A low-cost FPGA implementation of multi-channel FIR filter with variable bandwidth

Sang Yoon Park\textsuperscript{a)}

\textit{Department of Electronic Engineering and MPEES-ARC, Myongji University, 116 Myongji-ro, Cheoin-gu, Yongin-si, Gyeonggi-do 449–728, Korea}

\textsuperscript{a)} sypark@mju.ac.kr

Abstract: This paper proposes a low-cost implementation of a multi-channel FIR filter on FPGA, where each channel can have variable bandwidth and coefficients. New structures of the tapped-delay line and the coefficient bank unit based on time-division multiplexing are proposed. Pipelined adder tree is used to expedite the filtering process without disturbing generation of control signals for multi-channel data access. From implementation results, it is found that the proposed 39-tap FIR filter involves 32\% less number of slice registers as well as 65\% less number of DSP blocks than the Xilinx FIR Core 5.0, and also supports variable bandwidth.

Keywords: FIR filter, multi-channel, FPGA, reconfigurable, low-cost, variable bandwidth

Classification: Integrated circuits

References

[1] L. Ming and Y. Chao: Proc. IEEE CDCIEM (2012) 764. DOI:10.1109/CDCIEM.2012.187
[2] I. Hatai, I. Chakrabarti and S. Banerjee: Proc. IEEE IPDPSW (2013) 247. DOI:10.1109/IPDPSW.2013.188
[3] T. Hentschel, M. Henker and G. Fettweis: IEEE Personal Commun. Mag. 6 (1999) 40. DOI:10.1109/98.788214
[4] K.-H. Chen and T.-D. Chiueh: IEEE Trans. Circuits Syst. II, Exp. Briefs 53 (2006) 617. DOI:10.1109/TCSII.2006.875373
[5] D. Xu, E. Jung and X. Liu: IEEE Trans. Mobile Comput. 11 (2012) 1372. DOI:10.1109/TMC.2011.168
[6] C. Cicconetti, I. F. Akyildiz and L. Lenzini: Proc. IEEE INFOCOM (2007) 2018. DOI:10.1109/INFCOM.2007.244
[7] \textit{IP LogiCore FIR Compiler} Version 5.0 (Xilinx Inc., 2011).
[8] \textit{AN 421: Accelerating DUC and DDC System Designs for WiMAX} Version 2.2 (Altera, 2007).
[9] Reference Design RD1052, Multi-Channel Digital Up/Down Converter for WiMAX Systems (Lattice Semiconductor Corporation, 2009).
[10] K. K. Parhi: \textit{VLSI Digital Signal Processing Systems Design and Implementation} (John Willey and Sons, Inc., 1999)
1 Introduction

Multi-channel digital up converters (DUC) and digital down converters (DDC) are broadly used to implement the digital front end (DFE) of various standards such as Long-Term Evolution (LTE), Worldwide Interoperability for Microwave Access (WiMAX), Intelsat Earth Station Standards (IESS), and their software defined radio applications [1, 2, 3, 4]. A single channel DUC/DDC has intrinsically two separate channels for in-phase and quadrature (I/Q), thus N-channel DUC/DDC has 2N equivalent channels. The straightforward way to implement multi-channel DUC/DDC is to replicate the single channel 2N times, however, it wastes hardware resources. It is desirable to share resources over multiple channels as well as I/Q channels. A root-raised-cosine FIR filter and a half-band FIR filter are often employed in DUC/DDC design, which are the most expensive parts in the design. There exist efficient ways to share hardware resources over multiple channels for those FIR filters. Meanwhile, cognitive radio (CR) exploits temporal and spatial variations of the spectrum in the radio environment [5, 6]. CR improves spectrum efficiency by allocating different bandwidth per channel. Therefore, multi-channel FIR filter with dynamic bandwidth is required for the multi-channel digital transceiver using CR.

There are reference field programmable gate array (FPGA) designs of the FIR filter to support multiple channels [7, 8, 9]. However, such implementations support multiple channels which have only the same sampling rate and the same frequency response for all the channels. Therefore, the existing designs cannot support the different bandwidth per channel for CR applications. In this paper, therefore, we focus on a low-cost design of the multi-channel FIR filter accommodating the requirement of variable channel bandwidths. The proposed design is targeted at Xilinx FPGA, but can also be applied to FPGA devices from other leading vendors.

In the next section, we present the detail structure of the proposed FIR filter supporting multiple channels with variable bandwidths. The implementation results of the proposed FIR filter are provided in Section 3 with the performance comparison. Conclusions are stated in Section 4.

2 Proposed multi-channel FIR filter

In this Section, we propose a new structure of the proposed FIR filter supporting multiple channels so that each channel can have different bandwidth and coefficients. As shown in Fig. 1, the top module of the proposed FIR filter is implemented in a well-known direct-form structure [10, 11]. However, sub-modules of a tapped-delay line (TDL), a coefficient bank unit, an adder unit, and an
accumulator are implemented in different ways from the existing designs to support multiple and variable-bandwidth channels.

2.1 Structure of tapped-delay line

In the conventional full-parallel design of the $N$-tap FIR filter, the TDL is implemented as a cascade of $(N - 1)$ registers. In order to support multiple channels, each channel needs to have its own data path, each of which is equivalent to a single channel TDL. Fig. 2 depicts the structure of the multi-channel TDL based on register arrays when the number of channels is 8. As shown in Fig. 2, every tap consists of 8:1 de-multiplexor (DMUX), an array of 8 registers, and 8:1 multiplexor (MUX). A proper location to store the input data is selected by data-ch-id, and a 8:1 MUX selects one out of data stored in 8 registers, which is used as an input of the multiplier as well as an input of the DMUX for the next tap. An $N$-tap FIR filter supporting $M$ multiple channels requires $(N - 1)M$ registers. As $N$ or $M$ increases, the TDL requires more number of registers and bigger sizes of MUXes and DMUXes. Therefore, for large $N$ and $M$, the TDL would be the most expensive part in the design of the multi-channel FIR filter. In the proposed design, the register array with MUX and DMUX is replaced with a distributed RAM (DRAM). Since the minimum depth of Xilinx DRAM v5.1 is 16 [12], multiple

![Fig. 1. Architecture of the top module of the proposed FIR filter supporting multiple channels.](image)

![Fig. 2. Structure of a tapped-delay line based on register arrays.](image)
channels up to 8 with I and Q channels can be supported by the smallest DRAM unit. Fig. 3 shows the detail structure of TDL based on DRAM when $N = 19$. If the channel ID (denoted as data-ch-id in Fig. 3) of the current input sample (denoted as data-in) is $m$, all the DRAMs retrieve the registered data from address $m$ and provide them to the multipliers. If the FIR filter has linear phase property, the number of multipliers can be reduced by half using additional $N/2$ adders as shown in Fig. 3. Considering the ratio of the maximum usable frequency (MUF) of the FIR filter and the maximum sampling rate of the single channel, a multiplier can be shared over the several taps by splitting the sample period into a few time slots. For example, if the MUF of 240 MHz is reported and the maximum sampling frequency of the single channel is 80 MHz, one multiplier can work for three taps. In this case, three input signals of MUXes should be retained until the multiplier produces results for three-taps as shown in Fig. 3. Registers located between adders and 3:1 MUXes hold data during three clock periods for this purpose. Enable signals for those registers are obtained by delaying the write-enable of DRAM denoted as d-we, considering delays for retrieval from DRAM. In Fig. 3, four 3:1 MUXes and four multipliers are used when $N = 19$ and coefficients have odd-symmetric property.

### 2.2 Structure of coefficient bank unit

If one multiplier works for $K$ taps and the number of channels is $M$, each coefficient bank unit (CBU) should store $K \times M$ coefficients. The DRAM can also be used to store the coefficients in the CBU. Each CBU has its own DRAM and circuits to generate addresses and control signals. Fig. 4 shows the structure of the proposed CBU when $M = K = 4$. The most significant bits of addr are used to select one out of $K$ taps whereas the remaining bits are used as channel IDs. In Fig. 4, mac-id identifies which DRAM should be enabled to write the coefficient, and w-tap-id decides the location for the coefficient to be written with the coef-ch-id. When $M = K = 4$, address for each DRAM is 4 bit word where most significant 2 bits are for the tap ID, and remaining 2 bits are for the channel ID. The channel ID and tap...
ID to read the coefficient are derived from the channel ID of the corresponding input sample in the control unit.

2.3 Structure of adder unit

The output from multipliers are added together to produce the output of the filter. However, the repetitive additions without pipelining result in a long critical path. In order to expedite the process, registers need to be inserted after every addition, which is called as a systolic design (see Fig. 5 in [7]). However, in the design of the multi-channel FIR filter with variable bandwidths, the systolic design is not feasible to generate addresses and selection signals according to the pipelining. The proposed design, therefore, employs a pipelined adder tree whose pipelining does not affect to the generation of control signals for the TDL and the CBU. Fig. 5 shows the structure of the pipelined adder tree when the number of inputs is 8. Dashed lines represent possible locations of pipeline registers. The number of pipeline stages can be determined considering the required throughput of the FIR filter. The multipliers can be implemented using Xilinx DSP48 slice which consists of a multiplier, registers and adders. In that case, adders and pipeline registers in the first stage of the adder-tree can be absorbed into the Xilinx DSP48 slices to save resources as shown in Fig. 5.

2.4 Structure of accumulation unit

When one multiplier works for $K$ taps, an accumulation unit is required for the successive addition of $K$ multiplication results. Fig. 6 shows the structure of the accumulation unit. Before the first output is produced by the adder tree, the accumulator is reset by acc-rst, and the MUX selects the input from adder-tree, thereafter the output of the adder is chosen by the MUX. There are three registers in Fig. 6 where the Reg–1 is used for data accumulation of $K$ multiplication results. The Reg–2 collects the summation result with the enable signal at every $K$ clocks. The Reg–3 is used for the generation of the enable signal, which is delayed signal of acc-rst.
3 Result and performance

We have coded the proposed design in VHDL and implemented on Xilinx FPGA XC5VSX95T-1FF1136. The word-lengths of input sample and coefficient are chosen to be 12. The number of taps of the FIR filter is set to 39, and the number of channels is set to 8. Since each channel has I and Q components, filtering for 16 different channels are performed. The maximum sampling rate of the input is assumed to be 80 Mega samples per second (MSPS), that is, 40 MSPS for I and Q each. If all the channels have the same bandwidths, the sampling rate of each channel becomes 10 Msps. Based on the post place and route static timing analysis, the proposed FIR filter offers maximum usable frequency faster than 240 MHz. Therefore, one multiplier is implemented to operate for three taps, that is, $K = 3$.

In Table I, the proposed multi-channel FIR filter is compared with the existing reconfigurable FIR filters implemented on FPGA. As shown in Table I, the proposed FIR filter uses only 7 DSP48E slices whereas the Xilinx core generated by FIR compiler 5.0 uses 20 DSP48E slices [7]. Also, the proposed FIR filter offers 32% less number of slice registers as well as 28% less number of slice LUTs compared with the Xilinx core. Note that Xilinx FIR compiler 5.0 supports the multiple channels with only uniform bandwidth and the same coefficients set whereas the proposed FIR filter supports the multiple channels with different bandwidths. It should also be known that the proposed FIR filter can support filtering for 16 channels with a small overhead of only 7 DSP48Es and 5% more slice registers compared to the best of the existing systolic structures [13] whereas the design of [13] supports only the single channel.
In this paper, we propose low cost implementation of the multi-channel FIR filter with variable bandwidth on Xilinx FPGA. Unique features of the proposed FIR filter are summarized as follows.

1) Supporting multiple channels with different bandwidth per channel.
2) Supporting different coefficient set per channel.
3) Configurable channel bandwidth and coefficients.
4) High-speed direct-form structure with pipelined adder tree.
5) Saving of used DSP blocks by half using coefficient symmetric.
6) Saving of slice registers by using DRAM in the data-path and coefficient bank unit.
7) Saving of used DSP blocks sharing a multiplier over several taps.

Acknowledgments

This work was supported by the 2015 MPEES Advanced Research Center Fund of Myongji University.