

FPGA Based FRM GDFT Filter Banks

Fangzhou Wu, Rudi Villing
Department of Electronic Engineering
National University of Ireland Maynooth
Maynooth, Ireland
fwu@eeng.nuim.ie, rudi.villing@nuim.ie

Abstract—Efficient channelization in flexible, reconfigurable communications systems is an ongoing challenge. Previous work in our lab has shown that designs based on the GDFT-FB (Generalized DFT modulated Filter Bank) combined with FRM (Frequency-Response Masking) can reduce prototype filter complexity and improve hardware efficiency, permitting larger scale designs. In this work we examine the design and implementation of the full FRM GDFT-FB and narrowband FRM GDFT-FB on an FPGA and describe solutions to various design issues encountered. We evaluate the DSP performance and the FPGA resource usage using a concrete channelization problem based on TETRA 25 kHz channels.

Keywords—GDFT-FB, filter-bank, FRM, FPGA, polyphase

I. INTRODUCTION

Multi-rate signal processing and polyphase modulated filter banks can be used to efficiently implement the wideband channelization function of a base station receiver [1]. In previous work we have designed a number of filter banks [2] based on further development of these concepts. More recently, we have developed FPGA implementations based on the basic DFT and Generalized DFT (GDFT) filter bank (FB) designs and conducted a basic evaluation [3].

However, basic DFT-FB and GDFT-FB designs can require very large prototype filter lengths when many narrowband channels must be extracted [4]. These can be difficult to design and implement in practice. Potential solutions to this problem include multi-stage filtering [5] and a number of Frequency Response Masking (FRM) based designs [6, 7]. The FRM solutions in particular replace a single long prototype filter in a GDFT-FB with a cascade of interpolated filters and masking filters, each of which is shorter than the original filter and appear to provide comparable or better computation loading than the basic GDFT-FB designs. For this reason, this paper focuses on the development and evaluation of FPGA based implementations of these designs. The objective is to determine if theoretical benefits can be realized in practice.

The outline of the remainder of this paper is as follows: Section II provides a brief introduction to FRM concepts (particularly as they relate to filter banks) and Palomo Navarro's full FRM and narrowband FRM based GDFT-FBs. Section III examines the FPGA implementation of the

Full FRM GDFT-FB, while section IV focuses on the corresponding narrowband FRM implementations. Section V evaluates the DSP performance and hardware usage associated with these implementations using a concrete channelization specification. The conclusions are presented in section VI.

II. BACKGROUND

A polyphase filter bank implements the equivalent of multiple parallel band-pass operations using a single low-pass *prototype* filter and an efficient modulation operation that can filter multiple sub-bands at once. In the context of channelization these sub-bands are the desired narrow-band output channels [8]. The prototype filter for a K band DFT-FB, $H(z)$, is divided into K poly-phase components. The K sub-band filters are obtained by complex modulation of the prototype filter using the DFT algorithm [9] as:

$$H_k(z) = \sum_{p=0}^{K-1} E_p(z^K) z^{-p} W_K^{-kp}, \quad p, k = 0, \dots, K-1 \quad (1)$$

However, for polyphase filter-bank, where a large number of sub-bands are required, a very long prototype filter with a large number of coefficients will typically be needed. For example, [6] found that a 256-channel DFT-FB required a prototype filter with 8085 coefficients to meet the TETRA Voice and Data 25 kHz standard. FRM techniques can be used to overcome this problem.

The FRM technique for FIR filter design is used to replace a single high order filter with a cascade of lower order filters [10]. The basic operation is shown in Figure 1. A relaxed specification base and complementary filter are designed. These are then interpolated to achieve the required transition band characteristics. Finally relaxed specification masking filters are designed which select a subset of the interpolated base and complementary filter response. These masked outputs are added together to form a composite response which matches the original (single high order) filter specifications.

The transfer function of the FRM parallel structure is given by:

$$H(z) = H_a(z^L)H_{Ma}(z) + H_c(z^L)H_{Mc}(z) \quad (2)$$

where $H_a(z)$ is the base filter, and $H_c(z)$ is the complementary filter, and $H_{Ma}(z)$, $H_{Mc}(z)$ are the masking filters for interpolated base filter and complementary filter respectively.

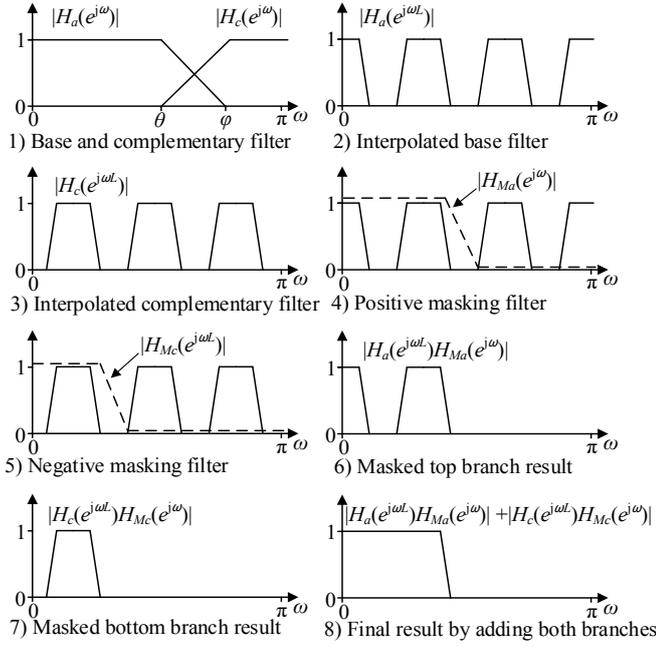


Figure 1 The process of two branches filtering based on FRM

A particularly efficient realization can be achieved by using a *subclass I filter* in which the base filter transition band includes the normalized frequency $\pi/2$ [11] and the complementary filter $H_c(z)$ is related to the base filter $H_a(z)$ by:

$$H_c(z) = H_a(-z) \quad (3)$$

The polyphase decomposition of $H_a(z)$ is:

$$H_a(z) = H_{a0}(z^2) + z^{-1}H_{a1}(z^2) \quad (4)$$

Therefore $H_c(z)$ can be expressed as:

$$H_c(z) = H_{a0}(z^2) - z^{-1}H_{a1}(z^2) \quad (5)$$

When (5) is substituted into (2), the efficient full FRM structure is obtained and may be implemented as illustrated in Figure 2 and $H_c(z)$ does not need to be explicitly designed.

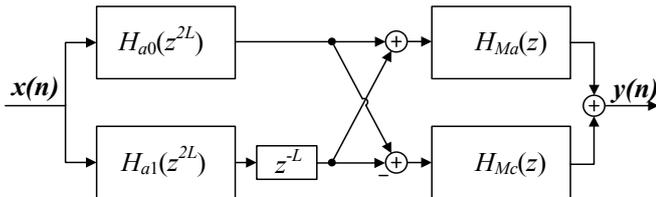


Figure 2 The Subclass I filter polyphase full FRM design

An even more efficient variant, called narrowband FRM, can be obtained if the lower (negative) branch in Figure 2 is discarded entirely, thus reducing the design to an

interpolated based filter followed by a single masking filter. In this case the masking filter selects only the DC centered image of the interpolated base filter (see Figure 1, steps 2 and 4) and the final transition band characteristics are determined by the interpolated base filter only.

In [6, 7] the FRM technique was used to reduce the prototype filter length of the GDFT-FB. The prototype filter $H(z)$ of the GDFT-FB is first expressed in FRM form as:

$$H(z) = H_{a0}(z^{2L})H_{Ma}(z) + z^{-L}H_{a1}(z^{2L})H_{Ma}(z) + H_{a0}(z^{2L})H_{Mc}(z) - z^{-L}H_{a1}(z^{2L})H_{Mc}(z) \quad (6)$$

To simplify this equation, we define

$$\begin{aligned} A(z) &= H_{Ma}(z) + H_{Mc}(z) \\ B(z) &= H_{Ma}(z) - H_{Mc}(z) \end{aligned} \quad (7)$$

Then substituting $A(z)$ and $B(z)$ into (6) yields

$$H(z) = H_{a0}(z^{2L})A(z) + z^{-L}H_{a1}(z^{2L})B(z) \quad (8)$$

Applying complex modulation to obtain the sub-band filters $H_k(z)$ we get:

$$H_k(z) = H_{a0}(z^{2L})A_k(z) + z^{-L}H_{a1}(z^{2L})B_k(z) \quad (9)$$

In which, $A_k(z) = A(zW_K^k)$ and $B_k(z) = B(zW_K^k)$. Applying the polyphase decomposition to both $A_k(z)$ and $B_k(z)$ yields

$$\begin{aligned} A(z) &= \sum_{i=0}^{K-1} z^{-i} E_{Ai}(z^K) \\ B(z) &= \sum_{i=0}^{K-1} z^{-i} E_{Bi}(z^K) \end{aligned} \quad (10)$$

At last, the bandpass filter in each sub-band of the FRM GDFT-FB can be expressed as:

$$\begin{aligned} H_k(z) &= H_{a0}(z^{2L}) \sum_{i=0}^{K-1} z^{-i} W_K^{-ki} E'_{Ai}(z^K) \\ &+ (-1)^k z^{-L} H_{a1}(z^{2L}) \sum_{i=0}^{K-1} z^{-i} W_K^{-ki} E'_{Bi}(z^K) \end{aligned} \quad (11)$$

where

$$\begin{aligned} E'_{Ai}(z^K) &= E_{Ai}(z^K W_K^{-k_0 K}) \\ E'_{Bi}(z^K) &= E_{Bi}(z^K W_K^{-k_0 K}) \end{aligned} \quad (12)$$

Figure 3 shows the full FRM GDFT-FB design. For odd-stacked channel allocation (i.e. channel centers offset by π/K relative to even-stacked) k_0 is $1/2$. For even stacked channels (i.e. the typical DFT-FB channel centers) k_0 is 0 and some elements of the structure simplify as a result. In the odd stacked configuration, it is also necessary to make the base filter odd by shifting its frequency response from DC to $\pi/2$. Note that the output phase shifts $e^{-j\pi}$ in Figure 3 correspond to $(-1)^k$ in (11).

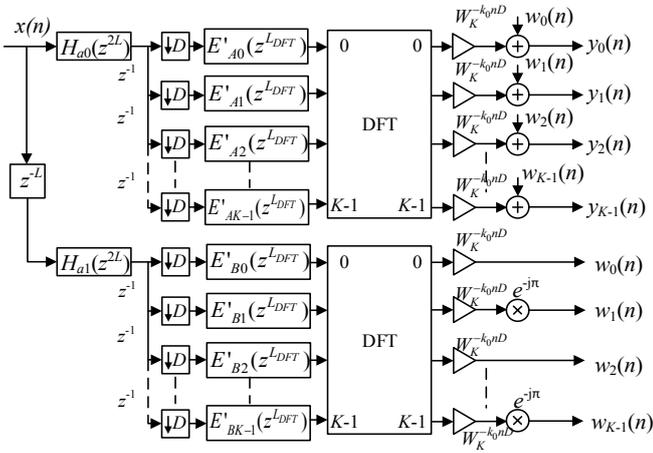


Figure 3 Full FRM GDFT-FB (odd stacked, $k_0 = 1/2$, even stacked, $k_0 = 0$)

In general, the GDFT-FB provides a large computational saving compared to direct per-channel filtering using the polyphase decomposition and noble identity [9]. Even so, further savings are possible.

In the alternative FRM configuration, the base filter can be moved to the output side of the GDFT-FB in order to operate at the lower sub-band output sample rate [6]. Consequently, the FRM interpolation factor L will be applied in a decimated form and this reduces the zero padding and sample delay required. In this configuration, the base filter coefficients will always be real-valued (regardless of channel stacking), the base filter no longer requires polyphase decomposition and therefore symmetric filters can be used to reduce the multiplications required.

The most efficient form is the alternative narrowband FRM GDFT-FB in which only the positive branch (the upper branch in Figure 2) is required. This form must use an oversampled configuration however [6], and the resulting design is shown in Figure 4. Every sub-band filter is given by:

$$H_k(z) = \sum_{i=0}^{K-1} z^{-i} W_K^{-k_0 i D} W_K^{-k i} E'_{Mai}(z^K) H_a(z^{L/D}) \quad (13)$$

where

$$E'_{Mai}(z^K) = E_{Mai}(z^K W_K^{-k_0 K}) \quad (14)$$

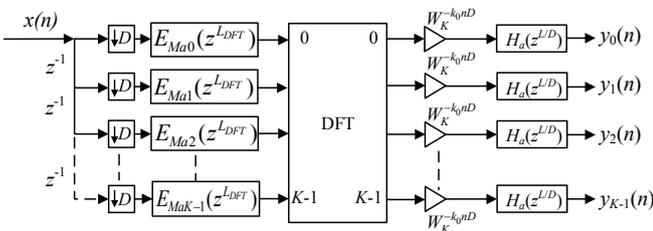


Figure 4 Efficient alternative narrowband FRM GDFT-FB (odd stacked, $k_0 = 1/2$, even stacked, $k_0 = 0$).

III. FPGA BASED FULL FRM GDFT-FB

The full FRM GDFT-FB was developed for the Xilinx FPGA family using the ISE (Integrated Software Environment) tool suite and the library of reusable IP cores.

Odd and even stacked designs were implemented and are treated separately because the modulations required for the odd stacked design result in complex valued filter coefficients. In addition, for the odd stacked channel allocation, the parameter k_0 in equation (12) is $1/2$ rather than 0, and base filter's response needs to be shifted from DC to $\pi/2$. Thus the base filters H_{a0} , H_{a1} and masking filters $A(z)$ and $B(z)$ are have complex values in odd-stacked configuration and real values in even-stacked configuration. The modifications needed to implement the FPGA channelizer for an odd stacked configuration is detailed described in our previous work [3], including a complex FIR implementation using 'cross-coupled' FIR compiler blocks and a frequency state-machine with the value of $W_K^{-k_0 i D}$.

A. Even stacked design

1) The high level FPGA design

The high level FPGA design is shown in Figure 5 and follows the general structure of the theoretical design in Figure 3. Each of the polyphase decomposed masking filters is implemented by a single FIR compiler IP core. The two polyphase components of the interpolated base filter are each implemented by a separate FIR compiler because the large delay (L) between branches makes a single FIR compiler impractical. All FIR coefficients must be generated and quantized offline before insertion into FIR compilers. The final implementation incorporates a number of optimizations described in more detail below.

2) Base filter delay with an arbitrary fractional clock divider

The delay z^{-L} prior to the lower path of the full FRM design in Figure 3 subjects this path to an L sample period delay. If we just use a shift register to implement the delay triggered by system clock, the required shift register depth will be $L \times \text{clock_rate} / \text{sample_rate}$. This base filter interpolation factor typical ranges from 10 ~ 100. Thus, the shift register depth on the separate I and Q paths will range from 10's to 100's of thousands of elements which is very inefficient.

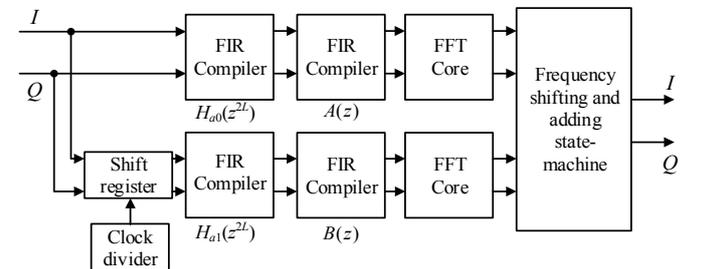


Figure 5 The FPGA based even stacked full FRM DFT-FB

The solution is to trigger the shift register at the input sample rate rather than the system clock rate by using a clock divider. Simple clock divisions, such as dividing by a power of 2 can be realized using a cascading d flip-flops structure [12], but this design is insufficient here. Instead we implement a fractional-n clock divider as described in [13]. After this change the shift register depth is reduced to L which saves resources.

3) Phase shifting and addition state machine

The phase shift by $e^{-j\pi}$ in every second sub-band in the H_{a1} path output shown in Figure 3, corresponds to negating every output sample on these sub-bands. The final filter bank output result is obtained by adding the H_{a1} path outputs to the H_{a0} path outputs. Remembering that the output of the FFT IP cores is a time division multiplexed serial stream, the addition of output sub-bands (with every second sub-band on the H_{a1} path negated) is efficiently implemented by state machine controlled addition. This reduces the number of hardware resources and processing delay required (in comparison to implementing the operations exactly as shown in Figure 3).

B. Odd stacked design

The implementation of the odd-stacked design contains many of the same elements as the even stacked design, including the more efficient delay line and phase shift and addition state machine described above. The complete design is shown in Figure 6 a).

For the odd-stacked design, k_0 is $\frac{1}{2}$ in (12). Thus the coefficients in the FIR compilers corresponding to $A(z)$ and $B(z)$ are complex values. In addition, the base filters H_a must be frequency shifted from DC to $\pi/2$ with the value of $e^{-j\pi/2}$ making these coefficients complex valued also. The FIR compiler IP core cannot handle complex coefficients so we must use the solution we have developed in [3], namely to use two cross-coupled FIR compiler IP cores to implement each complex valued filter, as shown in Figure 6 b). Additionally, the odd stacked modulations mean that a state machine is required to frequency shift the outputs of each DFT back to DC. Further details can be found in [3].

The complex modulation of all coefficients is applied offline at design time so that the modulated filter coefficients are supplied to the FIR compiler IP core.

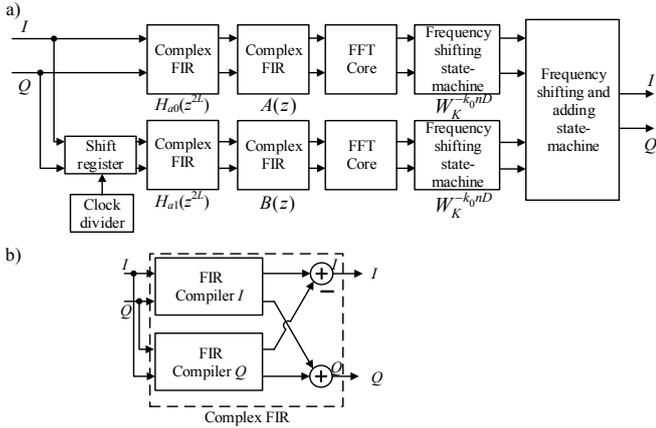


Figure 6 The FPGA based odd stacked full FRM GDFT-FB a) high level design. b) Inside of complex FIR.

IV. FPGA BASED ALTERNATIVE NARROWBAND FRM GDFT-FB

A. Even stacked design

Unlike the critically sampled ($K=D$) full FRM GDFT-FB, the alternative narrowband FRM is

oversampled ($L_{DFT} = K/D > 1$). For this reason the FPGA implementation shares elements of the oversampled DFT-FB design in [3]. In particular each oversampled polyphase FIR is implemented by two critically decimated FIR compiler cores with a selector state machine on the outputs to reconstruct the required oversampled input to the FFT core.

In the alternative narrowband FRM design, the base filter has been moved from the input side to the output side of the FFT on each sub-band. A naive implementation of this aspect of the theoretical design causes excessive resource use which is avoided using the solution described below. The overall design is shown in Figure 7.

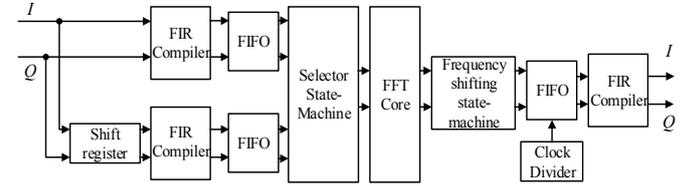


Figure 7 2x oversampled alternative narrowband FRM DFT-FB Efficient base filter implementation

The base filter in each output sub-band is implemented using an FIR compiler set to ‘single rate’ mode as multiple channels will be filtered using the same coefficients at the same sample rate. If the clock is fast enough, this IP core will process these samples serially by reusing DSP48 resources. Unfortunately, the FFT core outputs bursts of K samples at the system clock rate followed by inactivity while the next K samples are input and processed. This can prevent resource reuse by the FIR compiler so we introduce a FIFO to slow down the base filter input to the sub-band sample rate. This gives more clock cycles per sample and permits better resource reuse by the FIR compiler core.

B. Odd stacked design

The odd stacked alternative narrowband FRM design is based on the odd stacked oversampled GDFT-FB. The principal difference from the even stacked alternative narrowband FRM design is that the polyphase decomposed masking filter is both oversampled and complex valued and its implementation therefore requires 4 FIR compiler cores. The complete design is similar to the even stacked oversampled GDFT-FB, the difference is that the masking filters’ FIR compilers are replaced with the complex FIR shown in Figure 6 b).

V. EVALUATION AND RESULTS

To evaluate the FPGA based filter banks we use the filter specifications of the TETRA voice and data, 25 kHz channels: pass band ripple less than ± 2 dB and stop band attenuation at least 55 dB. Two configurations are used a 16 x 25 kHz and a 256 x 25 kHz filter bank.

All FPGA implementations use the Xilinx Virtex-6 ML605 and Xilinx ISE 14.3 design software. The FPGA filter-banks are programmed using Verilog HDL and use FIR compiler and FFT IP cores provided by Xilinx. All signals and filter bank internal values are quantized to 16-bit fixed point resolution. All filters are equi-ripple designs achieved using MATLAB’s FDATool. The pass band and stop band

frequencies used for all prototype filters were 9 kHz and 12.5 kHz.

There are three parts to the evaluation: DSP performance of the FRM designs in this paper compared to GDFT based designs (using the 16-channel filter bank), hardware resource usage of basic GDFT based designs compared to the FRM designs in this paper (again using the 16-channel filter bank), and finally resource usage and design practicality of scaled up designs for the most resource efficient designs (using the 256 –channel filter bank).

A. DSP performance of FRM based GDFT-FB designs

First we evaluated the DSP performance of the full FRM and alternative narrowband FRM GDFT-FBs in odd stacked configuration

With 16 channels, the sampling frequency F_s of wideband input signal was 400 kHz. In general the faster the system clock runs, the more DSP resources can be reused. In this test we used a 96 MHz clock rate.

As the alternative narrowband FRM approach to filter design is different to the full FRM approach two different prototype filter designs were created. The full FRM GDFT-FB was designed such that the order of the base filter H_a was 20, the order of the masking filters H_{ma} and H_{mc} was 145 each, and the FRM interpolation factor L was 24. In contrast, for the equivalent alternative narrowband FRM GDFT-FB design, the order of the base filter H_a was 41, the order of the masking filter H_{mc} was 41, and the FRM interpolation factor L was 8. The full FRM design was critically sampled whereas the alternative narrowband FRM design was oversampled by 2.

1) EVM result

The Error Vector Magnitude (EVM) of both designs was within specifications in the large signal to noise condition as shown in Table 1. The floating point columns indicate the results obtained for a Simulink implementation of the theoretical designs in Figure 3 and Figure 4 indicating that fixed point quantization did not have a significant effect.

Table 1 The EVM performance of both FPGA based designs: the 16-channel full FRM GDFT-FB and the alternative narrowband GDFT-FB

	Full FRM GDFT-FB		Alternative Narrowband FRMGDFT-FB		TETRA limits
	16-bit FPGA	Floating point	16-bit FPGA	Floating point	
Peak	0.1364	0.1197	0.1018	0.0807	0.3
RMS	0.0487	0.0464	0.0392	0.0354	0.1

2) Adjacent channel interference

Adjacent channel interference is caused by unwanted power from the signal in an adjacent channel intruding into the channel of interest. The ability of a system to reject adjacent channel interference ultimately affects its ability to deal with a mix of near (higher power) and far (lower power) transmitters.

The TETRA specification specifies the minimum requirement for the value of carrier to adjacent ratio is $C/I_a = -45$ dB. To test this, three adjacent channels were simulated: The middle channel was the channel of interest

and the two adjacent channels were interferers. The two interfering channels were set to the maximum amplitude at first while the channel of interest was attenuated and its EVM characteristics measured. Adjacent channel interference was evaluated at several levels of C/I_a , namely -40 dB, -45 dB and -50 dB.

Table 2 EVM results of the FPGA based full FRM GDFT-FB and alternative narrowband FRM GDFT-FB at different adjacent channel interference levels along with previously designed critically sampled GDFT-FB and oversampled GDFT-FB

C/I_a (dB)	Full FRM GDFT-FB		Narrowband FRM GDFT-FB		Critically sampled GDFT-FB		Oversampled GDFT-FB	
	Peak	RMS	Peak	RMS	Peak	RMS	Peak	RMS
-40	0.1858	0.0623	0.1374	0.0460	0.1539	0.0537	0.1374	0.0569
-45	0.2434	0.0829	0.1731	0.0578	0.2078	0.0749	0.1731	0.0756
-50	0.3326	0.1295	0.2396	0.0865	0.3229	0.1159	0.2396	0.1154

The detailed EVM results are presented in Table 2 along with the EVM results of critically and oversampled GDFT-FB. The data show that the expected results that EVM results degrade as the interference level increases. All the FPGA designs meet the TETRA specifications at -45 dB and the alternative narrowband FRM GDFT-FB still meets the TETRA limit at -50 dB adjacent channel level.

B. Hardware resource usage in 16-channel filter bank

The full FRM GDFT-FB uses multiple filters that each have fewer coefficients than the single filter design of the basic GDFT-FB. The alternative narrowband FRM design eliminates one branch of the full FRM design and moves the base filter to the output side of the DFT so that it operates at a lower sample rate, but the overall filter bank is 2x oversampled. The question is which filter bank designs require the fewest hardware resource and are thereby suitable for scaling to the largest number of channels.

Table 3 shows the hardware usage of multiple FPGA based filter banks. The data indicates that the alternative narrowband FRM GDFT-FB is the most hardware efficient FRM design and that the critically sampled filter banks appear to be most efficient overall. However it is possible that this result is biased by the small filter bank size evaluated. Therefore we next test with a larger filter bank.

Table 3 Hardware usage of all FPGA based 16-channel filter banks implemented to date

GDFT-FB Type	Register	LUTs	Block RAM 36	Block RAM 18	DSP48s
Critical samp. even	1223	838	0	8	5
Critical samp. odd	1435	1270	0	11	7
2x Even	1445	1503	0	7	8
2x Odd	2121	1570	0	12	15
full FRM even	2356	3379	0	6	33
full FRM odd	3098	3976	0	12	43
alt. narrowband even	1685	1814	2	0	9
alt. narrowband odd	2120	2529	0	5	16
Available	301440	150720	416	832	768

C. Hardware resource usage in 256 channel

The results to date showed that, for TETRA 25 KHz channels, the critically sampled DFT-FB and GDFT-FB performed as well as their oversampled equivalents, but with

less hardware resource usage. On the other hand, the alternative narrowband FRM GDFT-FB can achieved better adjacent channel interference resistance than the basic GDFT-FB and required far fewer filter coefficients. This latter feature is expected to make the prototype filters for channelizing large numbers of channels easier to design and implement. In addition, the alternative narrowband FRM did use not many more hardware resources than the critically sampled designs. For this reason, we evaluated the critically sampled GDFT-FB and alternative narrowband FRM GDFT-FB designs with a 256-channel filter bank.

The design and test setup is essentially the same as before except that the wide-band input signal contains 256 TETRA 25 kHz channels with and therefore the input sample rate is 6.4 MHz. While clock rate remains 96 MHz.

As the prototype filters are constructed in different ways for the different designs, the filter lengths differ. For the GDFT-FB, the prototype filter required 8704 coefficients (34 per channel). In contrast, the alternative narrowband FRM GDFT-FB requires only 768 coefficients for its masking filter (3 coefficients per channel). However there is also a base filter with 61 coefficients required in each channel, but it can take advantages of the symmetric FIR structure to save about half of the computational load. All filter coefficients in even stacked designs are real values whereas in odd stacked designs they are complex. This provides a greater benefit to the much shorter masking filter in the alternative narrowband FRM design. Moreover, even in the odd-stacked narrow band FRM GDFT-FB, the base filter coefficients are real-valued.

Table 4 Resource usage comparison of critically sampled GDFT-FB and alternative narrowband FRM GDFT-FB when configured for 256 channels

Filter-bank Type	Slice Register	Slice LUTs	Block RAM 36	Block RAM 18	DSP48s
Even stacked DFT-FB					
Critical sampled	1386	1594	19	0	12
Alt. Narrowband FRM	1765	2281	15	10	14
Odd stacked GDFT-FB					
Critically sampled	1859	2579	38	0	22
Alt. Narrowband FRM	2756	2989	17	12	20
Available	301440	150720	416	832	768

Table 4 shows the resource usage for the critically sampled GDFT-FB and alternative narrowband FRM GDFT-FB. The results suggest that the upper bound capacity of the Virtex-6 would be most constrained by the availability of DSP48 and Block RAM resources. For the even stacked channel allocation, the alternative narrowband FRM GDFT-FB will require slightly more resources: 2 more DSP48s are used compared to the critically sampled DFT-FB. However in the odd stacked channel allocation, the alternative narrowband FRM GDFT-FB has the advantage of requiring fewer complex operations. It uses 2 fewer DSP48s than the critically sampled GDFT-FB. In addition, it also saves more than 1/3 of the Block RAM, because Block RAM36 can be divided into 2 Block RAM18. This advantage could be expected to increase with larger numbers of channels and hence larger order prototype filters.

CONCLUSION

In this work, we have shown the implementation and evaluation of even and odd stacked full FRM GDFT-FB and alternative narrowband FRM GDFT-FB on an FPGA using reusable IP cores. The results indicate that the alternative narrowband FRM may have the best EVM performance and adjacent channel interference resistance. The critically sampled GDFT-FB and alternative narrowband FRM GDFT-FB have the best hardware resource efficiency. Among these two designs, the alternative narrowband FRM will use slightly more hardware than the critically sampled GDFT-FB in an even-stacked configuration, while in an odd-stacked configuration, the alternative narrowband FRM will use less hardware due to fewer complex operations. Therefore, if maximum spectrum utilization is required (suggesting the use of odd stacked channels), or adjacent channel interference is severe, then the alternative narrowband FRM GDFT-FB may be the preferred channelizer design.

REFERENCES

- [1] F. J. Harris, C. Dick, and M. Rice, "Digital receivers and transmitters using polyphase filter banks for wireless communications," *Microwave Theory and Techniques, IEEE Transactions on*, vol. 51, pp. 1395-1412, 2003.
- [2] A. Palomo, R. Villing, and R. Farrell, "Overlapped Polyphase DFT Modulated Filter Banks Applied to TETRA/TEDS SDR Base Station Channelization," presented at the RIA Colloquium on Communications and Radio Science 2010, Dublin, Ireland, 2010.
- [3] F. Wu, N. Palomo, and R. Villing, "FPGA realization of GDFT-FB based channelizers," in *Signals and Systems Conference (ISSC), 2015 26th Irish*, 2015, pp. 1-6.
- [4] A. P. Navarro, T. Keenan, R. Villing, and R. Farrell, "Non-uniform channelization methods for next generation SDR PMR base stations," in *Computers and Communications (ISCC), 2011 IEEE Symposium on*, 2011, pp. 620-625.
- [5] A. P. Navarro, R. Villing, and R. J. Farrell, "Practical Non-Uniform Channelization for Multistandard Base Stations," *ZTE Communications*, vol. 09, pp. 15-24, 2011.
- [6] A. Palomo - Navarro, R. J. Farrell, and R. Villing, "Combined FRM and GDFT filter bank designs for improved nonuniform DSA channelisation," *Wireless Communications and Mobile Computing*, 2015.
- [7] A. P. Navarro, "Channelization for Multi-Standard Software-Defined Radio Base Stations," Unpublished doctoral thesis, National University of Ireland Maynooth, 2011.
- [8] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*: Prentice-Hall, 1983.
- [9] P. P. Vaidyanathan, *Multirate systems and filter banks*: Prentice-Hall, Inc., 1993.
- [10] Y. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters," *IEEE Transactions on Circuits and Systems*, vol. 33, pp. 357 - 364, 1986.
- [11] H. Johansson, "New classes of frequency-response masking FIR filters," in *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, 2000, pp. 81-84 vol.3.
- [12] L. Gu, N. Zhou, and H. Li, "Research of Frequency Divider Based on Programmable Logic Device," *Procedia Environmental Sciences*, vol. 10, pp. 820-824, 2011.
- [13] S. W. Zhang and C. Zhao, "Design for realizing arbitrary fractional divider based on FPGA which duty cycle is up to 50%," in *Applied Mechanics and Materials*, 2013, pp. 1653-1657.