

NON-MAXIMALLY DECIMATED FILTER BANKS ENABLE ADAPTIVE FREQUENCY HOPPING FOR UNMANNED AIRCRAFT VEHICLES

^aElettra Venosa, ^aBert Vermeire, ^aCameron Alakija, ^afred harris, ^aDavid Strobel,

^aSpace Micro, San Diego, CA

^bCharles J. Sheehe, Glenn Research Center, Cleveland, OH

^cMarwan Krunz, University of Arizona, Tucson, AZ

Abstract

In the last few years, radio technologies for unmanned aircraft vehicle (UAV) have advanced very rapidly. The increasing need to fly unmanned aircraft systems (UAS) in the national airspace system (NAS) to perform missions of vital importance to national security, defense, and science has pushed ahead the design and implementation of new radio platforms. However, a lot still has to be done to improve those radios in terms of performance and capabilities. In addition, an important aspect to account for is hardware cost and the feasibility to implement these radios using commercial off-the-shelf (COTS) components. UAV radios come with numerous technical challenges and their development involves contributions at different levels of the design. Cognitive algorithms need to be developed in order to perform agile communications using appropriate frequency allocation while maintaining safe and efficient operations in the NAS and, digital reconfigurable architectures have to be designed in order to ensure a prompt response to environmental changes. Command and control (C2) communications have to be preserved during “standard” operations while crew operations have to be minimized. It is clear that UAV radios have to be software-defined systems, where size, weight and power consumption (SWaP) are critical parameters.

This paper provides preliminary results of the efforts performed to design a fully digital radio architecture as part of a NASA Phase I STTR. In this paper, we will explain the basic idea and technical principles behind our dynamic/adaptive frequency hopping radio for UAVs. We will present our Simulink model of the dynamic FH radio transmitter design for UAV communications and show simulation results and FPGA system analysis.

I. Introduction

In this paper, we present an efficient digital implementation of a fully digital dynamic spread spectrum frequency hopping (SS-FH) transmitter for UAVs that is based on multirate signal processing techniques. Dynamic FH systems are a very good candidate for UAV radios because they enable flexible and agile multichannel communications. In our system, the pseudo-random noise (PN) sequence can be changed on the fly, allowing secure and efficient communications. Multirate signal processing allows us to optimize the signal sample rate at different points of the digital communication chain. This results in power saving when high-rate communications are not required. SS techniques are well known to be highly jamming-resistant, which is an important feature to have in UAV radios. Our proposed radio has been designed for COTS FPGAs and a preliminary system analysis shows that it actually utilizes only a small percentage of logic cells, digital signal processing (DSP) slices and memories available in a Xilinx Virtex 7 chip. This implies that size and weight of the proposed architecture are significantly reduced and well suited for UAVs. The efficiency of the proposed architecture comes from the fact that non-maximally decimated filter banks (NMDFBs) are used to perform digital frequency hopping. NMDFBs use inverse fast Fourier transform (IFFT) algorithms to hop the signals over different center frequencies. FH systems have been developed long time ago however, no fully digital solution for them exists yet. Standard FH systems are still developed in the analog domain because the spreading operation widens (spreads) the signal bandwidth to a point that cannot be handled by digital circuits. The fact that the signal spreading still happens in the analog domain limits the development of dynamic/adaptive FH systems.

While it is easy to change the PN sequence in the digital world, flexibility in the analog world is paid with the replication of hardware circuits which increases the SWaP. Because the processing speed of FPGAs is limited by the system clock, they cannot process data faster than their clock frequency. This generally limits the analog bandwidth that can be processed in the digital domain. By using NMDFBs we overcome this limitation. NMDFBs allow us to perform very wideband digital signal processing, enabling fully-digital dynamic/adaptive FH systems.

The implementation of the FH radio described in this paper is a small part of a wider picture. Our cognitive/agile radio is equipped with a wideband digital spectrum sniffer that is composed of an ensemble averager spectrum analyzer working on a windowed FFT signal. The outputs of the spectrum sniffer feed a cognitive engine in which a proactive sensing algorithm is implemented. The proactive sensing algorithm changes the PN sequence according to channel/interference conditions. The proposed radio adapts its internal states to the environmental conditions and by doing that is quite robust against adverse channel conditions.

This paper is organized as follows: in Section II the high level digital design of the proposed FH transmitter is provided. In Section III Matlab simulations are provided to demonstrate its correct functioning. In Section IV its Simulink model is presented while in Section V the summary of the VHDL design is given. The conclusions, along with suggestions for future developments, are given in Section VI.

II. High Level System Design

Spread spectrum (SS) techniques were, at the beginning, investigated for military applications because of their characteristic of being highly jamming resistant. Today they are applied in many other important areas like mobile communication, navigation, and test systems. Their name derives from the fact that the modulated signal is spread over a wider bandwidth before being transmitted. The spreading of the signal band provides a long list of benefits such as interference suppression, energy density reduction and fine time resolution. The signal spreading also allows sharing of a communication resource among numerous users in

a coordinated way (multiple access transmission techniques). This technique achieves frequency diversity gain over frequency selective fading channels and also has a low probability of interception.

In frequency hopping spread spectrum transmissions the carrier frequency of the signal is periodically changed before transmission. A frequency band, called hopping band, is accessed by a controlled pseudorandom sequence, called frequency hopping pattern, that shifts it to a different center frequency selected from N possible center frequencies. The number N is usually chosen to be very large because the bigger the number of possible hopping frequencies the better the FH system performs in terms of interference suppression, probability of interception and multiple access possibility. The set, of dimensionality N , containing all the possible center frequencies is usually referred to as hopset. The large dimensionality of the hopset is one of the reasons for which fully digital FH systems are not available yet. When the bandwidth increases the processing speed has to increase accordingly and, unfortunately, currently available FPGAs are limited by their own clock speed. Processing digital data serially will limit the development of wideband digital systems. Smarter parallel digital signal processing will allow the bandwidth increase opening the door for software and cognitive radios.

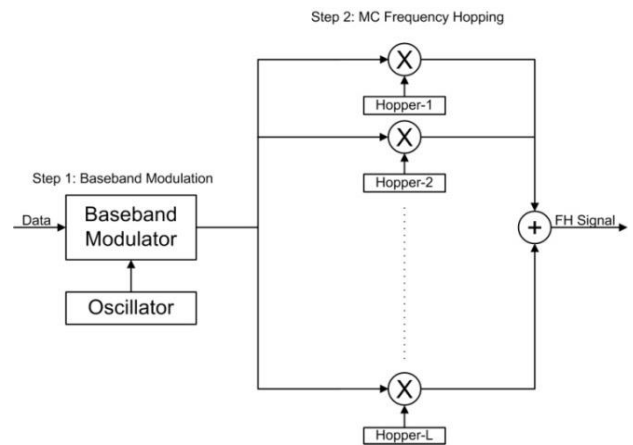


Figure 1: Standard MCFH Modulator.

In standard multi-carrier frequency hopping (MCFH) transmitters, the modulation process occurs in two steps. At first the input signal is baseband modulated (by using an analog or a digital modulator) and then, the complete hopping band is hopped independently, over the L sub-carriers, by L analog second tier up converters which are driven by pseudo random code generators. For each sub-carrier the transmitted symbols have different hopping patterns. At each hop time the pseudorandom code generators feed the frequency synthesizers a frequency word which dictates one of the possible center frequencies from the N/L hopsets.

Limiting factors in the development of MCFH systems are the dimensionality of the hopset and the number of multiple analog frequency hoppers or mixers, L , that have to be embedded in the system. Analog oscillators are costly and bulky. In spite of the efforts made in the direction of digitizing both the MCFH modulator and demodulator, today, frequency hopping systems are still implemented in the analog way. No computationally efficient solution has been found until now for performing the hopping of the baseband modulated signal digitally. Some hybrid solutions are present in the literature, but no fully digital modulator exists for performing multi-carrier frequency hopped transmissions.

In this paper we are showing the design a novel efficient fully digital architecture for multi-carrier frequency hopping modulators. The secret here is to design the digital architecture keeping in mind the clock limits of FPGAs and the opportunity of performing parallel signal processing by using polyphase filter banks. Contrary to standard solutions, the efficacy of the proposed architecture increases when the number of the subcarriers increases.

The key element of the proposed architecture is the M -path polyphase up converter channelizer. In its standard operating mode, an M -Path polyphase up converter channelizer, that is composed of an M -Point IFFT, an M -path partitioned filter and an output commutator, simultaneously performs three separate tasks. The first task is selection of the number of spectral Nyquist zones or channels. This is determined by M , the number of paths as well as the size of the

IFFT. The second task is channel shaping. This is determined by the low-pass prototype filter from which the M -path polyphase partition is formed. The third task is the upsampling operation which occurs in the output commutator. With all the input ports enabled, the M -path up converter channelizer shifts, by aliasing, the input base-band signals over M fixed, high order Nyquist zones. When only the input ports corresponding to the desired output channels are enabled, this engine shifts the input signal to the desired Nyquist zones, thus it represents a flexible fully digital, frequency selective, up converter and it can be used for building efficient (low workload), fully digital hopping structure.

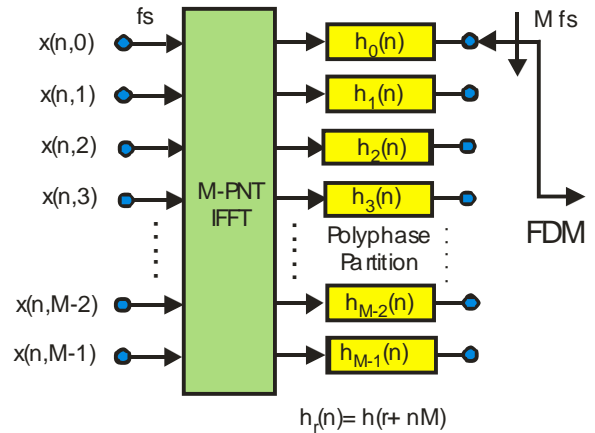


Figure 2: Standard M -path Polyphase Modulator; M -PNT IFFT, Polyphase Partitioned Filter and Output Commutator.

Figure 2 shows the architecture of a standard M -path up converter channelizer. In this engine, M -point IDFT, that for computational efficiency is implemented with the IFFT algorithm, performs two simultaneous tasks: an initial up sampling of 1-to- M which forms an M -length vector for each input sample $x(n,k)$ and further imparts a complex phase rotation of k cycles in M -samples on the up sampled output vector. The IFFT generates a weighted sum of complex vectors containing integer number of cycles per M -length vector while the polyphase filter forms a sequence of column coefficient weighted, MATLAB's dot-multiply, versions of these complex spinning vectors. The sum of these columns, formed by the set of inner products in the polyphase partitioned filter, is the shaped version of the up-converted M -length vector

output from the IFFT. The M-port commutator, at the end of these processes, takes M consecutive samples from the output ports of the M-path filter to deliver the 1-to-M interpolated, up converted and shaped time series formed by the channelizer.

Summarizing, we can describe the three basic operations performed by a standard polyphase up converter channelizer as: digital up conversion to higher Nyquist zones by the IFFT, spectral shaping and filtering due to the M-path partitioned filter weights, sample rate change due to the output commutator. These three operations are completely independent of each other and they can be modified to achieve different goals based on different channelizer applications. It is important to notice that the efficacy of this engine increases when the number of paths increases. The polyphase channelizer is more efficient, in terms of workload and power consumption, when the number of paths, M, is large. This is due to the IFFT embedded in this engine. When M is a small number ($M < 16$) there is no computational advantage in using channelizers even if, compared to a standard bank of filters, up converters and up samplers, they still provide more flexibility and compactness of design. Channelizers thus do not have limitations due to the signal bandwidth. The whole filter bank is built at a cost of a single low-pass prototype filter which is reshaped over M paths. This is why they are considered the most valid solution in handling wideband digital signal processing.

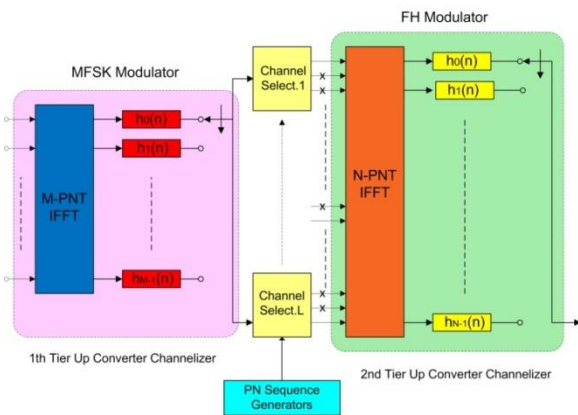


Figure 3: High-Level Block Diagram of Proposed FH Modulator.

For the purpose of designing a MCFH modulator we cascade two standard forms of the up converter channelizer that have equal channel spacing,

channel bandwidth and output sampling frequency. In Figure 3, the high-level block diagram of the proposed architecture is shown. Like its analog antecedent, this modulator is composed of two stages, each one formed of a polyphase up converter channelizer.

III. Matlab Simulations

Matlab simulations are the first step when designing digital systems. Matlab is the tool used in scientific communities for proving the correct functioning of theoretical concepts. It is generally used as a floating point tool and this means that it does not take into account the issues that arise when fixed point math is adopted.

All that we have in the hardware are bits. Bits travel from one point to another in a digital system. The floating point numbers are represented with a limited number of bits thus an approximation of them is obtained. The approximation becomes closer and closer to the floating point numbers when more and more bits are used to represent them. Unfortunately, it is not recommended to represent numbers with many bits because this will take away hardware resources and will increase the power consumption of the system. On the other side, using a limited number of bits causes a decrease in the performance. Often the effect of a limited number of bits is found to be similar to the effect of an increased noise power on the signal. Sometimes the increased noise level has a destructive effect on the correct functioning of the system. Selecting the appropriate number of bits to represent a digital signal is not an easy task. The result has to be a good trade-off between required performance and used resources.

For the proof of concept, on the transmitter side, we selected the 8-FSK modulation as first tier baseband modulation, thus the first tier channelizer is an 8-path polyphase engine, which performs the 8-FSK modulation of the input signal digitally. This channelizer, whose channel center frequencies are selected for matching the center frequencies of the FSK modulated signal, aliases the input signal to the selected center frequency among the 8 possible center frequencies. This task is accomplished by enabling the corresponding channelizer input port. After the baseband modulation has been performed

we still need to hop the signal onto the L sub-band according to the hopping patterns. The hopping process is nothing different but another up conversion process, so, by inputting the FSK modulated signals to the proper L ports of the second tier up converter channelizer, we acquire the capability to hop the modulated signal over the L possible hopping center frequencies.

Notice that those simulations only have the scope of proving that our concept is right and the dimensionality of the channelizer little matters for that. In the final system, multiple base-band modulation options will be provided. Our system is a software radio and the waveform should change according to environmental condition. This is not difficult to achieve: multiple modulation blocks will be programmed and embedded in the system and the parameter which selects one or the other will be an output of the adaptive algorithm developed for us by the University of Arizona.

The number of arms of the second tier channelizer, N , is selected according to the desired dimensionality of the hopset while the channelizer channels' center frequencies are designed to match the frequencies composing the hopset. The dimensions (number of paths and IFFT block) of the two tier channelizers composing the proposed architecture are of course different. The number of paths in the first one is selected according to the desired baseband modulation level while the number of paths of the second one is selected according to the dimensionality of the hopset. If a different baseband modulation is selected the proposed scheme might be further simplified and, for example in the case of BFSK modulation, the first tier up converter channelizer could be avoided.

For simulation and demonstration purposes and, according to our customer suggestions and to our hardware requirements, we selected the number of paths, N , of the second channelizer, to be equal to 32. Note that increasing the number of paths of the second channelizer only slightly affects the total workload of the proposed modulator. This is a clear consequence of the fact that the IFFT block embedded in the polyphase channelizer provides its best performance with higher dimensionality. Channelizers with a larger number of paths are currently implemented in a multitude of digital radios.

L channel selectors, controlled by pseudo noise sequence generators, and placed between the two engines, deliver the samples to the proper input ports of the second up converter channelizer for performing the desired L hops. No analog frequency synthesizers are required for this task. The hopping is performed by enabling and disabling the appropriate links between the first and second tier channelizers.

Figure 4 shows, in the upper subplot, the impulse response of the low pass prototype filter which defines the desired FSK bandwidth while, in the lower subplot, embraced in the red dotted line, are all the possible 8 FSK bandwidths to be selected.

Figure 5 describes the FSK bandwidth we selected for this Matlab simulation while, Figure 6 shows the upconverted 8-FSK modulated time domain signal and spectrum. The output bandwidth is 500MHz.

Figure 7 shows the impulse response and the frequency response of the low pass prototype filter for the second channelizer. In particular, in the upper subplot its impulse response is shown while, in the lower subplot, its frequency response is reported. Notice that the out-of-band attenuation is more than 80dB.

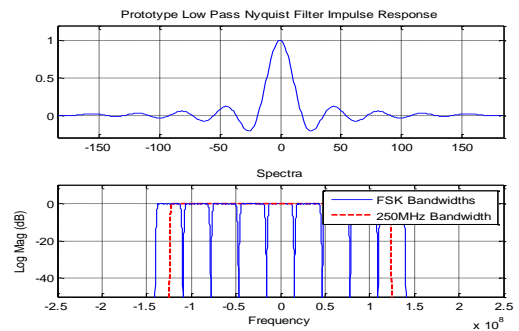


Figure 4: Impulse Response and Frequency Response of Proposed FSK Modulator.

Figure 8 shows the FH filter bank frequency response. As a trade-off between the desired number of hops and our hardware capabilities, we defined the dimensionality of this engine to be 32. To select a larger hopset would mean to make impossible an early hardware demonstration using the hardware we have at SMI. As specified before, those numbers can be easily modified when more

powerful hardware equipment will be available to us. It is not feasible to buy new hardware given the budget limitation of a phase I.

Figure 9 shows the impulse response and the frequency response of the FH modulated signal. In this case, the signal has been simultaneously hopped on two center frequencies.

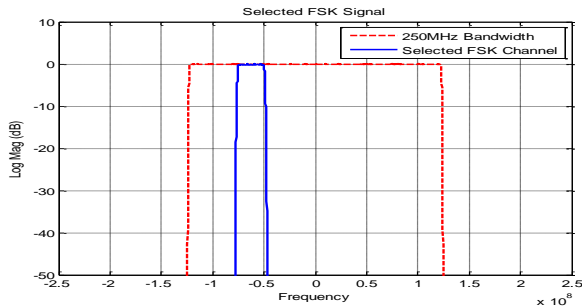


Figure 5: Selected FSK Band.

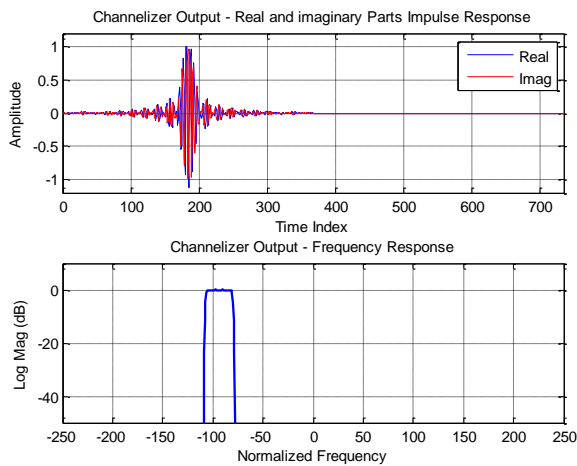


Figure 6: Modulated FSK Time Domain Signal (Upper Subplot) and Spectrum (Lower Subplot).

IV. Simulink Model

Figure 10 depicts the theoretical model of a non maximally decimated polyphase up converter channelizer. It is important to understand that this version of polyphase filter banks allow us a full usage of the sampled input bandwidth. Maximally decimated filter banks do not allow using the part of the spectrum that falls in the transition bandwidths of the aliased prototype filter. For a fully flexible software defined radio, which has no limits on the

bandwidth to be used, it is compulsory to adopt this modified version of the up converter channelizer.

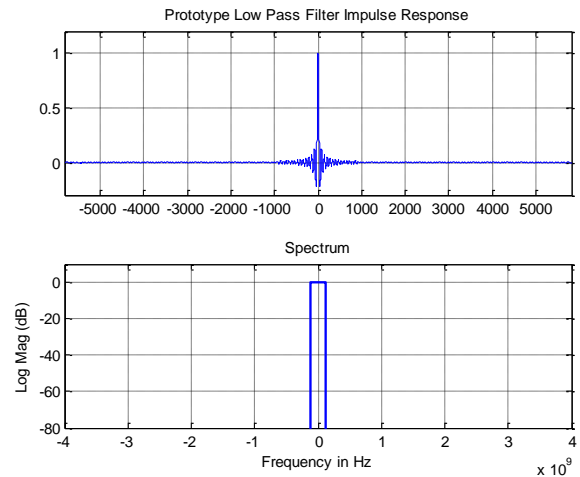


Figure 7: Impulse (In the Upper subplot) and Frequency (In the Lower Subplot) Responses of the Low Pass Prototype Filer for the FH Up-converter Channelizer.

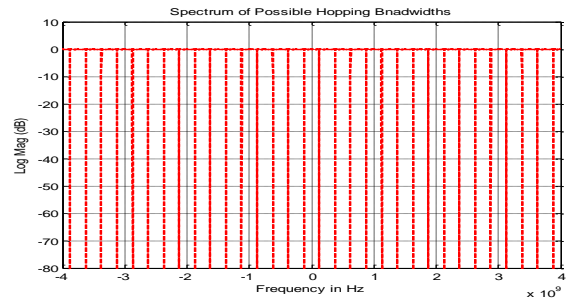


Figure 8: Frequency Plot of the Hopset.

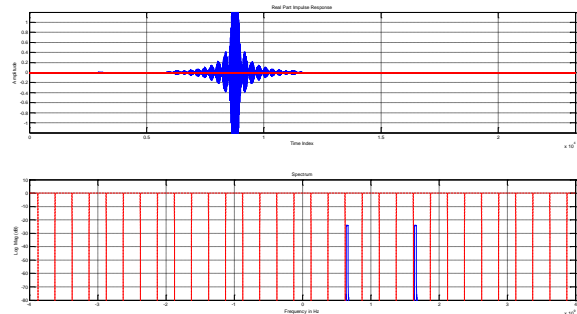


Figure 9: Impulse Response (Upper Subplot) and Frequency Response (Lower Subplot) of the Hopped Signal with Overlapped all the Possible Hopping Bands.

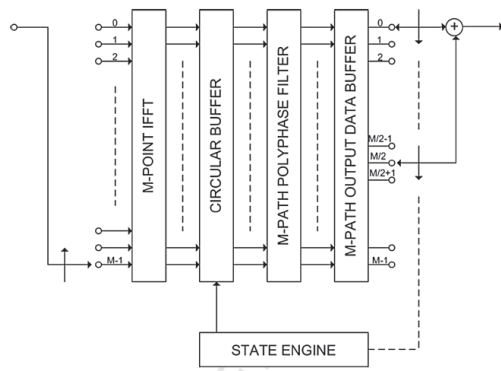


Figure 10: High Level Block diagram of Non Maximally Decimated Up Converted Channelizer.

Figure 11 depicts the high-level Simulink model of the digital up converter channelizer that is depicted in Figure 10. This is the main engine composing the dynamic, fully-digital FH modulator. It is easy to see that between the practical implementation and the theoretical model there are many differences. It is not trivial, even having deep theoretical knowledge, to implement a system digitally. Digital implementation has constraints that derive from the particular design application and from the selected hardware.

Simulink and the Xilinx System Generator (SysGen) for DSP are the fundamental tools to be used when implementing digital circuits on FPGAs. In fact, the SysGen can generate, directly from the Simulink model, Very high speed integrated circuits Hardware Description Language (VHDL) to be loaded onto a Xilinx digital board. If different FPGAs are used, which is often the case for space applications, the generated VHDL file can be manually modified as needed.

In Figure 11 we can see that two identical branches are present. In fact, the signal is made complex by the phase rotators and, when this happens, two identical paths need to be implemented one for the real component and the other for the imaginary one.

Notice that the IFFT, present in Figure 10, is absent in Figure 11. This is because we do not use all the channels simultaneously. FH systems only hop on one frequency at the time. We might decide to hop the signal on multiple center frequencies but it never happens that it is hopped on all the center

frequencies simultaneously. Thus it is much more efficient to implement the needed phase rotators rather than a whole IFFT.

Each branch in Figure 11 composes a 16-path up converter channelizer which performs a 1:8 up sampling of the input signal. Time scopes and spectral scopes are inserted in the model to make sure that desired results are obtained. Notice that a channel selector is present in the lower left corner of Figure 11. This allows us to hop onto different frequencies manually. Eventually, the cognitive engine developed by the University of Arizona will drive the channel selection according to the proactive sensing algorithm. Notice also that a DC canceller block has been included in the design. This block is absolutely needed. DC arises because of the fixed-point math and because of non-idealities in the hardware. It is a one tap filter that adaptively cancels the DC term from the input signal. It is our custom to insert such a block in our radios.

Figure 12 depicts the spectrum of the input signal. In this case, white noise is inserted in the system. The reason for that is that we want to show how well our filter bank works. Notice that our code has an option of having a real sinewave as input signal. This has been done for testing purposes. It is much easier to test a sinewave in the time domain because we know what the next samples have to look like.

Figure 13 depicts the output spectrum when channel zero is selected. To further demonstrate the capabilities of the up converter engine, the same signal is hopped, in Figures 14 and 15, on channels 5 and 15 respectively. For generating those plots, we have manually changed the input to the channel selector in Simulink while the simulation was running. We imagine that this selector will be driven by a cognitive algorithm whose output will be an input variable to the selected hardware.

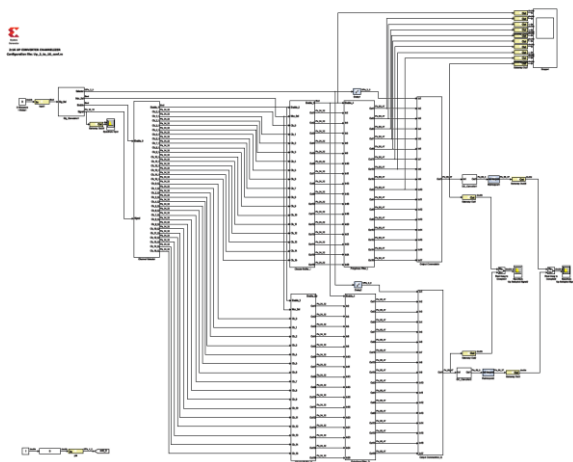


Figure 11: High Level Digital Design of a Complex (Real & Imaginary) Up Converter Channelizer.

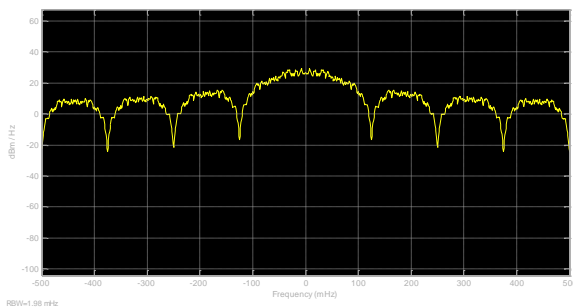


Figure 12: Spectrum of the Input Signal.

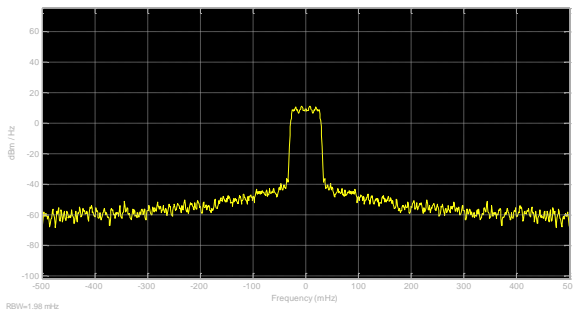


Figure 13: Spectrum of the Output Signal when Hopped onto DC.

Figure 16 depicts the time scope, which is in Figure 11. Unfortunately this plot is not very readable given the density of samples thus a zoomed version of it is presented in Figure 17 where on the lower subplot can be recognized the upsampled sinewave. While for the spectrum we used white noise in order to test the hopping capability, for the time scope it is better to use sinewaves so that the samples can be tracked individually. By looking at

those plots we are fully confident about the correct functioning of the proposed architecture.

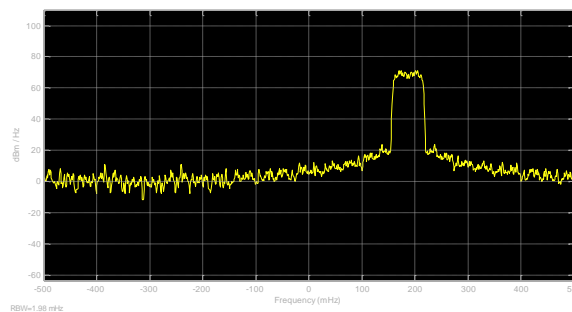


Figure 14: Spectrum of the Output Signal when Hopped on Frequency 5.

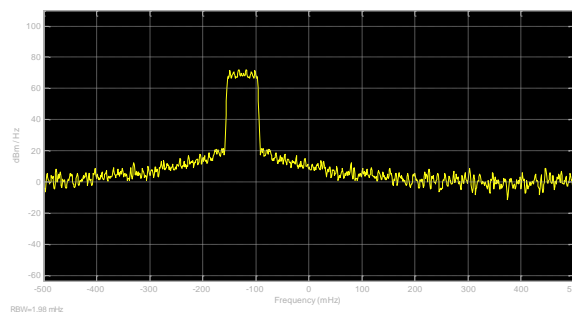


Figure 15: Spectrum of the Output Signal when Hopped on Frequency 15.

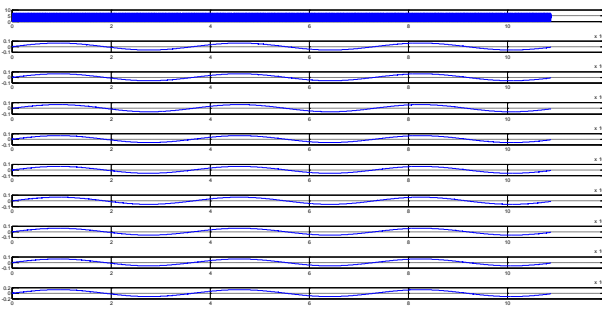


Figure 16: Time Scope Depicted in Figure 11.

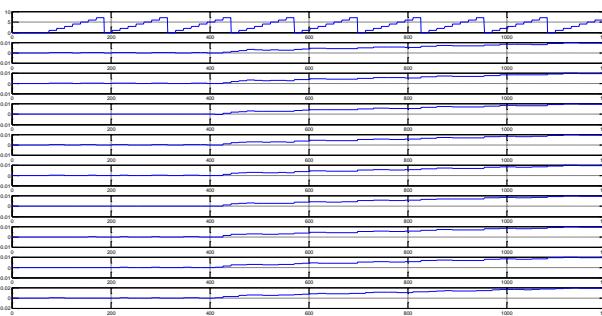


Figure 17: Zoom In of Time Scope Depicted in Figure 16.

up_2_to_16_cw Project Status (01/04/2016 - 12:14:36)

Project File:	up_2_to_16_cw.xise	Parser Errors:	No Errors
Module Name:	up_2_to_16_cw	Implementation State:	Programming File Generated
Target Device:	xc7vx980t-1ffg1928	• Errors:	
Product Version:	ISE 14.3	• Warnings:	
Design Goal:	Balanced	• Routing Results:	All Signals Completely Routed
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	All Constraints Met
Environment:	System Settings	• Final Timing Score:	0 (Timing Report)

Device Utilization Summary



Slice Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Registers	76,220	1,224,000	6%	
Number used as Flip Flops	76,220			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	55,629	612,000	9%	
Number used as logic	38,674	612,000	6%	
Number using O6 output only	37,973			
Number using O5 output only	14			
Number using O5 and O6	687			
Number used as ROM	0			
Number used as Memory	15,781	221,400	7%	
Number used as Dual Port RAM	0			
Number used as Single Port RAM	0			
Number used as Shift Register	15,781			
Number using O6 output only	3,749			
Number using O5 output only	0			
Number using O5 and O6	12,032			
Number used exclusively as route-thrus	1,174			
Number with same-slice register load	1,173			
Number with same-slice carry load	1			
Number with other load	0			
Number of occupied Slices	17,453	153,000	11%	
Number of LUT Flip Flop pairs used	63,656			
Number with an unused Flip Flop	2,944	63,656	4%	
Number with an unused LUT	8,027	63,656	12%	
Number of fully used LUT-FF pairs	52,685	63,656	82%	
Number of unique control sets	6			
Number of slice register sites lost to control set restrictions	15	1,224,000	1%	
Number of bonded IOBs	118	480	24%	
Number of RAMB36E1/FIFO36E1s	60	1,500	4%	
Number using RAMB36E1 only	60			
Number using FIFO36E1 only	0			

Number of RAMB18E1/FIFO18E1s	0	3,000	0%
Number of BUFG/BUFGCTRLs	1	32	3%
Number used as BUFGs	1		
Number used as BUFGCTRLs	0		
Number of IDELAYE2/IDELAYE2_FINEDELAYS	0	900	0%
Number of ILOGICE2/ILOGICE3/ISERDESE2s	0	900	0%
Number of ODELAYE2/ODELAYE2_FINEDELAYS	0	900	0%
Number of OLOGICE2/OLOGICE3/OSERDESE2s	0	900	0%
Number of PHASER_IN/PHASER_IN_PHYS	0	72	0%
Number of PHASER_OUT/PHASER_OUT_PHYS	0	72	0%
Number of BSCANs	0	4	0%
Number of BUFHCEs	0	216	0%
Number of BUFRs	0	72	0%
Number of CAPTUREs	0	1	0%
Number of DNA_PORTS	0	1	0%
Number of DSP48E1s	1,504	3,600	41%
Number of EFUSE_USRs	0	1	0%
Number of FRAME_ECCs	0	1	0%
Number of GTHE2_CHANNELS	0	72	0%
Number of GTHE2_COMMONS	0	18	0%
Number of IBUFDS_GTE2s	0	36	0%
Number of ICAPs	0	2	0%
Number of IDELAYCTRLs	0	18	0%
Number of IN_FIFOs	0	72	0%
Number of MMCME2_ADVs	0	18	0%
Number of OUT_FIFOs	0	72	0%
Number of PCIE_3_0s	0	3	0%
Number of PHASER_REFS	0	18	0%
Number of PHY_CONTROLS	0	18	0%
Number of PLLE2_ADVs	0	18	0%
Number of STARTUPs	0	1	0%
Number of XADCs	0	1	0%
Average Fanout of Non-Clock Nets	1.55		

Performance Summary [-]			
Final Timing Score:	0 (Setup: 0, Hold: 0, Component Switching Limit: 0)	Pinout Data:	Pinout Report
Routing Results:	All Signals Completely Routed	Clock Data:	Clock Report
Timing Constraints:	All Constraints Met		

Failing Constraints [-]
All Constraints Were Met

Clock Report [-]

Clock Net	Resource	Locked	Fanout	Net Skew(ns)	Max Delay(ns)
clk_BUFPG	BUFGCTRL_X0Y31	No	18367	1.170	2.530

Detailed Reports [-]

Report Name	Status	Generated	Errors	Warnings	Infos
Synthesis Report	Current	Mon Jan 4 09:33:10 2016	0	24796 Warnings (0 new)	47 Infos (0 new)
Translation Report	Current	Mon Jan 4 09:43:14 2016	0	2173 Warnings (0 new)	0
Map Report	Current	Mon Jan 4 10:20:36 2016			
Place and Route Report	Current	Mon Jan 4 10:44:30 2016	0	0	0
Power Report					
Post-PAR Static Timing Report	Current	Mon Jan 4 10:49:56 2016	0	0	3 Infos (0 new)
Bitgen Report	Current	Mon Jan 4 12:14:02 2016	0	119 Warnings (119 new)	1 Info (0 new)

Secondary Reports [-]

Report Name	Status	Generated
Physical Synthesis Report	Out of Date	Mon Jan 4 10:20:12 2016
WebTalk Report	Current	Mon Jan 4 12:14:14 2016
WebTalk Log File	Current	Mon Jan 4 12:14:26 2016

Date Generated: 01/04/2016 - 12:50:51

Figure 48: Design Summary of NMDFB in FPGA.

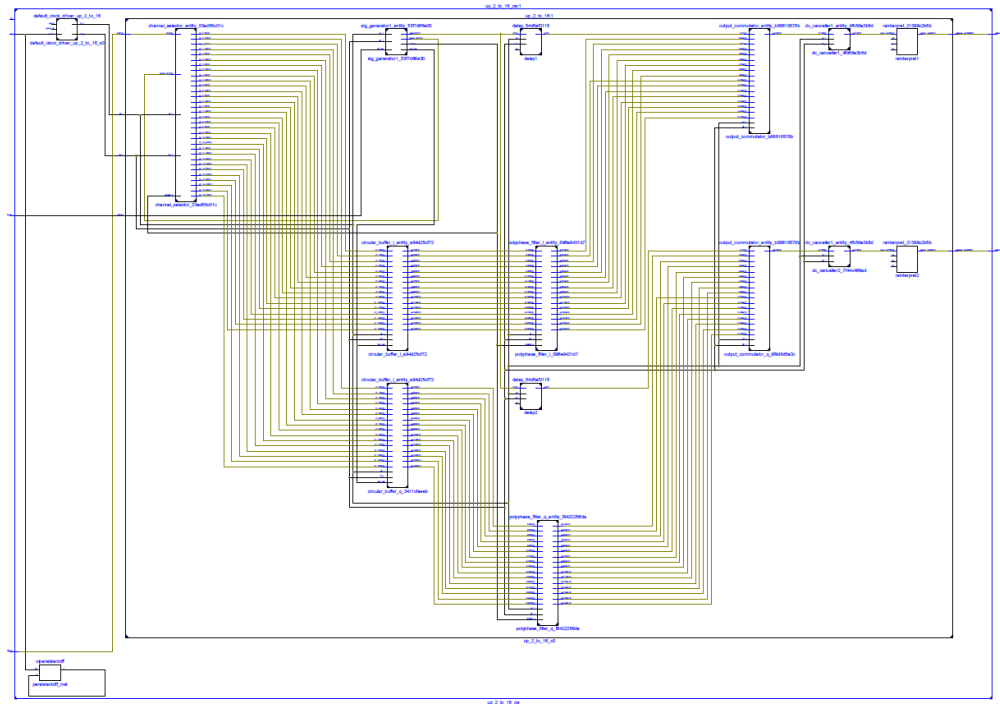


Figure 19: RTL Schematic of NMDFB with Phase Rotators.

V. Preliminary FPGA Results

Figure 18 shows the device utilization summary as it has been generated by the Xilinx System Generator. The way in which we can achieve the hardware resources estimation is by inserting the appropriate device information in the Xilinx software. The software analyzes the design, generates the VHDL code and analyzes the resources. Notice that the resources utilized for implementing a 16-path upconverter channelizer are very limited compared to what is available. FPGAs today are very capable. The RTL diagram of the 16-path NMDFB is shown in Figure 19. Even this diagram has been generated by the Xilinx software. It is interesting to notice the differences between Figures 7, 11 and 18. They all represent different steps in the NMDFB implementation process.

VI. Conclusions

In this paper we have shown preliminary design efforts for the implementation of a FH radio designed with NMDFBs. NMDFB allows us to implement the whole system design in the digital domain. Current FH radios are still implemented in the analog domain. The digital implementation, of course, carries advantages in terms of performance and SWAP-C compared to the analog counterpart. This makes them very much suitable for UAV radios where size, power consumption and spectrum awareness are very important parameters.

Frequency Hopping radios are highly jamming resistant and also, given their “multi-channel, multi-frequency” nature, are highly suited for highly crowded UAVs networks.

VII. References

[1] E. Venosa, f. harris and F. Palmieri, On Software Radio: Sampling Rate Selection, Design and Synchronization, Springer Science + Business Media, LLC 233 Spring Street, New York, NY 10013, USA.
[2] E. Venosa, X. Chen, f. harris, “Polyphase Analysis Filter Bank Down-Converts Unequal Channel Bandwidths with Arbitrary Center Frequencies-Design II”, SDR 2010, Washington DC, December 2010.
[3] Harris, F., Dick, C., Chen, X., & Venosa, E. “M-path channelizer with arbitrary center frequency

assignments,” WPMC 2010, Recife, Brazil, October 11–14, 2010.

[4] fred harris, Elettra Venosa, Xiaofei Chen, “Multirate Signal Processing for Software Radio Architecture,” Chapter 7 on Signal Processing Theory and Machine Learning, Academic press Library in Signal Processing, 2014

[5] Xiaofei Chen, fred harris, Chris Dick and Elettra Venosa, “A New Channelizer-Based Up Converter Architecture for Transmit Downlink of Combined 3GPP LTE and WCDMA Radios,” Proc. of SDR’13, January 8 –10, 2013, Washington DC.

[6] Elettra Venosa, fred harris, Chris Dick and Xiaofei Chen, “A New Channelizer-Based Down Converter Architecture for Receive Uplink of Combined 3GPP LTE and WCDMA Radios,” Proc. of to SDR’13, January 8 –10, 2013, Washington DC.

[7] fred harris, Xiaofei Chen, Elettra Venosa, “Polyphase Channelizers for Fully Digital Frequency Hopping Systems,” invited paper on Analog Integrated Circuits and Signal Processing, Springer Journal of.

[8] f. harris, X. Chen, E. Venosa, “An Efficient Full Digital Frequency Hopping Demodulator Based on Polyphase Filter Banks”, outstanding paper award , Proc. of SDR’11, November 30 – December 3, 2011, Washington DC.

[9] f. harris, E. Venosa, X. Chen, “Polyphase Up Converter Channelizer for Fully Digital Frequency Hopping Modem”, Proc. of SDR’11, November 30 – December 3, 2011, Washington DC.

[10] fred harris, Elettra Venosa, Xiaofei Chen, “Polyphase Up Converter Channelizers enable Fully Digital Multi-Carrier Frequency Hopping Modulators,” Proc. of Milcom 2015, October 26-28, Tampa, FL.

[11] fred harris, Xiaofei Chen, Elettra Venosa, “Efficient Implementation of Multicarrier Frequency Hopping Receiver via Polyphase Channelizer,” Proc. of Milcom 2015, October 26-28, Tampa, FL.