Si446x AFC Operation

1. Introduction

This document describes the operation of the Automatic Frequency Control circuitry on the Si446x RFIC family. In a radio link it is inevitable that there is some residual frequency offset between the transmit and receive side. This frequency offset can disturb the detection of an (G)FSK modulated digital transmission. The Automatic Frequency Control (AFC hereafter) in the receive side works towards eliminating this frequency offset to maintain good reception and sensitivity.

Fundamentally the AFC is built up from two main blocks: the frequency error detector device and the error compensation device. The two devices can be arranged in feedback configuration (in a control loop fashion) or in feed-forward configuration (the error compensation device will not modify the original frequency error). There are three kinds of frequency error detectors on the Si446x RFIC and both feedback and feed-forward compensation can be set.

The selected detector and compensation circuits are driven by the data packet structure which the chip is configured to receive. The receiver cannot acquire and track frequency errors immediately; it always needs a certain amount of time (and even better, some knowledge of the packet's beginning) to achieve frequency alignment. Therefore a preamble is always necessary in a packet before the real data fields begin. (The preamble is also used to settle the other two control algorithms in the receive side: Automatic Gain Control (AGC hereafter) and the Bit Clock Recovery (BCR hereafter). As a rule of thumb the longer the preamble and the more data transition it contains the more robust the link will become.

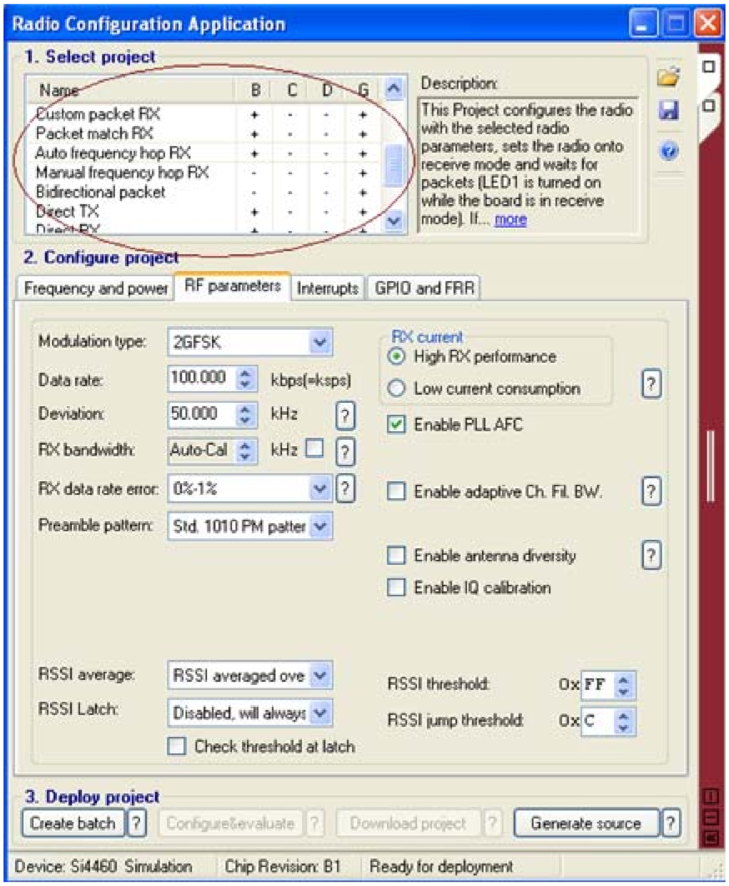
The AFC only works in (G)FSK mode; OOK signal detection is not sensitive to frequency offsets as long as the signal stays within the receive channel. However, when operating in OOK mode the receiver bandwidth is typically set wide enough to incorporate even relatively big frequency misalignments.

From above, one can conclude that there are a lot of configuration possibilities on the RFIC. However, all these settings will be automatically generated by the Wireless Development Suite (WDS hereafter) software based on higher level input parameters. References will be made in each section as to how to achieve a certain set of settings using WDS and which API properties contain the relevant settings.

In the following sections the different modes of operations, their prerequisites, and their performances are discussed in detail.

2. Overview of Application Modes

The application mode of the chip will be set by selecting a project on the Radio Configuration Application (RCA hereafter) interface of WDS (see the encircled area in the WDS screen shot below). Fundamentally there are two different application modes: packet mode and direct mode.



Name	B	C	D	G
Custom packet RX	+	-	-	+
Packet match RX	+	-	-	+
Auto frequency hop RX	+	-	-	+
Manual frequency hop RX	-	-	-	+
Bidirectional packet	-	-	-	+
Direct TX	+	-	-	+
Direct RX	+	-	-	+

Description: This Project configures the radio with the selected radio parameters, sets the radio onto receive mode and waits for packets (LED1 is turned on while the board is in receive mode). If... more

Modulation type: 2GFSK

Data rate: 100.000 kbps(=ksps)

Deviation: 50.000 kHz

RX bandwidth: Auto-Cal kHz □

RX data rate error: 0%-1%

Preamble pattern: Std. 1010 PM patter

RX current
● High RX performance
○ Low current consumption

☑ Enable PLL AFC

□ Enable adaptive Ch. Fil. BW.

□ Enable antenna diversity

□ Enable IQ calibration

RSSI average: RSSI averaged ove

RSSI Latch: Disabled, will alway:

□ Check threshold at latch

RSSI threshold: 0x FF

RSSI jump threshold: 0x C

Device: Si4460 Simulation | Chip Revision: B1 | Ready for deployment

Figure 1. Project (Application Mode) Selection in WDS

In packet mode the data stream gets loaded into the FIFO at the RX side upon reception. The Packet Handler features (packet decomposition / decoding / sync word detection) are or can be utilized in this mode. This kind of operation is used for standard packet based radio communication. (Preamble and Sync Word detection is fully configurable). Select any of the packet based projects for this mode of operation.

In direct mode at the receive side the Packet Handler features are disabled and the demodulated data stream has to go to one of the GPIO pins for further processing by the host MCU. No preamble detection or sync word detection is possible in this mode.

There are a couple of subtleties to the direct mode operation both at the TX and RX sides. Although the TX side is not within the scope of this application note it is worthwhile to discuss it here.

There are two different TX direct mode operations dependent on how the TX data is given to the RFIC. Synchronous direct mode is an application where the transmitted bit stream is supplied by an MCU upon bit clock transitions provided by the RFIC. Asynchronous direct mode is an application where the transmitted bit stream is supplied by an MCU without any bit clock. One restriction in this mode is that no GFSK modulation can be selected as the RFIC has no prior knowledge on the data rate, hence it cannot fit a Gaussian filter onto the data stream.

Similarly, at the RX side one can define two kinds of direct operation modes. In one mode the digital bit stream is synchronized to a bit clock generated by the chip itself. The circuitry responsible for this task is the BCR that can acquire small deviations (<10%) from the nominal data rate and can track small changes while the packet is being received. In another mode the digital bit stream is not synchronized to any bit clock, the BCR circuitry is simply bypassed. One might call the 1st operation mode synchronous direct RX analogous to the synchronous TX direct mode operation, and the 2nd one asynchronous direct RX mode. This differentiation, however, does not appear in WDS. The only difference between the two operation modes is where exactly the received data is taken from the demodulator chain. In the 1st case it is taken from after the BCR block, whereas in the 2nd case it is taken from before the BCR block. As mentioned before in WDS there is no differentiation between the two operation modes described above. Both can be achieved by utilizing a “DIRECT RX” project and setting one of the GPIO pins to either “RX_DATA” (synchronous mode) or “RX_RAW_DATA” (asynchronous mode). Both can be accessed simultaneously on two different GPIO pins. (Refer to the “GPIO and FRR” tab of the RCA.)

By the same token there is nothing stopping the IC from being able to output the received data stream to one of the GPIOs while in packet mode operation. If for some reason the packet handler features are needed in a direct mode operation (where a host MCU further processes the received data) this is the mode to be set.

The AFC operation is partially driven by the mode of operation, however it is more dependent on the preamble structure that is defined in the Preamble pattern control on the RF Parameters tab in the RCA panel. In the following section details are given as to how the AFC is configured upon different selections of operation modes and preamble patterns.

3. AFC in Synchronous Modem Operation

3.1. Overview of Synchronous and Asynchronous Operation

There are two demodulation paths implemented on the RFIC: synchronous and asynchronous. The AFC operation is grossly different for each, therefore it is important to have an overview of these demodulator paths.

When the modem is in synchronous operation mode the baseband signal gets converted to phase domain directly for the decision making (slicer hereafter) and the BCR circuits. The BCR provides the bit clock to the slicer so the demodulation is synchronous to the bit clock. In this mode the slicer cannot work without the BCR so it follows that there is no RAW_DATA access in this path. If however RAW_DATA is set to be output to one of the GPIOs (see the command details below in this section) the WDS calculator will select/force the asynchronous path.

In asynchronous operation mode the baseband signal gets converted to frequency (vs. time) domain and the decision is made there without the use of a bit clock. So the digital data stream at this point is not synchronized to the bit clock. This signal is the one referred to as RAW_DATA that can be multiplexed out to one of the GPIO pins for further processing. Then the asynchronously demodulated signal gets converted to phase domain again and is passed through the slicer and BCR blocks thus providing a synchronized data stream.

It is important to understand that whether the modem is in synchronous or asynchronous operation mode the RX_DATA will always be synchronized to the bit clock. In asynchronous mode the RAW_DATA is also available if one wishes to implement their own BCR algorithm. The synchronized RXDATA is available on a GPIO output pin by setting GPIO_PIN_CFG = 0x14 = 20d, while the RAW_DATA is available by setting GPIO_PIN_CFG = 0x15 = 21d. (Refer to the “GPIO and FRR” tab of the RCA.)

The synchronous demodulator path will be selected if the modulation index of a (G)FSK signal to be received is less than or equal to two and at least 32 standard (1010 pattern) preamble bits are available for the receiver and no RAW_DATA access is required. The asynchronous path will be selected for all other cases.

- Modulation index is calculated as 2 x outer_deviation/symbol_rate. For a 2(G)FSK signal the equation is simplified to 2 x deviation/DR.
- In OOK mode the asynchronous path is always selected. There the demodulation happens in the power vs. time domain. The power signal is the RSSI (Radio Signal Strength Indicator) value that gets calculated after the signal has passed through the baseband receive filter.

3.2. Frequency Error Detector in Synchronous Mode

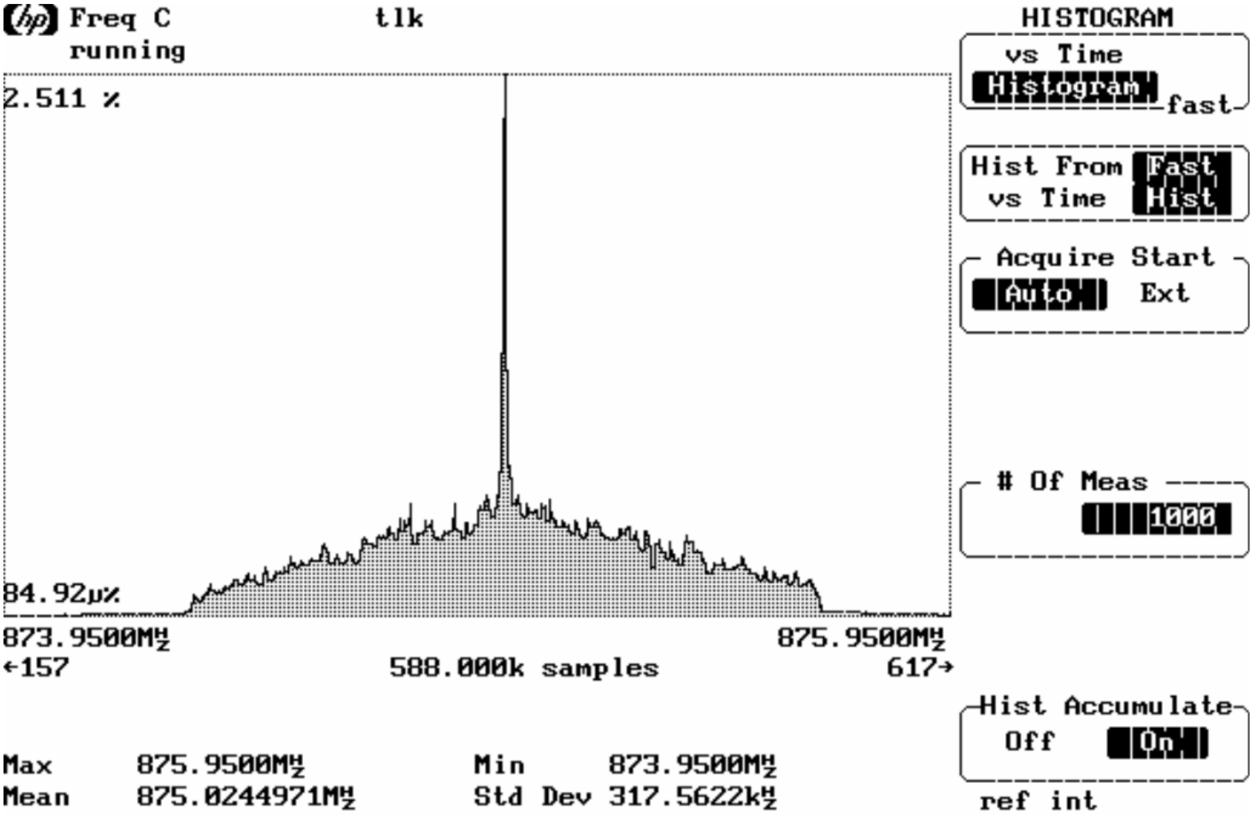
In synchronous mode there is only one frequency error detector, referred to as the 2Tb detector (Tb refers to the bit time). It works in the phase (vs. time) domain, rectifies the modulation, and measures the phase deviation from the expected phase trajectory.

This detector requires a 1010 preamble pattern which also makes it apparent that synchronous demodulation will only be selected if a sufficient number of 1010 preamble patterns are repeated at the beginning of the packet.

The 2Tb detector is enabled by setting bit5 ‘en2Tb_est’ in the MODEM_AFC_MISC property. This bit enables only the frequency error detection or estimation circuitry; compensation of the frequency error is enabled by other bits (to be discussed shortly).

MODEM_AFC_MISC							
7	6	5	4	3	2	1	0
enafcfrz	enfbpll	en2Tb_est	enfzpmend	enafc_clksw	ook_zerog	non_frzen	
1	0	1	0	0	0	0	

When the receiver is enabled the frequency error detector will start running immediately. If the transmit packet has not yet arrived, the frequency error detection circuitry in the receiver is most likely tracking on noise. If the PLL feedback compensation circuitry (more details on this will come in the next section) is additionally enabled to track the resulting estimated frequency error at the LO, the VCO will “wander” as the detector tracks on noise. The graph below shows the frequency distribution of the VCO (which is proportional to the frequency estimator’s output) in such a case.



```
Max     875.9500MHz          Min      873.9500MHz
Mean    875.0244971MHz       Std Dev  317.5622kHz
```

Figure 2. VCO Frequency Distribution While Tracking on Noise in Synchronous Mode

There is a discrete spectral component at the nominal LO frequency as the error estimation circuitry always gets reset whenever the measured offset value exceeds a certain threshold (see section 4 for details on this). This is also the reason for the slightly higher probability of the frequency staying closer to the center desired frequency.

The 2Tb detector can inherently track CW (unmodulated) signals so if a CW portion precedes the packet the AFC will start the settling on that.

3.3. Frequency Error Compensation in Synchronous Mode

As mentioned in section 1 there are two ways of compensating for the frequency error: feedback and feed-forward (also referred to as modem internal) compensation. The modem internal compensation is always enabled by default both in synchronous and asynchronous modem operation. However, the mechanism of modem internal compensation is different in synchronous and asynchronous modes. The BCR and Slicer blocks work in the phase domain so they can take phase skew information provided by the 2Tb detector directly (as the 2Tb detector also works in phase domain). In asynchronous mode, the offset detection happens in the frequency (vs. time) domain so the measured value cannot be applied to the BCR/Slicer block directly. More on that will be discussed in the asynchronous section.

The modem internal compensation in synchronous mode can be enabled by clearing bits ‘bcrfbbyp’ and ‘slicefbbyp’ in property MODEM_BCR_MISC1. These bits refer to bypassing the phase compensation in the BCR and Slicer blocks, thus they will have to be set in asynchronous mode. See section 4.3 for details.

MODEM_BCR_MISC1							
7	6	5	4	3	2	1	0
bcrfbbyp	slicefbbyp	0	rxncocomp	Rxcomp_lat	Crgainx2	Dis_midpt	Esc_midpt
1	1	0	0	0	0	0	0

Feedback compensation is done through the PLL; the LO signal gets detuned by as much as the offset amount (going through some gearing) measured by the frequency error detector. Feedback compensation is enabled by setting bit6 ‘enfbpll’ in MODEM_AFC_MISC property (see above).

The settling times of the feedback and modem internal compensations are 16Tbit and 8Tbit, respectively.

The following table summarizes how many bits the different control algorithms in the receiver require to achieve lock. These algorithms can only start if the preceding procedures have settled already (unless otherwise stated). That is to say, the BCR needs the AGC to have settled, the AFC needs the BCR to have settled, and so on. By adding up all the timing numbers one can conclude how many 1010 preamble bits are required in synchronous operation mode for successful reception and good sensitivity.

	AFC Feedback Comp Enabled	AFC Feedback Comp Disabled
AGC	1	1
BCR	8	8
AFC	16*note1	8*note1
Preamble Detection	20	20*note2
Margin	3	3
All	**40**	**32**

Notes:

1. 8Tbit occur simultaneously with the BCR (i.e., internal modem compensation settles with BCR).
2. This threshold for preamble detection reduces the probability of having a false preamble detect (triggered by noise) to nearly zero.

To sum it up, at least 40 bits of 1010 preamble are required for AFC feedback operation in synchronous mode and at least 32 bits for modem internal AFC operation.

The AFC PLL feedback compensation is a delicate control loop whose parameters are calculated by the WDS calculator. Here is the list of the related API properties:

MODEM_AFC_WAIT'

MODEM_AFC_GAIN_1'

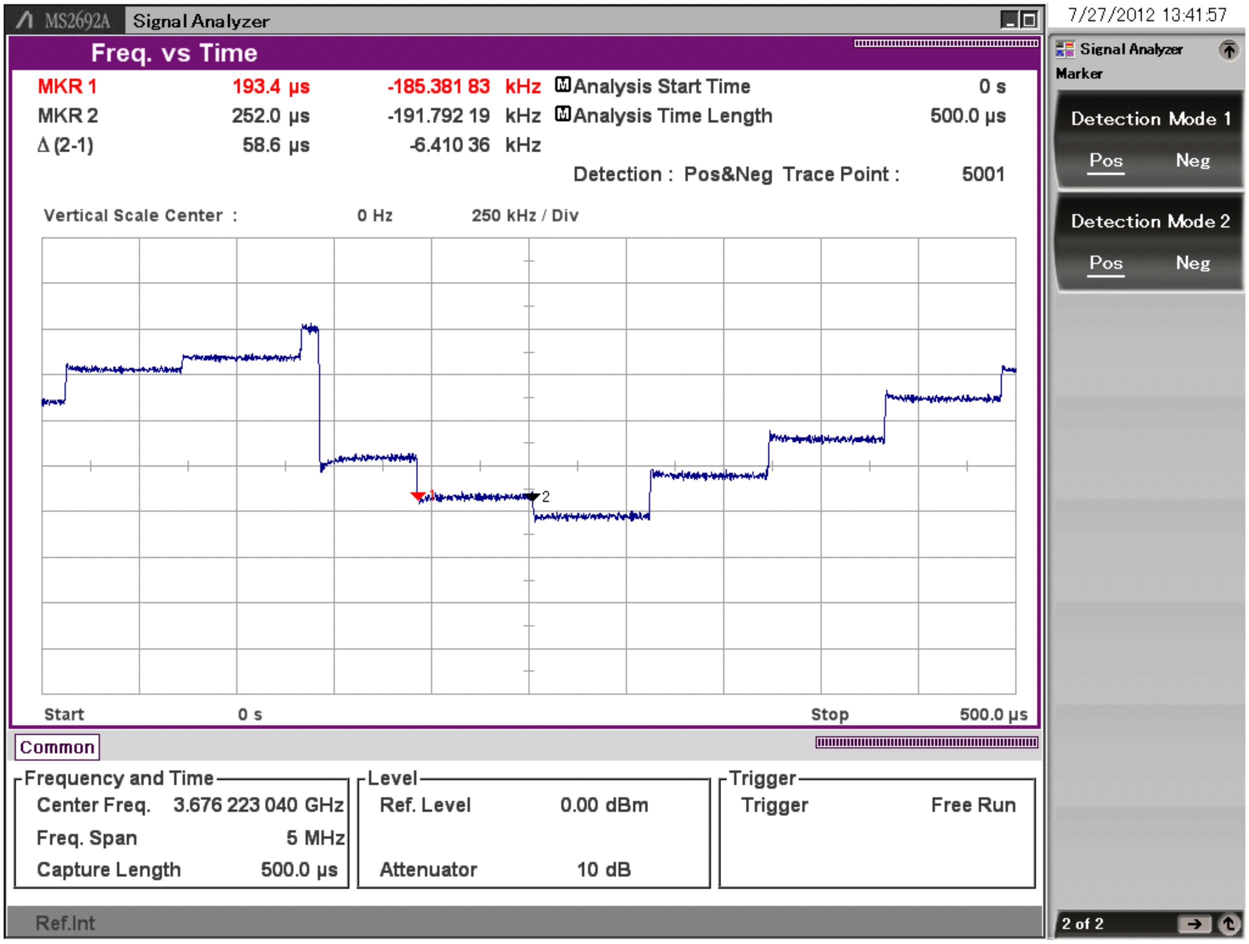
MODEM_AFC_GAIN_0'

MODEM_AFC_LIMITER_1'

MODEM_AFC_LIMITER_0'

These parameters only have an effect if the feedback compensation path is enabled. (Note that in property MODEM_AFC_GAIN1 apart from a gain setting there are two more parameters ‘enafc’ and ‘afcbd’ which will be covered later in this document.)

MODEM_AFC_WAIT determines the time between compensation steps applied to the PLL. This provides time for the PLL to settle after a frequency adjustment, and thus allows the detector to have an accurate measurement after the PLL frequency has been changed. The graph below shows the VCO’s frequency trajectory with the AFC operating on noise in synchronous mode. The modulation scheme the receiver is set to is 2GFSK with a 100 kbps data rate. The wait time is set 6Tbit (i.e., 60 us at 100 kbps).



MS2692A Signal Analyzer

7/27/2012 13:41:57

Freq. vs Time

MKR 1 193.4 µs -185.381 83 kHz
MKR 2 252.0 µs -191.792 19 kHz
Δ (2-1) 58.6 µs -6.410 36 kHz

Analysis Start Time 0 s
Analysis Time Length 500.0 µs

Detection : Pos&Neg Trace Point : 5001

Vertical Scale Center : 0 Hz 250 kHz / Div

Start 0 s Stop 500.0 µs

Common

Frequency and Time
Center Freq. 3.676 223 040 GHz
Freq. Span 5 MHz
Capture Length 500.0 µs

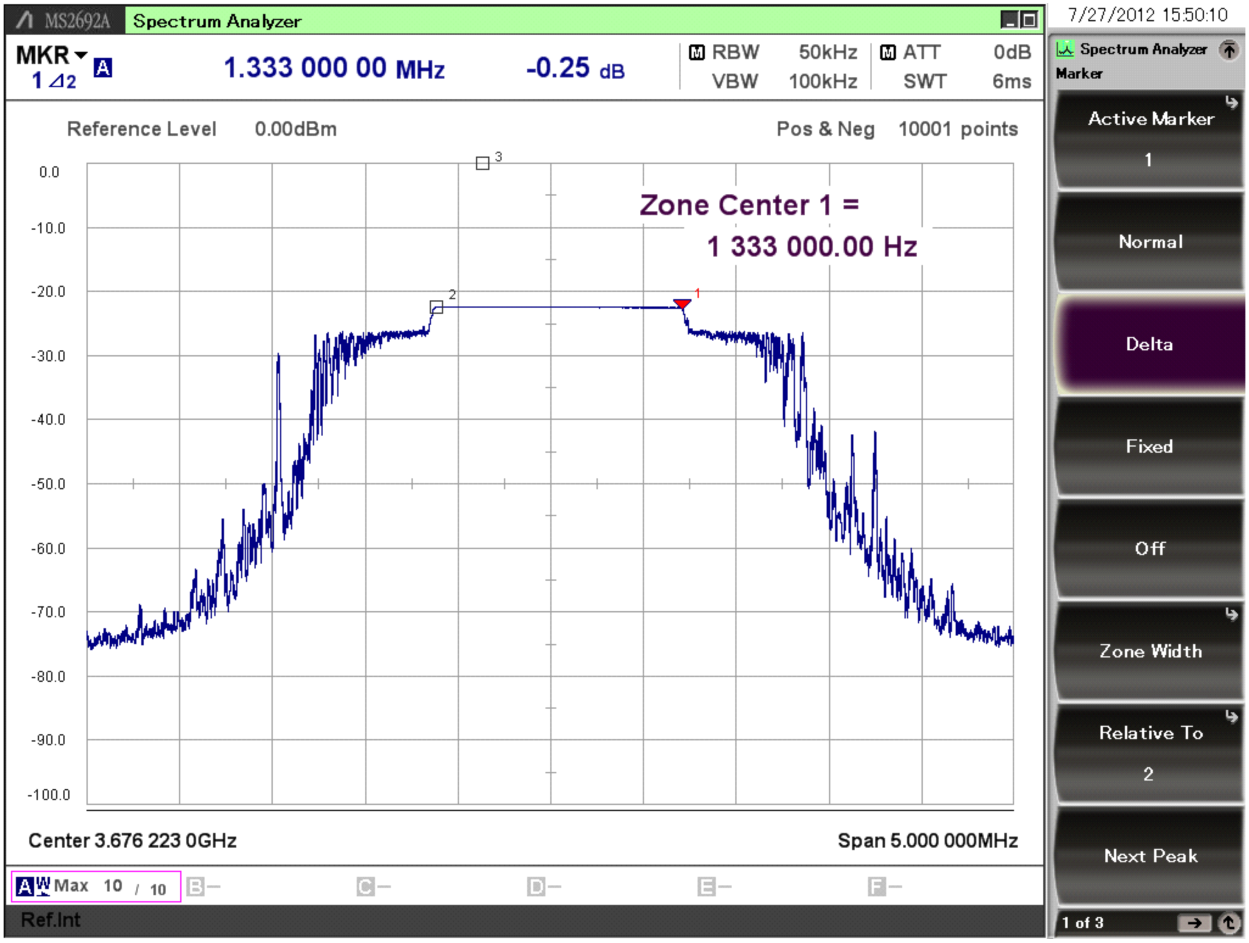
Level
Ref. Level 0.00 dBm
Attenuator 10 dB

Trigger
Trigger Free Run

Ref.Int

Figure 3. VCO Frequency Trajectory While Tracking on Noise

MODEM_AFC_LIMITER_1 and MODEM_AFC_LIMITER_2 contain a frequency offset value above which the AFC resets itself to 0. This prevents the AFC algorithm from “wandering off” too far from the nominal LO when running on noise (see the frequency distribution plot in the previous section). This mechanism can be seen on the graph above where a relatively short spike occurs (at approximately T=140 µs) after which the frequency goes back to its center value. The feature can also be captured on a spectrum plot of the VCO with the AFC running on noise. RX frequency is 460 MHz which is 1/8th of the VCO frequency.



MS2692A Spectrum Analyzer

7/27/2012 15:50:10

MKR 1 Δ2 1.333 000 00 MHz -0.25 dB

RBW 50kHz ATT 0dB

VBW 100kHz SWT 6ms

Reference Level 0.00dBm

Pos & Neg 10001 points

Center 3.676 223 0GHz

Span 5.000 000MHz

Max 10 / 10

Ref.Int

Zone Center 1 =
1 333 000.00 Hz

Figure 4. VCO “Max Hold” Spectrum Plot While Tracking on Noise

At this example the limiter frequency calculated by WDS is 82439 Hz. As the VCO is running 8 times the LO frequency (460 MHz) the measured single-sided limiter frequency is 1.333 MHz / 8 /2 = 833125 Hz. The limiter frequency value is proportional to the RX BW which is determined by many system-level input parameters (DR, deviation, Xtal accuracy).

MODEM_AFC_GAIN_0 and MODEM_AFC_GAIN_1 contain the loop gain of the feedback compensation (in other words how big a step is applied to the PLL for 1 unit change of frequency error). Modifying this value from that suggested by the WDS Calculator is not recommended as the AFC feedback loop can become unstable.

3.4. Gear Switching in Synchronous Mode

As the 2Tb detector can only work properly on 1010 patterns it is apparent that the AFC should be frozen for data reception during the Sync Word and Payload fields, otherwise it would pull the PLL which in turn would result in lost bits. By the same token there are additional properties in the chip (e.g., BCR loop gain) which may be changed for optimal reception of the data fields. Usually control loops are made slower in data reception mode hence the “gear switching” terminology for the feature.

A mechanism is implemented on the Si446x RFICs that can register an event and change the configuration of certain blocks on the fly (i.e., “switch gears”). The event that triggers the changes can be set in property MODEM_AFC_GEAR.

MODEM_AFC_GEAR							
7	6	5	4	3	2	1	0
gear_sw[1:0]		afc_fast[2:0]			afc_slow[2:0]		
0x0		0x0			0x0		

- gear_sw[1:0] - default:0x0 AFC and BCR gear switch control source.
  - 0 = Preamble detection – switch gears after detection of Preamble
  - 1 = Sync word detection – switch gears after detection of Sync Word
  - 2 = Mid-Point frequency error detection – switch gears when the estimated frequency error of the Mid-Point detector in the Asynchronous Demodulator is less than the half eye-threshold for a consecutive number of search periods. NOTE: The consecutive number of search periods is defined by the sch_frzth[2:0] field in property MODEM_RAW_SEARCH, while the length of each search period (in bits) is defined by the schprd_hi[1:0] field within the same property.
  - 3 = Preamble detection – switch gears after detection of Preamble (same as gear_sw[1:0] = 2'b00)

In synchronous mode the event is always set to preamble detection. (The rest of the options will be discussed in the asynchronous section.)

MODEM_AFC_GEAR also contains the weighing factors for the feedback loop's gain in the two different modes. The 'afc_fast[2:0]' field determines the feedback loop gain in fast mode prior to gear switching, while the 'afc_slow[2:0]' field determines the feedback loop gain in slow mode after gear switching. It is not recommended to change these values from those suggested by the WDS Calculator.

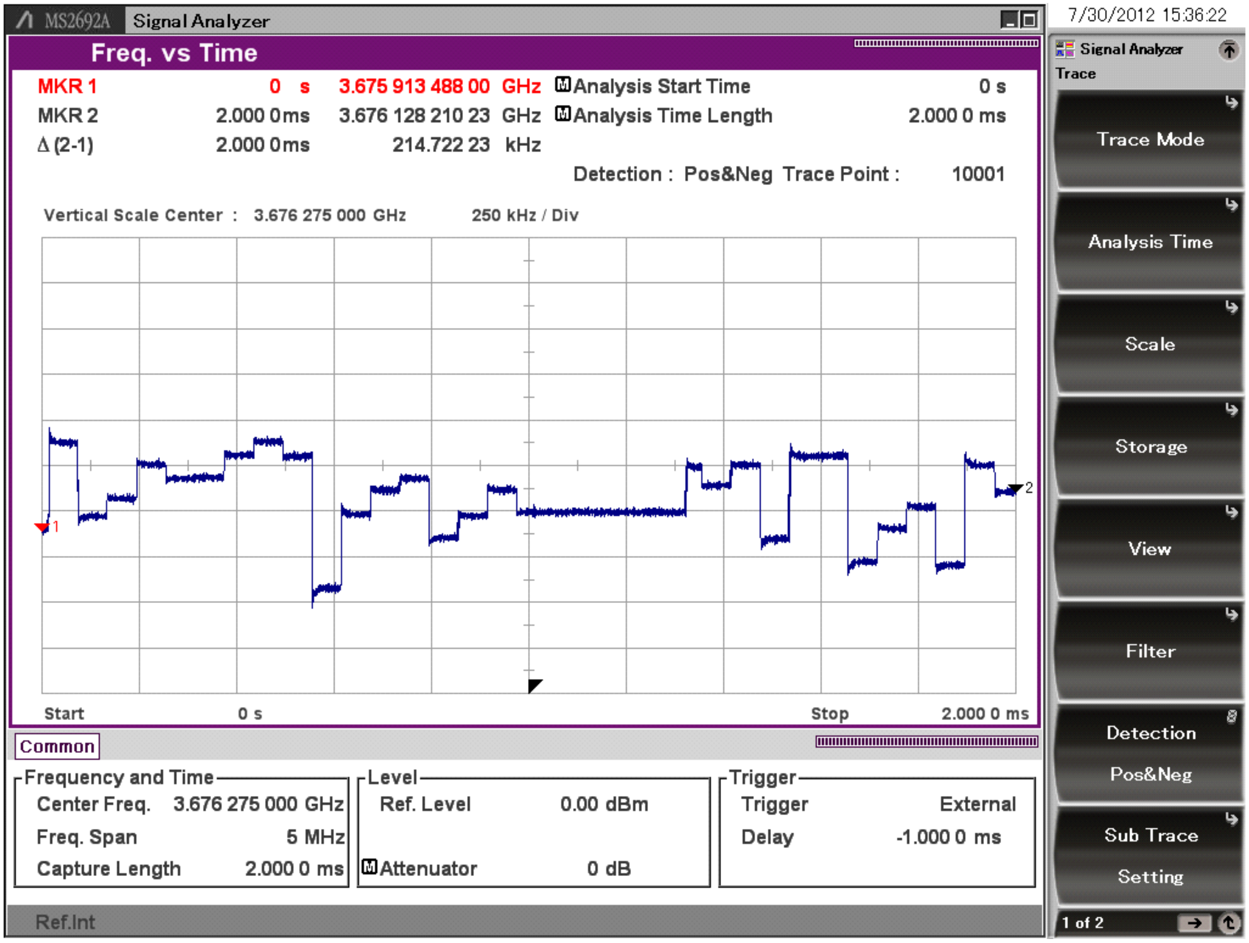
Features that are or can be affected by gear change in synchronous mode:

- Freezing the AFC detector

  This feature will stop the frequency error detector from updating after the gear switch event has occurred. It can be activated by setting bit7 'enafcfrz' in property MODEM_AFC_MISC. In synchronous mode this feature is always enabled as the 2Tb detector cannot work on random data (WDS will enable this bit automatically). Once the detector is frozen it can only be reactivated by exiting and then re-entering RX state. Automatic exit and re-entry into RX mode may be configured by providing the appropriate parameter bytes to the START_RX command (i.e., configuring the RXVALID_STATE and RX_INVALID state parameters). Alternatively, the chip may be manually commanded to READY or SLEEP mode then RX shall be enabled at a later time. (If in packet mode gear switching is set to preamble detection and sync word is not detected within the sync word detect time-out period, the receiver will automatically go back to preamble search state and release the lock from the AFC detector.

MODEM_AFC_MISC							
7	6	5	4	3	2	1	0
enafcfrz	enfbpll	en2Tb_est	enfzpmend	enafc_clksw	ook_zerog	non_frzen	
1	0	1	0	0	0	0	

The signal analyzer screenshot shown below (VCO frequency vs. time) demonstrates the freezing of the AFC operation upon preamble detection. The input to the receiver is a PN9 sequence and preamble detection threshold is set to 8 bits. The 2Tb detector generally cannot track the PN9 sequence until a 1010 pattern arrives; this is what happens immediately before the trigger event (center of the display, marked by small black triangle at the bottom of the plot). The trigger signal to the analyzer is the preamble detection signal itself taken from one of the GPIO pins. As sync word is not found within the PN9 sequence, the sync word timeout occurs and makes the receiver go back to preamble search state and the whole procedure starts again.



MKR 1 0 s 3.675 913 488 00 GHz

MKR 2 2.000 0ms 3.676 128 210 23 GHz

Δ (2-1) 2.000 0ms 214.722 23 kHz

Analysis Start Time 0 s

Analysis Time Length 2.000 0 ms

Detection : Pos&Neg Trace Point : 10001

Vertical Scale Center : 3.676 275 000 GHz 250 kHz / Div

Frequency and Time

Center Freq. 3.676 275 000 GHz

Freq. Span 5 MHz

Capture Length 2.000 0 ms

Level

Ref. Level 0.00 dBm

Attenuator 0 dB

Trigger

Trigger External

Delay -1.000 0 ms

Figure 5. VCO Frequency Trajectory at AFC Freezing

- Changing the gear (i.e., gain) of the feedback compensation loop

  This feature will change the feedback loop's gain. As discussed before, it is not recommended to change these values. (This is handled by WDS automatically.)

- Changing the gear (i.e., gain) of the BCR loop

  The gain of the BCR feedback loop is not closely related to AFC operation. However, the trigger event for gear switching of the BCR loop gain is the same as for AFC (i.e., controlled by the 'gear_sw[1:0]' field in MODEM_AFC_GEAR). It is therefore worth mentioning to get an overview of the interactions in the modem. See MODEM_BCR_GEAR property for details. (This is handled by WDS automatically.)

- Changing the RX BB filter bandwidth

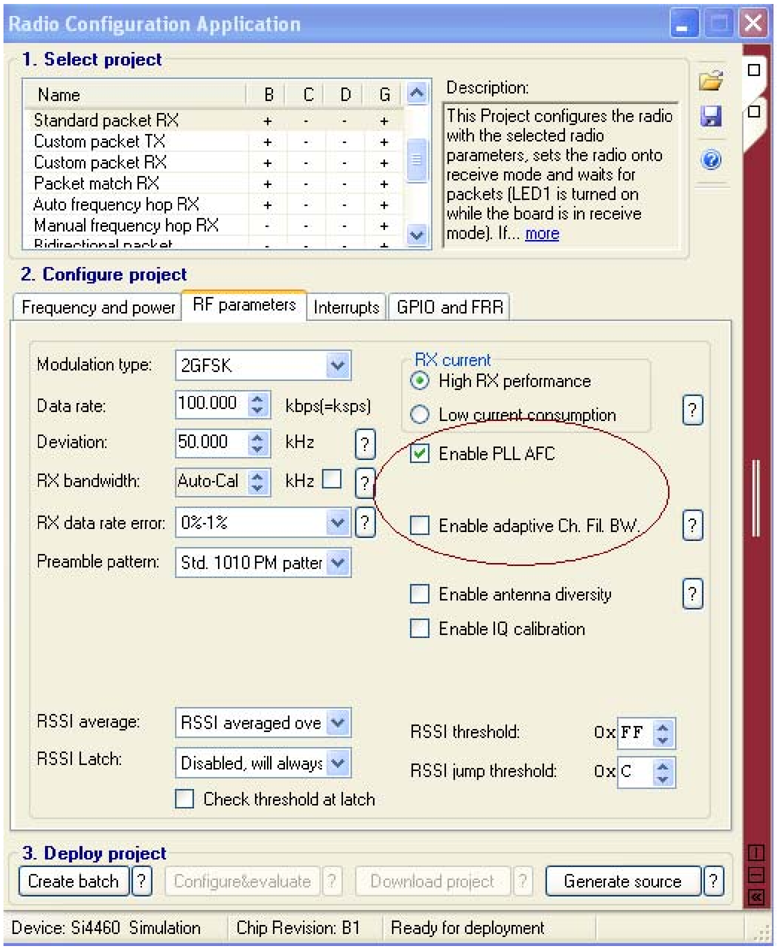
  There are two sets of RX channel filter coefficients. The 2nd one can be made to take effect after the gear switch event has occurred. This filter is the narrowest filter possible for the given modulation format (the BW of this filter is the narrowest that is wider than the modulation BW of 2 x deviation +DR). It is assumed that the frequency offset has been eliminated by the time the gear switch event occurs and therefore there is no need to keep a wider bandwidth which is primarily useful for frequency acquisition purposes. This feature will enhance sensitivity by reducing the amount of noise entering into the receiver in data reception mode.

  For this feature to work properly the PLL feedback compensation must be enabled (the "Enable AFC" check box must be ticked in WDS) as the feed-forward compensation on its own does not change the LO frequency. If not enabled, the LO frequency remains at an offset referred to the input signal and therefore reducing the filter bandwidth (after gear switching) may mean losing some of the signal energy.

  The feature can be activated by setting bit6 ('afcbd') in the MODEM_AFC_GAIN_1 property.

MODEM_AFC_GAIN_1							
7	6	5	4	3	2	1	0
enafc	afcbd	0	afcgain_12_8[4:0]				
1	0	0	0x03				

WDS will automatically set this bit if the “Enable adaptive Ch. Fil. BW.” (and the “Enable AFC”) check box is ticked. See below.



Radio Configuration Application

1. Select project

Name	B	C	D	G
Standard packet RX	+	-	-	+
Custom packet TX	+	-	-	+
Custom packet RX	+	-	-	+
Packet match RX	+	-	-	+
Auto frequency hop RX	+	-	-	+
Manual frequency hop RX	-	-	-	+
Bidirectional packet	-	-	-	+

Description:
This Project configures the radio with the selected radio parameters, sets the radio onto receive mode and waits for packets (LED1 is turned on while the board is in receive mode). If... more

2. Configure project

Modulation type: 2GFSK
Data rate: 100.000 kbps(=ksps)
Deviation: 50.000 kHz
RX bandwidth: Auto-Cal kHz □
RX data rate error: 0%-1%
Preamble pattern: Std. 1010 PM patter

RX current
● High RX performance
○ Low current consumption

☑ Enable PLL AFC
□ Enable adaptive Ch. Fil. BW.
□ Enable antenna diversity
□ Enable IQ calibration

RSSI average: RSSI averaged ove
RSSI Latch: Disabled, will always
□ Check threshold at latch

RSSI threshold: 0x FF
RSSI jump threshold: 0x C

3. Deploy project

Device: Si4460 Simulation | Chip Revision: B1 | Ready for deployment

Figure 6. WDS Controls for Enabling RX Bandwidth Gearing

3.5. Measurement Showcases in Synchronous Mode

The graph below was taken on a Si4460 RFIC at 434 MHz using 2GFSK modulation with a data rate of 100 kbps and a frequency deviation of 50 kHz. The receiver BW was set to 206 kHz (calculated by WDS).

The packet has 40 bits of preamble, 2 bytes of sync word, and 8 bytes of payload.

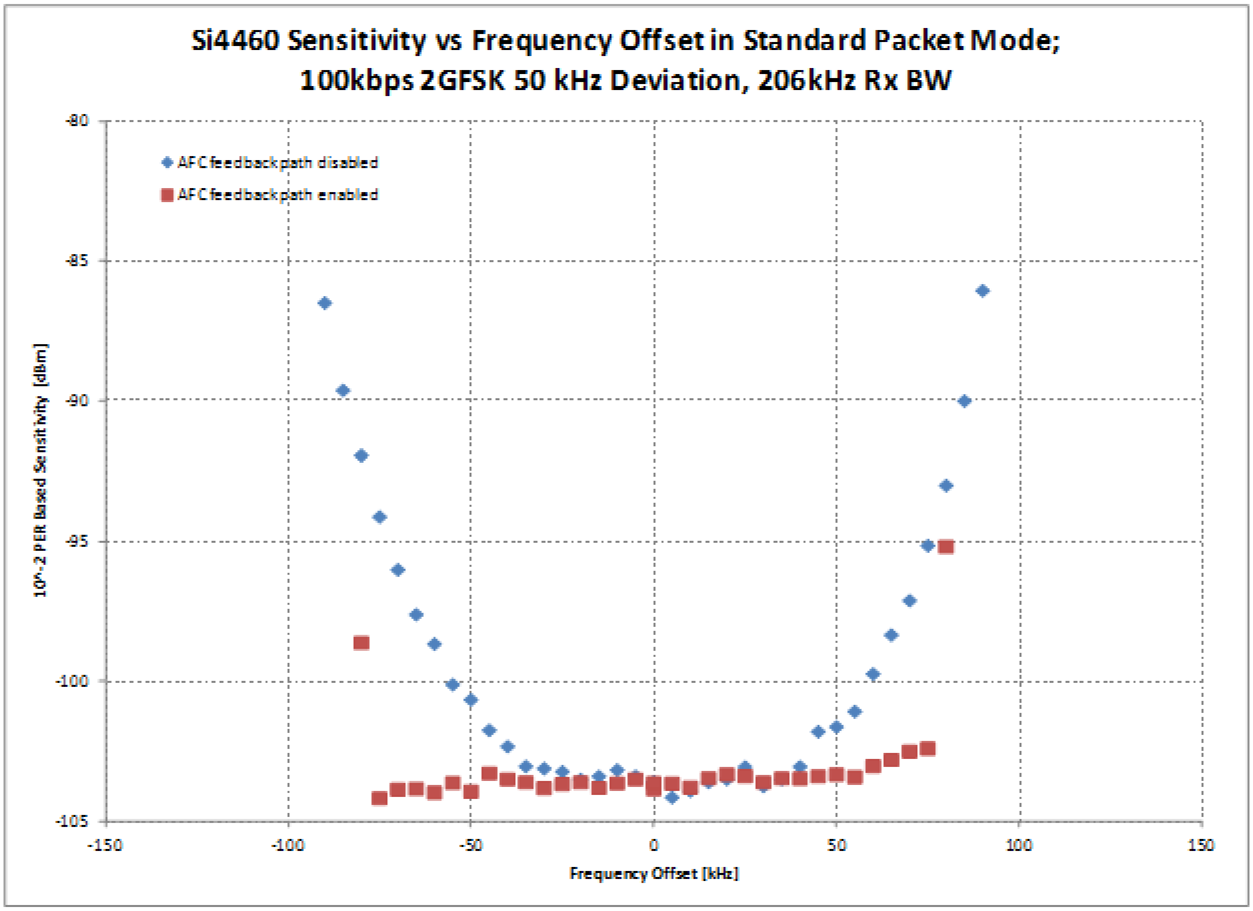


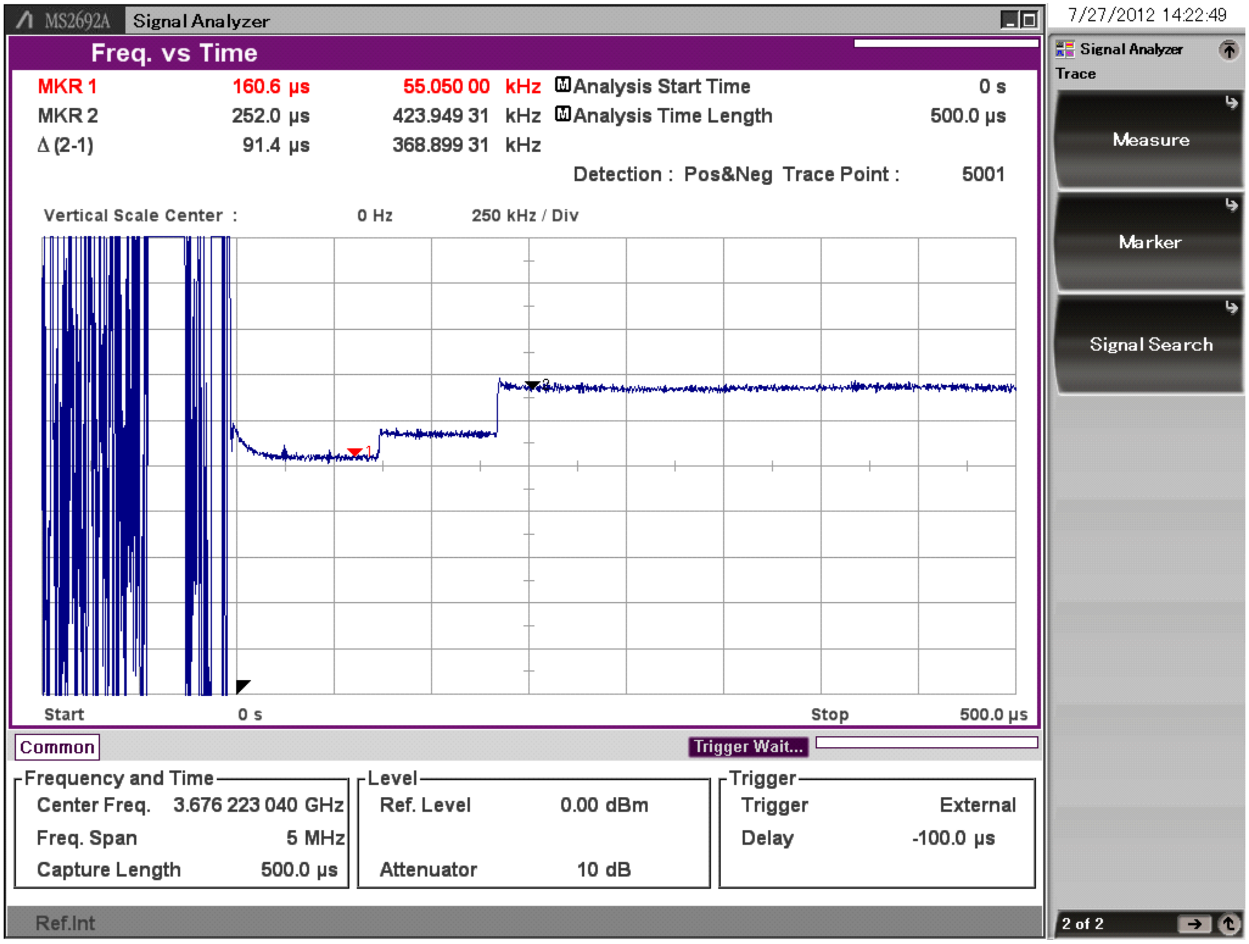
Figure 7. AFC "Bucket Curve" in Synchronous Mode

With the feedback compensation enabled the AFC has got a (3 dB) tracking range of ±75 kHz which is 72.8% of the filter bandwidth. With the feedback compensation disabled this range is ±55 kHz which is 53.5% of the filter bandwidth.

As also stated in the data sheet the part can track frequency errors falling in the range of ±35% of the RX filter BW with the AFC feedback compensation enabled and ±25% if it is disabled. These numbers are held if the signal to be received has a modulation index of 1 and the receive filter is as wide as the modulation bandwidth itself. (The example above is one such case.) If, however, the receive bandwidth is wider it will also make the AFC range wider. The selected deviation will also have an impact on the AFC range. As a rule of thumb when the feedback compensation is enabled the AFC will be able to acquire the incoming signal even if only one of the symbols are within the filter bandwidth. This range improvement, however, is restricted by the MODEM_AFC_LIMITER values (see section 3.3 for details). The MODEM_AFC_LIMITER values are set such that the settling would always be completed in 16Tb. (So even though the AFC could acquire a signal that is outside of the range of the MODEM_AFC_LIMITER values it is not guaranteed it would settle within the target of 16Tb.) In conclusion, when the feedback compensation is enabled, the AFC range will always be set to ±35% of the receive filter bandwidth driven by the MODEM_AFC_LIMITER values.

If the feedback compensation is not enabled the rule of thumb is that the AFC will acquire signals where both of the symbols are within the filter bandwidth. In this mode there is no limiter mechanism applied within the chip so an arbitrarily wide AFC range can be achieved with an arbitrarily wide filter (at the price of sensitivity degradation). The modem internal compensation will always settle in 8Tb.

A typical acquisition trajectory of the VCO (PLL feedback compensation enabled) is shown below. A continuous 1010 pattern is present at the antenna input when the chip is sent to RX from READY state. The offset at the wanted frequency is +50 kHz (+400 kHz at the VCO frequency, for the selected frequency of operation).



```
MKR 1       160.6 µs    55.050 00  kHz  Analysis Start Time      0 s
MKR 2       252.0 µs   423.949 31  kHz  Analysis Time Length   500.0 µs
Δ (2-1)      91.4 µs   368.899 31  kHz
                                        Detection : Pos&Neg  Trace Point :  5001

Vertical Scale Center :      0 Hz      250 kHz / Div
```

```
Frequency and Time                     Level                          Trigger
  Center Freq.    3.676 223 040 GHz      Ref. Level     0.00 dBm        Trigger      External
  Freq. Span                  5 MHz                                     Delay        -100.0 µs
  Capture Length           500.0 µs      Attenuator     10 dB
```

Figure 8. VCO Frequency Trajectory on a Standard Preamble in Synchronous Mode

3.6. 4 (G)FSK Mode

4 (G)FSK is only supported in the synchronous demodulation path as only the synchronous slicer can distinguish between four frequency levels (phase states). However, the preamble sequence must be a two-level modulation format (i.e., 2 (G)FSK using the outer deviation points of the 4 (G)FSK signal) as the frequency error detector cannot acquire and track four-level frequency signals. It follows then that all the considerations regarding 2 (G)FSK will hold for 4 (G)FSK too as long as the AFC is frozen upon preamble detection.

4. AFC in Asynchronous Modem Operation

4.1. Overview

The modem is typically configured to operate in synchronous mode if the modulation index is less than 2 and there are at least 32 bits of 1010 preamble pattern. If either of these conditions is not met it is recommended that the modem operate in asynchronous mode. The feedback compensation is always disabled by WDS in asynchronous mode unless there are at least 32 bits of standard 1010 preamble.

This mode is mostly used by so called “direct mode” (usually legacy) applications where the structure of the preamble will not allow the use of the synchronous demodulator. Quite often the requirement against such applications is that the receiver is on for a significant amount of time and therefore the AFC is expected to be running all the time without being frozen.

4.2. Frequency Error Detector in Asynchronous Mode

Frequency error detection happens in the frequency (vs. time) domain when the modem is in asynchronous mode. There are two detectors that can be utilized for the task.

- Moving Average (MA) detector
- Mid-Point detector

Note: In other documentation the Mid-Point detector may get referred to as MEAN or Min-Max detector, Mid-Point, however describes the detector’s operation much better.

Both detectors work towards determining the average frequency of the data stream. The deviation of this average frequency from 0 (dc) is the frequency error. The MA detector can only be used if the data stream is dc balanced (i.e., Manchester encoded), otherwise the data content itself will modulate the measured result. The Mid-Point detector takes the maximum and the minimum frequency within an observation window and calculates the mid-point value of the two numbers. This detector will work if there is at least one of each type of symbol in the observation window. The MA detector is less noisy as it uses the whole signal for determining the mean value, whereas the Mid-Point detector uses only two points.

As opposed to the 2Tb detector a few properties can be configured on the MA and Mid-Point detectors. The 1st parameter is the observation window. This can be set in the MODEM_RAW_SEARCH property.

MODEM_RAW_SEARCH							
7	6	5	4	3	2	1	0
sch_frzen	sch_frzth[2:0]			schprd_hi[1:0]		schprd_low[1:0]	
0	0x5			0x1		0x2	

There are two values for the observation window,; see the gear change related section for details. The window size can be set to 2, 4, 8 and 16Tbit. The setting of this value is a compromise between detector settling time and measurement accuracy. The longer the observation time the less noisy the measurement will become which in turn will cause less desensitization due to uncorrected frequency offset (remember the internal modem compensation is always running). However, a longer preamble sequence is required to allow for the longer measurement time. The graph below shows how sensitivity degrades by shortening the measurement time of the Mid-Point detector from 16 to 2Tbit at three different data rates. WDS will always set this value to 4Tbit assuming (worst case) short preambles; however, if an application has more preamble bits it is recommended that this parameter be increased.

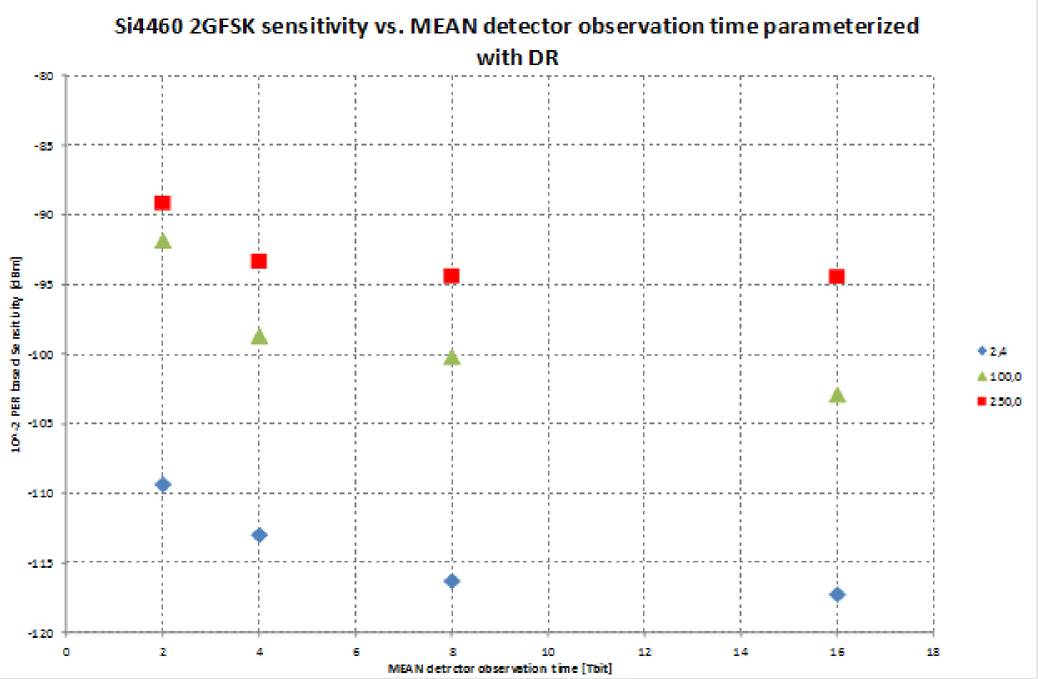


Figure 9. Sensitivity vs Mid-Point Detector Measurement Window

If there are consecutive one or zero symbols in the data stream that are longer than the observation window the detectors will not be able measure the frequency offset correctly. (They would simply return the average frequency of one symbol). There is a feature on the IC that stops the detector output from getting updated if more consecutive symbols arrive than as many as would fit into the window. This feature is enabled by default.

4.3. Frequency Error Compensation in Asynchronous Mode

The parameters of the modem internal frequency compensation in asynchronous mode cannot be adjusted. If there are a sufficient number of preamble bits available, the PLL feedback compensation may also be enabled and the same properties will describe the behavior as outlined in section 2.3.

As mentioned in section 2.3, the modem internal compensation mechanism is different between synchronous and asynchronous modes. As in asynchronous mode the detection happens in the frequency domain the measured frequency offset gets applied to the RAW data (which is still in the frequency domain) before it is reconverted to phase domain to feed the BCR/Slicer block. It is self-evident that the phase compensation of the BCR /Slicer block must be disabled in this mode as there is no phase skew information from the asynchronous detectors. (bcrfbbyp and slicefbbyp field must be set in the MODEM_BCR_MISC1 property.)

To enable either of the asynchronous detectors the 2Tb detector must be disabled and non-standard packet mode must be enabled by setting unstdpk in the MODEM_RAW_CONTROL property.

MODEM_RAW_CONTROL							
7	6	5	4	3	2	1	0
unstdpk	conschk_byp	0x0		pm_pattern[1:0]		rawgain[1:0]	
0	0	0x0		0x0		0x2	

To select between the two detectors field ‘detector [1:0]’ must be set appropriately in property MODEM_OOK_MISC. A value of 0x2 will select the MA detector whereas a value of 0x3 will select the Mid-Point detector. (The rest of the settings are only applicable in OOK mode.)

MODEM_OOK_MISC							
7	6	5	4	3	2	1	0
ookfastma	0x00					detector[1:0]	
0	0x00					0x3	

4.4. Gear Switching in Asynchronous Mode

The gear switching mechanism works much the same as it does in synchronous mode, although the parameters upon which it has an effect are different. Apart from preamble detection and sync word detection there is a third available condition that can trigger the gear switching event. In legacy “direct mode” applications it is conceivable that there is not enough preamble and sync word bits to utilize either of the detection functions (preamble and/or sync word), so if one still would like to make the gear change happen this third option can be used.

- gear_sw[1:0] - default: 0x0 AFC and BCR gear switch control source.
  - 0 = Preamble detection – switch gears after detection of Preamble
  - 1 = Sync word detection – switch gears after detection of Sync Word
  - 2 = Mid-Point frequency error detection – switch gears when the estimated frequency error of the Mid-Point detector in the Asynchronous Demodulator is less than the half eye-threshold for a consecutive number of search periods. Note: The consecutive number of search periods is defined by the sch_frzth[2:0] field in property MODEM_RAW_SEARCH, while the length of each search period (in bits) is defined by the schprd_hi[1:0] field within the same property.
  - 3 = Preamble detection – switch gears after detection of Preamble (same as gear_sw[1:0] = 2’b00)

WDS calculates an eye opening threshold based upon the input values of data rate and deviation. (This eye opening value is specified in the MODEM_RAW_EYE_1 and MODEM_RAW_EYE_0 properties.) When the Mid-Point detector is used and the measured frequency error is consecutively less than half the eye opening threshold for a specified number of bits, the gear change will get triggered. The number of bits through which the condition has to be true can be set in the ‘sch_frzth[2:0]’ field in the MODEM_RAW_SEARCH property.

MODEM_RAW_SEARCH							
7	6	5	4	3	2	1	0
sch_frzen	sch_frzth[2:0]			schprd_hi[1:0]		schprd_low[1:0]	
0	0x5			0x1		0x2	

Features that are /can be affected by gear change in asynchronous mode:

- Freezing the AFC detector

  This is the same feature as in synchronous mode. When the MA detector is selected the same control bit (enafcfrz in property MODEM_AFC_MISC) controls this feature. When the Mid-Point detector is selected the bit that enables / disables this feature is sch_frzen, the MSB of property MODEM_RAW_SEARCH (see above).

MODEM_AFC_MISC							
7	6	5	4	3	2	1	0
enafcfrz	enfbpll	en2Tb_est	enfzpmend	enafc_clksw	ook_zerog	non_frzen	
1	0	1	0	0	0	0	

A note of caution when this feature is used in an application where the receiver is on for a significant amount of time (i.e., direct mode application): Once the AFC detector is frozen there is no means of enabling it again unless the receiver is restarted. It can happen that the AFC locks onto the 1st packet and freezes the AFC, thus effectively disabling it for the rest of the packets to be received while the RX remains on. In such an application this feature should be disabled. In direct mode projects, WDS will implicitly disable the gear switching by setting the gear switching event to sync word detection which will never come about (with the packet handler being disabled).

- AFC detector search window gear change

  The AFC detector may be configured to have different search windows (i.e., measurement times) before and after gear shifting. The search period prior to gear shifting is typically set to a smaller value (shorter time length) than the search period after gear shifting, thus resulting in faster acquisition time at the trade-off of increased noise in the detected value of frequency error. The search period prior to gear shifting is set by the ‘schprd_hi[1:0]’ field, while the search period after gear shifting is set by the ‘schprd_low[1:0]’ field. See MODEM_RAW_SEARCH property above. (WDS will handle this automatically.). This search window property is applicable to both the MA and Mid-Point detectors.

- Changing the gear of the BCR loop

  As mentioned previously, the trigger event for gear shifting of the gain of the BCR loop is the same as for the AFC. Although this feature is not closely related to AFC operation it is worth mentioning to get an overview of the interactions in the modem. See MODEM_BCR_GEAR property for details. (WDS will handle this automatically.)

- Changing the RX BB filter bandwidth

  This feature works the same way as it does in synchronous mode; see the details in that section.

4.5. AFC Configurations from WDS

Table 1 summarizes how the AFC configuration changes with WDS input parameters.

Table 1. AFC Configuration Changes with WDS Input Parameters

WDS Input				AFC Configuration			
Application Mode	**Preamble Pattern**	**Modulation Index**	**Machester Coding on Whole Packet****	**Demod Path**	**AFC Detector**	**Preamble Detection**	**Gear Switching Event**
Direct	NA	≥2	NA	async	Mid-Point	no	no
Direct	NA	<2	NA	async	Mid-Point	no	no
Packet	Std. 1010 PM pattern (>32 bit)	≥2	NO	async	MA	in modem	preamble detect
Packet	Std. 1010 PM pattern (<32 bit)	≥2	NO	async	Mid-Point	no	sync detect
Packet	Non std. Pattern (>32 bit)	≥2	NO	async	Mid-Point	in PH*	preamble detect
Packet	Non std. Pattern (<32 bit)	≥2	NO	async	Mid-Point	no	sync detect
Packet	Std. 1010 PM pattern (>32 bit)	<2	NO	sync	2Tb	in modem	preamble detect
Packet	Std. 1010 PM pattern (<32 bit)	<2	NO	async	Mid-Point	no	sync detect
Packet	Non std. Pattern (>32 bit)	<2	NO	async	Mid-Point	in PH*	preamble detect
Packet	Non std. Pattern (<32 bit)	<2	NO	async	Mid-Point	no	sync detect
Packet	Std. 1010 PM pattern (>32 bit)	≥2	YES	async	MA	in modem	preamble detect
Packet	Std. 1010 PM pattern (<32 bit)	≥2	YES	async	MA	no	sync detect
Packet	Non std. Pattern (>32 bit)	≥2	YES	async	MA	in PH*	preamble detect
Packet	Non std. Pattern (<32 bit)	≥2	YES	async	MA	no	sync detect
Packet	Std. 1010 PM pattern (>32 bit)	<2	YES	sync	2Tb	in modem	preamble detect
Packet	Std. 1010 PM pattern (<32 bit)	<2	YES	async	MA	no	sync detect
Packet	Non std. Pattern (>32 bit)	<2	YES	async	MA	in PH*	preamble detect

Table 1. AFC Configuration Changes with WDS Input Parameters

Packet	Non std. Pattern (<32 bit)	<2	YES	async	MA	no	sync detect

Notes:

1. When preamble detection is done in the PH (referred to as the non-standard preamble detection mode) it is the user's responsibility to set it up from WDS. Select a customizable preamble pattern on the preamble field on the "Packet" tab on the RCA panel.
2. There is no control in WDS that would enable/disable Manchester coding/decoding for the whole packet. It has to be set individually on each field including the preamble and the sync word.
3. When BERT mode is ticked ("Direct RX" project only) the whole AFC circuitry will be disabled. See section 4.2 for details.

4.6. Disabling the AFC

For debug purposes it may be desirable to disable the whole AFC circuitry. This can be done by clearing the MSB bit of property MODEM_AFC_GAIN1. This is how the AFC gets disabled when the “Enable BER mode” check box is ticked. In normal operation mode, however, this bit is always set by WDS.

MODEM_AFC_GAIN_1							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
enafc	afcbd	0	afcgain_12_8[4:0]				
1	0	0	0x03				

4.7. Reading Back the Measured Frequency Error

It is possible to read back the frequency error measurement result from the chip. This feature however only works when the PLL feedback compensation is enabled. The frequency error value is updated when the AFC gets frozen at the gear switching event. The value can be read back as the last two bytes of the reply to the “GET_MODEM_STATUS” command.

The value is in twos complement code. Its unit is the PLL frequency resolution that can be calculated as follows:

PLL frequency resolution [Hz] = XO_frequency[Hz]/2^18/output_divider_ratio

The output_divider_ratio parameter is frequency band dependent and can be read back (or checked in a batch file) from property “MODEM_CLKGEN_BAND”.

MODEM_CLKGEN_BAND							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0x0				sy_sel	BAND[2:0]		
0x0				1	0x0		

- BAND[2:0] - default:0x0
  - 0 = Output is FVCO/4.
  - 1 = Output is FVCO/6.
  - 2 = Output is FVCO/8.
  - 3 = Output is FVCO/12.
  - 4 = Output is FVCO/16.
  - 5 = Output is FVCO/24.
  - 6 = Output is FVCO/24.
  - 7 = Output is FVCO/24.

4.8. Detector Characteristics vs. Input Signal Deviation

The following frequency offset tolerance curves have been taken on a Si4463 RFIC at 868 MHz configured to a modulation format of 2GFSK with 100 kbps DR and 50 kHz deviation. The test signal deviation has been varied and the AFC “bucket curve” (sensitivity vs. frequency offset) has been recorded at each deviation setting.

The test packet contained 40 bits of preamble, 16 bits of sync word, 9 bytes of payload and a 2 byte CRC at the end. Sensitivity is measured as the input power level to the RFIC where the Packet Error Rate (PER) is 1%.

The following two charts show the AFC behavior when the synchronous demodulation path is selected with no PLL FB compensation (1st graph) and with the PLL FB compensation enabled (2nd graph). In this mode the 2Tb error detector is deployed.

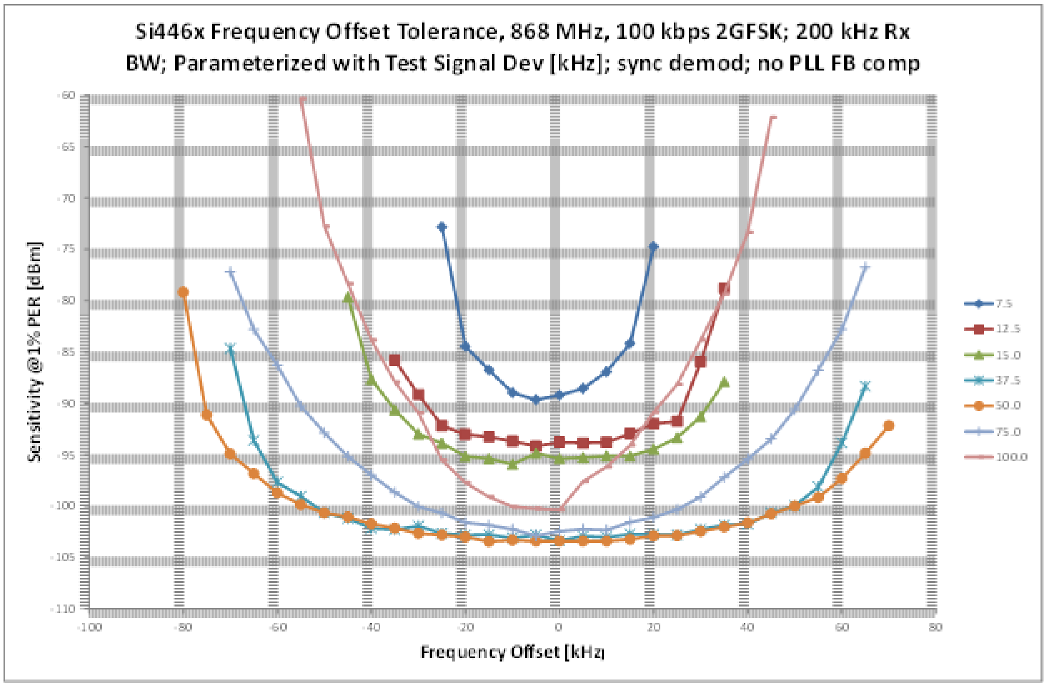


Figure 10. AFC “Bucket” Curves vs Test Signal Deviation without FB Compensation; Synchronous Mode

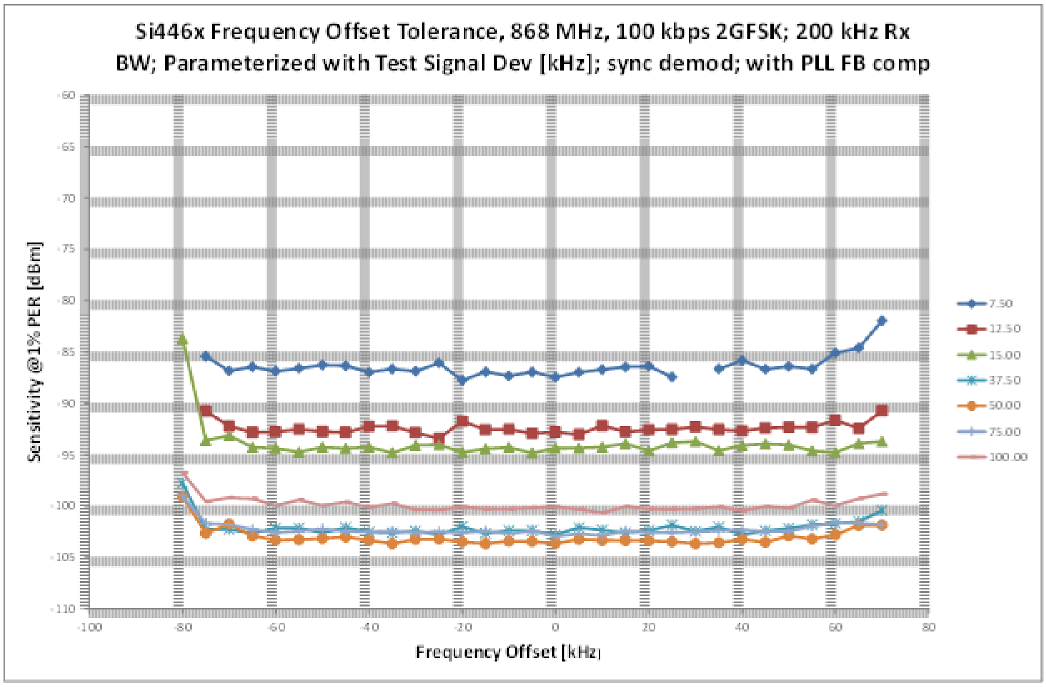


Figure 11. AFC “Bucket” Curves vs Test Signal Deviation with FB Compensation; Synchronous Mode

The next graph shows the MA detector’s characteristics for different test signal deviations.

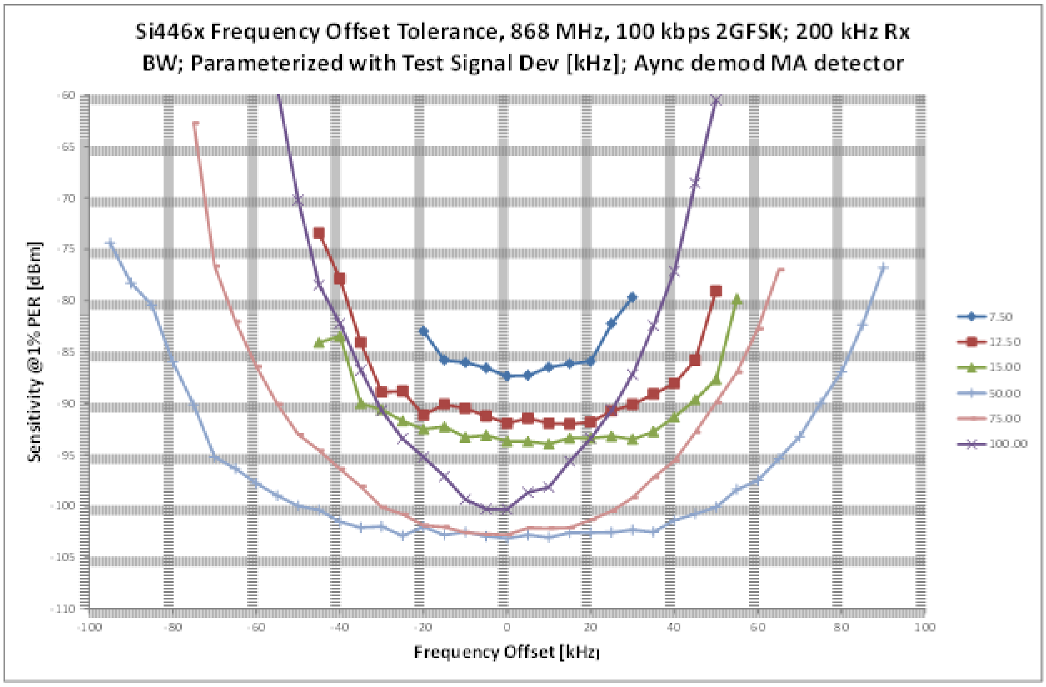


Figure 12. AFC "Bucket" Curves vs Test Signal Deviation with FB Compensation; Asynchronous Mode; MA Detector

And finally see below the same set of curves for the Mid-Point detector.

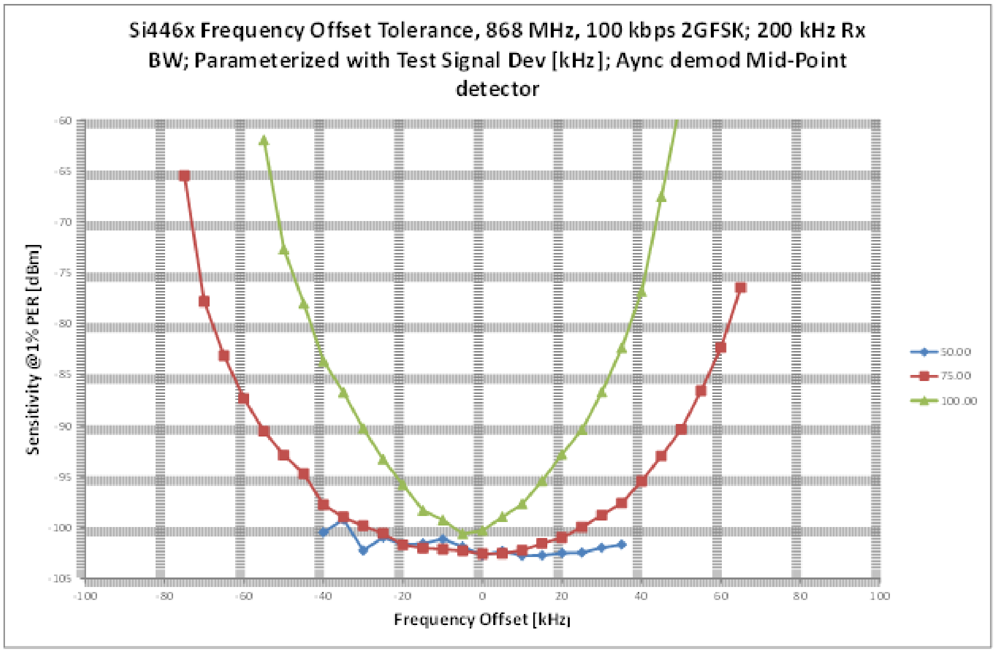


Figure 13. AFC "Bucket" Curves vs Test Signal Deviation with FB Compensation; Asynchronous Mode; Mid-Point Detector

At lower deviation values (7.5, 12.5, 15 kHz) there is a PER floor using the Mid-Point detector; the 1% PER value cannot be reached. The Mid-Point detector is more sensitive to eye narrowing at any given nominal deviation setting.

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701
Tel: 1+(512) 416-8500
Fax: 1+(512) 416-9669
Toll Free: 1+(877) 444-3032

Please visit the Silicon Labs Technical Support web page:
https://www.silabs.com/support/pages/contacttechnicalsupport.aspx
and register to submit a technical support request.

Patent Notice
Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog-intensive mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team.

The information in this document is believed to be accurate in all respects at the time of publication but is subject to change without notice. Silicon Laboratories assumes no responsibility for errors and omissions, and disclaims responsibility for any consequences resulting from the use of information included herein. Additionally, Silicon Laboratories assumes no responsibility for the functioning of undescribed features or parameters. Silicon Laboratories reserves the right to make changes without further notice. Silicon Laboratories makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Silicon Laboratories assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. Silicon Laboratories products are not designed, intended, or authorized for use in applications intended to support or sustain life, or for any other application in which the failure of the Silicon Laboratories product could create a situation where personal injury or death may occur. Should Buyer purchase or use Silicon Laboratories products for any such unintended or unauthorized application, Buyer shall indemnify and hold Silicon Laboratories harmless against all claims and damages.