A High-Performance FIR Filter Architecture for
Fixed and Reconfigurable Applications

Abstract:

Transpose form finite-impulse response (FIR) filters are inherently pipelined and support multiple constant multiplications (MCM) technique that results in significant saving of computation. However, transpose form configuration does not directly support the block processing unlike direct form configuration. In this paper, we explore the possibility of realization of block FIR filter in transpose form configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. Based on a detailed computational analysis of transpose form configuration of FIR filter, we have derived a flow graph for transpose form block FIR filter with optimized register complexity. A generalized block formulation is presented for transpose form FIR filter. We have derived a general multiplier-based architecture for the proposed transpose form block filter for reconfigurable applications. A low-complexity design using the MCM scheme is also presented for the block implementation of fixed FIR filters. The proposed structure involves significantly less area delay product (ADP) and less energy per sample (EPS) than the existing block implementation of direct-form structure for medium or large filter lengths, while for the short-length filters, the block implementation of direct-form FIR structure has less ADP and less EPS than the proposed structure. Application specific integrated circuit synthesis result shows that the proposed structure for block size 4 and filter length 64 involves 42% less ADP and 40% less EPS than the best available FIR filter structure proposed for reconfigurable applications. For the same filter length and the same block size, the proposed structure involves 13% less ADP and 12.8% less EPS than that of the existing direct-form block FIR structure. The proposed architecture of this paper analysis the logic size, area and power consumption using Xilinx 14.7.

Existing Method:

FIR filters of large order to meet the stringent frequency specifications. Very often these filters need to support high sampling rate for high-speed digital communication. Filter coefficients very often remain constant and known a priori in signal processing applications. This feature has been utilized to reduce the complexity of realization of multiplications. Several designs have been suggested by various researchers for efficient realization of FIR filters (having fixed coefficients) using distributed arithmetic (DA) and multiple constant multiplication (MCM) methods. DA-based designs use lookup tables (LUTs) to store
precomputed results to reduce the computational complexity. The MCM method on the other hand reduces the number of additions required for the realization of multiplications by common subexpression sharing, when a given input is multiplied with a set of constants. The MCM scheme is more effective, when a common operand is multiplied with more number of constants. Block-processing method is popularly used to derive high-throughput hardware structures. It not only provides throughput-scalable design but also improves the area-delay efficiency. The derivation of block-based FIR structure is straightforward when direct-form configuration is used, whereas the transpose form configuration does not directly support block processing. But, to take the computational advantage of the MCM, FIR filter is required to be realized by transpose form configuration.

**Proposed Method:**

In this paper, we explore the possibility of realization of block level Reconfigurable FIR filter in transpose form configuration in order to take advantage of the MCM schemes and the inherent pipelining for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications.

**Applications:**

1. Digital Signal Processing.
2. Speech processing.
3. Loud speaker equalization.
4. Echo cancellation and adaptive noise cancellation.
Advantages:

High throughput,
Reduced number of additions.