interrupt	void BrownOut_Handler (void)
ClkMon_IRQn	1	Clock monitor interrupt	void ClkMon_Handler (void)
-	2	*Reserved*	
PIN_IRQn	3	Pin change interrupt	void PIN_Handler (void)
RFRE_IRQn	4	RF: Rising Edge base band signal reception interrupt	void RFRE_Handler (void)
RFFE_IRQn	5	RF: Falling Edge base band signal reception interrupt	void RFFE_Handler (void)
TX_done_IRQn	7	*RF: Burst transmission done interrupt*	void TX_done_IRQ_Handler (void)
TX_reload_IRQn	6	*RF: Transmission FIFO reload interrupt*	void TX_reload_IRQ_Handler (void)
UART_IRQn	8	*UART*	void UART_Handler (void)

-	9	*Reserved*	void Default_IRQ_Handler (void)
SPI_IRQn	10	*SPI*	void SPI_Handler (void)
-	11	*Reserved*	void Default_IRQ_Handler (void)
RFMSG_IRQn	12	RF: Message received interrupt	void RFMSG_Handler (void)
IRQ13_IRQn to IRQ15_IRQn	13-15	*Reserved*	void Default_IRQ_Handler(void)
TIMER0_IRQn	16	Timer0 interrupt	void Timer0_Handler (void)
TIMER1_IRQn	17	Timer1 interrupt	void Timer1_Handler (void)
TIMER2_IRQn	18	Timer2 interrupt	void Timer2_Handler (void)
WATCHDOG_IRQn	19	Watchdog timer interrupt	void Watchdog_Handler (void)

*Note: For more information see *Cortex-M0 Devices – Generic Users Guide (ARM DUI 0497A (ID112109))* at: http://infocenter.arm.com/help/topic/com.arm.doc.dui0497a/DUI0497A_cortex_m0_r0p0_generic_ug.pdf

14.0 RF Receiver Subsystem

Herzog implements a programmable ISM (Industrial, scientific and medical band, 300-450MHz) OOK (on-off keying) low-IF receiver. The local oscillator is generated using a fully integrated fractional-N PLL referenced to an external crystal reference. The received data is digitized using analog to digital converters before being processed by an autonomous digital section.

The receiver uses Weaver architecture for image rejection, primarily to avoid noise imaging. After amplification through an LNA, a RF mixer is used to generate I/Q signals at the IF frequency of 795KHz, where it is filtered to ~500KHz bandwidth. After the second frequency conversion, the I and Q signals are filtered to ~150kHz bandwidth. The wide bandwidth relative to the data symbol rate is necessary to accommodate manufacturing variation in the transmit and receive frequency references.

The frequency generation for the local oscillators is accomplished using a PLL locked to the crystal frequency. The VCO is a low current quadrature ring oscillator. It is expected that 30MHz crystal in combination with a divide by 8 prescaler stage will be utilized for frequency reference (3.130MHz). In the default frequency plan, the first LO is generated from 138.62*fXO = 433.17MHz using the PLL in frac-N mode.

The second LO is generated by dividing the first LO, first by a high speed divide-by-eight prescalar, followed by a programmable divider and a quadrature divide-by-four. The default frequency plan uses divide by 18 for the programmable part, for a second LO of 752.03kHz. Therefore the overall division ratio for the LO2 signals equal 576 (8x18x4), to be verified in the frequency table below.

f_RX	f_XTAL	F_EN	NX	NF	HISD	DIV_LO2	f_LO1	f_IF	f_LO2	freq_error
MHz	MHz	{0,1}	{1..255}	{0..255}	{0,1}	{0..3}	MHz	kHz	kHz	kHz
433.92	3.1250	1	138	157	1	2	433.1665	753.5	752.03	1.471

Due to the slight difference between the LO and IF frequencies, there is a 1.5kHz frequency offset in the baseband data, which appears as a slight additional transmit frequency error to the decoder. For other crystal frequencies, different settings will need to be programmed for the PLL and divider as described below.

After analog filtering, the baseband signal is then digitized at 298kS/s using a 12bit ADC. The digitized signal is dc-offset corrected and AM detected using a CORDIC to produce an AM baseband signal, which is filtered and decimated to an approximately 5kHz bandwidth with 18.6kS/s data rate.

Data is digitally extracted from filtered baseband signal using a digital bit slicer. An integrated decoder may be utilized to decode 1/3-2/3 duty-cycle encoded data. Decoded bits are stored in a bit buffer with capability to store messages as long as 80 bits. Once an entire valid message is stored in the RF bit buffer, an interrupt is generated. The receiver then enters an armed state, but with decoder inactive until the microcontroller re-enables the receiver to receive subsequent messages. The microcontroller should read any data from the bit buffer before re-enabling or else it will be lost.

Alternatively, if an application requires a coding scheme other than 1/3-2/3 coding, the slicer digital output may be made available in real time for the micro to decode the signal by software. The raw signal is guaranteed glitch-free, allowing a simple decode.

The receiver can be run in an autonomous “sniffing” mode with, for example, a 5% on-time duty cycle, in order to save power. Whether in sleep mode or not, intervention by the microcontroller is only required upon reception of an entire message with valid number of bits. The micro can therefore be asleep during normal RF reception, and only needs be awoken by interrupt after an entire message arrives, allowing significant power savings.

Many parameters in the RF are controllable by software. Additionally, functional control such as enabling and disabling the receiver, and the received bits are controllable through registers. The register section describes the functions of various registers related to the receiver. All control registers preset when power-on reset or software reset is asserted.

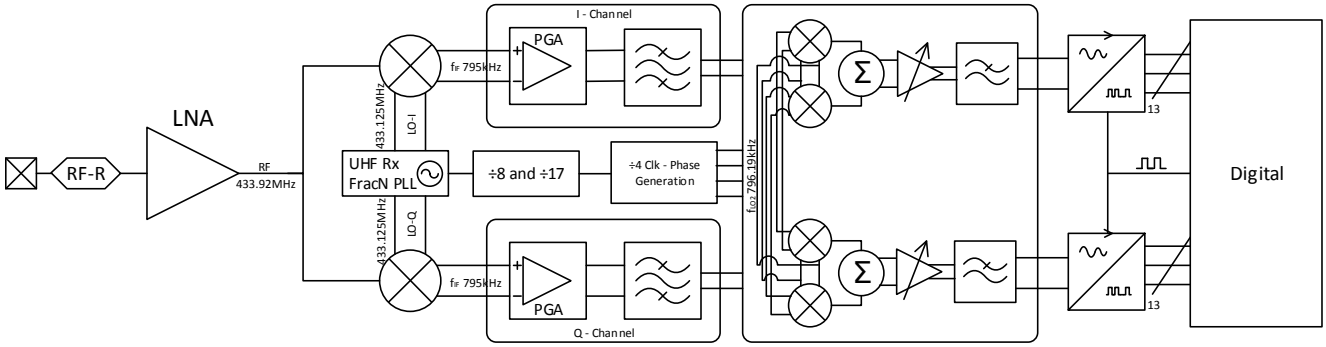


Figure 4 – ASK Receiver

Table 10 – RF specification, recommended operating conditions unless otherwise specified					
Parameter	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
LNA input impedance			10-129j		Ω
Sensitivity			-110		dBm
Frequency Range		300		470	MHz
Data Rate				5	kbps
Maximum input signal				-10	dBm
Total Gain	Voltage gain from RF input to I or Q IF outputs at default gain programming	59	71	74	dB
Spurious Emission				-60	dBm

14.1RF Receiver Usage Description

The following code fragment, shows how to configure the radio receiver to a wanted receive frequency of 433.92MHz with a minimum message length of 40 bits. Additionally, it enables message reception interrupts and employs the interrupt handler via the NVIC block in the microcontroller, and to perform a basic response to the interrupt

* TBD * (code to be inserted)

14.2 RF Receiver Registers

Table 11 – Receiver Register Map

Address	Register Name	Field Name	Description
0x50000020	RF_Buff0-F	RXDATA 0-127	RF Buffer Registers containing received bits.
0x50000030	RF_NUMB	RFNUMB0-7	Returns the number of bits contained in the bit buffer
0x50000031	RF_STATUS	RF_SLEEP	RF sleep mode indicator
		SLC_OUT	Slicer output
		DCDMD0	Determines whether state machine transitions on edge or level
		DCDMD1	Determines what to do if there is a overly-long non-guard band element
		ADC_FLG	Overflow indicator
		MSG_RDY	Message ready indicator
		SNF_EN	Enables sniff/sleep mode in the receiver

Table 11 – Receiver Register Map			
		RF_EN	Enables RF blocks
0x50011000	RF_NBMIN	Low_BPS	Indication to interpret all bit timings as 2X (slow transmitter)
		NBMIN[6:0]	Minimum number of bits for a valid transaction
0x50011001	RF_AGCCTRL	AGC_EN	Automatic gain control enable
		AGCTRM[6:0]	Controls gain of analog blocks
0x50011002	RF_SLCCTRL	ALPHA[1:0]	Controls decay time of slicer level
		BETA[1:0]	Controls attack time of slicer level
		FTIME[1:0]	Contols the time to allow the slicer to fast bias
		DR_SYM[1:0]	Sets post-CORDIC decimation rate
0x50011003	RF_SYSTIME	PLLTIME[1:0]	Controls time to wait for PLL to bias
		SLEEPTIME[2:0]	Controls sleep time between sniff cycles
		WAKETIME[2:0]	Controls the time to stay awake after seeing a valid guard band
0x50011004/5	RF_DCDTIME	MIN_GB[5:0]	Minimum number of additional samples after passing MIN_TE and MAX_TE for a low element to be considered a valid guard band length
		MAX_TE[5:0]	Maximum number of additional samples after passing MIN_TE for an edged to be considered short enough to be valid
		MIN_TE[3:0]	Minimum number of samples for a valid element time
0x50011006	RF_SNIFMODE	RT_SEL[1:0]	Selects a source for real time output
		RF_ON	Force analog RF to stay on
		SNIFF_NE[3:0]	Number of edges to check in each sniff cycle before committing to a long wake cycle
0x50011007	RF_AGCMON	AGCDATA[6:0]	RF gain control value
0x50011008/9	RF_SIGI	CAL_I[11:0]	Channel I DC calibration value
0x5001100A/B	RF_SIGQ	CAL_Q[11:0]	Channel Q DC calibration value

Table 11 – Receiver Register Map			
0x5001100C/D	RF_SLCHI	PDHI[15:0]	Peak detector high value
0x5001100E/F	RF_SLCLO	PDLO[15:0]	Peak detector low value
0x50018004	RF_NX	NX[7:0]	Integer part of the divider value for PLL divider
0x50018005	RF_NF	NF[7:0]	Fractional part of the divider value for PLL divider
0x50018006	RF_FETRIM0	HI_SD	Controls image rejection mixer to use high side or low side mixing
		FE_EN	Controls fractional mode
		CP_TRIM	Adjusts charge pump current in PLL for optimum settling time
		LF_TRM	Adjust loop filter stabilization resistor to control overshoot
0x50018007	RF_FETRIM1	DIV_LO2	Divider value from PLL frequency to LO2 frequency
		BBGAIN[1:0]	Sets baseband amplifier gain
		LNA_DRES	LNA drain resistor
		LNA_PG[2:0]	Adjust …

The following registers define the behavior of the RX-RF:

RF_BUFF0-F: RF Buffer Registers containing received bits.							
RF_BUFF0-F		0x50000020-2F			0xXX		
R	R	R	R	R	R	R	R
RXDATA7	RXDATA6	RXDATA5	RXDATA4	RXDATA3	RXDATA2	RXDATA1	RXDATA0
RXDATA15	RXDATA14	RXDATA13	RXDATA12	RXDATA11	RXDATA10	RXDATA9	RXDATA8
:	:	:	:	:	:	:	:
RXDATA119	RXDATA118	RXDATA117	RXDATA116	RXDATA115	RXDATA114	RXDATA113	RXDATA112

RF_BUFF0-F: RF Buffer Registers containing received bits.							
RXDATA127	RXDATA126	RXDATA130	RXDATA124	RXDATA123	RXDATA122	RXDATA121	RXDATA120
MSB							LSB

Bit127-0 **RXDATA[7:0]**: Received data bits. Most recently received bit is stored in RXDATA0. This register should be read when a complete message is ready (determined by reading MSG_RDY bit or having received an interrupt from the RF system) for repeatable results

RF_NUMB: Returns the number of bits contained in the bit buffer.							
RF_NUMB		0x50000030			0xXX		
R	R	R	R	R	R	R	R
RFNUMB7	RFNUMB6	RFNUMB5	RFNUMB4	RFNUMB3	RFNUMB2	RFNUMB1	RFNUMB0
MSB							LSB

Bit7-0 **RFNUMB [7:0]:** Returns the number of received data bits contained in the bit buffer. For repeatable results, it is recommended to only read this register when a complete message is ready

RF_STATUS: RF Status Register.							
RF_STATUS		0x50000031			0xXX		
R/W	R/W	R	R	Reserved	Reserved	R	R
RF_EN	SNF_EN	MSG_RDY	AGC_FLG	DCDMD1	DCDMD0	SLC_OUT	RF_SLEEP
MSB							LSB

Bit7 **RF_EN**: Enables RF block.
0 = Disable the receiver
1 = Enable the receiver

Bit6 **SNF_EN**: Enables sniff/sleep mode in the receiver

RF_STATUS: RF Status Register.
0 = Continuous mode 1 = Sniff/Sleep mode
Bit5 **MSG_RDY**: Message ready indicator. Reading returns one if a complete message is available and the receiver is in armed mode, otherwise returns zero. Write is ignored unless in armed mode, where writing a zero causes the receiver to leave the armed state and start decoding message again
Bit4 **AGC_FLG**: AGC overflow indicator. Returns one if an overflow(signal too large) has occurred in receiver. Intended for use in continuous mode, where the micro may want to reduce the gain setting
Bit3 **DCDMD1**: Determines what to do if there is an overly-long non-guard band element 0 = reset the decoder and start looking for new message 1 = wait until a guardband is received before resetting state machine
Bit2 **DCDMD0**: Determines whether state machine transitions on edge or level 0 = transition on edge 1 = transition on level
Bit1 **SLC_OUT**: Slicer Output
Bit0 **RF_SLEEP**: RF sleep mode indicator 0 = RF RX is active 1 = RF RX is sleeping

RF_NBMIN: RF NB MIN Register.							
RF_NBMIN		0x50011000			0x28		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
LOW_BPS	NBMIN6	NBMIN5	NBMIN4	NBMIN3	NBMIN2	NBMIN1	NBMIN0
MSB							LSB
Bit7 **LOW_BPS**: Indication to interpret all bit timings as 2X (for slow transmitters) 0 = normal rate 1 = count bit timings at half rate							
Bit6-0 **NBMIN[6:0]**: Minimum number of bits for a valid message							

RF_AGCCTRL: RF Automatic Gain Control Register.							
RF_AGCCTRL		0x50011001			0x98		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
AGC_EN	AGCTRM6	AGCTRM5	AGCTRM4	AGCTRM3	AGCTRM2	AGCTRM1	AGCTRM0
MSB							LSB

Bit7 **AGC_EN**: Automatic Gain Control Enable

0 = Fixed gain mode

1 = AGC enabled

Bit6-0 **AGCTRM[6:0]**: Controls gains of analog blocks

If AGC_EN is zero, then the bits in the control registers are used to directly control the AGC trim of the analog blocks.

If AGE_EN is one, then the bits sets the default gain and reduction step

Bit4 (0 = gain steps down the table in increment of 1; 1= gain steps down the table in increment of 2)

Bit3-0	st3	st2	LNA	gain	delta
0000	00	00	000	-4.7dB	
0001	00	00	010	-1.0dB	3.7dB
0010	00	00	011	4.5dB	5.5dB
0011	00	00	100	9.7dB	5.2dB
0100	00	00	101	15.4dB	5.7dB
0101	00	00	110	21.0dB	5.6dB
0110	00	00	111	26.8dB	5.8dB
0111	00	10	111	32.4dB	5.6dB
1000	10	11	111	38.6dB	6.2dB (default)
1001	11	11	111	45.5dB	6.9dB
1010	11	11	111	52.6dB	7.1dB
1011	not allowed				
11xx	not allowed				

RF_SLCCTRL: RF Slicer Control Register							
RF_SLCCTRL		0x50011002			0xAB		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ALPHA1	ALPHA0	BETA1	BETA0	FTIME1	FTIME0	DR_SYM1	DR_SYM0
MSB							LSB

Bit7-6 **ALPHA[1:0]**: Controls decay time of slicer level. When input is inside slicer levels, slicer decays according to equation (clocked at decimated data rate):

y[n] = (1-ALPHA)*y[n-1] + ALPHA*x[n]

00 = 1/306 (fastest decay rate)

01 = 1/512

10 = 1/1024

11 = 1/2048 (slowest decay rate)

Bit5-4 **BETA[1:0]**: Controls attack time of slicer level. When input is outside slicer levels, slicer grows according to equation (clocked at decimated data rate):

y(n) = (1-BETA)*y[n-1] + BETA*x[n]

00 = 1/2 (fastest attack rate)

01 = 1/4

10 = 1/8

11 = 1/16 (slowest attack rate)

Bit3-2 **FTIME[1:0]**: Controls the time to allow the slicer to fast bias, measured in clock cycles of fxo/64

00 = 1 cycle (17us for 3.75MHz clock (30MHz crystal / 8))

01 = 32 cycles (546us for 3.75MHz clock)

10 = 48 cycles (819us for 3.75MHz clock)

11 = 64 cycles (1.09ms for 3.75MHz clock)

Bit1-0 **DR_SYM[1:0]**: Sets post-CORDIC decimation rate.

00 = 13X

01 = 14X

10 = 15X

11 = 16X

RF_SYSTIME: RF System Time Register.							
RF_SYSTIME		0x50011003			0x5A		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PLLTIME1	PLLTIME0	SLPTIME2	SLPTIME1	SLPTIME0	WAKETIME 2	WAKETIME1	WAKETIME0
MSB							LSB

Bit7-6 **PLLTIME[1:0]**: Controls the time to wait for PLL to bias. Measured in clock cycles of the fxo/64

00 = 32 cycles (546us for 3.75MHz clock (30MHz crystal / 8))

01 = 48 cycles (819us for 3.75MHz clock)

10 = 64 cycles (1.09ms for 3.75MHz clock)

11 = 128 cycles (2.18ms for 3.75MHz clock)

Bit5-3 **SLPTIME[2:0]**: Controls the sleep time between sniff cycles.

Measured in clock cycles of the fxo/64.

000 = 4*1024 cycles (70ms for 3.75MHz clock (30MHz crystal / 8))

001 = 6*1024 cycles (105ms for 3.75MHz clock)

010 = 8*1024 cycles (140ms for 3.75MHz clock)

011 = 10*1024 cycles (175ms for 3.75MHz clock)

100 = 12*1024 cycles (209ms for 3.75MHz clock)

101 = 14*1024 cycles (245ms for 3.75MHz clock)

110 = 32*1024 cycles (559ms for 3.75MHz clock)

111 = 128*1024 cycles (2.27s for 3.75MHz clock)

Bit2-0 **WAKETIME[2:0]**: Controls the time to stay awake after seeing a valid guard band.

Measured in clock cycles of the fxo/64.

000 = 8*1024 cycles (140ms for 3.75MHz clock (30MHz crystal / 8))

001 = 10*1024 cycles (175ms for 3.75MHz clock)

010 = 12*1024 cycles (209ms for 3.75MHz clock)

011 = 14*1024 cycles (245ms for 3.75MHz clock)

100 = 16*1024 cycles (280ms for 3.75MHz clock)

101 = 18*1024 cycles (315ms for 3.75MHz clock)

110 = 24*1024 cycles (419ms for 3.75MHz clock)

111 = 32*1024 cycles (586ms for 3.75zMHz clock)

RF_DCDTIME: RF Decode Time Control Register.							
RF_DCDTIME		0x50011004/5			0x3614		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MAX_TE1	MAX_TE0	MIN_GB5	MIN_GB4	MIN_GB3	MIN_GB2	MIN_GB1	MIN_GB0
MIN_TE3	MIN_TE2	MIN_TE1	MIN_TE0	MAX_TE5	MAX_TE4	MAX_TE3	MAX_TE2
MSB							LSB

Bit5-0 **MIN_GB[5:0]**: Minimum number of ADDITIONAL samples after passing MIN_TE and MAX_TE for a low element to be considered a valid guard band length. Default setting with 3.75MHz clock (30MHz crystal / 8) corresponds to minimum guard band time of 3.4ms

Bit11-6 **MAX_TE[5:0]**: Maximum number of ADDITIONAL samples after passing MIN_TE for an edge to be considered short enough to be valid. Default setting with 3.75MHz clock (30MHz crystal / 8)
Corresponds to maximum element time of 1.4ms

Bit15-12**MIN_TE[3:0]**: Minimum number of samples for a valid element time. Counted in decimated data rate (fxo/(12*DR_SYM)). Default setting with 3.75MHz clock (30MHz crystal / 8) corresponds to minimum element time of 153us

RF_SNIFMODE: RF Sniff Mode Register.							
RF_SNIFMODE		0x50011006			0x0B		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
RT_SEL1	RT_SEL0	RF_ON	Reserved	SNIFF_NE3	SNIFF_NE2	SNIFF_NE1	SNIFF_NE0
MSB							LSB

RF_SNIFMODE: RF Sniff Mode Register.
Bit7-6 **RT_SEL[1:0]**: Selects a source for real-time output. 00 = supervisor clock 01 = decimator output (serialized stream) 10 = slicer output 11 = PLL_EN (high if analog blocks enabled)
Bit5 **RF_ON**: Force analog RF to stay on (for test). 0 = Normal 1 = RF stays on for test
Bit3-0 **SNIFF_NE[3:0]**: Number of edges to check in each sniff cycle before committing to a long Wake cycle. When in sniff mode, the first SNIFF_NE edges are tested for valid timing. If any one of these first edges is badly timed, then the receiver will go to sleep

RF_AGCMON: AGC Monitor Register							
RF_AGCMON		0x50011007			0xXX		
R	R	R	R	R	R	R	R
Reserved	AGCDATA 6	AGCDATA 5	AGCDATA 4	AGCDATA 3	AGCDATA 2	AGCDATA 1	AGCDATA0
MSB							LSB
Bit6-0 **AGCDATA[6:0]**: RF gain control value							

RF_SIGI: I Channel Calibration Monitor Register							
RF_SIGI		0x50011008/9			0xXXXX		
R	R	R	R	R	R	R	R
CAL_I7	CAL_I6	CAL_I5	CAL_I4	CAL_I3	CAL_I2	CAL_I1	CAL_I0
Reserved	Reserved	Reserved	Reserved	CAL_I11	CAL_I10	CAL_I9	CAL_I8
MSB							LSB

RF_SIGI: I Channel Calibration Monitor Register
Bit11-0 **CAL_I[11:0]**: Channel I DC calibration value

RF_SIGQ: Q Channel Calibration Monitor Register							
RF_SIGQ		0x5001100A/B			0xXXXX		
R	R	R	R	R	R	R	R
CAL_Q7	CAL_Q6	CAL_Q5	CAL_Q4	CAL_Q3	CAL_Q2	CAL_Q1	CAL_Q0
Reserved	Reserved	Reserved	Reserved	CAL_Q11	CAL_Q10	CAL_Q9	CAL_Q8
MSB							LSB
Bit11-0 **CAL_Q[11:0]**: Channel Q DC calibration value							

RF_SLCHI: Peak Detector High Value Monitor Register							
RF_SLCHI		0x5001100C/D			0xXXXX		
R	R	R	R	R	R	R	R
PDHI7	PDHI6	PDHI5	PDHI4	PDHI3	PDHI2	PDHI1	PDHI0
PDHI15	PDHI14	PDHI13	PDHI12	PDHI11	PDHI10	PDHI9	PDHI8
MSB							LSB
Bit15-0 **PDHI[15:0]**: Peak detector high value							

RF_SLCLO: Peak Detector Low Value Monitor Register							
RF_SLCLO		0x5001100E/F			0xXXXX		
R	R	R	R	R	R	R	R
PDLO7	PDLO6	PDLO5	PDLO4	PDLO3	PDLO2	PDLO1	PDLO0
PDLO15	PDLO14	PDLO13	PDLO12	PDLO11	PDLO10	PDLO9	PDLO8

RF_SLCLO: Peak Detector Low Value Monitor Register							
MSB							LSB
Bit15-0 **PDLO[15:0]**: Peak detector low value							

RF_NX: Controls the integer portion of the PLL feedback divider.							
RF_NX		0x50018004			0x79		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NX7	NX6	NX5	NX4	NX3	NX2	NX1	NX0
MSB							LSB

Bit7-0 **NX[7:0]**: Integer part of the divider value for PLL divider.
If (F_EN = 0) $f_{LO} = f_{xo} * NX$
If (F_EN = 1) $f_{LO} = f_{xo} * (NX + NF/306)$							

RF_NF: Controls the fractional portion of the PLL feedback divider							
RF_NF		0x50018005			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NF7	NF6	NF5	NF4	NF3	NF2	NF1	NF0
MSB							LSB

Bit7-0 **NF[7:0]**: Fractional part of the divider value for PLL divider.
If (F_EN = 0) $f_{LO} = f_{xo} * NX$
If (F_EN = 1) $f_{LO} = f_{xo} * (NX + NF/306)$							

RF_FETRIM0: RF Front-end Trim0 Register.							
RF_FETRIM0		0x50018006			0x9A		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
HI_SD	F_EN	CP_TRM1	CP_TRM0	LF_TRM3	LF_TRM2	LF_TRM1	LF_TRM0
MSB							LSB

Bit7 **HI_SD**: Controls image reject mixer to control whether to use high-side or low-side mixing

0 = low-side

1 = high-side

Bit6 **F_EN**: Controls fractional mode

0 = integer-N mode

1 = fractional-N mode

Bit5-4 **CP_TRM[1:0]**: Adjust charge pump current in PLL for optimum settling time

Bit3-0 **LF_TRM[3:0]**: Adjust loop filter stabilization resistor in PLL to control overshoot

RF_FETRIM1: RF Front-end Trim1 Register.							
RF_FETRIM1		0x50018007			0x4A		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
DIV_LO21	DIV_LO20	BBGAIN1	BBGAIN0	LNA_DRES	LNAB_PG2	LNAB_PG1	LNAB_PG0
MSB							LSB

RF_FETRIM1: RF Front-end Trim1 Register.	
Bit7-6	**DIV_LO2[1:0]**: Divide value from PLL frequency to LO2 frequency ($f_{LO2} = f_{LO1}/(32*DIV\_LO2)$) 00 = 8*15*4 01 = 8*16*4 10 = 8*17*4 11 = 8*18*4
Bit5-4	**BBGAIN[1:0]**: Sets baseband amplifier gain 00 = 0.6 dB 01 = 6.0 dB 10 = 11.3 dB 11 = 15.8 dB
Bit3	**LNA_DRES**: LNA drain resistor 0 = 2kΩ 1 = 1kΩ
Bit2-0	**LNAB_PG[2:0]**: Adjust loop filter stabilization resistor in PLL I = 200uA*Bit0 + 500uA*Bit1 + 1mA*Bit2

15.0 RF Transmitter Subsystem

Herzog implements a powerful RF transmitter operating in the ISM (**I**ndustrial, **S**cientific and **M**edical) band and capable of transmitting ASK/OOK modulations. Its main characteristics are:

- Precise fractional-N PLL referenced to a 30MHz crystal oscillator, which drives into a modulation control circuit, and then into a high power output stage
- High power output stage
- Integrated power control
- Autonomous state machine which controls transmit bursts

The following block diagram details the main structures used by the transmitter.

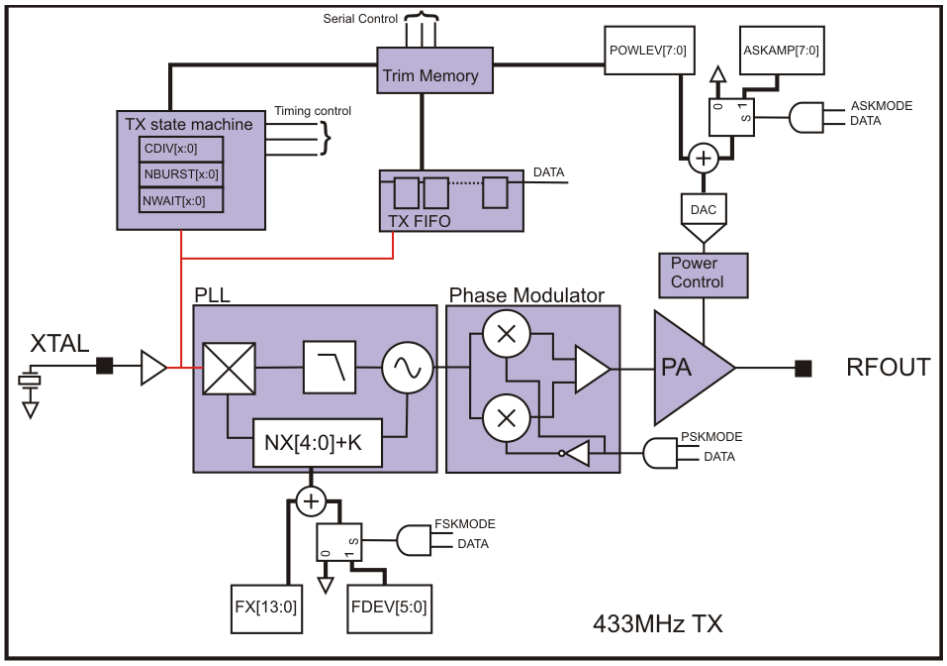


Figure 5 – ASK Transmitter

The output stage amplifies modulated data up to a maximum power level of +15dBm. The device has a single output pin, which requires external matching. The PA output power is regulated by an integrated power control stage. This power control circuit takes an internal measurement, which correlates with the output power of the high power output stage, and compares it to an internal reference, which is set by a digital control register.

The RF transmit frequency is generated by a fractional-N PLL. The PLL may be operated in integer mode by setting bit F_EN to logic 0, or in fractional mode by setting bit F_EN to logic 1. The frequency generated is calculated according to the following formula:

In integer mode, $f_{VCO}=f_{XTAL}*(NX[4:0]+K)$, where K=8

or in fractional mode, $f_{VCO}=f_{XTAL}*(NX[4:0]+K+(NF[13:0]/2^{14}))$, where K=9

Table 12 – Transmitter Performance Specification

Name	Conditions	Min.	Typ.	Max.	Unit
Output Power Range	V3p3ANA=3.3V, T_A=27°C	-13.5		+15.1	dBm
Output Power Step	Valid from +11.5 to -9.5dBm output power		1.5		dB
Operating Frequency	433.92 MHz		433.92		MHz
Current Consumption	Pout=max (+13dBm)		18		mA
Sleep Current Consumption	All blocks disabled			100	nA
Data Rate	Output Power below 0dBm for ASK/OOK			50	kbps
Data Rate	Output Power higher than 0dBm for ASK/OOK			30	kbps
Output Noise	fc = 434.07 MHz, frequency offset=1.6 MHz compliant with EN 300 220-1 (2000.09)			-36	dBm
Harmonics, Conducted 2nd Harmonic 433 MHz 3rd Harmonic 433 MHz			 -45.1 -49.4		dBm
Frequency stability	Depends on crystal employed			13	ppm

15.1 Fractional PLL

The RF transmitter frequency is generated by a fractional-N PLL circuit. The default reference frequency is 30 MHz. The PLL may be operated in integer mode by clearing the FRACEN bit (**TXCTRL0** register) or in fractional mode by setting FRACEN bit (**TXCTRL0** register).

The frequency generated is calculated depending on the PLL setup of NX and NF bits (located in **PLLCTRL0**, **PLLCTRL1** and **PLLCTRL2** registers):

PLL in integer mode:

$$Fvco = Fxtal \times (NX[4...0] + 8)$$

PLL in fractional mode:

$$Fvco = Fxtal \times (NX[4...0] + 9 + \frac{NF[13...0]}{2^{14}})$$

In the ISM band reaches from 433.05 to 434.79 MHz. Herzog can generate nominal frequencies from 433.049927 MHz up to 434.789429 MHz in steps of ~1831Hz.

Code Example: Setting the RF Transmitter to transmit at 433.92 MHz in fractional mode from a 30 MHz clock:

```
test = RF_Set_Frequency( 30000000, 433920000, PLL_FRACTIONAL);
```

15.2 Transmission State Machine Operation

The Herzog transmitter operates as a FSM (**F**inite **S**tate **M**achine) implemented in hardware. This FSM provides a group of registers that allow for a very flexible and powerful yet simple transmission of data. The FSM provides automatic transmission of up to 16 bits without reloading transmission registers. The FSM may interact with the microcontroller via 2 interrupts.

- The interrupts are:
  1.) Single Byte Left: when the buffer has only a single byte left to transmit (so that the micro can choose to refill both bytes as they are double-buffered)
  2.) End of Burst: After sending all bits the FIFO is cleared and the transmitter waits NWAIT samples. At this moment an interrupt is generated. The micro can refill the data buffer with samples and reprogram the clock division rate for the next burst if required or, if no new data is to be sent then it can disable the system.

The steps to properly set this FSM are as follows:

Select the modulation scheme ASK/OOK, using the MODUL[1…0] bits (TXCTRL0 register).

Set the PLL for the required frequency.

Select the bit rate using TXCDIV[10…0]. (TXCLKDIV and TXCTRL3)

Define the total number of bits per burst with NTXBITS[9…0] bits (NBURST and TXCTRL2).

Define the number of bits between bursts using NWAIT[5…0] (TXCTRL2).

Select the appropriate values for the different timing elements: TDET1, TDET2, TDET3, TDET4, TDET5 and TPATCH. (TXCTRL3, TXCTRL4, TXCTRL5)

If required, enable the interrupts associated with the transmission (end of burst and end of first byte transmission).

Load the first two bytes of data to be transmitted in TXDATA[15…0] bits (TXDATL and TXDATH).

Set the STARTX bit (TXCTRL0 register) to initiate transmission.

15.2.1 ASK Modulation Selection

Select ASK/OOK by simply loading the MODUL[1…0] bits in the TXCTRL0 register :

Code example 1:

(To set the ASK modulation now using access functions)

```
ret = RF_Set_Modulation( ASK );
```

15.2.2 Chip Rate Selection

The required chip rate can be calculated applying following equation:

$$chiprate = \frac{1.25MHz}{\text{TXCDIV}}$$

For instance, to obtain a chip rate of 19200bits/sec:

$$chiprate = \frac{1.25MHz}{\text{TXCDIV}} = 19200 \Rightarrow \text{TXCDIV} = \frac{1.25MHz}{19200} \cong 65$$

(This in fact will generate a chip rate of ~19231bits/sec, an error of 0.16%)

Code example: (From the calculation above)

```
TXTIMING->CLKDIV.HOWRD = 0xF800; // Clear CLKDIV
TXTIMING->CLKDIV.BYTE[0] = 65;   // Load the proper value in the lower 8 bits
```

Note:

Loading TXDIV with '0' will result in the maximum divider (1024), resulting in the minimum chip rate of ~1220bits/sec.

Baud rate will be fraction of above chip rate and depends on the format of coding used. For example if the coding used is Manchester, baud rate would be chip rate divided by two. If coding used is 1/3 2/3 coding, baud rate will be chip rate divided by three.

15.2.3 Burst Size Selection

The number of bits to be transmitted per burst is defined in NTXBITS[9…0]. The following code example selects a 700 bits/burst transmission: (700 = 0x2BC)

```
*TXCTRL2 &= 0xFC;            //Clear the upper 2 bits of the burst size
*TXCTRL2 |= 0x02;            //Load 0x02 into them
*NBURST = 0xBC;              //Load 0xBC into lower 8 bits (700d = 0x2BC)
```

Example Code using access function:

```
RF_Set_BurstSize( 700 );
```

15.2.4 Inter-Bursts Bit-Number Selection

The time waited between bursts is defined in number of bits at the current bit-rate. This time is defined as a function of NWAIT and the timing elements. The calculation of the total time between bursts is explained in the following.

Code Example: To create a 7 bits NWAIT time:

```
*TXCTRL2 &= ~0xFC;            //Clear the 5 bits of NWAIT
*TXCTRL2 |= (0x07<<2);        //Load 7d (0x07) into them
```

15.2.5 Timing Elements Selection

The RF transmitter is made of several subsystems. In order to operate properly they have to be activated and the proper settling time defined for each one of them. The figure below shows the several timing elements:

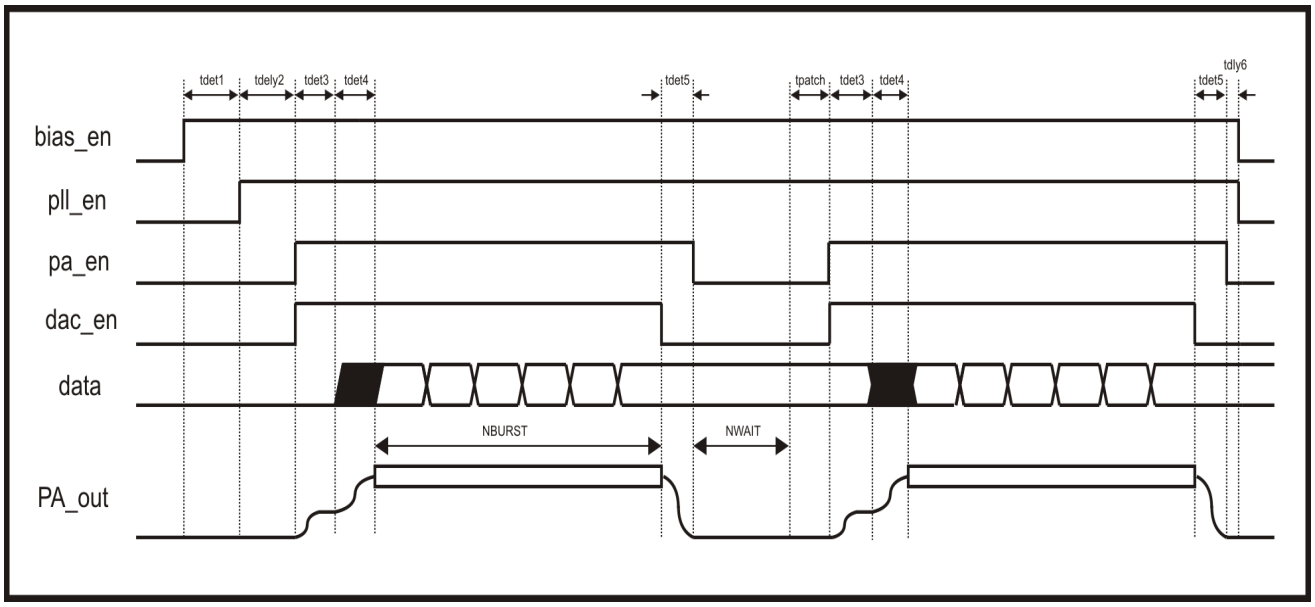


Figure 6 – Transmitter Timing

Once the transmission is started several subsystems are automatically activated:

PA (**P**ower **A**mplifier)

PLL (**P**hase **L**ocked **L**oop)

DAC (**D**igital-**A**nalog **C**onverter)

Figure 6 shows different delay times each of this subsystems require to reach their proper operational condition. These delay times are defined by timing elements as follows:

Tdet1 – This time element allows for the PA bias to reach the proper levels. The minimum recommended time for this delay is 1usec, therefore selecting TDET1 = 'b0 (1.6usec) is adequate.

Tdet2 – This time element allows for the PLL to be enabled and stabilize properly. The minimum time for it to do so is 1msec. and the recommended value for this field is TDET2 = 'b010 (1.6msec).

Tdet3 – This time element defines the "Pedestal" period. This time is necessary to ensure the PA reached a midway power level prior to being enabled into full power. The recommended time for this parameter is 'b101 (4usec).

Tdet4 and Tdet5* – These time elements are responsible for allowing for the minimum delay required to bring the PA to full power and minimum power. Both time elements should in principle be given the same value. The recommended value is 'b10100 (32usec).

*TDET5 is not directly accessible in Herzog and it reflects the value of TDET4.

Tpatch – This time element is responsible for adjusting the wait time between bursts to create an integer multiple of bit times interval.

Therefor the interval between bursts is equal to:

$$T = t\det 5 + NWAIT + tpatch + t\det 3 + t\det 4 = NWAIT + tpatch + t\det 3 + 2 \times t\det 4$$

Assuming the recommended values for Tdet3 and 4 we have then:

$$T = NWAIT + tpatch + 68u\sec$$

For a given number N of bit-times required between bursts the calculation becomes:

$$T = N \times \frac{TXCDIV}{1.25 \times 10^6} = NWAIT + tpatch + 68u\sec$$

And:

$$Tpatch = 781 \times 10^{-9} \times (TPATCH[10...0]) + 0.8 \times 10^{-6}$$

The following flowchart defines the procedure to adjust the parameters:

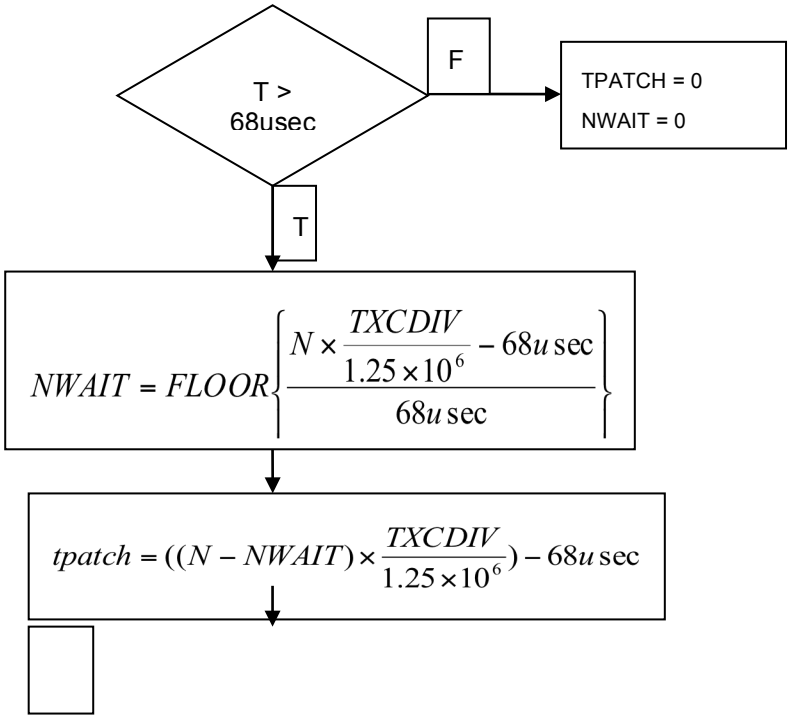


Figure 7 – Calculation of Interval between Bursts

For instance, if the user wants a baud rate of 19200bits/sec and 7 bit-time of interval between bursts assuming the recommended values described above the calculation becomes:

$$T = (7) \times \frac{65}{1.25 \times 10^6} = 364u\,\sec$$

T is larger than 68usec, therefore we can go ahead:

$$NWAIT = FLOOR\frac{\left[7 \times \frac{65}{1.25 \times 10^6} - 68u\,\sec\right]}{68u\,\sec} = 4$$

We have then 4 bit-time waits. Now calculating the tpatch time:

$$tpatch = \left[(7-4)\times\frac{65}{1.25\times10^{6}}\right] - 68u\,\text{sec} = 88u\,\text{sec}$$

$$TPATCH = ROUND\left[\frac{88u\,\text{sec} - 0.8u\,\text{sec}}{781n\,\text{sec}}\right] = 112$$

This gives us the following:

$$T = 68u\,\text{sec} + 4\times\frac{65}{1.25\times10^{6}} + 781\times10^{-9}\times112 + 0.8u\,\text{sec} = 364.3u\,\text{sec}$$

Code Example:

```
*TXCTRL3 &= 0x07;              //Clear upper 5 bits
*TXCTRL3 |= (0x10<<3);         //Set TPATCH = 112d = 0x70 (Low 5 bits)
*TXCTRL1 &= 0xC0;              //Clear TPATCH[10…5] bits
*TXCTRL1 |= 0x03;              //Load 0x03 (High 6 bits)
*TXCTRL4 &= 0xF0;              //Clear lower 4bits
*TXCTRL4 |= (0x02<<1)+0x00;    //Set TDET2='b010  and TDET1=0
*TXCTRL5 |= (0x05<<5)+0x14;    //Set TDET3='b101  and TDET4=0x14
```

15.3 Transmitter Registers

The transmitter provides several registers that provide a flexible and simple to use interface:

Table 13 – Transmitter Register Map

Address	Register Name	Field Name	Description
0x50000032	TXCTRL0	STARTX	Single bit which starts the transmission
		NXTBURST	Next Burst Flag
		MODUL	Modulation Selection
		FRACEN	PLL Fractional Enable bit
		[2…0]	Reserved
0x50000033	TXCTRL1	PEDES	Pedestal Time
		TPATCH	Delay Time between disabling the PLL and turning off the bias circuit
0x50000034	POWLEV	POWLEV	Power Level
0x50000035	POWAP	POWAP	Power level for ASK/OOK transmission of '1' (High Level)
0x50000036	TXDATL	TXDATA	Low Byte of Transmission Data
0x50000037	TXDATH	TXDATA	High Byte of Transmission Data
0x50000038	NBURST	NTXBITS	Number of bits of transmission in the next burst low byte
0x50000039	TXCTRL2	NWAIT	Number of bits between bursts
		NTXBITS	Number of bits of transmission in the next burst higher 2 bits
0x5000003A	TXCLKDIV	TXCDIV	Low byte of clock divider
0x5000003B	TXCTRL3	TPATCH	Lower 5 bits of the delay between disabling the PLL and turning off the bias
		TXCDIV	Upper three bits of the Transmission Clock Divider. (data rate is the 1.30Mbps/CDIV)
0x5000003C	TXCTRL4	TRAMP	Prevents the PA output power ramp up/down when the data bit changes.
		TDET2	Delay between enabling the PLL and enabling the power control DAC & power amplifier

Table 13 – Transmitter Register Map			
		TDET1	Delay between enabling the bias and enabling the PLL
0x5000003D	TXCTRL5	TDET3	Delay between enabling the power control DAC and enabling the power amplifier
		TDET4	PA output power rising and falling time
0x50010003	MDRCTRIM	MDRC[7:0]	ASK/OOK RC filter capacitor trim
0x50018000	PLLCTRL0	NF[7:0]	Fractional N portion of feedback divider lower bits
0x50018001	PLLCTRL1	NX[1:0]	Lower two bits of the Integer portion of feedback divider
		NF[13:8]	Six higher bits of the Fractional Portion of Feedback Divider
0x50018002	PLLCTRL2	LPFTRIM[1:0]	Low pass filter trim bits
		CPTRIM[2:0]	Charge pump trim bits
		NX[4:2]	Integer portion of feedback divider high bits
0x50018003	PLLCTRL3	DF[5:0]	Frequency deviation control
		LPFTRIM[3:2]	Low pass filter trim bits

A more detailed description of each register follows below:

TXCTRL0: First Transmitter Control Register							
TXCTRL0		0x50000032			0x08		
R/W	R/W	R/W	R/W	R/W	Reserved	Reserved	Reserved
STARTX	NXTBURST	MODUL1	MODUL0	FRAC_EN	RF_SW	-	TX_ON
MSB							LSB

Bit7 **STARTX**: Start Transmission.

0 = Do not start transmit

1 = Start Transmitting

Bit6 **NXTBURST**: Set to indicate there is a next burst.

0 = No next burst

TXCTRL0: First Transmitter Control Register
1 = There will be a next burst Bit5-4 **MODUL[1:0]:** Modulation selected. 00 = Reserved 11 = ASK/OOK Bit3 **FRAC_EN**: Fractional PLL enabled. 0 = Fractional PLL disabled 1 = Fractional PLL enabled Bit2 **RF_SW**: Toggles RF Switch between RX and TX Mode 0 = RX Mode 1 = TX Mode Bit1 **Reserved** Bit0 **TX_ON**: TX on for constant transmission 0 = Constant transmission disabled 1 = Constant transmission enabled

TXCTRL1: Second Transmitter Control Register							
TXCTRL1		0x50000033			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PEDES1	PEDES0	TPATCH10	TPATCH9	TPATCH8	TPATCH7	TPATCH6	TPATCH5
MSB							LSB
Bit7-6 **PEDES[1: 0]**: Pedestal Time: 00 = 12.8usec. 01 = 14.4usec 10 = 16usec. 11 = 17.6usec. Bit5-0 **TPATCH[10:5]**: Delay between disabling the PLL and turning off the bias: 0x000 = 0.8usec							

TXCTRL1: Second Transmitter Control Register
: 0x7FF = 1.6msec

POWLEV: Power Level of ASK/OOK Transmission in Level “0”							
POWLEV		0x50000034			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POWLEV 7	POWLEV6	POWLEV5	POWLEV4	POWLEV3	POWLEV2	POWLEV1	POWLEV0
MSB							LSB

Bit7-0 **POWLEV[7:0]**: Power Level, defined as follows:

POWLEV[7:0] (HEX)	Pout (dBm)
06-00	<-11
0A-07	-9.5
0B-08	-8
10-09	-6.5
12-0A	-5
12-11	-3.5
17-12	-2
18-14	-0.5
1B-18	+1
1F-1B	+2.5
24-1F	+4
28-26	+5.5
36-31	+7
45-37	+8.5
55-39	+10
70-54	+11.5

POWAP: Power Level of ASK/OOK transmission in level "1"							
POWAP		0x50000035			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
POWAP7	POWAP6	POWAP5	POWAP4	POWAP3	POWAP2	POWAP1	POWAP0
MSB							LSB
Bit7-0 **POWAP[7:0]**: Refer to **POWLEV** register.							

TX TXDATL: TX data lower byte register							
TXDATL		0x50000036			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TXDATA7	TXDATA6	TXDATA5	TXDATA4	TXDATA3	TXDATA2	TXDATA1	TXDATA0
MSB							LSB
Bit7-0 **TXDATA[7:0]**: Lower byte of data to be transmitted.							

TXDATU: TX data upper byte register							
TXDATU		0x50000037			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TXDATA15	TXDATA14	TXDATA13	TXDATA12	TXDATA11	TXDATA10	TXDATA9	TXDATA8
MSB							LSB
Bit7-0 **TXDATA[15:8]**: Upper byte of data to be transmitted.							

NBURST: Number of bits in the next burst							
NBURST		0x50000038			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NTXBITS7	NTXBITS6	NTXBITS5	NTXBITS4	NTXBITS3	NTXBITS2	NTXBITS1	NTXBITS0
MSB							LSB
Bit7-0 **NTXBITS[7:0]**: Lower byte of the counter of number of bits to be transmitted in the next burst.							

TXCTRL2: Third TX Control Register for RF Transmission							
TXCTRL2		0x50000039			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NWAIT5	NWAIT4	NWAIT3	NWAIT2	NWAIT1	NWAIT0	NTXBITS9	NTXBITS8
MSB							LSB
Bit7-2 **NWAIT[5:0]**: Number of bits (time) between bursts. Bit1-0 **NTXBITS[9:8]**: Upper two bits of the counter of number of bits to be transmitted in the next burst.							

TXCLKDIV: Transmission Clock Divider							
TXCLKDIV		0x5000003A			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TXCDIV7	TXCDIV6	TXCDIV5	TXCDIV4	TXCDIV3	TXCDIV2	TXCDIV1	TXCDIV0
MSB							LSB
Bit7-0 **TXCDIV[7:0]**: Low byte of clock divider. (data rate is defined as 1.30Mbps/ TXCLKDIV)							

TXCTRL3: Fourth TX Control Register for RF Transmission							
TXCTRL3		0x5000003B			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TPATCH4	TPATCH3	TPATCH2	TPATCH1	TPATCH0	TXCDIV10	TXCDIV9	TXCDIV8
MSB							LSB

Bit7-3 **TPATCH[4:0]**: Lower 5 bits of the delay between disabling the PLL and turning off the bias.

0x000 = 0.8usec

:

0x7FF = 1.6msec

Bit2-0 **TXCDIV[10:8]**: Upper three bits of the Transmission Clock Divider. (data rate is 1.30Mbps/CDIV)

Example: If we want to transmit data at a rate of 19200bits/sec we should load **TXCDIV** with:

$$baud = \frac{1.25MHz}{\mathrm{TXCDIV}} = 19200 \Rightarrow \mathrm{TXCDIV} = \frac{1.25MHz}{19200} \cong 65$$

This in fact will generate a baud rate of ~19231bits/sec. (0.16% error)

NOTE: The equation for TPATCH is:

$$Tpatch = 78 \times 10^{-9} \times (TPATCH[10...0]) + 0.8 \times 10^{-6}$$

TXCTRL4: Fifth TX Control Register for RF Transmission							
TXCTRL4		0x5000003C			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TRAMP3	TRAMP2	TRAMP1	TRAMP0	TDET2_2	TDET2_1	TDET2_0	TDET1
MSB							LSB

Bit7-4 **TRAMP[3:0]**: Prevents the PA output power ramp up/down when the data bit changes. This delay is calculated as:

$$Delay = \frac{TRAMP}{SysClk}$$

TXCTRL4: Fifth TX Control Register for RF Transmission

This leads to the following values: (Assuming System Clock = 30 MHz)

0000 = 0usec

:

1000 = 0.27usec

:

1111 = 0.5usec

Bit3-1 **TDET2[2:0]**: Delay between enabling the PLL and enabling the power control DAC & power amplifier. This delay is calculated as:

$$Delay = \frac{TDET2}{1.25MHz} \times 1024$$

This leads to the following values:

000 = 0msec

:

100 = 3.28usec

:

111 = 5.73usec

Bit0 **TDET1**: Delay between enabling the bias and enabling the PLL:

0 = 1.6usec

1 = 8usec

TXCTRL5: Sixth TX Control Register for RF Transmission

TXCTRL5		0x5000003D			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
TDET3_2	TDET3_1	TDET3_0	TDET4_4	TDET4_3	TDET4_2	TDET4_1	TDET4_0
MSB							LSB

TXCTRL5: Sixth TX Control Register for RF Transmission
Bit7-5 **TDET3[2:0]**: Delay between enabling the power control DAC and enabling the power amplifier: $Delay = \frac{TDET3}{1.25MHz}$ This leads to the following values: 000 = 0usec : 100 = 3.2usec : 111 = 5.6usec Bit4-0 **TDET4[4:0]**: PA output power rising delay time: $RiseFall = \frac{TDET4}{1.25MHz} \times 2$

MDRCTRIM: Modulation RC filter capacitor trim							
MDRCTRIM		0x50010003			0x44		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDRC7	MDRC6	MDRC5	MDRC4	MDRC3	MDRC2	MDRC1	MDRC0
MSB							LSB
Bit7-0 **MDRC[7:0]**: ASK/OOK RC filter capacitor trim bits.							

PLLCTRL0: PLL Control Register 0							
PLLCTRL0		0x50018000			0x06		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NF7	NF6	NF5	NF4	NF3	NF2	NF1	NF0

PLLCTRL0: PLL Control Register 0							
MSB							LSB
Bit7-0 **NF[7:0]**: Fractional Portion of Feedback Divider lower bits.							

PLLCTRL1: PLL Control Register 1							
PLLCTRL1		0x50018001			0x2D		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
NX1	NX0	NF13	NF12	NF11	NF10	NF9	NF8
MSB							LSB
Bit7-6 **NX[1:0]**: Integer portion of feedback divider low bits.							
Bit5-0 **NF[13:8]**: Fractional Portion of feedback divider high bits.							

PLLCTRL2: PLL Control Register 2							
PLLCTRL2		0x50018002			0xCB		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
LPFTRIM1	LPFTRIM0	CPTRIM2	CPTRIM1	CPTRIM0	NX4	NX3	NX2
MSB							LSB
Bit7-6 **LPFTRIM[1:0]**: Low pass filter trim bits.							
Bit5-3 **CPTRIM[2:0]**: Charge pump trim bits.							
Bit2-0 **NX[4:2]**: Integer portion of feedback divider high bits.							

PLLCTRL3: PLL Control Register 3							
PLLCTRL3		0x50018003			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DF5	DF4	DF3	DF2	DF1	DF0	LPFTRIM3	LPFTRIM 2
MSB							LSB

Bit7-2 **DF[5:0]**: Frequency deviation control

Bit1-0 **LPFTRIM[3:2]**: Low pass filter trim bits.

PACTRL0: Power Amplifier Control Register 0							
PACTRL0		0x5001800E			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-	-	-	-	-	-	-	-
MSB							LSB

PACTRL1: Power Amplifier Control Register 1							
PACTRL1		0x5001800E			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
-	-	-	-	-	-	-	-
MSB							LSB

16.0 RF Switch

Herzog implements an RF Switch to allow for time duplexed reception and transmission sharing a single antenna.

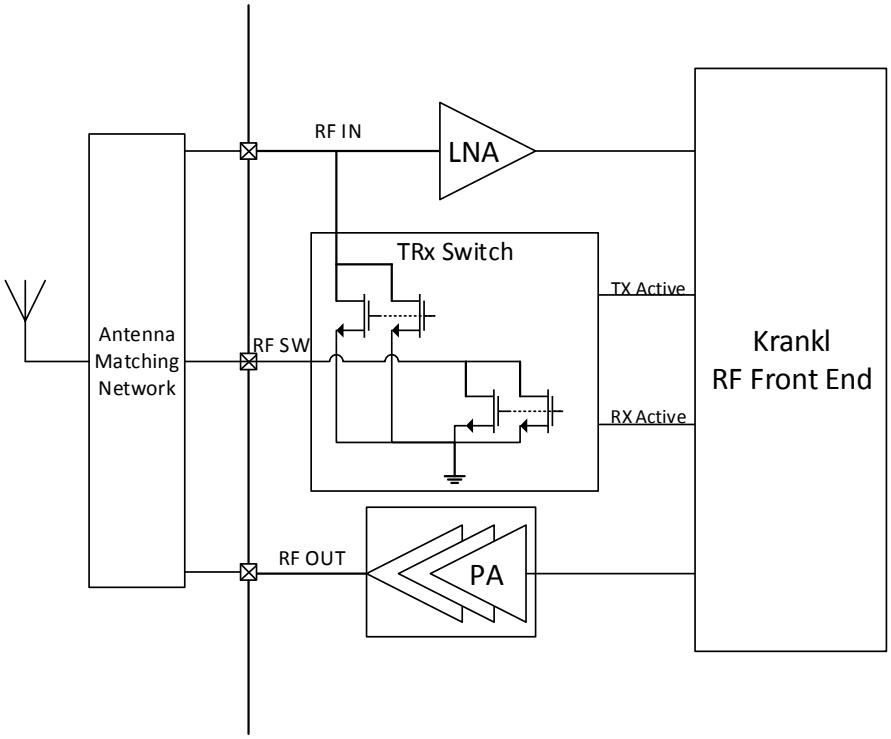


Figure 8 – RF Switching Circuit Concept

During transmission the TX Active signal will be enabled allowing for accurate matching of the antenna impedance with the PA output impedance. For proper reception the RX Active signal will be switched on which results in a matched impedance network for the RX bath.

17.0 Device Peripherals

17.1 General Purpose ADC 8 Bit

Herzog includes an 8-bit analog to digital converter with single ended input. The main features are described below:

- 8-bit resolution
- Single ended input
- Up to 80 ksps
- Configurable reference (VREF = VREFHI-VREFLO)
  - Either based on the bandgap voltage(VBG) or supply voltage(VDD)
  - Reference may be scaled in order to provide more resolution around a smaller input voltage range
- Maximum ADC input range is from 0V to its supply voltage
- It may read from a total of 20 channels (19 GPIOs and Supply Voltage)
- Using VBG as the reference, supply voltage can be measured in calibration mode

17.1.1 ADC Description

Herzog ADC uses a standard charge redistribution technique, with a single-ended input and internally generated positive and negative reference voltages. Each ADC accommodates 20 analog input channels. The user can select which input channel to be sampled by setting the ADCCHANNEL register. The ADC has its own internally generated reference voltages (VREFHI and VREFLO). The performance table is shown below.

Table 14 – ADC Performance Specification, Recommended Operating Conditions

Parameter	Conditions	Min.	Typ.	Max.	Unit
Conversion speed	Signal source resistance less than 20kΩ			80	ksps
Clock Frequency				1	MHz
Input voltage range		0		VDD	V
Resolution				8	bits
INL				1	LSB
DNL				1	LSB

There are several steps required for the user to use the ADC. The general sequence is described below:

- Step 1: Select the input channel to be measured
- Step 2: Configure ADC settings.
    a. Set ADC clock frequency.
    b. Configure references by programming the ADCREFHI, ADCREFLO, ADCPGN, and ADCREFS bits.
- Step 3: Start the ADC conversion.
- Step 4: Check the ADC status bit and read the data.

The following section will describe each configuration step in detail.

All GPIOs (PA[6:0], PC[1:0], PD[7:0], PE[7:0] and PF[2:0]) and the Supply Voltage reference are available as inputs to the ADC. The user can control which input is connected for the conversion by programming the control bits, **ADDCH[7:0]**, in the **ADDCHANNEL** register.

Code Example: Selecting PA0 for ADC input channel

```
ADC_Select_Channel( ADC_PA0 );
```

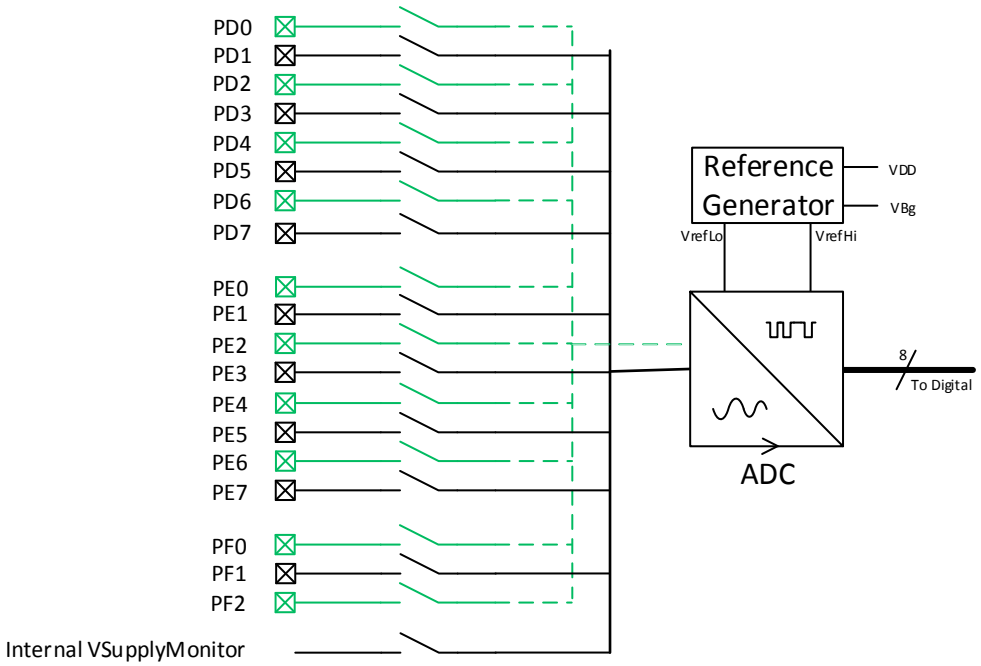


Figure 9 – ADC Multiplexing

17.1.2 ADC Clock and Sampling Period

The conversion algorithm has a basic period of 9 cycles (one for sampling, one for each bit). There is a two-cycle latency from the last bit measurement until the data becomes available to the user. Additionally, there is a single idle cycle to allow biasing before any conversion is initiated. Thus a single conversion will take 13 cycles.

The converter will use a single clock cycle to sample the input into an input capacitor. When the channel is selected, the source must drive the sample/hold capacitor through the source resistance of the signal to be measured. The sampling time varies with this source resistance. The input to ADC must have sufficiently low driving impedance and settling time to settle the input to within 1 LSB of the data conversion during the input sampling stage. An equivalent circuit and related equations are depicted in **Error! Reference source not found.**.

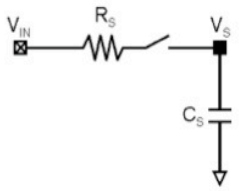
The ADC clock frequency can be programmed through the **ADCCLKDIV** register. As an example, if ADC clock is derived from a 12MHz oscillator divided by 16, the input has 1.3μs to settle. Since the maximum value of the internal sample/hold capacitor, C_s is 10pF, the maximum source resistance of the signal to be sampled to guarantee 8-bit performance can be calculated as below:

$$R_s = \frac{1.3\mu s}{10pF \times 5.5} = 24.2k\Omega$$

If the source impedance is larger, the user can reduce the ADC clock frequency in order to retain the conversion accuracy.

Code Example: Configure ADC clock to 12MHz RC oscillator frequency divided by 16.

```
ADC_ClkDiv (16);
```



$$V_S = V_{IN}(1 - e^{-\frac{t}{\tau}}) \quad , where\ \tau = R_S C_S$$

V_S *is within 1 LSB of* V_{IN} *after* $t = 5.5\tau$

Figure 10 – ADC Input Settling Time

17.1.2.1 Configuration of Reference Voltages for the ADC

The ADC can generate its own reference voltages (VREFHI and VREFLO) from two different sources, its supply voltage (VDD) or the internal bandgap reference voltage (VBG). The **ADCREFS** bit in the **ADCREG3** register selects the source. Once the reference source is selected, the reference voltages can be programmed through the **ADCREFHI**, **ADCREFLO**, and **ADCPGN** bits according to **Error! Reference source not found.**.

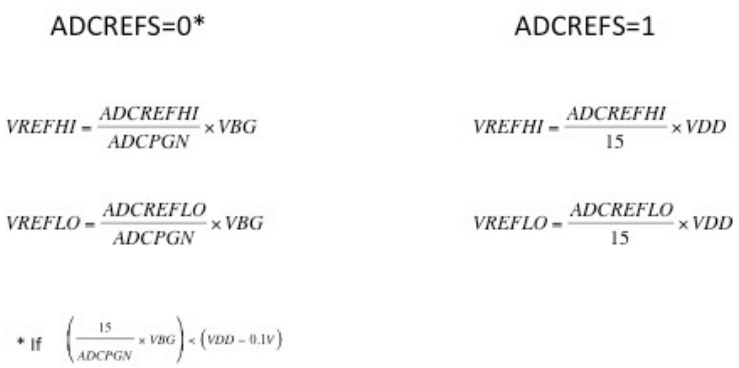


Figure 11 – ADC Reference Voltage

Once the reference voltages are established, the ADC conversion equation for input voltage (VIN) can be defined as:

$$ADCDT = floor\left(255 \times \frac{(VIN - VREFLO)}{(VREFHI - VREFLO)}\right)$$

Here are a few examples:

- Example 1: In a system operating with VDD=3V, there is a signal that moves between 0V and 2.94V. In this case it is recommended that VDD be selected as the reference source and the following setting be made, **ADCREFH**=15, **ADCREFL**=0, and **ADCPGN**=15. This selection would allow for the maximum range of measurement (0V to VDD).
  - Code Sample `ADC_Reference_Config( ADC1, 15, 0, 15, ADCREFVDD);`
- Example 2: In a system operating with VDD=3V and VBG= 1.21V, there is a signal that moves between 0V and 2.5V. In this case VBG can be selected as the reference source with settings of **ADCREFH**=13, **ADCREFL**=0 and **ADCPGN**=7. This configuration would allow the signal range of measurement to be 0V to 2.6V:
  - Code Sample `ADC_Reference_Config( 13, 0, 7, ADCREFBG);`

- Example 3: In a system operating with VDD=3V and VBG=1.21V, there is a signal that moves between 1.71V and 2.2V. In this case selecting VBG as the reference source and settings of **ADCREFH**=15, **ADCREFL**=11, and **ADCPGN**=8 we can achieve higher resolution:
    - Code Sample `ADC_Reference_Config( 15, 11, 8, ADCREFBG);`

The resolution in Example 3 can be calculated as follows:

$$RESOLUTION = \left(\frac{VREFHI - VREFLO}{255}\right) = \left(\frac{\left(\frac{15}{8}\times 1.21\right) - \left(\frac{11}{8}\times 1.21\right)}{255}\right) = \frac{2.27 - 1.66}{255} = 2.38mV$$

Note that in this particular case we have the 8-bit ADC effectively generating a digital value with the precision of a 10-bit ADC operating from 0V to VDD.

It is clear from the examples how flexible the ADC can be in a range of applications. The user can devise several schemes to cleverly measure the range of signal of interest and then narrow the reference values to get the optimum resolution if the conversion time is acceptable.

17.1.2.2 ADC Start and Status

Before starting the conversion, the ADC must be enabled and biased. The ADC is enabled by the **ADCEN** bit in register **ADCREG3**. The START bit (**ADCSTART**) starts the conversion process. Once completed the value of the conversion is loaded into the **ADCDATA** register.

Code Example: Enable the converter and start conversion

```
ADC_Enable(ADCEN);
ADC_Start();                              //ADC enabled and start
while ( ADC_ConversionComplete() == 1 );  //Wait until completed
```

17.1.3 ADC Registers

The following registers control the behavior of the ADC:

Table 15 – ADC Control Register Map				
Address	Register Name	Description	Reset Value	Reference
0x5000000C	ADCCHANNELS			
0x5000000D	ADCSTART			
0x5000000E	ADCDATA			
0x5000000B	ADCCLKDIV			
0x50018008	ADCTRIM0		0xF0	
0x50018009	ADCTRIM1		0x9F	
0x5001800A	ADCTRIM2		0x00	
0x5001800B	ADCTRIM3		0x00	

A more detailed description of each register follows below:

ADCCHANNELS: ADC Channel Select Register							
ADCCHANNELS		0x5000000C			0x00		
Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
-	-	-	ADC_CH4	ADC_CH3	ADC_CH2	ADC_CH1	ADC_CH0
MSB							LSB
Bit3-0 **ADC_CH[4:0]**: Channel Selection for ADC:							

ADCCHANNELS: ADC Channel Select Register
ADC_CH[4:0] 00000 = PD0 00001 = PD1 … 00110 = PD6 00111 = PD7 01000 = PE0 01001 = PE1 … 01110 = PE6 01111 = PE7 01000 = PF0 01001 = PF1 01010 = PF2

ADCSTART: ADC Start Register							
ADCSTART		0x5000000D			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	R/W
–	–	–	–	–	–	–	START
MSB							LSB
Bit0 **START**: Writing one starts the conversion. Reading returns the status of conversion; '0' means conversion Is finished and '1' means the conversion is ether pending or in progress							

ADCDATA: ADC1 Result Register							
ADC_DATA		0x5000000E			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ADC1DT7	ADC1DT6	ADC1DT5	ADC1DT4	ADC1DT3	ADC1DT2	ADC1DT1	ADC1DT0
MSB							LSB
Bit7-0 **ADC1DT[7:0]**: ADC Result							

ADCCLKDIV: ADC Clock Divider Control Register							
ADCCLKDIV		0x5000000B			0x6F		
R/W	Reserved	R/W	R/W	R/W	R/W	R/W	R/W
SWMODE	-	ADCDIV0	ADCDIV0	ADCDIV0	ADCDIV2	ADCDIV1	ADCDIV0
MSB							LSB

Bit5-0 **ADCDIV[5:0]:** ADC Clock divider

00000 = System Clock/2

00001 = System Clock/4

11111 = System Clock/64

Bit7 **SWMODE:** ADC Mux Switch Mode

0 = ADC Mux switch is open after the conversion

1 = ADC Mux switch is closed to the selected input after the conversion

ADCTRIM0: ADC Trim0 Register							
ADCTRIM0		0x50018008			0xF0		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PHI5	PHI4	PHI3	PHI2	PLO5	PLO4	PLO3	PLO2
MSB							LSB

Bit7-4 **PHI[5:2]:** ADC Reference High Setting

Bit3-0 **PLO[5:2]:** ADC Reference Low Setting

ADCTRIM3: ADC Trim3 Register							
ADCTRIM3		0x50018009			0x9F		
R/W	R/W	R/W	R/W	R/W	R/W	Reserved	R/W
ADC_SW1	ADC_SW0	ADC_CAL	ADC_FulRg	PGN5	PGN4	PGN3	PGN2
MSB							LSB

Bit7-6 **ADC_SW[1:0]:** ADCs Enable Bit.

00 = Correlated double sampling off

01 = Input offset calibration on

1x = Correlated doubling sampling on (default)

Bit5 **ADC_CAL:** ADC Calibration

0 = Normal

1 = Calibration mode

Bit4 **ADC_FulRg:** ADC output range

0 = output reference based on fixed band gap voltage range

1 = output voltage full range based on VDD

Bit3-0 **PGN[5:2]:** ADC Reference DC gain

17.2 PIR Interface

The PIR Interface is intended for connection to an external Pyro Infra Red (PIR) sensor, for the purpose of human motion detection. It consists of two signal conditioning operational amplifiers, which are intended to be configured, as bandpass filters with approximately 40dB of gain per stage between 0.1 and 10Hz. The gain and roll-off are set by external components, due to the low frequencies and low bias currents required. The output of the filter stages is DC coupled to a window comparator, which determines if the wanted signal exceeds a preset threshold. The output of the window comparator feeds a digital section where its pulse length is measured. If its pulse length exceeds a programmable value, then an event is signaled. If an event is signaled a second digital block times a period through which new events are ignored. This second period is known as the Inhibit length, which may also be programmed. This block will be enabled after power on reset, but may be disabled by software. This block runs directly from the battery voltage with no internal regulation required. The block diagram is shown below:

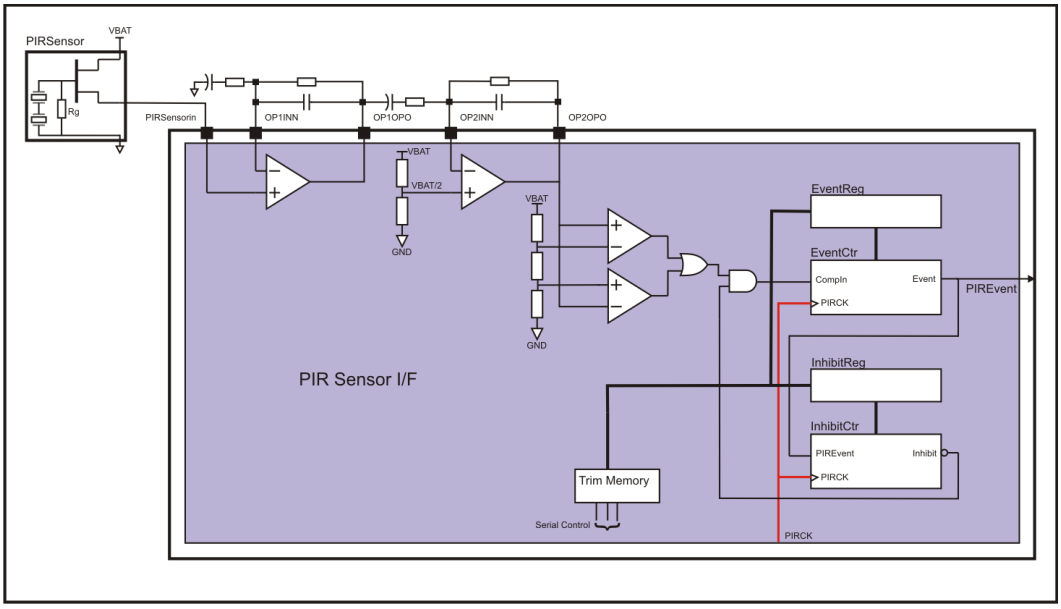


Figure 12 –PIR Sensor interface Block Diagram

A timing diagram showing simple PIR detection is shown below:

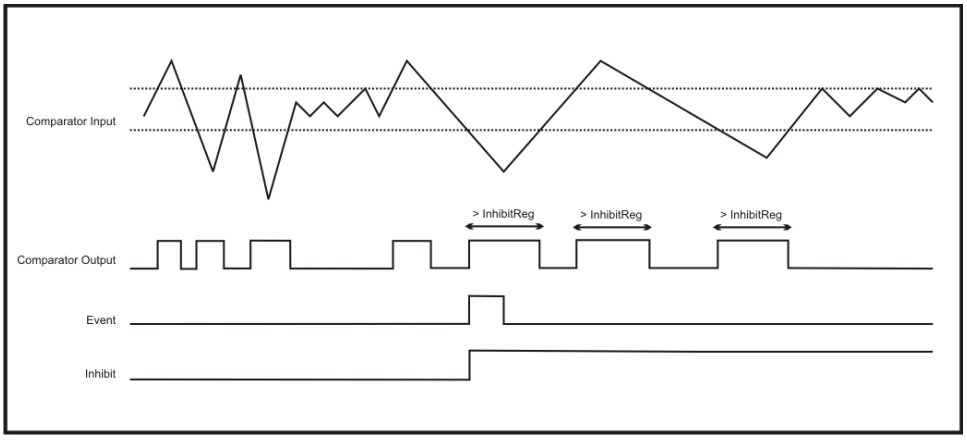


Figure 13 –PIR Sensor Timing Diagram

The register settings for the PIR sensor interface are shown below

Table 16 – PIR Interface Register Map

Address	Name	Description
tbd	PIR_En	PIR sensor interface enable, Logic1 is enabled, Logic0 is disabled
tbd	EventCtr[9:0]	Sets the length of the comparator pulse required to signify movement detection
tbd	InhibitCtr[3:0]	Sets the length of the time after an event during which further events are ignored
tbd	Event	Bit signifying that a movement event has occurred

17.2.1.1 First Stage Operational Amplifier

The first stage operational amplifier is fully uncommitted, with inverting, non-inverting and output available at pins. This stage is intended to be configured as a bandpass filter with around 40dB of in-band gain. The input is P-type only, and as such has a limited common-mode input range. As it is P-type only, noise and offset are improved compared to the second stage. The output stage will support rail to rail operation. The quiescent current of this stage is around 500nA.

Table 17 – Recommended Operating Conditions First Stage Op Amp

Name	Conditions	Min.	Typ.	Max.	Unit
Vbat voltage		2.5	3.0	3.2	V
Operating Temp.		-10	30	60	°C
Current consumption	Enabled		500	tbd	nA
Gain bandwidth product		tbd	15		kHz
Slew Rate				3	V/ms
Input referred offset			±3	±7	mV
Input referred noise	At 0.1Hz		500	800	nV/√Hz
Input referred noise	At 10Hz			100	nV/√Hz
CMRR		60	75	80	dB
Input voltage range		-0.5		1.8	V
Open Loop DC gain			100		dB
Output voltage Range		10		VBAT-10	mV
Linear Output voltage Range		100		VBAT-100	mV

17.2.1.2 Second Stage Operational Amplifier

The second stage operational amplifier is semi uncommitted, with only inverting and output available at pins. The non-inverting input is internally biased to VBAT/2. This stage is also intended to be configured as a bandpass filter with around 40dB of in-band gain. The input is both N-type and P-type, which means it will operate rail-to-rail. The output stage will also support rail to rail operation. The quiescent current of this stage is also around 700nA.

Table 18 – Recommended Operating Conditions Second Stage Op Amp					
Name	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
Vbat voltage		2.5	3.0	3.2	V
Operating Temp.		-10	30	60	°C
Current consumption	Enabled		700	tbd	nA
Gain bandwidth product		tbd	15		kHz
Slew Rate				3	V/ms
Input referred offset			±5	±15	mV
Input referred noise	At 0.1Hz		800	tbd	nV/√Hz
Input referred noise	At 10Hz			tbd	nV/√Hz
CMRR		60	75	80	dB
Input voltage range		-0.5		VBAT	V
Open Loop DC gain			100		dB
Output voltage Range		10		VBAT-10	mV
Linear Output voltage Range		100		VBAT-100	mV

17.2.1.3 Window Comparator

The window comparator consists of two dynamic comparators, which are clocked with a divided version of the RC oscillator clock. As the comparators are dynamic, they consume negligible current. The threshold for the comparators are set at vdd/3 and 2vdd/3 via a resistor string. If the wanted signal exceeds the higher threshold or is lower than the lower threshold then the comparator will output a logic hi.

Table 19 – Recommended Operating Conditions					
Name	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
Vbat voltage		2.5	3.0	3.2	V
Operating Temp.		-10	30	60	°C
Low Threshold		VBAT/3-5%	VBAT/3	VBAT/3+5%	V
High Threshold		2*VBAT/3-5%	2*VBAT/3	2*VBAT/3+5%	V
Current consumption	Enabled			100	nA

17.2.1.4 Event and Inhibit Counters

The Event Counter counts as long as one of the comparator outputs is a logic high. If the pulse length from the comparator output is longer than a pre-programmed value, the event flag is set. If the event flag is set, the Inhibit counter begins to count for a pre-programmed period, during which all subsequent event alarms are ignored. The inhibit counter may be disabled via software, in which case the micro must decide whether to ignore subsequent alarms.

Table 20 - Recommended Operating Conditions Event controller

Name	Conditions	Min.	Typ.	Max.	Unit
Vbat voltage		2.5	3.0	3.2	V
Operating Temp.		-10	30	60	°C
Event length		tbd		tbd	s
Inhibit length		tbd		tbd	s

17.3 Herzog Serial Interfaces

17.3.1 UART

Herzog includes a UART (**U**niversal **A**synchronous **R**eceiver **T**ransmitter) module. The main characteristics are defined below:

- Interrupt available for transmission, reception and error events
- Reception timeout timer
- Programmable break reception and transmission
- Programmable parity with “sticky” parity option
- Selectable number of bits from 5 to 8
- Selectable number of stop-bits: 1, 1 ½, 2
- Programmable loop-back
- Swappable TXD and RXD (PE[4] and PE[5])
- Transmitter Polarity selection

17.3.1.1 UART Operation

The UART protocol requires two wires (UTXD and URXD). PE[4] and PE[5] are configured as UTXD and URXD when the MDUART bit is set in the pin configuration register, PCONF. In order to use the UART, the following steps must be followed:

- Step 1: Select the pins position (normal or swapped) of the interface and also its polarity. The normal position (not swapped) is TX= PE[4] and RX = PE[5].

Code Example: Selecting UART with normal polarity and swapped: (UART pins swapped: TX=PE[5] and RX = PE[4])

```
UART_Setup(UARTSWAP_EN, UARTPOL_NORMAL, UART_MODE_EN);
```

- Step 2: Define the following parameters:
  - Loop back: Used mainly in tests, internally connects the output to the input.
  - Break enable: Pulls the output down while asserted, raising the output once de-asserted.
  - Sticky parity: Forces the parity to stay stable in one direction.
  - Even/Odd parity selection and enable: Selection and enable of Even or Odd parity bits.
  - Number of stop bits: Selection of 1 (default), 1½ (5-bit communications only) or 2 stop bits.
  - Data size in bits (5,6,7,8): Selection of the number of bits used in the communication

Code example: Setting the above parameters: (no loop-back, no break signal, no sticky parity, no parity, one stop-bit, 8-bit communications)

```
UART_Ctrl(UART_LBDIS, UART_BREAKDIS, UART_STPDIS,UART_ODDEN, UART_PARDIS, \
          UART_10STOP, UART_8BITS);
```

Define the baud rate. The baud rate is calculated as follows: (UARTDIV is a 16-bit register)

$$Baud = \frac{Fclk}{16 * (UARTDIV + 1)}$$

Assuming a 4MHz system clock the following table provides some register values, baud rates and related errors:

Table 21 – UART baud rates, divider values and errors

Baud	UARTDIV	Real Baud	Error (%)
300	832	300.1	0.04
600	416	599.5	0.08
1200	207	1202	0.16
2400	103	2404	0.16
4800	51	4808	0.16
9600	30	9615	0.16
19200	12	19231	0.16

Code Example: Setting the UART to operate at 9600 baud:

```
UART_BaudRateDivider(30);
```

- Step 3: Enable the UART and its interrupt: The UART may generate an interrupt for events related to:
  - Transmission completed.
  - Reception: Timeout of ~40 bit-times without reception, and data received.
  - Errors detected: Framing error, parity error, and overrun error.
  - Break signal detected (received).

Code example: Enabling the UART with no timeout interrupts, errors, transmission and reception interrupts with interrupts once 1 byte is received.

```
//Timeout interrupt disabled, Error enabled, TX enabled, RX enabled

UART_Interrupt_Control(UART_TOUTDIS,UART_ERREN,UART_TXEN,UART_RXEN);
```

```
//UART enabled

      UART_Ctrl1( UARTEN);

//Enabling interrupt from UART at the microcontroller

      NVIC_EnableIRQ(UART_IRQn);         //Enable UART interrupt
```

Processing of the UART interrupt:

```
void UART_Handler( void )           // IRQ 8 UART
{
      switch(UART_Interrupt_Status())
      {
            case  UART_ERROR:
                  //Process reception error here
                  SWITCH(UART_CheckError())
                  {
                        case UART_FRAMING_ERROR: // Process this error here
                              break;
                        case UART_PARITY_ERROR: // Process this error here
                              break;
                        case UART_OVERRUN_ERROR: // Process this error here
                              break;
                        case 0: // No error, exit
                        default:
                              break;
                  }
                  break;
            case  UART_RXRDY:
                  //Process data received
                  mydata = *UARTDATA;     //Read data received by uart into mydata
                  break;
            case  UART_TIMEOUT:
                  //Process reception timeout
                  break;
            case  UART_TXDONE:
                  //Process transmission complete
                  break;
            case  UART_NOINT: //No int.
            default:
            //If no interrupt asserted or something else happened nothing to do
                        break;
      }
}
```

17.3.2 UART Registers

The following registers for the UART are defined in Herzog:

Table 22 – UART Control Register Map				
Address	Register Name	Description	Reset Value	Reference
0x50000010	UARTDATA	UART Data Register	0x00	
0x50000011	UARTICTRL	UART Interrupt Control Register	0x00	
0x50000012	UARTLCTRL	UART Line Control Register	0x00	
0x50000013	UARTEN	UART Enable Register	0x00	
0x50000014	UARTLSTAT	UART Line Status	0x00	
0x50000016-7	UARTDIV	UART Baud Rate Divider	0x0000	

UARTDATA: UART Data Register							
UARTDATA		0x50000010			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
UARTD7	UARTD6	UARTD5	UARTD4	UARTD3	UARTD2	UARTD1	UARTD0
MSB							LSB
Bit7-0 UARTD [7:0]: UART data, both received and to be transmitted.							

UARTICTCL: UART Interrupt Control Register							
UARTICTCL		0x50000011			0x00		
R	R	R	R	R/W	R/W	R/W	R/W
UISTTS3	UISTTS2	UISTTS1	UISTTS0	UTOUTIEN	URXERREN	UTXIEN	URXIEN
MSB							LSB

UARTICTCL: UART Interrupt Control Register
Bit7-4 **UISTTS [3:0]:** UART Interrupt status: 0001 = No Interrupt asserted 0010 = Transmission completed 0100 = Data received 0110 = Reception error 1100 = Reception timeout (~40 bit-time)
Bit3 **UTOUTIEN**: UART time-out interrupt enable bit: 0 = Time-out interrupt disabled 1 = Time-out interrupt enabled
Bit2 **URXERREN**: UART reception error interrupt enable bit: 0 = Reception error interrupt disabled 1 = Reception error interrupt enabled
Bit1 **UTXIEN**: UART transmission completed interrupt enable bit: 0 = Transmission completed interrupt disabled 1 = Transmission completed interrupt enabled
Bit0 **URXIEN**: UART reception interrupt enable bit: 0 = Reception interrupt disabled 1 = Reception interrupt enabled

UARTCTRL: UART Line Control Register							
UARTLCTRL		0x50000012			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
ULOOPEN	UBREAKEN	USTICKEN	UPARITY	UPAREN	USTOP	USTOP	USIZE
MSB							LSB
Bit7 **ULOOPEN**: UART loop back enable: 0 = UART loop back disabled 1 = UART loop back enabled							

UARTCTRL: UART Line Control Register	
Bit6	**UBREAKEN:** UART break enable: 0 = UART break disabled 1 = UART break enabled
Bit5	**USTICKEN**: UART sticky parity enable bit: 0 = Sticky parity disabled 1 = Sticky parity enabled
Bit4	**UPARITY**: UART parity bit: 0 = Odd parity 1 = Even parity
Bit3	**UPAREN**: UART parity enable bit: 0 = Parity disabled 1 = Parity enabled
Bit2	**USTOP**: UART stop bit: 0 = One stop bit 1 = If a 5-bit transmission it selects1.5 stop bits, otherwise 2 stop bits (6, 7 and 8 bits)
Bit1-0	**USIZE**: UART transmission size: 00 = 5-bit data 01 = 6-bit data 10 = 7-bit data 11 = 8-bit data

UARTCTRL1: UART Enable Register							
UARTCTRL1		0x50000013			0x00		
Reserved	Reserved	Reserved	Reserved	R/W	Reserved	Reserved	Reserved
-	-	-	-	UARTEN	-	-	-
MSB							LSB

UARTCTRL1: UART Enable Register
Bit3 **UARTEN**: UART enable: 0 = UART disabled 1 = UART enabled

UARTSTATUS: UART Line Status Register							
UARTSTATUS		0x50000014			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
UERR	UTXEMPTY	UTXFFEMPTY	UBREAKINT	UFRMERR	UPRTYERR	UOVRUNERR	UDTRDY
MSB							LSB

Bit7 **UERR**: UART error:
0 = No error
1 = Error in UART

Bit6 **UTXEMPTY:** UART transmission empty:
0 = UART transmitter not empty
1 = UART transmitter empty

Bit5 **UTXFFEMPTY**: UART transmission buffer empty:
0 = TX buffer not empty
1 = TX buffer empty

Bit4 **UBREAKINT**: UART break interrupt:
0 = No break interrupt
1 = Break interrupt

Bit3 **UFRMERR**: UART framing error:
0 = UART no framing error
1 = UART framing error

Bit2 **UPRTYERR**: UART parity error:
0 = No parity error
1 = Parity error

Bit1 **UOVRUNERR**: UART overrun error:

UARTSTATUS: UART Line Status Register	
	0 = No overrun error
	1 = Overrun error
Bit0	**UDTRDY**: UART data ready:
	0 = No data ready (reception)
	1 = Data ready (reception)

UARTDIV: UART Baud Rate Divider Register (16-bit)							
UARTDIV		0x50000016			0x0000		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
UDIV7	UDIV6	UDIV5	UDIV4	UDIV3	UDIV2	UDIV1	UDIV0
UDIV15	UDIV14	UDIV13	UDIV12	UDIV11	UDIV10	UDIV9	UDIV8
MSB							LSB
Bit15-0 UDIV [15:0]: UART clock divider							

17.3.3 SPI Interface

The Serial Peripheral Interface (SPI) is a synchronous full-duplex serial interface. It communicates in master/slave mode where the master initiates the data transfer. In Herzog, the SPI is implemented as a master. The module is compatible with an industry standard SPI interface. There are many references available, but a simple overview can be found at:

http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus

Herzog SPI module's main features are defined below:

- Compatible with Industry Standard SPI interface
- Four bytes deep reception FIFO
- Four bytes deep transmission FIFO
- Interrupt upon events related to transmission, reception and error:
  - Write Collision
  - Transmission FIFO full and empty
  - Reception FIFO full and empty

The SPI protocol requires four wires (SCK, MISO, MOSI, and SS). The pins PE[3:0] are configured as the SPI bus when the MDSPI bit is set in the pin configuration register, PCONF. The following table describes how each pin is connected:

Table 23 – SPI Interface Signals

Name	Pin Number	Pin Name	Comments
MISO	26	PE1	Mater In Slave Out
MOSI	28	PE3	Master Out Slave In
SCLK	27	PE2	Serial Clock
SS	30	PE0	Slave Select

17.3.3.1 SPI Operation

Only the master mode is implemented in Herzog. Herzog configures the clock frequency and generates the serial clock (SCK) for the interface. The data transfer is synchronous through SCK. The SPI is a full-duplex system; data is transmitted and received simultaneously. Herzog sends the information to the slave device through the MOSI line and receives the data through MISO line. The CPOL and CPHA bits in the SPI control register determine when to sample the data.

When CPOL=0, the base value of the clock is logic '0'. In this case, if CPHA=0, data is captured on the rising edge of SCK and data is propagated on the falling edge of SCK. For CPHA=1, data is captured on the falling edge of SCK and data is propagated on the rising edge of SCK.

If CPOL=1, the base value of clock is logic '1'. In this case, if CPHA=0, data is captured on the falling edge of SCK and data is propagated on the rising edge of SCK. For CPHA=1, data is captured on the rising edge of SCK and data is propagated on the falling edge of SCK.

The timing diagram is shown below:

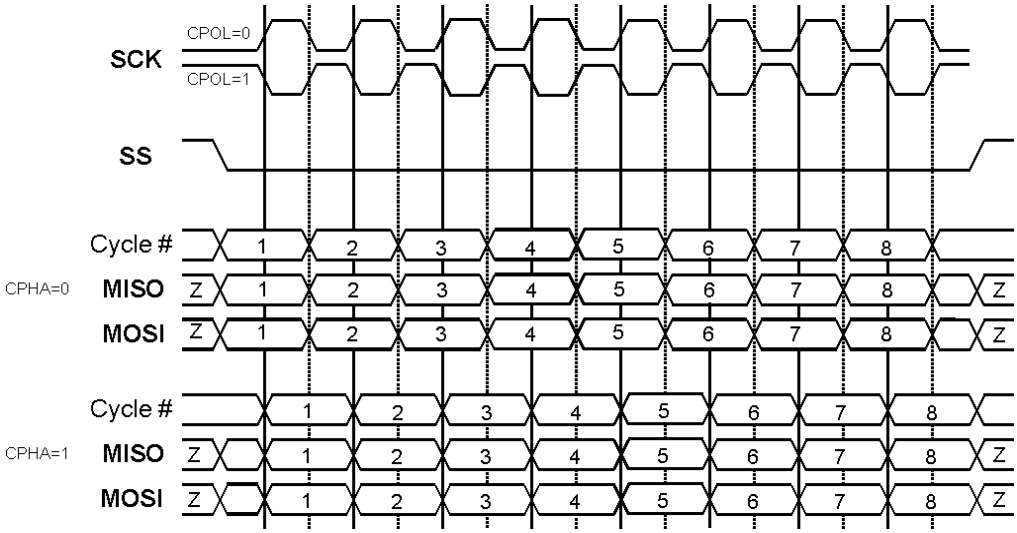


Figure 14 – SPI Timing Diagram

After a desired configuration is set through configuration registers, a transfer is initiated by writing to the SPI Data Register (SPDR). The data is input into a 4-deep FIFO before it is transmitted. When the data is transmitted, the slave also transmits the data simultaneously for Herzog to receive. The received data is stored in a separate 4-deep FIFO. The data is accessed by reading SPDR register.

To operate it properly the following steps must be performed:

- Step 1: Configure and enable the SPI: Select if the interrupt is enabled, set the polarity, phase and the clock divider:

  Code Example: Enabling the SPI interface with interrupt enabled, clock polarity negative (base value = 0), phase1 (data captured on the clock's falling edge and propagated on the rising edge), and divider = 2.

```
SPI_Config(SPI_INT_EN,SPI_EN,SPI_CPOL_NEG,SPI_PHASE1,2);
```

- Step 2: Enable the interrupt (if required):

  Code Example:

```
//Enabling interrupt from SPI at the microcontroller

NVIC_EnableIRQ(SPI_IRQn);          //Enable SPI interrupt
```

- Step 3: Process the Interrupt (if required) and detect the reason for the interrupt (error, transmission or reception related and act accordingly):

  Code Example: Processing SPI interrupt:

```
void SPI_Handler( void )            // IRQ A SPI
{
      uint8_t status;
      if ( (status = SPI_ReadStatus() ) & SPI_INT_FLAG )
      {
            if (status == SPI_WCOL)
            {
            //Process write collision (data is written to the SPI data
            //register while a SPI data transfer is in progress)

            }
            if (status == SPI_TX_FIFO_FULL)
            {
            //Process Transmission FIFO full
            }
            if (status == SPI_TX_FIFO_EMPTY)
            {
            //Process end of transmission of data previously
            //in transmission FIFO
                  *SPIDATA = outdata[j++];
                  *SPIDATA = outdata[j++];
                  *SPIDATA = outdata[j++];
            }
            if (status == SPI_RX_FIFO_FULL)
            {
            //Process reception FIFO full, normally by reading
            //all bytes of data
                  for ( I = SPI_ReadRxFifoSize(); I > 0; i--)
                        mydata[i++] = *SPIDATA;
            //Other processing here
            }
            if (status == SPI_RX_FIFO_EMPTY)
            {
            //Process when no more information is available (received)
            }
      }
}
```

17.3.4 SPI Registers

Table 24 – SPI Control Register Map				
Address	Register Name	Description	Reset Value	Reference
0x5000001C	SPCR	SPI Control Register	0x10	
0x5000001D	SPSR	SPI Status Register	0x00	
0x5000001E	SPDR	SPI Data Register	0x00	
0x5000001F	SPER	SPI Extension Register	0x00	

SPCR: SPI Control Register							
SPCR		0x5000001C			0x10		
R/W	Reserved	Reserved	R	R/W	R/W	R/W	R/W
SINTE	-	-	MSTR	CPOL	CPHA	SCKSTD1	SCKSTD0
MSB							LSB

Bit7 **SINTE**: SPI Interrupt enable

0 = Interrupt is disabled

1 = Interrupt is enabled

Bit4 **MSTR:** Master Mode Select Bit

SPI is always in master mode in Herzog, and therefore, it is always set to logic '1'.

Bit3 **CPOL** SPI clock polarity

0 = The base value of the clock is zero

1 = The base value of the clock is one

Bit2 **CPHA**: SPI clock phase

0 = data is captured on clock transition from base and data is propagated on the clock transition to base

1 = data is captured on clock transition to base and data is propagated on the clock transition from base

Bit1-0 **SCKSTD[1:0]**: SPI standard clock divider selection

Please refer to SPER register for system clock

SPSR: SPI Status Register							
SPSR		0x5000001D			0x00		
R/W	R/W	Reserved	Reserved	R/W	R/W	R/W	R/W
SINTF	SWCOL	-	-	STXFF	STXFE	SRXFF	SRXFE
MSB							LSB

Bit7 **SINTF**: SPI interrupt flag

0 = Interrupt not asserted

1 = Interrupt asserted

Bit6 **SWCOL:** SPI write collision is set when the SPDR register is written to while the transmit FIFO is full

0 = No collision

1 = collision

Bit3 **STXFF**: SPI transmit FIFO full

0 = transmit FIFO not full

1 = transmit FIFO full

Bit2 **STXFE**: SPI transmit FIFO empty

0 = transmit FIFO not empty

1 = transmit FIFO empty

Bit1 **SRXFF**: SPI reception FIFO full

0 = reception FIFO not full

1 = reception FIFO full

Bit0 **SRXFE**: SPI reception FIFO empty

0 = reception FIFO not empty

1 = reception FIFO empty

SPDR: SPI Data Register							
SPDR		0x5000001E			0xXX		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
SPID7	SPID6	SPID5	SPID4	SPID3	SPID2	SPID1	SPID0
MSB							LSB
Bit7-0 **SPID[7:0]**: SPI data, used in both transmission and reception							

SPER: SPI Extension Register							
SPER		0x5000001F			0x00		
R/W	R/W	Reserved	Reserved	R/W	R/W	Reserved	Reserved
SICNT1	SINCT0	-	-		SPE	SCKEXT1	SCKEXT0
MSB							LSB

Bit7-6 **SICNT[1:0]**: SPI Interrupt Counter Bits

00 = SINTF is set after every completed transfer

01 = SINTF is set after two completed transfers

10 = SINTF is set after three completed transfers

11 = SINTF is set after four completed transfers

Bit2 **SPE**: SPI Enable

0 = SPI module is disabled

1 = SPI module is enabled

Bit1-0 **SCKEXT[1:0]**: SPI extended clock divider

SCKSTD	**SCKEXT**	**Result Clock Divider**
00	00	= System Clock/2
01	00	= System Clock/4
10	00	= System Clock/16
11	00	= System Clock/32
00	01	= System Clock/8

SPER: SPI Extension Register		
01	01	= System Clock/64
10	01	= System Clock/128
11	01	= System Clock/306
00	10	= System Clock/512
01	10	= System Clock/1024
10	10	= System Clock/2048
11	10	= System Clock/4096
xx	11	= Reserved

18.0 GPIOs

Herzog provides 19 general-purpose 3.3V I/O pins with the following technical specification

Table 25 – General Purpose IOs						
I/O Type	**Name**	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
3V3IO	V_{IL}	Threshold Low		1.65		V
		Threshold High		4		V
	V_{IH}	Threshold Low		1.65		V
		Threshold High		1.65		V
	I_{OL}			10		mA
	I_{OH}			10		mA

18.1 GPIO Registers

The following registers control the behavior of the GPIO pins:

PORTD: Port D input and output register. When a bit is selected as output the corresponding pin will reflect the register's bit. When a bit is selected as input the bit will reflect the condition of the pin.							
PORTD		0x50000062			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
MSB							LSB
Bit7-0 **PD[7:0]**: Port D register bits. 0 = Pin state is '0' 1 = Pin state is '1'							

PORTE: Port E input and output register. When a bit is selected as output the corresponding pin will reflect the register's bit. When a bit is selected as input the bit will reflect the condition of the pin.							
PORTE		0x50000063			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
MSB							LSB
Bit3-0 **PE[7:0]**: Port E register bits. 0 = Pin state is '0' 1 = Pin state is '1'							

PORTF: Port F input and output register. When a bit is selected as output the corresponding pin will reflect the register's bit. When a bit is selected as input the bit will reflect the condition of the pin.							
PORTF		0x50000064			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W
-	-	-	-	-	PF2	PF1	PF0
MSB							LSB
Bit3-0 **PE[2:0]**: Port E register bits. 0 = Pin state is '0' 1 = Pin state is '1'							

PMODE: Pin Mode register.							
PMODE		0x50000065			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W
-	-	-	upswap	-	modeSPI	modeUART	utxpol
MSB							LSB

PMODE: Pin Mode register.
Bit0 **utxpol**: UART TX polarity 0 = non inverted 1 = inverted
Bit1 **modeUART**: pin configuration 0 = PE4 and PE5 as GPIOs 1 = PE4 and PE5 for UART
Bit2 **modeSPI**: pin configuration 0 = PE0-3 as GPIOs 1 = PE0-3 for SPI
Bit4 **upswap**: UART Pin Swap 0 = PE4 as UART_RX and PE5 as UART_TX 1 = PE5 as UART_RX and PE4 as UART_TX

PDINT: Port D Interrupt Enable Register.							
PDINT		0x50000068			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PDINTM7	PDINTM6	PDINTM5	PDINTM4	PDINTM3	PDINTM2	PDINTM1	PDINTM0
MSB							LSB
Bit7-0 **PDINTM[7:0]**: Port D Interrupt Enable bit. 0 = Pin interrupt disabled 1 = Pin interrupt enabled							

PEINT: Port E Interrupt Enable Register.							
PORTE		0x50000069			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
PEINTM7	PEINTM6	PEINTM5	PEINTM4	PEINTM3	PEINTM2	PEINTM1	PEINTM0

PEINT: Port E Interrupt Enable Register.							
MSB							LSB
Bit7-0 **PEINTM[7:0]**: Port E Interrupt Enable bit. 0 = Pin interrupt disabled 1 = Pin interrupt enabled							

PFINT: Port E Interrupt Enable Register.							
PORTF		0x5000006A			0x00		
Reserved	Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W
-	-	-	-	-	PFINTM2	PFINTM1	PFINTM0
MSB							LSB
Bit2-0 **PFINTM[2:0]**: Port F Interrupt Enable bit. 0 = Pin interrupt disabled 1 = Pin interrupt enabled							

18.2 GPIO Configuration Registers

18.2.1 GPIO Register Set D

PD0CFG: PD0 GPIO configuration register.							
PD0CFG		0x50018020			0x00		
Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
-	-	-	-	OE_PD0	RE_PD0	PU_n_PD0	PD_PD0
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PD1CFG: PD1 GPIO configuration register.							
PD1CFG		0x50018021			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PD1	RE_PD1	PU_n_PD1	PD_PD1
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PD2CFG: PD2 GPIO configuration register.							
PD2CFG		0x50018022			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PD2	RE_PD2	PU_n_PD2	PD_PD2
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PD3CFG: PD3 GPIO configuration register.							
PD3CFG		0x50018023			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PD3	RE_PD3	PU_n_PD3	PD_PD3
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PD4CFG: PD4 GPIO configuration register.							
PD4CFG		0x50018024			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	Reserved	-	-	OE_PD4	RE_PD4	PU_n_PD4	PD_PD4
MSB	-						LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PD5CFG: PD5 GPIO configuration register.							
PD5CFG		0x50018025			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	Reserved	-	-	OE_PD5	RE_PD5	PU_n_PD5	PD_PD5
MSB	-						LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PD6CFG: PD6 GPIO configuration register.							
PD6CFG		0x50018026			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PD6	RE_PD6	PU_n_PD6	PD_PD6
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PD7CFG: PD7 GPIO configuration register							
PD7CFG		0x50018027			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PD7	RE_PD7	PU_n_PD7	PD_PD7
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

18.2.1.1 GPIO Register Set E

PE0CFG: PE0 GPIO configuration register.							
PE0CFG		0x50018028			0x00		
Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE0	RE_PE0	PU_n_PE0	PD_PE0
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PE1CFG: PE1 GPIO configuration register.							
PD1CFG		0x50018029			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE1	RE_PE1	PU_n_PE1	PD_PE1
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PE2CFG: PE2 GPIO configuration register.							
PE2CFG		0x5001802A			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE2	RE_PE2	PU_n_PE2	PD_PE2
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PE3CFG: PE3 GPIO configuration register.							
PE3CFG		0x5001802B			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE3	RE_PE3	PU_n_PE3	PD_PE3
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PE4CFG: PE4 GPIO configuration register.							
PE4CFG		0x5001802C			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE4	RE_PE4	PU_n_PE4	PD_PE4
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PE5CFG: PE5 GPIO configuration register.							
PE5CFG		0x5001802D			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE5	RE_PE5	PU_n_PE5	PD_PE5
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PE6CFG: PE6 GPIO configuration register.							
PE6CFG		0x5001802E			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE6	RE_PE6	PU_n_PE6	PD_PE6
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PE7CFG: PE7 GPIO configuration register.							
PE7CFG		0x5001802F			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PE7	RE_PE7	PU_n_PE7	PD_PE7
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

18.2.1.2 GPIO Register Set F

PF0CFG: PF0 GPIO configuration register.							
PF0CFG		0x50018030			0x00		
Reserved	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
-	-	-	-	OE_PF0	RE_PF0	PU_n_PF0	PD_PF0
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PF1CFG: PF1 GPIO configuration register.							
PF1CFG		0x50018031			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PF1	RE_PF1	PU_n_PF1	PD_PF1
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

PF2CFG: PF2 GPIO configuration register.							
PF2CFG		0x50018032			0x00		
Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W	R/W
-	-	-	-	OE_PF2	RE_PF2	PU_n_PF2	PD_PE2
MSB							LSB

Bit3 **OE**: Output Select
0 = High impedance
1 = Output

Bit2 **RE**: GPIO read enable bit.
0 = Read Disabled
1 = Read Enabled

Bit1 **PU_n**: IO pin Pull-Up control bit.
0 = Pull-Up Active
1 = Pull-Up Inactive

Bit0 **PD**: GPIO read enable bit.
0 = Pull Down inactive
1 = Pull Down active

19.0 Clock Sources

Herzog provides four clock sources:

- Internal auxiliary oscillator running at 10kHz. (This oscillator is always running, even when the device is in sleep mode, but its power consumption is negligible)
- Internal RC oscillator running at 10MHz.
- Real Time Oscillator @ 32.768kHz
- Crystal oscillator running @ 30MHz

Herzog starts from power-on reset using the internal 10MHz RC oscillator. From this point on the user may select the crystal or the auxiliary oscillators.

The crystal oscillator is required for RF reception. The 10kHz oscillator may be used in power saving modes.

19.1 Clock Source Characteristics

The following table defines the main characteristics of the clock sources:

Table 26 – Clock Performance Specification, recommended operating conditions

Name	Conditions	Min.	Typ.	Max.	Unit
Crystal Oscillator frequency			30		MHz
Frequency stability	Using defined crystal			TBD	ppm
Auxiliary Oscillator	(Calibrated Frequency)		10		kHz
Auxiliary Oscillator accuracy	Post-calibration to 10KHz, T_A=27°C			5	%
RC Oscillator frequency			10		MHz
RC Oscillator accuracy	Post-calibration to 10MHz, T_A=27°C			5	%

19.2 Clock Source Operation and Description

Upon Reset or Power-On Reset the system starts using the internal RC 10MHz oscillator.

Depending on the application requirements the designer can:

- Enable or disable the internal RC oscillator
- Enable or disable the external crystal
- Select the system clock source: RC or Crystal

Example Code: Enable the Crystal oscillator.

```
uint8_t XTCLK_Init ( void )
{
  int i,j;

  // Initialize the Crystal Clock to run the system, divided by one (default)
  j=0;
  while ((*PMUCLK & (XTCLOCK|RC10MHZ)) != XTCLOCK )  //Wait for XT monitor ON
  {
      CLK_CrystalControl( XTON );          // Enable crystal oscillator
      for ( i = 0; i < 20000; i++);
      CLK_SelectClockSource( XTCLOCK );    // Selects crystal as clock source
      j++;
        if (j >= 10) return (FALSE);
  }
  return (TRUE);
}
```

19.3 Clock Related Registers

The following registers are used to control the behavior of the clock sources:

PMUCLK: Processor Control Register.							
PMUCLK		0x50000000			0x15		
R/W	R/W	R	R	R/W	R/W	R/W	R/W
CKD1	CKD0	XOMON	RCMON	XO_CK_ENB	RC_CK_ENB	CKSEL1	CKSEL0
MSB							LSB

Bit7-6 **CKD[1:0]**: Clock Frequency Divider
00 = Clock Divided by 1
01 = Clock Divided by 2
10 = Clock Divided by 4
11 = Clock Divided by 8

Bit5 **XOMON**: Crystal Oscillator Monitor
0 = Crystal Oscillator Inactive
1 = Crystal Oscillator Active

Bit4 **RCMON**: RC Oscillator Monitor
0 = RC Oscillator Inactive
1 = RC Oscillator Active

Bit3 **XO_CK_ENB**: Crystal Oscillator Control
0 = Crystal Oscillator Disable
1 = Crystal Oscillator Enable

Bit2 **RC_CK_ENB**: RC Oscillator Control
0 = RC Oscillator Disable
1 = RC Oscillator Enable

Bit1-0 **CKSEL[1:0]**: Clock Select
00 = 10 KHz Auxiliary Clock
01 = 10MHz RC Oscillator Clock*
10 = Crystal Oscillator Clock
11 = not used

PMUCLK: Processor Control Register.
*Note: This is the clock selected after Power-On-Reset and for clock fault condition.

RTOUT: Real Time Output Register							
RTOUT		0x50000005			0x00		
R/W	Reserved	R/W	R/W	R/W	R/W	R/W	R/W
SER_FAST	-	RTSELA2	RTSELA1	RTSELA0	RTSELB2	RTSELB1	RTSELB0
MSB							LSB

Bit2-0 **RTSELB[2:0]**: Real Time Output at PD1

001 : clk_full (system clock)

010 : clk_div

011 : clk_ser

100 : rto_adc

101 : rto_tx

Bit5-3 **RTSELA[2:0]**: Real Time Output at PD0

001 : clk_AUX (10 kHz clock)

010 : clk_RC (10 MHz clock)

011 : clk_XO (30 MHz clock)

110 : bor_n_mask

111 : rto_rx

20.0 Power Management Unit

Herzog implements a power management unit. Its main characteristics are:

- HW reset - Affects all aspects of Herzog
- SW reset - Does not affect clock nor brownout setup
- Selectable Sleep mode and Halt Mode
- Programmable brownout detector

20.1 PMU Operation and Description

The PMU module allows for the control of reset, deep sleep (halt), sleep, and brownout.

20.1.1 PMU control of Reset:

There are two forms of reset that can be issued:

- Hardware reset: In this reset all peripherals are reset, the 10 KHz clock is selected and all other clock sources are disabled, but the brownout selection is kept.
- Software reset: In this reset all peripherals are reset but the clock setup is kept unchanged along with the brownout selection.

Code Examples: HW reset and SW reset:

```
PMU_HwReset();
PMU_SwReset();
```

20.1.2 PMU control of sleep and deep sleep (halt) modes:

The PMU can set the system into sleep or deep sleep (halt) modes.

In the deep sleep mode:

- the CPU is halted
- Any enabled clock source will continue to operate

- The three timers (Timer0, Timer1 and Timer2) and the SysTick Timer will stop operating
- All other peripherals will keep running (if enabled and fed by their required source clock)
- The system will leave the deep sleep (halt) mode only through a reset or POR (**P**ower-**O**n **R**eset). The sources of a reset can be the wakeup timer or any peripheral that generates an interrupt independently of the interrupt being enabled by the NVIC module.(**N**ested **V**ector **I**nterrupt **C**ontroller)

Note: For those peripherals that have in their registers a bit that locally enables the interrupt this register has to be enabled in order to reset the system. Example: For the GPIOs ports PORTA, PORTB and PORTC the INTE bits of the I/O pins selected to reset the part upon change must be set.

Code Example: Enabling the reset by enabling an interrupt from PORTB[4] upon change. (Port B, bit4)

```
//PortB.4 interrupt enabled, pull up enabled
Port_Config(PTB, 0, 0x10, 0x00, 0x00, 0x10, 0x00, 0x00);
PMU_Deep_Sleep(); //System in deep sleep (halt)
```

In the sleep mode:

1. the CPU is halted
2. Any enabled clock source will continue to operate
3. All timers (Timer0, Timer1 and Timer2) and the SysTick Timer will continue operating
4. All other peripherals will keep running if enabled and fed by their required source clock
5. Besides a POR and/or reset the system will leave the sleep mode also through an interrupt; the sources of an interrupt can be any peripheral generating an interrupt.

   Note: The interrupt must be enabled by the NVIC module.(**N**ested **V**ector **I**nterrupt **C**ontroller) and for those peripherals that have in their registers a bit that locally enables the interrupt this register also has to be enabled in order to generate an interrupt and wakeup the system from sleep.

   Example: For the GPIOs ports PORTA, PORTB and PORTC the INTE bits of the I/O pins selected to reset the part upon change must be set.

20.1.3 PMU control of Brownout:

The PMU controls the brownout. It entails:

- Enabling or disabling the brownout circuit
- Selecting the behavior when a brownout is detected:
  - Generate an interrupt
  - Reset the system
- Selecting the brownout voltage level

Code Example: Enable the BOR, with interrupt but no reset and with 2.4V level.

```
BOR_ResetControl(BORRSTDIS);   //Brownout reset disabled
BOR_IntControl(BORINTEN);      //Brownout interrupt enabled
BOR_Level(BOR24V);             //Brownout level = 2.4V
BOR_Control(BOREN);            //Brownout enabled
```

20.2 PMU Registers

Herzog implements the following PMU registers:

PMURST: Processor control register.							
PMURST		0x50000001			0x01		
W	W	R/W	Reserved	Reserved	Reserved	R	R/W
HWRST	SWRST	DLEEP	-	-	-	BROUT	PORF
MSB							LSB

Bit7 **HWRST**: Hardware reset

0 = Idle

1 = Hardware reset (automatically cleared after reset process completed)

Bit6 **SWRST**: Software reset

0 = Idle

1 = Software reset (automatically cleared after reset process completed)

Bit5 **DLEEP**: Deep sleep (HALT) mode

Writing:

0 = Clear deep sleep flag

1 = Put the system in deep sleep - Halt

Reading:

0 = Flag cleared

1 = system in deep sleep mode

Bit1 **BROUT**: Brownout indicator

0 = No brownout

1 = Brownout

Bit0 **PORF**: Power-On Reset flag

Writing:

0 = Clear POR / 1 = No effect

Reading:

0 = POR flag already cleared by application

1 = The system just came out of POR or HW Reset

PMUBOR: BOR Control.							
PMUBOR		0x50000002			0x00		
R/W	Reserved	Reserved	Reserved	R/W	R/W	R/W	R/W
BORENB	-	-	-	BORRSTB	BORINT	BOUTVALUE1	BOUTVALUE0
MSB							LSB

Bit7 **BORENB**: Brownout enable (active low)

0 = Brownout enabled

1 = Brownout disabled

Bit3 **BORRSTB**: Brownout Reset enable (active low)

0 = Enable Brownout based reset

1 = Disable Brownout based reset

Bit2 **BORINT**: Brownout interrupt

1 = Brownout interrupt enabled

0 = Brownout interrupt disabled

Bit1-0 **BOUTVALUE [1:0]**: Brownout threshold value

00 = 2.0V

01 = 2.2V

10 = 2.4V

11 = 2.6V

21.0 Wake-Up Timer

In addition to the Timer0/1/2 Herzog implements a timer capable of waking-up the microcontroller from a sleep state.

The wake up timer is a timer used to allow for recovery from deep sleep, including when the microcontroller is disconnected from its power supply.

21.1 Wake-Up Timer Operation and Description

For instance, a value of 0x54 would give a time of: (Assuming the application is running from the 10 kHz internal oscillator)

WakeupPeriod=5*30/10kHz=16msec

```
WKP_Timing(5,4)          //Select the mantissa=5 and exponent=4
PMU_Deep_Sleep(SLEEPON);      //Put the part in deep sleep mode, it will
//Reset in 16msec.
```

21.2 Wake-Up Timer Register

The following register controls the wake-up timer:

WKPTIME: Wakeup timer control.							
WKPTIME		0x50000004			0x00		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MANT3	MANT2	MANT1	MANT0	EXP3	EXP2	EXP1	EXP0
MSB							LSB

Bit7-4 **MANT [3:0]**: Mantissa of the wakeup timer

Bit3-0 **EXP [3:0]**: Exponent of the wakeup timer (range: 0...12)

WakeupPeriod = Mantissa*$2^{(Exponent+1)}$/SystemClock

22.0 Reserved Registers

The following registers are for a reserved purpose, and although they are not material to the primary operation of the device, they must be regularly reprogrammed with their default values in order to ensure reliable device operation

Reserved Registers							
Name		Address			Default Value		
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
---	---	---	---	---	---	---	---
MSB							LSB

23.0 References

ISM Band references:

http://en.wikipedia.org/wiki/ISM_band

http://ecfr.gpoaccess.gov/cgi/t/text/text-idx?c=ecfr&sid=8a0249759d545fa2c9ea0840b08bb817&rgn=div5&view=text&node=47:1.0.1.1.14&idno=47)

24.0 Contacts

United States

32 Journey
Aliso Viejo, California 92656, USA
Tel: +1 949-608-0854
sales@indiesemi.com

China

D2-608, China Sensor Network Innovation Park, 200#,
LinHu Road, XinWu district, WuXi City, China.
Tel: +86 139 1837 6572

Scotland

5th Floor, The Auction House,
63a George Street
Edinburgh EH2 2JG
Tel: +44 131 718 6378

http://www.indiesemi.com/

Important Notice

indie semiconductor reserves the right to make changes, corrections, enhancements, modifications, and improvements to indie semiconductor products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on indie semiconductor products before placing orders. indie semiconductor products are sold pursuant to indie semiconductor's terms and conditions of sale in place at the time of order acknowledgement. Purchasers are solely responsible for the choice, selection, and use of indie semiconductor products and services described herein. indie semiconductor assumes no liability for the choice, selection, application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by indie semiconductor by this document.

The materials, products and information are provided "as is" without warranty of any kind, whether express, implied, statutory, or otherwise, including fitness for a particular purpose or use, merchantability, performance, quality or non-infringement of any intellectual property right. Indie semiconductor does not warrant the accuracy or completeness of the information, text, graphics or other items contained herein. indie semiconductor shall not be liable for any damages, including but not limited to any special, indirect, incidental, statutory, or consequential damages, including without limitation, lost of revenues or lost profits that may result from the use of the materials or information , whether or not the recipient of material has been advised of the possibility of such damage.

Unless expressly approved in writing by two authorized indie semiconductor representatives, indie semiconductor products are not designed, intended, warranted, or authorized for use as components in military, space, or aircraft, in systems intended to support or sustain life, or for any other application in which the failure or malfunction of the indie semiconductor product may result in personal injury, death, or severe property or environmental damage.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.