DIFFERENT METHODS OF HARDWARE IMPLEMENTATIONS DIGITAL FIR FILTER IN FPGA

Ing. Ondřej PIROCHTA, Doctoral Degreee Programme (1)
Dept. of Radio Electronics, FEEC, BUT
E-mail: xpiroc00@stud.feec.vutbr.cz

Supervised by Dr. Aleš Prokeš

ABSTRACT

This paper deals with various methods how to realize digital FIR filters in programmable logic devices. Their computing powers in dependency of their filter lengths are compared.

1 INTRODUCTION

The conventional single rate filter computes the convolution sum as defined in Eq. 1, where *N* is the number of filter coefficients [1], [2]:

$$y(k) = \sum_{n=0}^{N-1} a(n) \cdot x(k-n)\,. \tag{1}$$

The conventional tapped delay line realization of this inner product calculation is shown in Fig. 1.

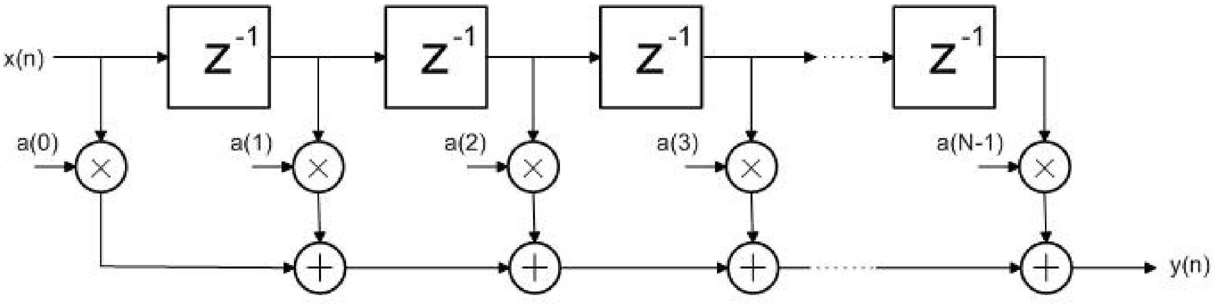

Fig. 1: *Conventional Tapped Delay Line FIR Filter Mechanization*

Although the diagram is a useful conceptualization of the computation performed by the filter, the actual FPGA realization can be quite different. The most common way of hardware implementation uses single or multiple time shared multiply accumulate (MAC) functional units to supply the *N* sum of product calculations of the filter. This kind of hardware realization is described in the following chapter.

2 HARDWARE IMPLEMENTATION OF DIGITAL FIR FILTER WITH MAC UNITS

In fig. 2 a block diagram of a single MAC engine is shown. The MAC FIR engine is composed of 5 main sections: input buffer, data storage, coefficient storage, a multiply accumulate unit and a control logic. Full precision results from the multiplier are summed in the accumulator. The accumulator has sufficient precision to ensure that no overflow can occur during the “sum-of-products” operation.

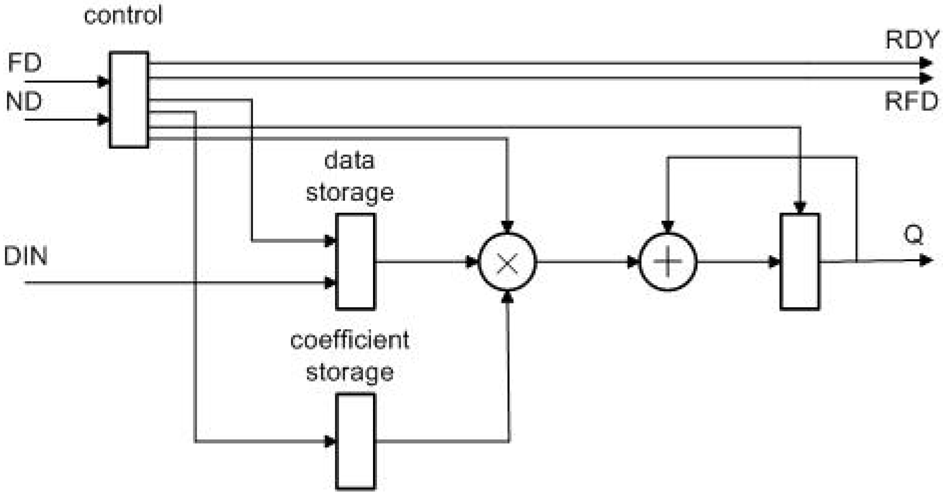

Fig. 2: *Block Diagram of a Single MAC Engine*

3 HARDWARE IMPLEMENTATION OF A FIR FILTER WITH DISTRIBUTED ARITHMETIC

A simplified view of a distributed arithmetic (DA) FIR is shown in fig. 3. In its most obvious and direct form, DA based computations are bit serial in nature serial distributed arithmetic (SDA) FIR. Extensions to the basic algorithm remove this potential throughput limitation [3]. The advantage of a distributed arithmetic approach is its efficiency of mechanization. The basic operations required are a sequence of table look ups, additions, subtractions and shifts of the input data sequence. All of these functions are efficiently mapped to the FPGAs. Input samples are presented to the input parallel-to-serial shift register (PSC) at the input signal sample rate. When the new sample is serialized the bit wide output is presented to a bit serial shift register or time skew buffer (TSB). The TSB stores the input sample history in a bit serial format and is used for creating the required inner product computation. The TSB itself is constructed using a cascade of shorter bit serial shift registers. The nodes in the cascade connection of TSBs are used as address inputs for a look up table (LUT) [4]. This LUT stores all possible partial products [3] over the filter coefficient space.

The throughput of the DA FIR can be improved by using parallel distributed arithmetics. Every bit of the input samples must be indexed in turn before a new output sample becomes available at the SDA FIR. When the input samples are represented with B bits of precision, B clock cycles are required to complete an inner product calculation (for a nonsymmetrical impulse response). Additional speed can be obtained in several ways. One approach is to partition the input words into M subwords and process these subwords parallel.

This method requires *M* times as many memory look up tables and results in an increased amount of storage facilities. Maximum speed is achieved by factoring the input variables into single bit subwords. The resulting structure is a fully parallel DA (PDA) FIR filter. With this implementation a new output sample is computed every clock cycle. PDA FIR filters provide an exceptionally high performance.

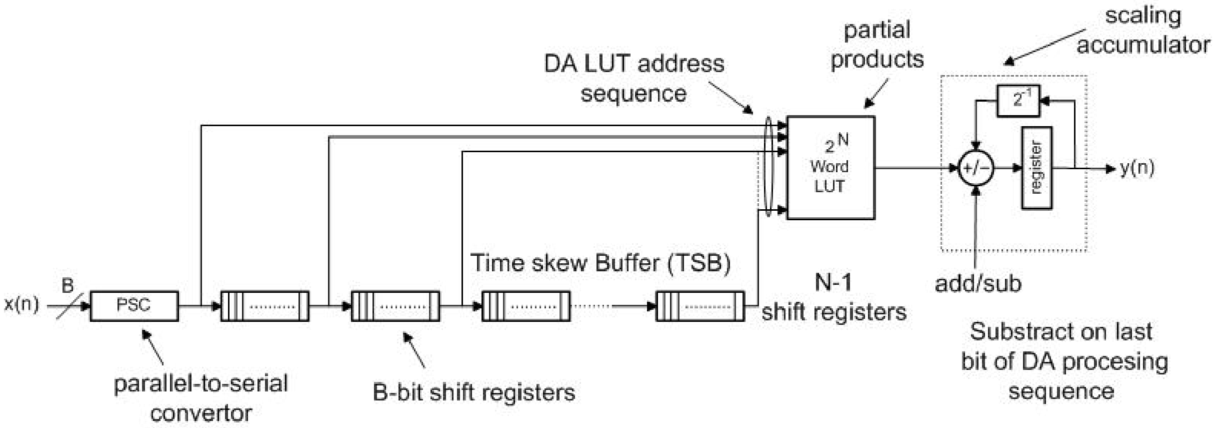

Fig. 3: *Serial Distributed Arithmetic FIR Filter*

4 COMPARISON BETWEEN SDA BASED FILTER AND FILTER WITH MAC

In a conventional MAC based FIR realization, the sample throughput is coupled to the filter length. With a DA architecture the system sample rate is related to the bit precision of the input data samples. Every bit of an input sample must be indexed and processed in turn before a new output sample is available. For *B* bit precision input samples *B* clock cycles are required to form a new output sample for a nonsymmetrical filter and *B+1* clock cycles are needed for a symmetrical filter. The rate at which data bits are indexed occurs at the bit clock rate. The bit clock frequency is greater than the filter sample rate (*fs*) and is equal to *B·fs* for a nonsymmetrical filter and (*B+1*)·*fs* for a symmetrical filter. In a conventional instruction set (processor) approach to the problem, the required number of multiply accumulate operations are implemented using a time shared MAC unit. The filter sample throughput is anti proportional to the number of filter taps. As the filter length is increased the system sample rate is proportionally decreased. This is not the case with DA based architectures. The filter sample rate is independent of the filter length. As the filter length is increased in a DA FIR filter more logical resources are consumed but the throughput is maintained.

In fig. 4 a comparison between a SDA FIR architecture and a conventional scheduled MAC based approach is shown. Values of maximal sample rate were obtained from simulations in the development environment Xilinx ISE 6. As target device the Virtex II from Xilinx was chosen. The dependency SINGLE MAC is for the filter with a single MAC unit with the precision of the input samples *B*=16. The Dependency of the maximal sample rate is independent of the precision of the input samples for the filter with a MAC unit. Several values of input sample precision for the SDA FIR are presented. The dependency of the SDA filter throughput on the sample precision is apparent from the plots. For 8-bit precision input samples the SDA FIR maintains a higher throughput for filter lengths greater than 8 taps than a filter with single MAC unit. When the sample precision is increased to 16 bits, the crossover point is 16 taps.

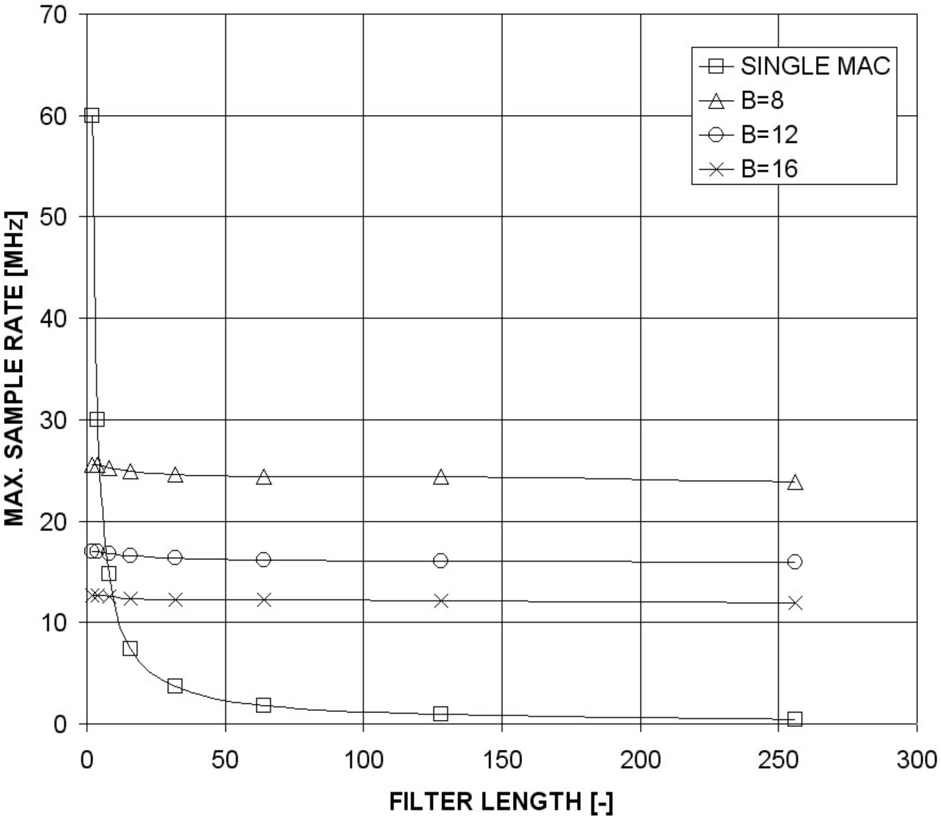

Fig. 4: *Dependency of Maximal Sample Rate on the Filter Length*

5 CONCLUSION

The paper provides a basic comparison of digital FIR filter realized with a single MAC unit and filters based on serial distributed arithmetic. Advantage of filters based on SDA is the independency of the computing power on the length of the filter. The length of the filter is dependent on the quantity of the occupied logic on the chip. Advantage of filters with MAC units is their invariable quantity of occupied logic on the chip. Computing power of filters with MAC units can be improved by using multiple MAC units or pipelining.

ACKNOWLEDGEMENT

This contribution has been supported by grant GA ČR No. 102/05/0732.

REFERENCES

[1] Vích, R., Smékal, Z.: Číslocové filtry, Academia, AV ČR, 2000

[2] Jan, J.: Číslicová filtrace, analýza a restaurace signálů, VUT Brno, 1997

[3] White, S. A.: Applications of Distributed Arithmetic to Digital Signal Processing, IEEE ASSP Magazine,Vol. 6(3), pp. 4-19, July 1989

[4] Wakerly, J. F.,: Digital design principles & practices, Prentice Hall, 2001