MiloSAR Documentation

Version 0.2

Darryn Jordan

March 20, 2018

Contents

Chapter 1

Introduction

Chapter 2

Synthesizer

2.1 μWire Interface

The LMX2492EVM is packaged with a USBtoMANY converter, enabling users to program the evaluation module through its μWire connector using TI's CodeLoader. This process is however tedious, especially when programming two synthesizers with different waveforms. The LMX2492 datasheet provides the timing requirements and clocking structure for programming the synthesizer through its μWire connector pins. GPIO pins on the Red Pitaya were used to program both synthesizers efficiently. Figure 2.1 illustrates the connection between the Red Pitaya and two LMX2492 evaluation boards.

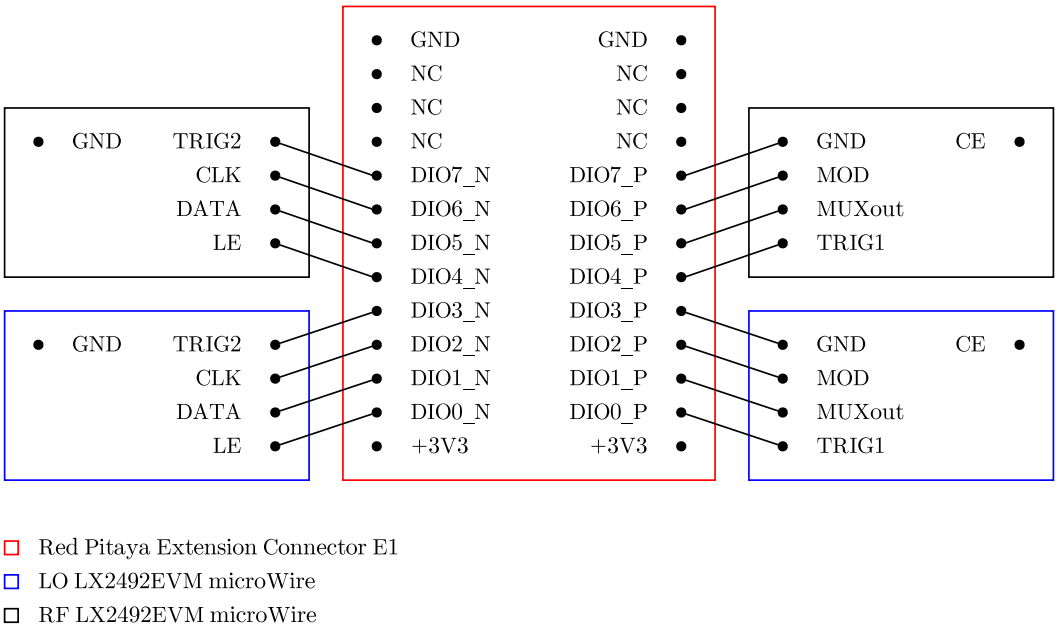

Figure 2.1: Wiring diagram for connection between Red Pitaya and the LMX2492.

Chapter 3

FPGA

3.1 Registers

3.1.1 Channel X Status

Both recording channels within the FPGA require a status register to provide the CPU with the location of the RAM writer's current pointer location.

Table 3.1: Channel X status register.

B3								B2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Channel X Pointer [31:16]															

B1								B0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Channel X Pointer [15:0]															

3.1.2 GPIO

The GPIO register is currently haphazard and contains the clock divisor value, which is used to set the PRF. Furthermore, an enable flag, bit 8, is used to enable the receive chain. It is suggested that both of these settings be moved the the configuration register once the decimation factor is fixed.

Table 3.2: GPIO register.

B3								B2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Clock Divisor [31:16]															

B1								B0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Clock Divisor [15:9]							E	GPION [7:0]							

3.1.3 Reference Signal Phase Increment

Provides the FPGA with the DDS phase increment used to generate the 50 MHz phase reference signal.

Table 3.3: Synthesizer reference signal phase increment register.

B3								B2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Phase Increment [31:16]															

B1								B0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Phase Increment [15:0]															

3.1.4 Cancellation Signal Phase Increment

Provides the FPGA with the DDS phase increment used to generate the IF cancellation signal.

Table 3.4: Cancellation signal phase increment register.

B3								B2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Phase Increment [31:16]															

B1								B0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Phase Increment [15:0]															

3.1.5 Cancellation Signal Phase Offset

Provides the FPGA with the DDS phase offset used to adjust the phase of the cancellation signal.

Table 3.5: Cancellation signal phase offset register.

B3								B2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Phase Offset [31:16]															

B1								B0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Phase Offset [15:0]															

3.1.6 Profile Integrator Configuration

Specify the number of samples to integrate over and the number of pulses to integrate. Number of samples is currently limited to 13 bits. This can be increased to 15 bits if the FIFO depth is increased in the FPGA design.

Table 3.6: Cancellation signal phase offset register.

B3								B2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Number of Pulses [31:16]															

B1								B0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			Number of Samples [12:0]												

3.1.7 Profile Integrator Index

Specify the start and end index of the integrated profile. **Note that sample indexing begins at one**.

Table 3.7: Profile integrator index register.

B3								B2							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
End Index [31:16]															

B1								B0							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Start Index [15:0]															

3.2 Digital Receiver Design

Design of the digital receiver begins with the selection of a suitable analogue bandpass filter. The SBP-10.7+ bandpass filter (BPF), procured from Mini-Circuits, features a 3 dB bandwidth of 3.8 MHz and lower-side cut-off frequency, $f_{3\,\mathrm{dB}}$, of 8.5 dB as illustrated in Figure 3.1. The SBP-10.7+ serves a dual role in suppressing feed-through and implementing sensitivity frequency-control.

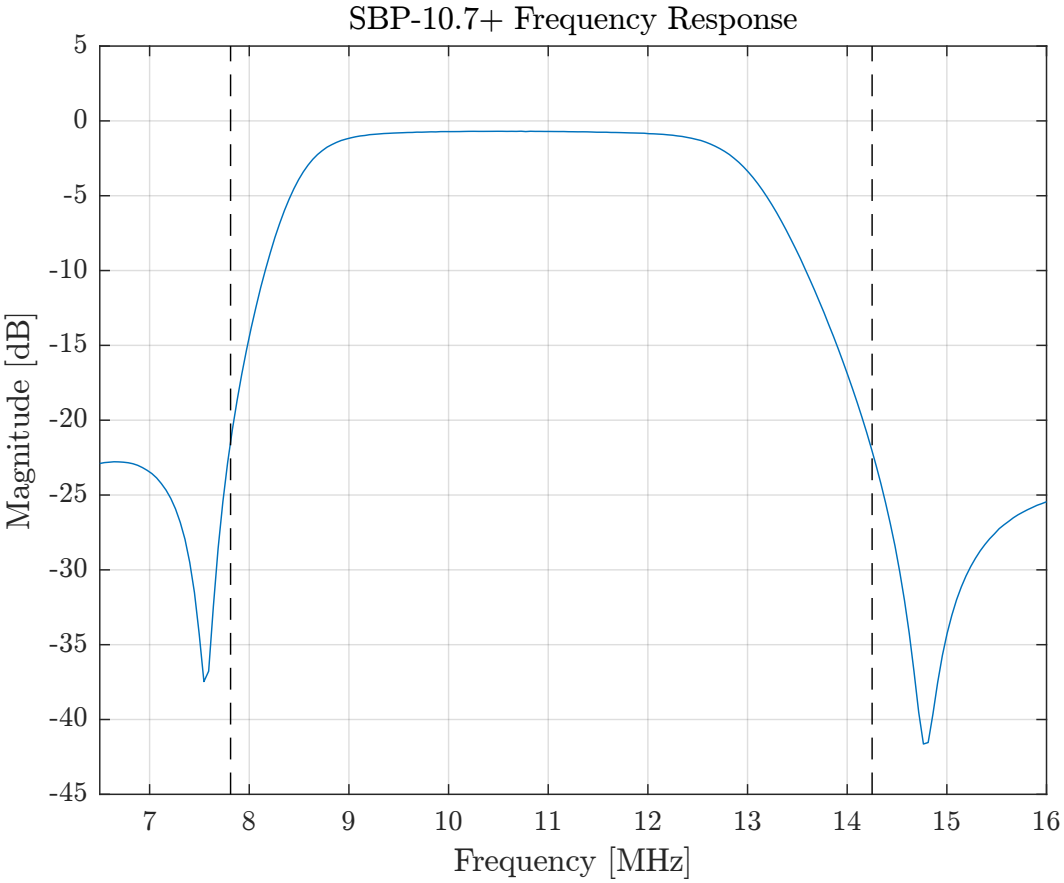

Figure 3.1: Frequency response of the SBP-10.7+ BPF with dashed lines indicating the Nyquist band of interest .

Following the SBP-10.7+, the beat spectrum is oversampled at a rate of 125 MHz using the Red Pitaya. This results in a spreading of quantization noise over the full sampled bandwidth (62.5 MHz). With appropriate filtering, quantization noise is greatly reduced. An additional benefit of oversampling is the ability to perform decimation through averaging in order to increase the effective number of bits (ENOB). Equation 3.1 reveals that an additional 1.5 bits are gained through decimation by a factor of 8.

$$n = \log_4(DF) \tag{3.1}$$

The SBP-10.7+ BPF is ideally positioned to make use of the second Nyquist zone

when using a decimation factor (DF) of 8. All signals of interest must therefore be contained within the $f_s/2$ to f_s band, or 7.8125 MHz to 15.625 MHz respectively, illustrated by the dashed lines in Figure 3.1. Components out of the Nyquist band of interest are therefore attenuated by more than 20 dB.

Having oversampled the beat spectrum at IF, the first step in the digital receive chain is pass the signal through the bandpass FIR decimation filter illustrated in Figure 3.2.

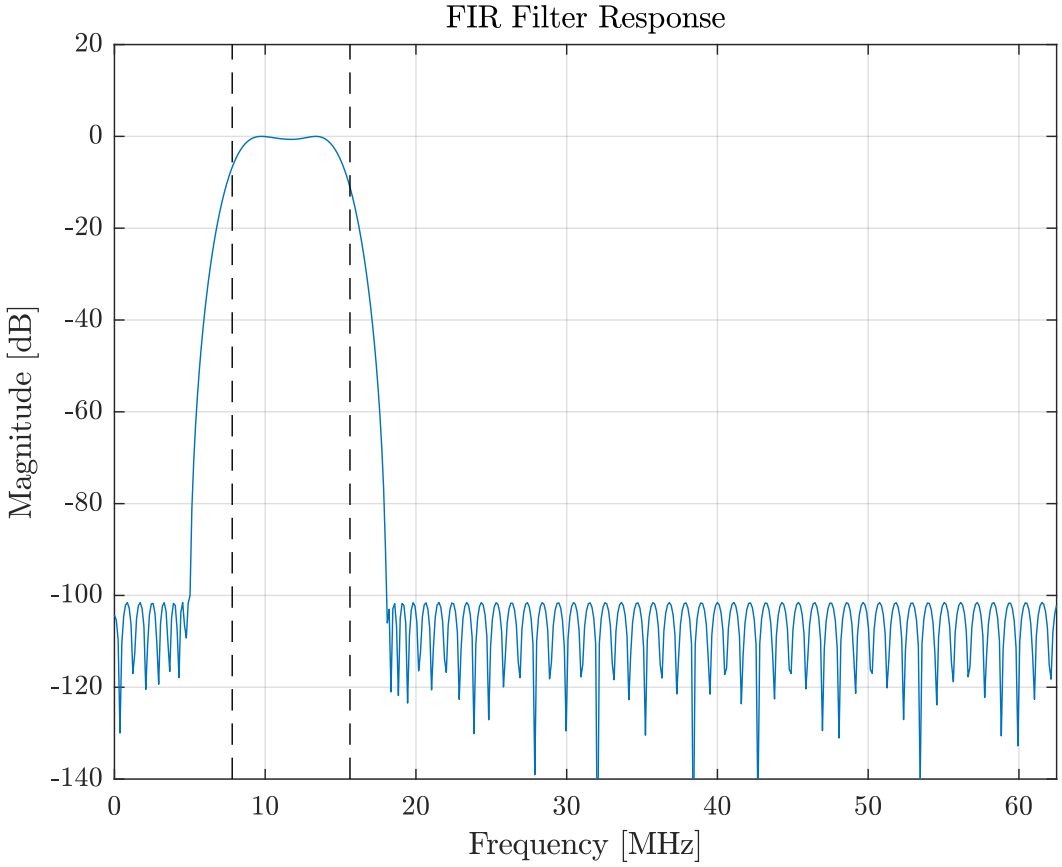

Figure 3.2: Frequency response of the FIR filter within the Red Pitaya.

It is important to note that the SBP-10.7+ analogue filter suppresses signals out of the second Nyquist band by an additional 20 dB. Over 100 dB of suppression is provided by this 115-tap FIR filter. In addition to this, the filter is responsible for implementing decimation through averaging. As explained previously, 1.5 bits are gained through averaging. The output samples are however limited to 15 bit samples, taking advantage of an extra 1 bit.

Both the main and reference channel pass through these filters and are saved to disk for further processing in Matlab. Figure 3.3 illustrates the frequency domain steps for demodulating the main channel using the reference channel.

The topmost two spectrums illustrate the process of Nyquist folding during undersampling. The analogue beat spectrum folds around $f_s/2$, resulting in a flipped digital replica of the spectrum. The third spectrum is simply the same result of Nyquist folding for the reference channel.

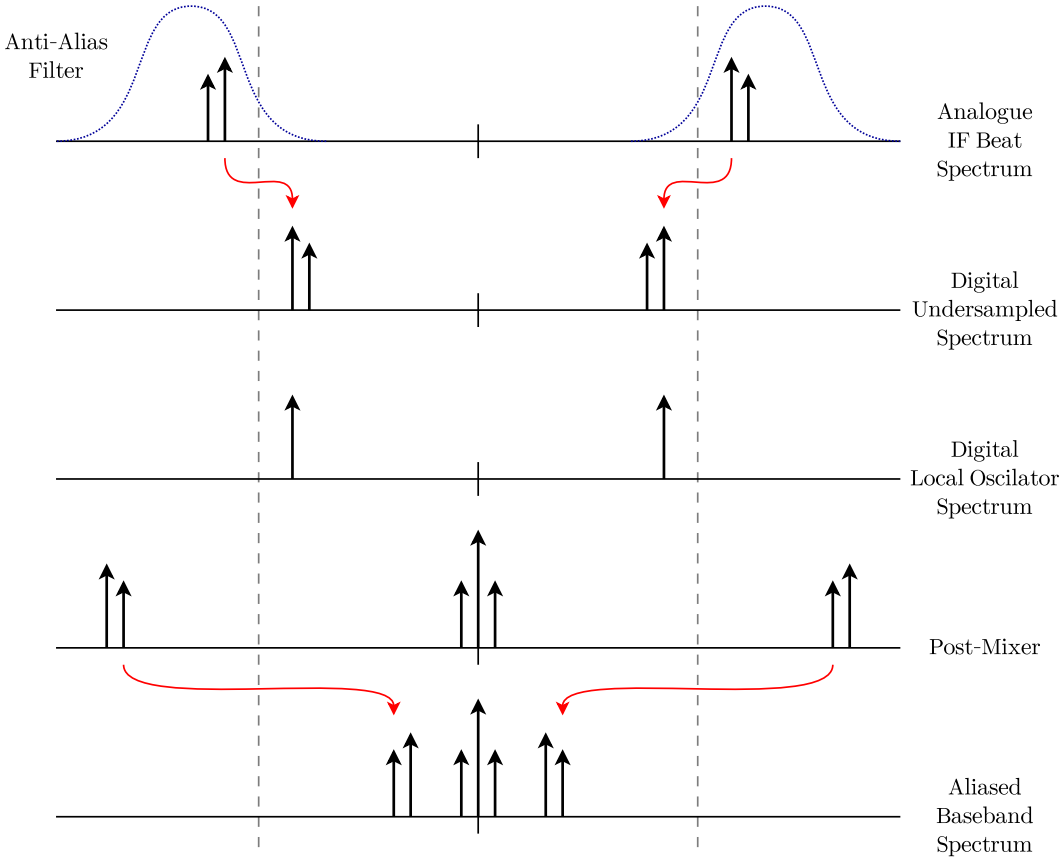

Figure 3.3: Down-conversion with aliasing owing to low sample rate.

The digitised main and reference channels are then multiplied together to perform demodulation, however, the post-mixer spectrum reveals a potential problem. The components shifted to a higher frequency fall out to the sampled bandwidth and therefore fold into the baseband spectrum. This unforeseen problem can however be avoided by zeroing the components that fold into the baseband spectrum before mixing.

Figure 3.4 illustrates the result of nullifying half of the main and reference channel before mixing. This process can be seen to prevent aliasing and avoid a doubling of the DC component.

The final processing step is then to preform integration of range profiles on the real data. Addition of two 15 bit numbers results in a single bit growth. Our data is now 16 bit and the PRF has been halved, resulting in a halving of the data rate.

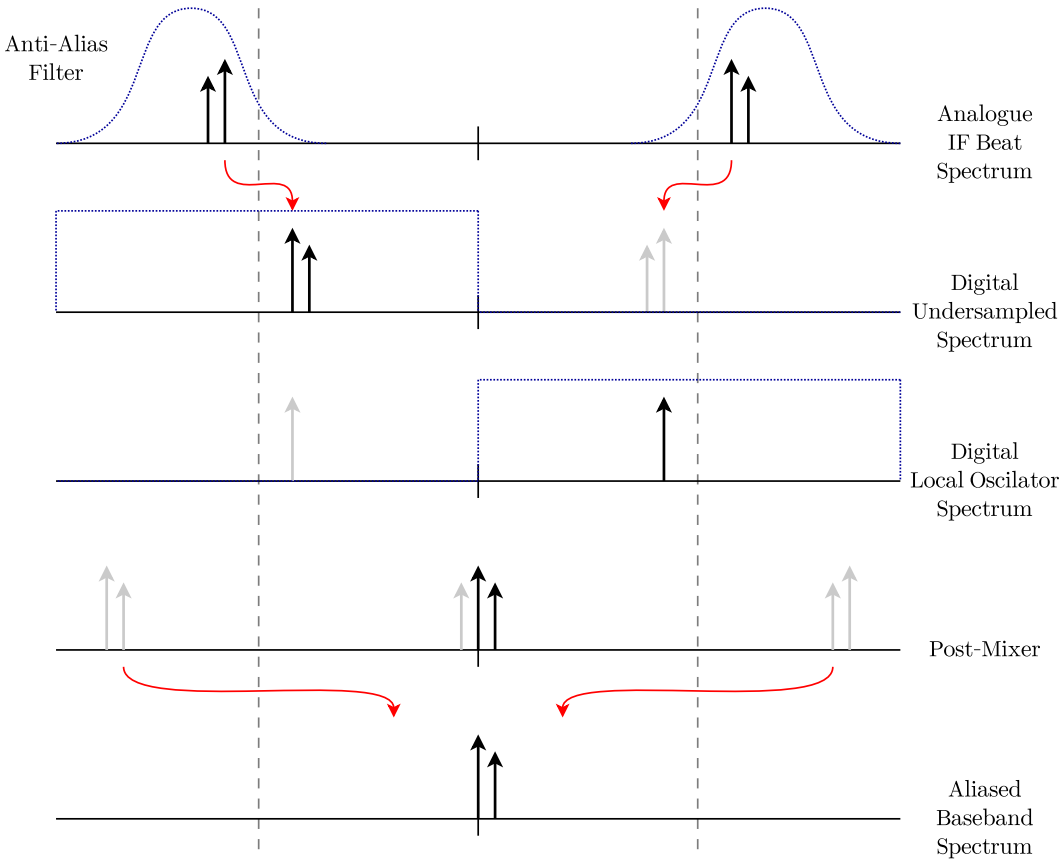

Figure 3.4: Down-conversion without aliasing.