# A STUDY ABOUT FPGA-BASED DIGITAL FILTERS

**Javier Valls**
Dept. Ingeniería Electrónica
Univ. Politecnica de Valencia
Camino de Vera s/n
46071 Valencia, Spain
**jvalls@eln.upv.es**

**Marcos M. Peiró**
Dept. Ingeniería Electrónica
Univ. Politecnica de Valencia
Camino de Vera s/n
46071 Valencia, Spain
**mpeiro@eln.upv.es**

**Trini Sansaloni**
Dept. Ingeniería Electrónica
Univ. Politecnica de Valencia
Camino de Vera s/n
46071 Valencia, Spain
**tmsansal@die.upv.es**

**Eduardo Boemo**
E.T.S.I. Informática
Univ. Autónoma de Madrid
Ctra. Colmenar Km.15
28049 Madrid, Spain
**eduardo.boemo@ii.uam.es**

**Abstract - In this paper, a set of operators suitable for digit-serial FIR filtering is presented. The canonical and inverted forms are studied. In each of these structures both the symmetrical and anti-symmetrical particular cases are also covered. All circuits have been implemented using an EPF10K50 Altera FPGA. Main results show that the canonical form presents the less occupation and the higher throughput. The 8-tap filter versions implemented can be applied in real-time processing with sample rate ranging up to 7MHz using the bit-serial versions and up to 25 MHz with the bit-parallel ones.**

## INTRODUCTION

The high-density of current FPGAs make the construction of custom DSP (CDSP) single-chip systems possible. The advantages of this approach are multiple:

- A higher speed with respect to the general-purpose DSP solution can be obtained.
- The FPGA reprogrammability can be exploited to construct reconfigurable systems.
- Future DSP ASICs can be fast-prototyped, different design options can be emulated, and exhaustive or heterogeneous simulations can be avoided.
- Off-chip interconnections and external components like FIFOs or RAMs can be integrated in the embedded application.
- Hard-wired DSP cores can be simplified and optimised for a given application (data rate and precision, peculiarities of the coefficient, etc.).

A custom solution also allows the designer to select different types of arithmetic and styles of implementations. In several cases, it is senseless to use conventional bit-parallel circuits: they have an important cost in area and run faster than the speed needed by the application. In this way, digit-serial architectures become an important alternative to efficiently implementing a wide range of real-time signal processing circuits. The digit-serial approach allows the designer to select an intermediate area-time figure, situated between the bit-parallel and the bit-serial implementations. In addition, a serial stream of data matches better with the structure of an FPGA; thus, in an actual implementation, the speed of a full serial circuit is not N times lower than the equivalent N-bit parallel approach [1]. Several researchers have studied the implementation of filters on FPGAs. In [2] and [3] the main architectural ideas are reviewed. The powers-of-two coefficients are exploited by Evans in [4]. The bit-serial approach is studied in [5], [6]. In [7] and [8], the canonic signed digit arithmetic is applied with the bit-serial style of implementation. [9] Constructs several circuits based on digit-serial systolic multiplier. In [10], [11], [13], [14], [15], [16], the distributed arithmetic is introduced. Finally, a comparison between serial and parallel approaches is presented in [1]. From the implementation side, references [2], [3], [4], [7], [8], [9], [10], [11], [12], [13], [14] use Xilinx chips; [15] and [16] Altera devices; [1] both Xilinx and Altera; [4] both Orca and Xilinx; and [5] uses CLI. In this work, complete sets of digit-serial operators and filters have been evaluated in Altera FPGA devices.

The organisation of this paper is as follow. In the next section the digit-serial architectures are briefly exposed and the basic digit-serial adder cell is explained. In Section 3, the multipliers utilised are presented, and the way they give the results of the computation is depicted. In the next section, the different canonical and inverted form topologies to design FIR filters are summarised. Section 5 presents the operators needed to design them using digit-serial style with each one of the previous structures. In section 6, the implementation results in an EPF10K50-3 are given. Finally, the conclusions are presented.

## DIGIT-SERIAL ARCHITECTURES

In digit-serial computation, data words of size W bits are partitioned into digits of size N bits (the digit-size, N, is divisor of the word-size, W) and are processed serially one digit at a time with the least significant digit first. A complete word is processed in P=W/N clock cycles and consecutive words follow each other continuously. The time of P cycles is named a sample period. In every digit-serial operator, it is necessary to add some control signal to indicate a new word entry. The digit-serial processors include parallel-serial and serial-parallel converters to process in digit format and to present the result in parallel format. The digit-serial operators are cascaded following the data-flow algorithm in a pipeline fashion. A detailed explanation of this kind of architectures can be found in [17], [18], [19], [20], [21].

A set of digit-serial architectures can be designed by using different digit-sizes. Each element of the family will have a specific size and throughput. Thus, it is possible to choose the digit-size that best suits the speed of the application, minimising the cost in terms of area.

### Digit-Serial Adder

In Fig. 1.b, the N=2 bits digit-size adder is shown. For an 8-bit word length operation, in the first clock cycle the first digit of the input words (the two least significant bits of each ones) are fed to the adder. The carry-in must be zero in the first digit computation. A ripple carry addition of the two bits is computed, and the two least significant bits of the addition together with a carry-out, are produced. The carry-out is delayed one clock cycle to be added to the next digit. Finally, in the fourth clock cycle the last (or more significant) digit of each input data enters into the circuit and they are added to the carry-out of the previous computation. During this clock cycle, the CONTROL signal must be high in order to avoid the propagation of the carry-out from the last digit of a word to the first digit of the next data.

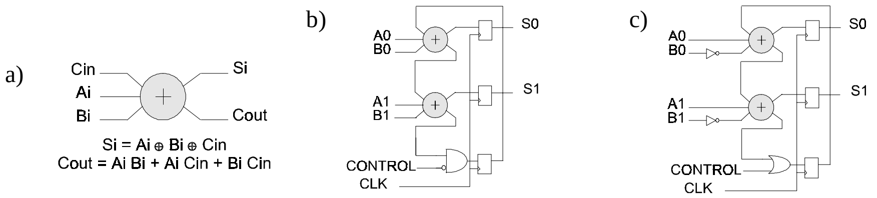


Fig.1: a) Full adder; b) digit-serial adder with N=2; c) digit-serial subtractor with N=2

## DIGIT-SERIAL MULTIPLIER

A family of digit-serial multipliers obtained by folding the two’s complement array multiplier has been selected because a double precision can be obtained without adding clock cycles. The design of this kind of digit-serial circuit has been highly detailed by Parhi, Hartley, and Corbet [18], [21]. It is shown in Fig.2, for 4 bits coefficient-size and digit-size N=2 bits.

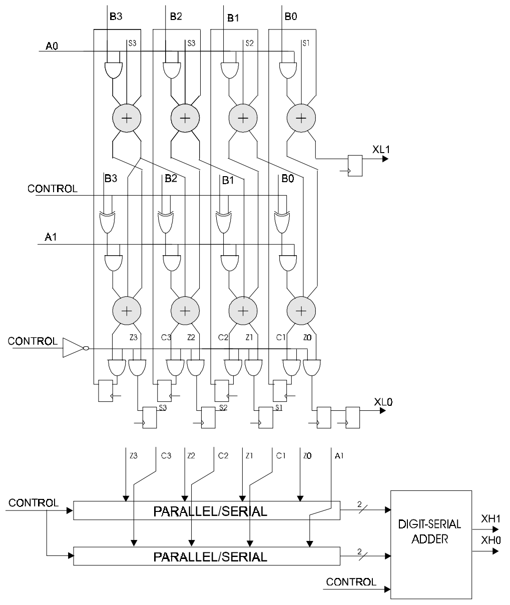


Fig.2. Digit-serial array multiplier with digit-size N=2.

This multiplier computes 2xAxB (by inserting a 0 in the LSB position and ignoring the MSB) and it produces a standard double precision result [22]. If data A and coefficient B are W-bit word size, the lower part of the product (W bit size) is outputted with a latency of 1 with respect to A. The higher part (W bit size) is produced by the digit-serial adder with a latency of W+1 with respect to A. This multiplier computes at the same time, the low order product of current input-word and the high order product of the previous one.

## FIR FILTER STRUCTURES

The equation for the computation of an N-taps FIR filter is:

$$y[n] = \sum_{k=0}^{N-1} h[n].\, x[n-k] \qquad (1)$$

From Eq.1, the direct form, transversal structure or canonical form (Fig. 3.a) can be generated, meanwhile the transposed direct form structure or inverted form (Fig. 3.b) is obtained by applying the Transposition Theorem [23].

a)

b)

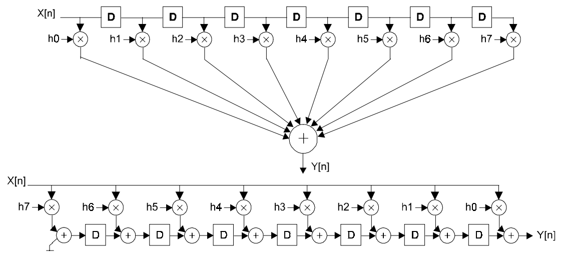
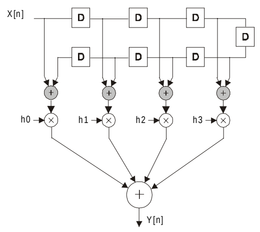


Fig.3. FIR filter structures: a) canonical form; b) inverted form.

a)



b)

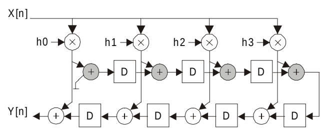


Fig.4. Symmetric FIR filter structures: a) canonical form; b) inverted form.

The main advantage of FIR filters is their linear-phase response. This property is achieved when the impulse response satisfies the symmetry or anti-symmetry conditions. Symmetric FIR filter topologies for both canonical and inverted forms are shown in Fig. 4. These structures can be transformed in the equivalent anti-symmetric forms by replacing some of the adders (those which are shadowed in Fig. 4) by subtractors. The number of multipliers in these circuits is significantly reduced compared with the previous structures: (N-1)/2 if N is odd and N/2 if N is even.

## DESIGN OF DIGIT-SERIAL FIR FILTER

In this section, a set of blocks to perform the digit-serial versions of the above structures is presented. For the sake of clearness, the following examples are based on the multiplier of Fig.2 and correspond to filters with:

- 8-tap programmable coefficients
- 2-bit digit-size (N=2)
- 8-bit Input data and coefficient size (W=8)

All filters keep full precision along the whole datapath. It is guaranteed if 19 bits word length is adopted ($W_{in\text{-}data}+W_{coefficient}+log_2\, M=19$, where M is the number of filter stages).

## Inverted form digit-serial FIR filter

The basic structure in the inverted form FIR filter is the multiply-and-accumulate shown in Fig. 5.

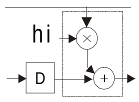
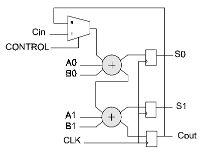


Fig.5. Basic structure of the multiply-and-accumulate

In order to design the digit-serial accumulator, several modifications of the adder cell (Fig.1.b) are necessary. Each accumulator cell must propagate its carry-out to the next cell, as well as feed the carry-out from the previous cell. During the first digit of any word, the CONTROL signal is high. As a consequence, the latched carry-out of the previous cell is fed to its least significant full adder. The final accumulator block is depicted in Fig. 6. It has a latency of 1.

a) 

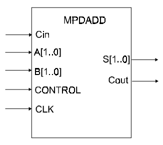
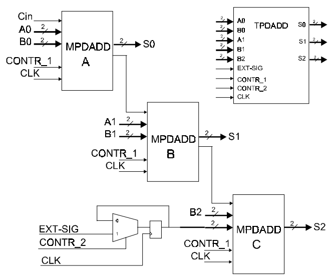
b) 

Fig.6: Basic digit-serial accumulator cell: a) scheme, b) symbol of the cell.

A multiple precision accumulator is constructed by connecting several of these cells and adding some circuitry to keep the sign extension. For implementing 8-taps FIR filters with N=2 bits and W=8 bits, three cells are needed and up to 24 bits of precision are guaranteed. The triple precision digit-serial accumulator is shown in Fig. 7. This circuit performs the addition of digits of three different words at a time. While intermediate digits of each word are computed, the carry-out in every cell is fed back and latched in order to be added to the following digits in the next clock cycle. During the first digit of each word, the latched carry-out of the last digit is propagated to the next cell through the carry-in input. The sign is extended by latching the sign bit of the double precision word and feeding it to the input of one of the digit inputs. It happens during the last digit of every word, when CONTR_2 signal is high.



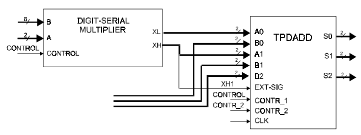


Fig.7. Triple precision digit-serial accumulator with digit-size N=2: scheme and symbol.

Fig.8. Digit-serial MAC

The MAC cell is illustrated in Fig. 8. The multiplier low word output (XL) is connected to the least significant accumulator (MPDADD A). The multiplier high-word output (XH) is connected to MPDADD B. The most significant bit of XH is connected to the EXT_SIGN input. Finally, the other digit inputs are connected to the one-sample-delayed outputs of the previous digit-serial MAC.

Two digits of successive samples with the same weight have to be fed into the accumulator at the same time. To achieve the right scheduling without adding latency, the digit-serial delay operator has to be designed with N lines of W/N-1 cascade registers.

## Linear-phase inverted form digit-serial FIR filter

The symmetric inverted form digit-serial FIR filters can be designed utilising the MAC described above and the structure shown in Fig. 4.b.

The anti-symmetric inverted form digit-serial FIR filters require the use of some subtractors (shadowed adders in Fig. 4.b) instead of adders. They can be done by modifying the circuit in Fig. 7. The inputs A0, A1 and EXT-SIGN are inverted and the carry-in of the least significant MPADD is fixed to “1”.

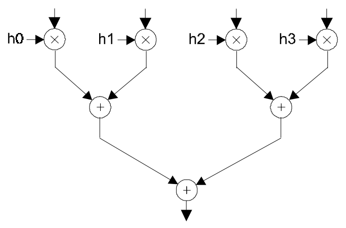


Fig.9. Four multipliers MAC.

## Canonical form digit-serial FIR filter

The basic structure in the canonical form FIR filter is the four-multipliers MAC shown in Fig. 9. The first line of adders has to guarantee the sign extension of the two input data. The structure of this cell is illustrated in Fig. 10. On the contrary, the last adder does not require sign manipulation, it only has to add the resulting data of the previous adders (Fig. 11). Two of these MACs are needed in an 8-taps FIR filter. Their outputs can be added using the block depicted in Fig. 11.

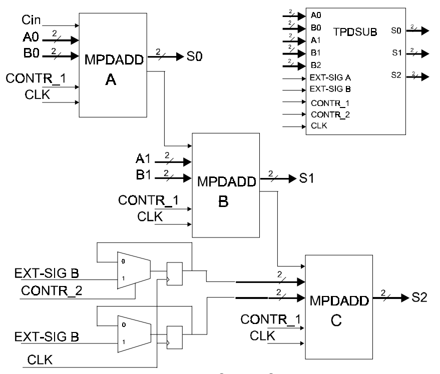


Fig.10. Structure of the first adders of the MAC.

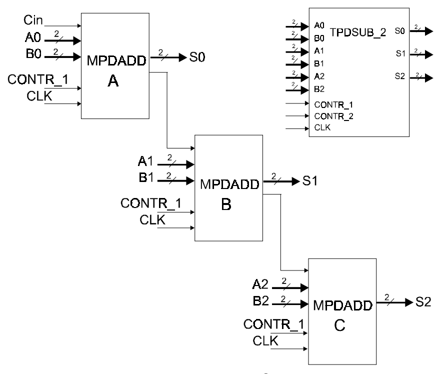


Fig. 11. Structure of the other adders of the MAC.

## Linear-phase canonical form digit-serial FIR filter

The structure of the symmetric digit-serial canonical form FIR filter is shown in Fig. 4.a. These kind of filters can be implemented using the MACs explained above. To add the symmetric samples, a digit-serial adder (Fig. 1.b) has to be used. For the anti-symmetric form, a digit-serial subtractor (Fig. 1.c) is needed instead of the adder. The problem in these filters is that the adder or the subtractor can overflow. Nevertheless, it could be avoided if the input samples are divided by two, although it will produce a quantization error increment.

## FPGA IMPLEMENTATION

Several families of 8-taps digit-serial FIR filters using the structures explained above have been implemented using an EPF10K50GC403-3 Altera FPGA [24]. In all cases, default placement and routing compilation options have been used. The main results can be observed in Table-I, where, for each structure and digit-size, the maximum clock rate, the sample period, the area occupied, and the area-time product are presented.

| | Digit-size (Bits) | Canonical Form | Inverted Form | Symmetric Canonical Form | Symmetric Inverted Form | Anti-symmetric Canonical Form | Anti-symmetric Inverted Form |
|---|---|---|---|---|---|---|---|
| Chip Speed (MHz) | 1 | 51,02 | 49,75 | 57,8 | 52,63 | 63,69 | 49,01 |
| | 2 | 44,84 | 41,84 | 50,76 | 44,84 | 50,76 | 42,91 |
| | 4 | 30,21 | 30,12 | 30,3 | 33 | 31,84 | 31,34 |
| | 8 | 23,8 | 24,21 | 25,51 | 27,77 | 24,15 | 24,93 |
| Filter Speed (MHz) | 1 | 6,38 | 6,22 | 7,23 | 6,58 | 7,96 | 6,13 |
| | 2 | 11,21 | 10,46 | 12,69 | 11,21 | 12,69 | 10,73 |
| | 4 | 15,11 | 15,06 | 15,15 | 16,50 | 15,92 | 15,67 |
| | 8 | 23,8 | 24,21 | 25,51 | 27,77 | 24,15 | 24,93 |
| Area (LEs) | 1 | 671 | 770 | 383 | 513 | 383 | 513 |
| | 2 | 837 | 915 | 470 | 595 | 470 | 595 |
| | 4 | 1169 | 1209 | 646 | 761 | 646 | 761 |
| | 8 | 1470 | 1377 | 811 | 840 | 811 | 865 |
| Area x Time | 1 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 |
| | 2 | 0,71 | 0,71 | 0,70 | 0,68 | 0,77 | 0,66 |
| | 4 | 0,74 | 0,65 | 0,80 | 0,59 | 0,84 | 0,58 |
| | 8 | 0,59 | 0,46 | 0,60 | 0,39 | 0,70 | 0,41 |

Table I. Implementation results.

All circuits exhibit a similar throughput (Fig. 12). The achieved sample frequency goes from 6MHz in the bit-serial version to 25MHz in the bit-parallel one.

The main differences among the design options can be highlighted only if small digit-sizes are considered. The filter clock rates for the digit-size 1, 2 and 4 bits are shown in Fig. 13. The canonical forms always achieve higher frequency than the inverted ones. It is a consequence of the absence of global lines in canonical structures. Furthermore, maximum values correspond to the anti-symmetrical and symmetrical canonical forms. That is because their occupation level is lower than it is in the other structure. On the other hand, if greater digit-sizes are considered, those differences disappear. The higher delay of these global lines is masked for dense FPGA occupations. In that case, even local wires can reach higher interconnection delay than global ones. The results also illustrate a well-known rule of actual FPGA implementations: the smaller the circuit, the faster the clock rate.

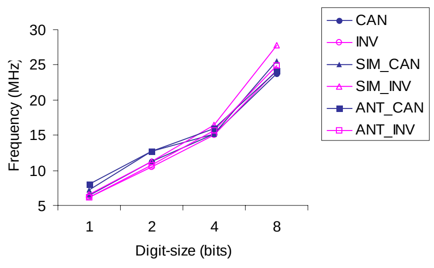


Fig.12. Throughput vs. digit-size.

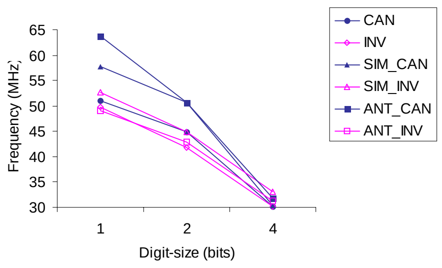


Fig.13. Filter clock-rates vs. digit-size.

The relative speedup in each circuit with respect to its bit-serial version is depicted in Fig. 14. The speed increment achieved with the bit-parallel versions goes from 3 times higher than the bit-serial one in the anti-symmetric canonical structure to 4 times higher in the symmetric inverted topology. In the abstract sense, it should be 8 times higher. This effect is the consequence of the influence of the routing process in the FPGA. In high digit-size circuits, a large set of paths exists, and each of them is a candidate to become the critical path after the implementation. The limited resources of interconnection in the FPGA make it difficult for the router to assign a low delay value to each of these paths. On the contrary, relatively few paths determine the delay in serial circuits.

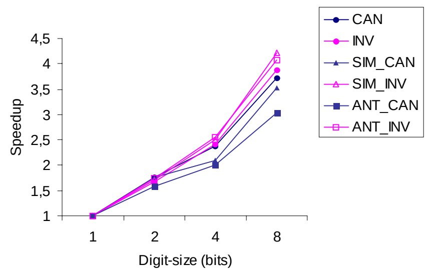


Fig. 14: Relative Speedup respect to N=1 operation.

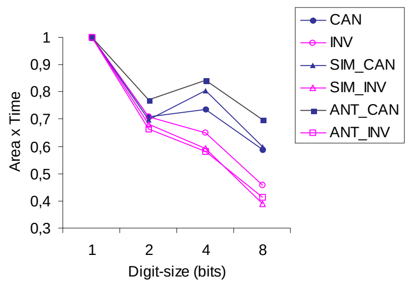


Fig.15. Area x Time vs. Digit-size

As it is expected, the area is reduced by nearly a half when symmetrical structures are used. In the small digit-size cases the canonical forms filters make use of less FPGA resources than the inverted form ones. That difference in the occupation of the chip is mainly caused by the registers used in the implementation of the delays. In the inverted form, the triple precision data has to be delayed, and the number of registers needed are given by (P-1)*3*(M-1). Meanwhile, in the canonical one, just the input data has to be delayed, resulting in a register count of 8*(M-1).

In Fig.15 the area-time product of the circuits for each digit-size is shown. The area is measured as the number of LEs, and the time is the inverse of the sample period or throughput. As can be seen the canonical forms are more efficient than the inverted ones for all digit-sizes. Considering only the canonical form filters, those with 2 bits digit-size have better area-time product than those with 1 and 4 bits digit-size have. If inverted forms are considered it does not happen in this way, they are more efficient according to digit-size increases.

## CONCLUSIONS

A study of full precision digit-serial FIR filters with programmable coefficients has been presented. Canonical and inverted form, and their symmetrical and anti-symmetrical special cases have been considered. The design methodology using each of these structures has been detailed and, finally, the results of their implementation in an EPF10K50GC403-3 Altera FPGA have been given.

The best results in area-time correspond to canonical form structures, being those with 2-bits digit-size the most efficient. Nevertheless, if symmetric or anti-symmetric structures are required, the inverted form offers less quantization error.

The throughput achieved lets the filters being used in applications where the sample rate range goes up to 7 MHz using the bit-serial circuits and up to 25MHz with bit-parallel one. The occupation obtained is relatively small for the gate density of current FPGAs.

All filters have been automatically implemented using the default parameters of the partitioning, placement, and routing tools. Thus, in critical applications, an important area reduction and speedup can be expected if some of these tasks are manually performed.

Future research includes the extension of this study to Xilinx chips, the power-figure measurement and a full characterisation of each design option at layout level.

## References

[1] R. Petersen and B. Hutchings, “An Asssesment of the Suitability of FPGA-based Systems for Use in DSPs”, in Lecture Notes in Computer Science, nº975, pp.293-302, Springer-Verlag, Berlin: 1995.

[2] Chi-Jui Chou, Satish Mohanakrishnan, and Joseph B. Evans, “FPGA Implementation of Digital Filters”, Proc. Int. Conf. Signal Proc. Appl. & Tech. (ICSPAT’93), 1993.

[3] J. Isoaho, J. Nousiainen and O. Vainio, “FPGA-implementable Digital Filters”, More FPGAs, Eds. W.R. Moore and W. Luk, Abindon EE&CS books, 1994.

[4] Joseph B. Evans, “Efficient FIR Filter Architectures Suitable for FPGA Implementation”, IEEE Trans. Circuits & Systems, July 1994.

[5] R.J.Andraka, “FIR filters fits in an FPGAs using a Bit-Serial approach”, 3rd PLD Conference, Mar 1993.

[6] J. Hancq, H. Leich, “Implementation of digital filters on reconfigurable Field-Programmable Gate Arrays”, Signal Processing VII: Theories and Applications, Eds. M.

Holt, C. Cowan, P. Grant, W. Sandham, 1994.

[7] L.E. Turner, P.J.W. Graumann, and S.G. Gibb, “Bit-serial FIR Filters with CSD Coefficients for FPGA”, in Lecture Notes in Computer Science, nº975, pp.310-320, Springer-Verlag, Berlin: 1995.

[8] Shousheng He, Mats Torkelson, “FPGA Implementation of FIR Filters Using Pipelined Bit-Serial Canonical Signed Digit Multipliers”, IEEE 1994 Custom Integrated Circuits Conference.

[9] H. Lee and G. Sobelman, "FPGA-Based FIR Filters Using Digit-Serial Arithmetic,", Proc. of the Tenth Annual IEEE International ASIC Conference and Exhibit, pp. 225-228, 1997.

[10] Les Mintzer, “FIR Filters with Field-Programmable Gate Arrays”, Journal of VLSI Signal Processing, vol. 6, pp. 119-127, 1993

[11] Xilinx, “The fastest filter in the west”, Xilinx Inc., http:\www.xilinx.com.

[12] “The role of the distributed aritmetic in FPGA-based signal processing”, Xilinx Inc., http:\www.xilinx.com.

[13] G.Goslin. A Guide to Using FPGAs for Application-Specific Digital Signal Processing Peformance. Xilinx Inc.

[14] G.Goslin, and and Bruce Newgard, “16-Tap, 8-bit FIR Filter Application Guide”, Xilinx Inc., http: \www.xilinx.com, 1994.

[15] Altera, “Implementing FIR Filters in FLEX Devices”, A-AN-073-01, www.altera.com.

[16] Altera, “FIR Filters”,A-FS-01-01, http:\ www.altera.com, 1996.

[17] S.G. Smith and P.B. Denyer, Serial Data Computation, Kluwer Academic, Boston, MA, 1988.

[18] R. Harley and P.Corbet, “Digit-serial processing techniques”, IEEE Trans. On Circuits and Systems, Vol. 37, no. 6, pp. 707-719, June 1990.

[19] K.K. Parhi and C. Wang, “Digit-serial DSP architectures”, in Proc. of Int. Conf. On Application Specific Array Processors, pp. 341-351, September 1990.

[20] K.K. Parhi, “A systematic approach for design of digit-serial signal processing architectures”, IEEE Trans. Circuits and Systems, Vol. 38, pp.358-375, April 1991.

[21] R.I. Hartley and K.K. Parhi, Digit-Serial Computation, Kluwer Academic, Boston, MA, 1995.

[22] P. Denyer and D. Renshaw, VLSI SIGNAL PROCESSING: A Bit-Serial Approach, Addison-Wesley, 1985.

[23] Lars Wanhammar, “DSP Integrated Circuits”, Linköping University, 1997.

[24] ALTERA, “Data Book”, 1996.