Efficient Implementation of Digital Filters with Use of Advanced Synthesis Methods Targeted FPGA Architectures

Mariusz Rawski, Paweł Tomaszewicz, ♣Henry Selvaraj, Tadeusz Łuba

Warsaw University of Technology, Institute of Telecommunications
Nowowiejska 15/19, 00-665 Warsaw, Poland

♣ *University of Nevada, Las Vegas*
4505 Maryland Parkway, Las Vegas, NV
89154-4026, USA

Abstract

This paper presents an efficient method for implementation of digital filters targeted FPGA architectures. The traditional approach is based on application of general purpose multipliers. However, performance of multipliers implemented in FPGA architectures does not allow to constructs high performance digital filters. In this paper application of distributed arithmetic is demonstrated. Since in this approach general purpose multipliers are replaced by combinational LUT blocks, it is possible to construct digital filters of very high performance. However LUT blocks can be of considerable size thus advanced synthesis methods have to be used to map them efficiently into FPGA resources. In this paper an application of the functional decomposition based synthesis has been investigated. This method is recognised as the best synthesis method targeted FPGA architectures and allows significant improvements in digital filters implementation. The paper presents many examples confirming that decomposition allows reduction of logic cell utilisation of filter implementation based on distributed arithmetic concept with no performance degradation and even increasing it.

1. Introduction[1]

Digital filters are used to modify attributes of signal in the time or frequency domain trough the process called linear convolution. With advances in digital technology they are rapidly replacing analogue filters, which were implemented with RLC components. In recent years digital filters have been recognised as primary digital signal processing (DSP) operation. They are typically implemented as multiply-accumulate (MAC) algorithms with use of special DSP devices [6, 7, 8]. Such devices are based on the concept of RISC processors with an architecture consisting of fast array multipliers. By using pipeline architecture the speed of such implementation is limited by the speed of array multiplier.

Progress in development of programmable architectures observed in recent years resulted in digital devices that allow building very complex digital circuits and systems at relatively low cost in a single programmable structure. Programmable technology provides possibility to increase the performance of digital system by implementation of multiple, parallel modules in one chip. This technology allows also application of special techniques such as distributed arithmetic (DA) [3, 13]. DA technique is extensively used in computing sum of product in filters with constant coefficients. In such a case partial product term becomes a multiplication with constant (i.e. scaling). DA approach significantly increases the performance of implemented filter, by removing general purpose multipliers and introducing combinational blocks that implement the scaling. These blocks have to be efficiently mapped onto FPGA's logic cells. This can be done with use of such advanced synthesis methods as functional decomposition [15].

In case of applications targeted to FPGA structures based on look-up tables (LUT), the influence of advanced logic synthesis procedures on the quality of hardware implementation of signal and information processing systems is especially important. Direct cause of such situation is imperfection of technology mapping methods that are currently widely used, such as minimization and factorization of Boolean function, which are traditionally adapted to be used for structures based on standard cells. These methods transform Boolean formulas from the form of sum-of-products into multilevel, highly factorized form that is then mapped into LUT cells. This process is at variance with nature of LUT cell, which from the logic synthesis' point of view is able to implement any logic function of limited input variables. For this reason in case

[1] This paper was supported by Polish State Committee for Scientific Research financial grant for years 2003-2004 number 4 T11D 014 24