Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Area Efficient Implementation of Polyphase Channelizer for Multi-Standard Software Radio Receiver

Awan, Mehmood-Ur-Rehman; Alam, Muhammad Mahtab; Koch, Peter; Behjou, Nastaran

Published in:
Proceedings of the 5th Karlsruhe Workshop on Software Radios

Publication date:
2008

Document Version
Peer reviewed version

Citation for published version (APA):
Awan, M-U-R., Alam, M. M., Koch, P., & Behjou, N. (2008). Area Efficient Implementation of Polyphase Channelizer for Multi-Standard Software Radio Receiver. In Proceedings of the 5th Karlsruhe Workshop on Software Radios (pp. 123-130). Institut für Nachrichtentechnik, Universität Karlsruhe (TH).

Area Efficient Implementation of Polyphase Channelizer for Multi-Standard Software Radio

Mehmood-Ur-Rehman Awan, Muhammad Mahtab Alam, Peter Koch, Nastaran Behjou
Department of Electronic Systems, Technology Platform Section
Aalborg University Denmark
Email: (mura, mma, pk, nab)@es.aau.dk

Abstract*—Software Defined Radio architectures for multi-standard receiver i.e. UMTS and WLAN are proposed in this work. To extract the 12 and 3 channels of UMTS and WLAN respectively, we propose Polyphase techniques being used in the channelizer. This is a resource efficient way of implementing the multi-rate filters and further to extract the channels. The aim of this paper is to present the continuation of our research for optimizing and tunning the system design for the multi-standard software radio receiver and to illustrate an area efficient implementation of the Polyphase channelizer, where the target chip is a Virtex IV FPGA from Xilinx. The bandpass sampling technique is used at 630MHz to intentionally alias the combined band of WLAN and UMTS. The polyphase channelizer works at much lower rate which results in lowering the filter lengths. In the implementation, different structures for Polyphase channelizer are considered, such as standard structure, symmetric property based structure with shared adders and multipliers structure and serial Polyphase structure with serial and parallel Multiplier and Accumulator (MAC). The complexity analysis (in terms of hardware resources and operating frequency) of these structures is conducted. The Serial Polyphase structure with parallel MAC is selected since it requires fewer resources and also operates at the rate same as input. The FPGA implementation of the WLAN polyphase channelizer requires 11% of the slices and 14% of the embedded multipliers (DSP48s) and maximum operating frequency is 101.930MHz, which is above the desired operating frequency of 90MHz. UMTS channelizer requires 33% of the slices and 15% of the embedded multipliers (DSP48s). Its maximum operating frequency is 102.566MHz, which is above the desired operating frequency of 70MHz. Finally, the upsampler(UMTS) requires 11% and 8% of Slices and DSP48s respectively which can be operated up to 98.96MHz.

I. INTRODUCTION

A software-defined radio (SDR) system is a radio communication system which can tune to any frequency band and receive any modulation across a large frequency spectrum by means of a programmable hardware which is controlled by software. A Software radio is an enabling technology for future radio transceivers, allowing the realisation of multi-mode, multi-band, and reconfigurable base stations and terminals. However, considerable research efforts and breakthroughs in technology are required before the ideal software radio can be realised. An ideal software radio (ISR) samples the signal at Radio Frequency (RF), just after the antenna, whereas the realizable version of the software radio is the one that solve the problem of sampling the RF signal (according to minimum nyquits criteria, i.e. to sample at twice the maximum frequency of the incoming signal).

Recent developments and increasing trend toward a single device integrating several features and capabilities encourage the companies and research centers to develop the multi-standard multi-mode ”all-in-one” front-ends. High level of integration and small size are precedence objectives in these types of mobile applications. In order to acheive those objectives it is feasible to move most of the data processing to digital domain through shifting the analogue to digital converter (ADC) as close to antenna as possible [1]. This imposes more stringent performance requirements on the analog-to-digital (A/D) conversion, where a high dynamic range must be combined with a high sampling rate [2].

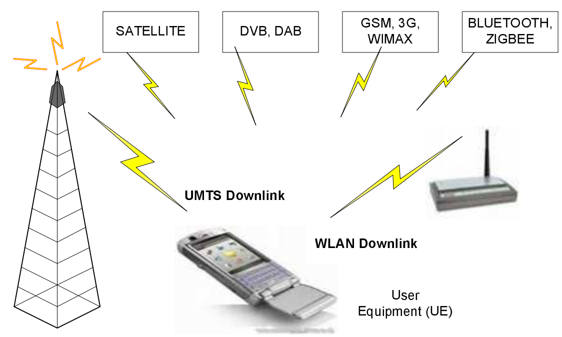

Fig. 1. A scenario of multi-standard multi-mode ”all-in-one” front-ends user equipment. It highlights the user equipment capbale of receiving two standards i.e.UMTS and WLAN.

This paper is a continuation of our research in the domain of Multi-Standard Software Radio Receiver (MSSRR). A scenario of multi-standard multi-mode is shown in Fig. 1. This scnario has been scale down to the UMTS and WLAN standards, [1] which actually fits to the mobile application. The RF spectral location for UMTS and WLAN standards are shown in Fig. 2. UMTS has a bandwidth of 60MHz for downlink having 12 channels and WLAN has 84.5MHz of bandwidth having 3 non-overlapped channels. It is required to downsample and to downconvert these channels to baseband.

A. Architecture for Software Radio

The selection of the hardware architecture is not easy for SDR based applications in an era where the number of transistors in an integrated circuit are increasing by a factor of

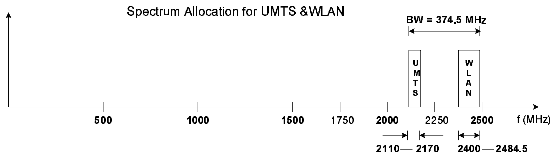

Fig. 2. Spectrum Allocation for UMTS and WLAN standards. UMTS has a bandwidth of 60MHz for downlink having 12 channels and WLAN has a 84.5MHz of bandwidth having 3 non-overlapped channels.

two every second year (Moores Law) as shown in the Fig. 3. As signal processing tasks (the algorithms) are getting more and more complex which is at the same time putting high requirements on the technologies platform with increasing demand of the MIPS (million of instructions per second). So the software solution for SDR makes it possible to make the transition from dedicated, single-purpose hardware (ASICs, etc.) to highly versatile general-purpose hardware such as FPGAs and DSPs, and even to general-purpose processors whose functionality is defined solely by their software configuration. This in turn paves the way for high-volume/low-cost production, making it financially viable to embed autonomous radio communication devices in a wide range of new kinds of devices and applications [3].

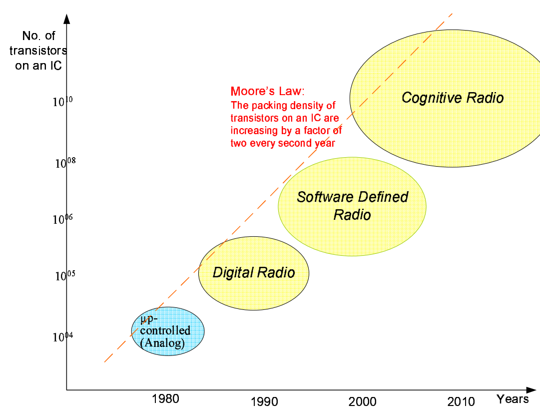

Fig. 3. Gorden E. Moore Law: The number of transsitors are increasing by a factor of 2 after every 18 to 24 months, due to increasing demand of applications. The complexity of the overall systems are increasing but with the demands of minimum cost, minimum size, faster execution time and least power dissipation.

Bandpass sampling and direct conversion are two receiver architectures that are suitable for software radios [4]. The sampling of bandpass signals can be carried out at rates lower than conventional lowpass Nyquist sampling, causing intentional aliasing the signal. Bandpass sampling can allow for received signals to be digitized closer to the antenna using manageable sampling rates and hence could be favourable for downconversion in software radios. The bandpass sampling architecture is shown in Fig. 4. According to bandpass sampling, the sampling frequency should be twice the signal bandwith rather than twice the maximum frequency component as in the case of Nyquist sampling. So the sampling frequency for the combined band of UMTS and WLAN must be atleast 749MHz to have non-overlap alaises. Todays technology set a limit to achieve such a high sampling rate. Significant improvement in ADC performance is required for sampling at RF.

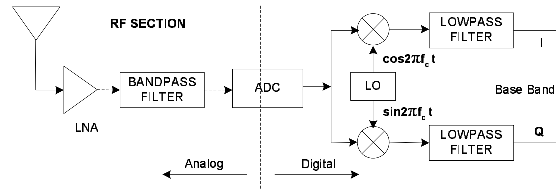

Fig. 4. Bandpass sampling archhitecture of Software Defined Radio

Direct conversion which is shown in Fig. 5, also sometimes called zero-IF, due to the lack of an intermediate frequency, converts the received RF signal direct to baseband. This is particularly attractive for the use in wireless systems, especially in handsets since direct conversion receivers lend themselves more easily to monolithic integration than heterodyne architectures, since the IF components are replaced by lowpass filters and baseband amplifiers. Direct conversion exhibits immunity to the problem of image since there is no IF [5]. There are a number of design issues associated with the direct conversion architecture. The most serious problem is DC offset in the baseband, following the mixer. This offset appears in the middle of the downconverted signal spectrum, and may be larger than the signal itself. This phenomenon can be caused by local oscillator leakage and self-mixing [5].

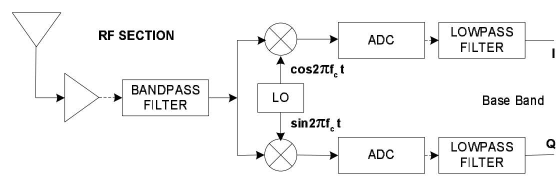

Fig. 5. Bandpass sampling archhitecture of Software Defined Radio

Bandpass sampling architecture does not require additional circuits for downconversion prior to quantization [4]. This leads to all the processing required for bandpass sampling architecture to be implemented on FPGA which can be reconfigured to different radio configuration. A software radio receiver architecture is presented in [1] which is shown in Fig. 6.

II. System Design

Polyphase channelizer are most efficient in term of computations and required hardware resources as compared to standard channelizer [4]. Based on the unique features of the polyphase channelizer, we have choosen it, to implement the System Design. The relation between the sampling frequency, channel spacing and number of channels for the polyphase channelizer is [6]:

$$f_s \quad = \quad N \times \Delta f \tag{1}$$

where f_s is the input sampling frequency, N is number of channels/transform size and Δf is the inter channel spacing.

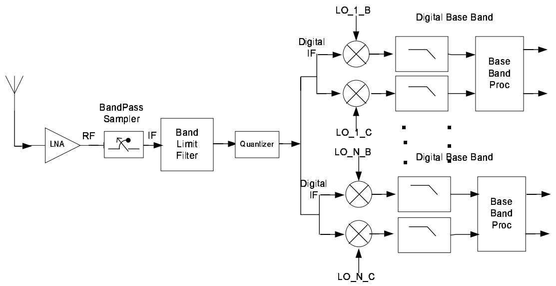

Fig. 6. The proposed architecture of the software radio, where sampling is done at RF just after the LNA which is the only analog component in this architecture.

There are two constraints that have to meet, one is that the (N) number of channels should be an integer, and second that the channels to be down-sampled and down-converted to baseband should be centered on to the multiples of the channel spacing or on to the multiple of quarter of their channel spacing respectively. The complete system have been designed with sampling frequency of 840MHz which is selected after examinig different sampling frequency, that fulfils the two contraints. This design is explained in [1], and the block level diagram is shown in Fig. 7.

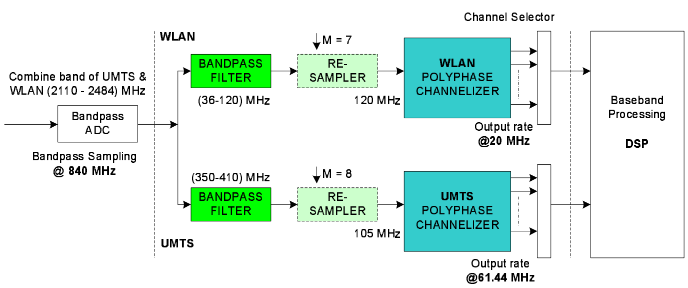

Fig. 7. System block diagram having re-samplers prior to UMTS and WLAN channelizers.

In this design, Polyphase channelizer is not used to its level best advantages of extracting all the channels at the same time. This is due to the fact the different standards have different channel bandwidth and inter-carrier spacing. Even for one of the standards, all of its sub-channels are not converted at the same time. This is due to the unequal channel spacing (offset from the DC). The polyphase channelizer can be used to its level best features that is extracting all of the channels for any standard, by having a heterodyning at the input of the polyphase channelizer, and heterodyning-carrier is selected such that the translated channels have equal channel spacing. This case will result in extracting all the channels of a standard, just by using standard polyphase channelizer, not by its variant to compensate the offsets of multiples of quarter of channel spacing which is explained in [1]. These modifications result in an optimized system design of MSSRR and is shown in Fig. 8.
The combined band of WLAN and UMTS is bandpass sampled at 630MHz and its aliases are overlapped but the individual bands of UMTS and WLAN are non-overlapped.

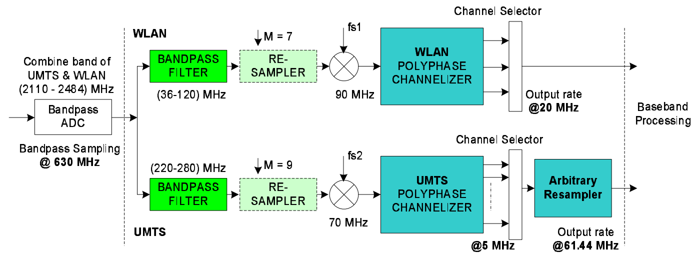

Fig. 8. System block diagram having spectrum translation prior to UMTS and WLAN channelizers. UMTS channelizer is followed by Arbitrary Resampler to achive the target rate of 61.44MHz.

WLAN band aliases to 36-120MHz and UMTS aliases to 220-280MHz in the Nyquist zone. WLAN band is spectrally inverted. Bandpass sampling alaises are shown in Fig 9.

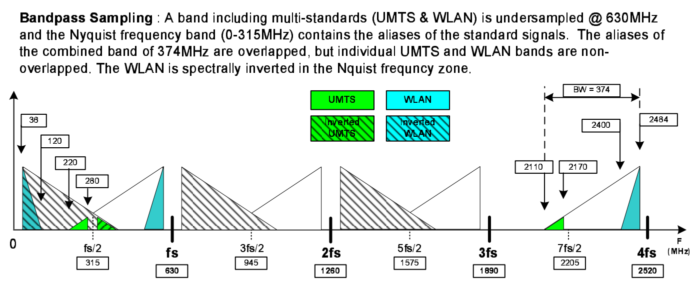

Fig. 9. The combined spectrum of UMTS and WLAN is bandpass sampled at 630MHz, and the resulted aliases in the Nyquist zone. WLAN alias is spectrally inverted.

At 630MHz, polyphase channelizer for WLAN has 21 channels of 30MHz and UMTS has 126 channels of 5MHz, but the required channels for WLAN and UMTS are only 3 and 12 respectively. This puts an extra load on the filtering process in terms of high clock speed requirement and large memory storage for filter coefficients. One of the techniques is to resample the data before the polyphase channelizer as shown in Fig. 8 in a similar way as it has been explained in [1]. The sampled signal can be resampled by large factors such that the resultant sampling frequency is above the total signal bandwidth, if the incoming signal is image free. The resampling process in this case is simply the spectrum translation [4]. Based on this technique, the WLAN and UMTS bandpass filters are made complex and the resultant image free signals for WLAN and UMTS are tried by different resampling factors to have the minimum possible sampling frequencies, So finally the resampling factor comes out to be 7 and 9 for WLAN and UMTS respectively and is summerized in Table I. This results in a new sampling frequency of 90MHz for WLAN, with 3 channels of 30MHz and it fits well to the non-overlapped channel criterion. In order to have desired WLAN rate of 20MHz, an embedded resampling factor of 9/2 is required. For the case of UMTS, with 9 being the resampling factor, it results in the new sampling frequency of 70MHz which will have 14 channels of 5 MHz band. In order to have target UMTS rate of 61.44MHz, UMTS channelizer is used as maximally decimated system to have the output of 5MHz, which is further upsampled to target rate of 61.44MHz by using an arbitrary resampler as shown in Fig. 8.

Cases	Sampling rate (MHz)	Channel Spacing (MHz)	No. of Channels
UMTS	70	5	14
WLAN	90	30	3

TABLE I
SPECIFICATIONS FOR THE CHANNELIZER FOR UMTS AND WLAN

After bandpass sampling, individual WLAN channels are centered at (48, 78 and 108)MHz, and translated to (-42, -12 and 18)MHz after downsampling by a factor of 7 which are further aligned having equal inter-channel spacing with respect to zero by digital down-conversion resulting the channels centered at (30, -30, 0)MHz at 90MHz of sampling frequency. The modified channelizer for WLAN is shown in Fig. 10 with reduced numbers of polyphase sub-filters.

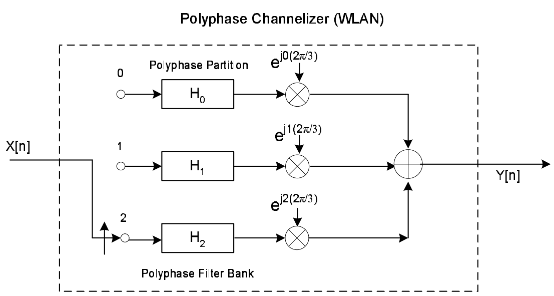

Fig. 10. WLAN channelizer.

For WLAN, with sampling frequency of 90MHz and channel spacing of 30MHz, the number of channels becomes 3 which is the number of the polyphase decomposition. To achieve the target rate of 20MHz, a downfactor 9/2 is required which is embedded in the polyphase arms. This is realized by upsampling the data with zero packing and then downsampling by serpentine shifting the data through the filter in stride of length 9. The process is illustrated for two data load iterations in Fig 11.

There is no actual zero packing in the final configuration. In the first data load, 5-actual data samples are delivered to the 3 register addresses, while in the second load 4-actual data samples are delivered to the 3 register addresses. The data loading procedure is found to be periodic in 2-load cycles for which it will require 2-states to control the process. (The least common multiple of 3 and 9 is 9, and since 9 zeropacked inputs are delivered at a time, results in 1 states. For upsampling factor of 2, the LCM of 1 and 2 becomes 2, which is the periodic interval). Table II lists the memory loading instructions for the process that anchors the data registers and cycles the data load and coefficient sets. Note that in the 2-states, a total of 9 inputs are delivered and 2 outputs are taken from the polyphase engine to realize the desired embedded 9/2 resampling. The loading scheme is seen to be a constant offset of -2 modulo 3 within a sequence as well as in the transition between sequences. The -2 offset is a consequence of the 1-to-2 up sampling represented by the zero packing but not actually implemented in the process.

State Machines for Register Load Sequence

State	No. of Inputs	Loading Sequence
0	5	R1, R0, R1, R2, R0
1	4	R1, R2, R0,, R1

TABLE II
POLYPHASE FILTER'S DATA LOADING SEQUENCE FOR THE WLAN WITH THE STATE MACHINE

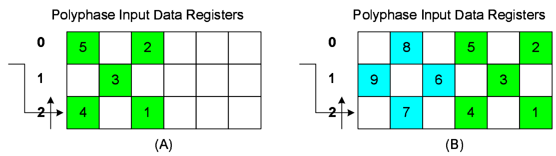

Fig. 11. Successive serpentine data shifts in polyphase memory and data load for 9:2 re-sampling in a 3-stage polyphase filter. It shows two data load operations.

Because of the 1-to-2 up sampling implemented by the zero packing, only one half of the weights in each stage actually contributes to the subfilter output. Thus each stage is further partitioned into 2 sub sets of weights, which results in a total of $3 \times 2 = 6$ filter weight sets. These sets are denoted by C0, C1,...., C5 where the integer is the starting index from the original non-partitioned prototype filter. Table III lists the filter assignment to the 3-successive data registers for 2-states of the process. It shows that in a given state the successive filter index increments by 4 modulo-6 and between states, the filter index increments by 9 modulo-6. The integer 4 is the offset between two data samples in the zero-packed load in two adjacent rows. The 9 index is the number of zero-packed data points introduced per data load cycle. The prototype filter has to be designed to operate at 2 times f_s or 180 MHz due to up sample the data by a factor of two on the way into the filter. Consequently, the filter becomes two times longer than the standard design but since only one-half of it is used per processing cycle so no processing penalty is paid.

State Machines for FILTER Co-efficients

State	Filter Co-efficients sets
0	C0, C4, C2
1	C3, C1, C5

TABLE III
FILTER CO-EFFICIENTS LOADING SEQUENCE WITH THE STATE MACHINE

In UMTS case, after bandpass sampling individual channels centered at (222.5 to 277.5)MHz are translated to (12.5 to 32.5 and -32.5 to -2.5)MHz by downsampling by a factor of 7 which are further aligned having equal inter-channel spacing with respect to zero by digital down conversion resulting the channels centered at (15,20,25,30,35,-30,-25,...-5,0)MHz at 70MHz of sampling frequency. The modified channelizer for UMTS is shown in Fig. 12 with reduced numbers of polyphase sub-filters.

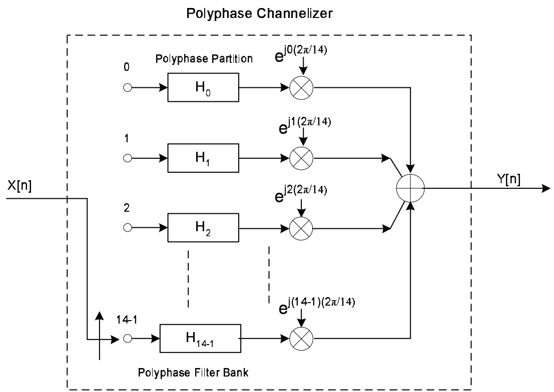

Fig. 12. UMTS channelizer

In UMTS channelization, with the sampling frequency of 70MHz and channel spacing of 5MHz, the number of channels become 14 which is the number of the polyphase decomposition. The exact target rate of 61.44MHz cannot be achieved by embedding the resampling in the polyphase arms, so the resampling process is achieved in two steps. First, the channels are downsampled to 5MHz by using the UMTS channelizer in maximally decimated mode and is achieved by the shifted the data through the filter in stride of length 14 same as polyphase partition. This process is illustrated for one data load iteration in Fig. 13 and the remaining data load will be similar. Next in the second step, 5MHz signal is upsampled to the target rate by using arbitrary interpolator.

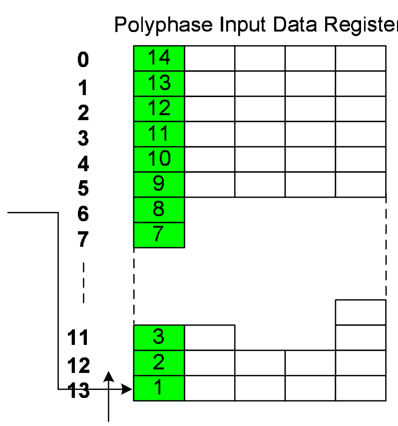

Fig. 13. Data load operations in a 14-stage polyphase filter. It shows just one data load operation and remaining load operation will be similar.

Here in polyphase channelizer acting as a maximally decimated filter, the data loading procedure is found to be periodic in 1-load cycles for which it will require 1-state to control the process. So register loading is very straight forward from (R13,...R0) as shown in Table IV.

State Machines for Register Load Sequence

State	No. of Inputs	Loading Sequence
0	14	R13, R12, R11, R10, R9, . . . R1, R0

TABLE IV
POLYPHASE FILTER'S DATA LOADING SEQUENCE FOR THE UMTS WITH THE STATE MACHINE

Arbitrary polyphase upsampler is an interpolator filter with upsampling factor of 16/1.302083. The 5MHz signal is upsampled to 80MHz and then downsampled by 1.302. The prototype filter is designed at upsampled frequency i.e. 80MHz and partitioned into 16 sub-filters. An efficient implementation structure of 1:M polyphase interpolator as described in [8] is shown in Fig. 14. As seperate filters all contain the same input data and differ by only their unique coefficient sets, so the M-path version of the polyphase filter can be replaced with a single stage filter with M-coefficient sets that are sequentially presented to the filter to compute successive outputs. In this case, coefficient sets are sequentailly incremented with step size of 1.302 modulo 16 to get the target sampling rate.

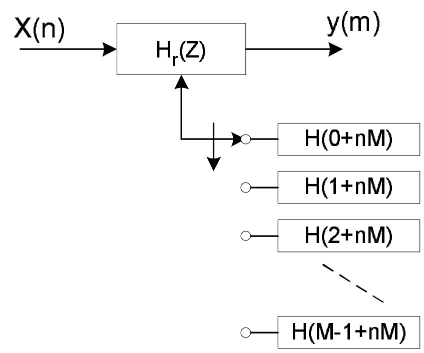

Fig. 14. Efficient Implementation Structure of 1:M Polyphase Interpolator

III. IMPLEMENTATION

In the implementation phase, polyphase channelizers are analyzed in terms of the required components, consisting of demultiplexer as commutator, a filter bank having polyphase filters, and finally the coherent phase summation. There are different structural techniques which can be used to carry out the implementation. To select the best technique for the designed receiver, general polyphase structure, optimized structures - symmetric property based structure, adder shared structure, serial polyphase structures with serial and parallel MAC are considered. Based on the complexity analysis as shown in Table V, serial polyphase structure with parallel MAC is selected for the final implementation, as shown in Fig. 15.

In the individual sub-filter implementation, different implementation structures are considered. These being Parallel Multipliers and Accumulate, Distributed Arithmetic, Fast FIR,

Cases	# Mults	#Adders	#Regs	Clock speed
Polyphase General (Transpose form)	N	((N/M)-1)M	N	f_s/M
Symmetric form (Shared Multipliers)	N/2	((N/M)-1)M	N	$2f_s$/M
Symmetric form (Shared Multipliers & Adders)	N/2	((N/M)M)/2	Nx2	$2f_s$/M
Serial Polyphase (Serial MAC)	1	1	N	f_s(N/M)
Serial Polyphase (Parallel MAC)	N/M	(M/N)-1	N	f_s

TABLE V
COMPLEXITY ANALYSIS FOR POLYPHASE FILTER BANK, IN TERMS OF MULTIPLIERS, ADDERS, REGISTERS, AND CLOCK REQUIREMENTS.

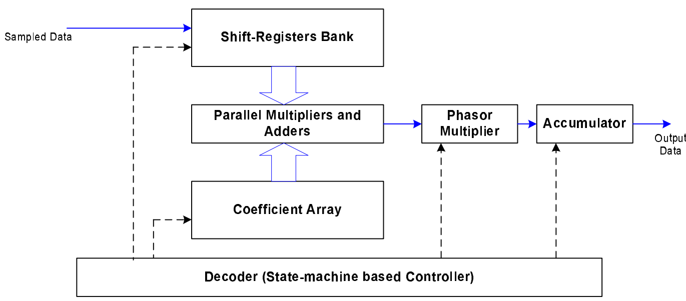

Fig. 15. The basic building blocks of Serial Polyphase Channelizer with Parallel MAC. It consists of Shift Register Bank, Filter's Coefficient Bank, Parallel Multiply and Accumulate, Phasor Multiplication, Accumulator and Decoder (state-machine based controller) to control the filtering operation.

Frequency domain filtering and Multiplier less filtering techniques. Each structure and its variants are analyzed in terms of hardware resources. The analysis is based on the approximation for the area requirements for multipliers, adders and registers etc [4]. The focus of the above techniques is to use multipliers as little as possible, to save the area. But due to technology advancement, the modern FPGAs have dedicated multiplier blocks which are more efficient than the CLB-slices based multipliers, mainly in terms of operating speed and reduced power requirements. Xilinx FPGA, Virtex-IV has XtremeDSP blocks that can perform multiplication up to 500MHz. The system performance is increased by using these blocks. Each XtremeDSP block has two DSP48 slices [9].
The hardware consist of two fundamental blocks, bandpass filters, and polyphase channelizers for UMTS and WLAN. The design process includes system breakup into functional units in *top-down* fashion. Each functional unit is further splitted into smaller processing blocks. Each functional block is coded in VHDL and simulated in ModelSim. All the functional units are then interconnected together in a *bottom-up* approach to get the final system. The system is tested and verified with the MatLab generated data. At the end, resource utilization of the target chip is presented.
In the polyphase channelizers, UMTS polyphase filter bank has 14 sub-filters each having length of 11 taps, whereas WLAN polyphase filter bank has 3 sub-filters each having length of 10 taps. We will focus to the design of polyphase channelizer for WLAN. (The polyphase channelizer for UMTS will be little modification of the WLAN channelizer interms of polyphase filter bank, and state machine based controller.) The basic design parameters are:

- Input Data-width: 16 Bits (1 sign , 7 Integer, 8 Fraction)
- Filter's Coefficient Data-width: 12 Bits (1 sign , 0 Integer, 11 Fraction)
- Input data is complex and Filter Coefficients are Real
- Complex Phasors : 16 Bits (1 sign , 1 Integer, 14 Fraction)
- Complex Output : 28 Bits (1 sign , 9 Integer, 18 Fraction)

The serial polyphase structure with parallel MAC is splitted in to sub-blocks as shown in Fig. 15. It consists of:

- Shift Register Bank: To store the incoming data for each sub-filter
- Filter's Coefficient Bank: To store the filter coefficients
- Decoder: To generate and decode the control signals for operating sub-modules of the channelizer like shift-register bank, filter's coefficient bank etc.
- Parallel Multiply and Accumulate : To perform the convolution operation between sub-filter's data and corresponding cofficients.
- Phasor Multiplication : To perform the beam forming operation. The phasors are selected based on the signal generated by decoder.
- Accumulator : To perform the addition of the beam forming process. It is reset after every coherant phase summation.

The decoder and the control sequence act as a commutator to the polyphase sub-filters. In Shift-register Bank, there are shift-register arrays equal to the number of the polyphase sub-filters which are multiplexed to give their output to the parallel multiply and accumulate block. In polyphase channelizer for WLAN, there are 3 sub filters, so we need to have 3 shift register arrays each of length 10. The data loading operation is controlled by the control sequence generated by the decoder. The input data is fed to all the shift-register array, but get loaded only in one, activated by the control sequence. So in order to have a shift-register bank, we need to have shift-register arrays and multiplexers, as shown in Fig. 16.
In the hardware implementation, seperate modules for multiplexer, decoder and shift-register array are formed, tested and integrated to form the shift-register bank. In shift-register array, registers are cascaded to form an array. On every activated clock cycle it shifts the data to the next register in the row. In the decoder (state-machine based controller), counters and comparators are used to generate and keep track on the output controlled sequence. In polyphase channelizer for WLAN, after feeding data samples to the shift-register bank, the operations of coefficient multiplication, addition and phasor multiplication are performed. The array registers are loaded in a cyclic fashion, so their addresses are generated by another counter. Now the blocks are combined in *bottom-up* fashion to form the register-bank. It uses 3 shift-register

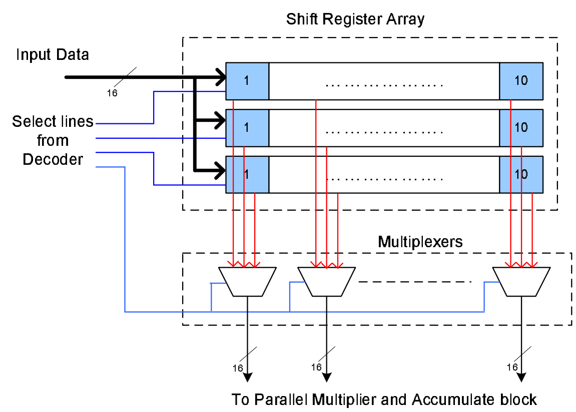

Fig. 16. Shift-register bank, consisting of shift-register arrays and multiplexers. Three(3) shift-register arrays of length ten(10) are multiplexed to give data to parallel multiply and accumulate block. The input data is fed to all the shift-register array, but get loaded only in one, activated by the control sequence generated by decoder.

arrays, 10 multiplexers and 1 decoder.

In the parallel multiplier and accumulate block, multipliers equal to the sub-filter's length are required. So 10 multipliers are required. To accumulate the products (from shared multipliers) summer tree network is used. They are operated in pipelined stages to increase the throughput, as shown in Fig. 17. In the first stage, all the multiplication operations are performed at once and then in the next four(4) stages, products are accumulated to form the output.

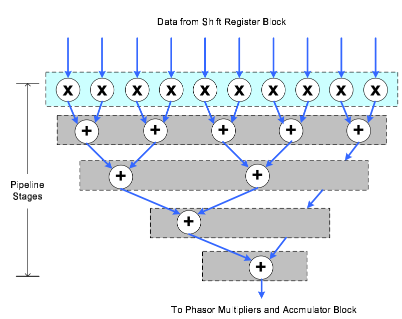

Fig. 17. Parallel multiplier and accumulate block: It require 10 multipliers and summer tree network. They are operated in pipelined stages to increase the throughput. In the first stage, all the multiplication operations are performed at once and then in the next four(4) stages, products are accumulated to form the output.

In phasor multiplication block, the phasors corresponding to the channel desired at the output, is already stored in an array. They are accessed by the control sequence same as of the sub-filter bank, but with the delay equal to the pipeline operation. By having different channel phasors, stored in array, different channels can be obtained at the output. The phasor multiplication is a complex data operation which requires two clock cycles, one for four(4) multiplications at a time and one for two(2) addition/subtraction as shown in Fig. 18. The phasor multiplication process is pipelined to increase the system throughput.

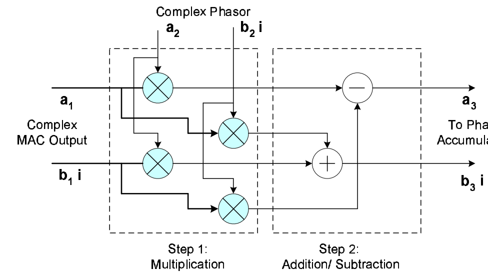

Fig. 18. Complex phasor multiplication with complex MAC data. It requires two clock cycles, one for four(4) multiplications at a time and one for two(2) addition/subtraction.

In the accumulator block, the phase coherant summation take place. It is reset after accumulating every 3 operations of the phasor multiplication block, to accumulate for the new output sample. Accumulation process is also controlled by state machine based controller to synchronized it with the MAC and phasor multiplication operations.

In the top-level module, all the sub modules are combined to form the final polyphase channelizer for WLAN. Filter coefficient bank, and the phasor coefficient bank are implemented with storage registers. They can be implemented by using dedicated memory block in the FPGA.

The module is finally tested and verified by the fixed-point data generated by MatLab for the WLAN channels. The resource utilization of polyphase channelizer for WLAN is tabulated in Table VI. The resource utilization is small in terms of Slices, and it uses 14% of the embedded multipliers (DSP48s). The maximum operating frequency comes out 101.930MHz, which is above the desired operating frequency of 90MHz.

Resource Ultilization for WLAN Channelizer

Selected Device	Virtex-IV 4vsx35ff668-10	
Number of Slices:	1842 out of 15360	11%
Number of Slice Flip Flops:	2261 out of 30720	7%
Number of DSP48s:	28 out of 192	14%

TABLE VI
RESOURCE ULTILIZATION: IT IS SMALL IN TERMS OF SLICES, AND IT USES 14% OF THE EMBEDDED MULTIPLIERS (DSP48S).

The module for UMTS channelizer along with arbitrary polyphase interpolator is also implemented, tested and verified. Polyphase channelizer has 14 sub-filters each having 11 taps, and arbitrary upsampler filter partitioned into 16 sub-filters each of length 11 taps. The resource utilization of polyphase channelizer for UMTS is tabulated in Table VII. The maximum operating frequency comes out 102.566MHz, which is above the desired operating frequency of 70MHz.

The resource utilization of arbitrary polyphase interpolator for UMTS is tabulated in Table VIII. The maximum operating frequency comes out 98.961MHz, which is above the desired operating frequency of 80MHz.

Resource Ultilization for UMTS Channelizer

Selected Device	Virtex-IV 4vsx35ff668-10	
Number of Slices:	5202 out of 15360	33%
Number of Slice Flip Flops:	6367 out of 30720	20%
Number of DSP48s:	30 out of 192	15%

TABLE VII
RESOURCE ULTILIZATION: IT REQUIRES 33% OF SLICES, AND 15% OF THE EMBEDDED MULTIPLIERS (DSP48S).

Resource Ultilization for UMTS Upsampler

Selected Device	Virtex-IV 4vsx35ff668-10	
Number of Slices:	1242 out of 15360	8%
Number of Slice Flip Flops:	1677 out of 30720	5%
Number of DSP48s:	22 out of 192	11%

TABLE VIII
RESOURCE ULTILIZATION: IT REQUIRES 8% OF SLICES, AND 11% OF THE EMBEDDED MULTIPLIERS (DSP48S).

IV. CONCLUSION

We presented a dual-standard software radio receiver architecture. A system designed with resource efficient technique 'polyphase channelizer' is used to extract the 12 UMTS and 3 WLAN Channels with desired rate at the baseband. The sampling frequency is a critical parameter in the whole system design. By having multiple bands the spectrum is much wider, so in order to fulfill the Nyquist criterion of $fs \geq 2B$, higher sampling frequency is required. This puts more limitations on the selection of hardware platform with high speed ADCs, technology with higher switching speed. The system has been optimized at 630MHz, where the polyphase channelizer works at its best level by extracting all the channels at the same time. Lowering down the sampling frequency helps to reduce the filter Taps and therefore, the computational complexity of the polyphase channelizer is much reduced in comparision to the previous design [1]. Serial Polyphase filter structure with parallel MAC is considered for the FPGA implementation. The critical analysis in terms of hardware area is carried out, which reflects that Distributed Arithmetic or Dedicated Xtreme DSP48 blocks are optimal solution for polyphase channelizer. The FPGA implementation of the WLAN polyphase channelizer requires 11% of the slices and 14% of the embedded multipliers (DSP48s) and maximum operating frequency is 101.930MHz, which is above the desired operating frequency of 90MHz. UMTS channelizer requires 33% of the slices and 15% of the embedded multipliers (DSP48s). Its maximum operating frequency is 102.566MHz, which is above the desired operating frequency of 70MHz. Finally, the upsampler(UMTS) requires 11% and 8% of Slices and DSP48s respectively which can be operated up to 98.96MHz.

ACKNOWLEDGMENT

The research described in this publication is carried out in Center for Software Defined Radio at Aalborg University. A special thanks to Prof. fredric j harris, San Diego State University (USA), for his valuable guidance for setting up the system design.

REFERENCES

[1] Mehmood-ur-Rehman Awan, Muhammad Mahtab Alam, ”Design & Implementation of FPGA-based Multi-standard Software Radio Receiver”, in *Proc. IEEE Norchip Conference on microelectronic*, Nov. 2007
[2] B. Razavi, ”Challenges and trends in RF design”, in *Proc. IEEE ASIC Conf. and Exhibit*, 1996, pp. 81-86
[3] Stream radio goes digital. http://www.csdr.dk.
[4] Mehmood-ur-Rehman Awan, Muhammad Mahtab Alam, ”Design & Implementation of FPGA-based Multi-standard Software Radio Receiver”, Master of Engineering Thesis, Aalborg University Denmark, June 2007
[5] Patel, M. and Lane, P., ”Comparison of downconversion techniques for software radio”, Department of Electronics and Electrical Engineering, University College London. www.ee.ucl.ac.uk/lcs/papers2000/lcs050.pdf.
[6] Fredric J. Harris, Chris Dick and Micheal Rice, ”Digital receivers and Transmitters using Polyphase Filter Banks for Wireless Communications”, in *IEEE Transaction on microwave theory and techniques*, Vol. 51, No. 4 April 2003
[7] Chris Dick, Fredric J. Harris. ”Performing Simultaneous Arbitrary Spectral Translation and Sample Rate Change, in Polyphase Interpolating or Decimating Filters in Transmitters and Receivers”. http://www.xilinx.com/products/logicore/dsp/sdr_paper_1.pdf.
[8] Fredric J. Harris, *Multirate Signal Processing for Communication Systems*, Prentice Hall, 2006
[9] Variable Parallel Virtex Multiplier V2.0. [online]. Available: http://www.xilinx.com/ipcenter/catalog/logicore/docs/mult_vgen_v2_0.pdf.