IEEE International Conference on Electronics, Circuits and Systems, 1998.
Volumen 2, IEEE, 1998. 191-194.

DOI: http://dx.doi.org/10.1109/ICECS.1998.814860

Design and FPGA implementation of Digit-Serial FIR filters

Javier Valls*, Marcos Martínez. Peiró*, Trini Sansaloni* and Eduardo Boemo**

* Departamento de Ingeniería Electrónica, Universidad Politécnica de Valencia,
Camino de Vera s/n, 46071 Valencia, Spain. E-mail: {jvalls, mpeiro, tmsansal}@eln.upv.es,
** Escuela Técnica Superior de Ingeniería Informática, Universidad Autónoma de Madrid,
Ctra. Colmenar Km.15, 28049 Madrid, Spain. E-mail: eduardo.boemo@ii.uam.es

Abstract-

In this paper the design of a family of digit-serial 8th-order FIR filters with programmable coefficients is presented. Both input data and coefficient size are 8 bits, but every filter of the family allows the computation with full precision of the intermediate data. The output data is truncated to 8 bits. The design of both, the digit-serial multiple precision multiply-and-accumulate and the digit-serial multiple-to-single-precision converter, is detailed. All filters were implemented using an ALTERA FPGA being useful in applications with sample rate range from 5 to 22 MHz.

1.- Introduction

The high-density of current FPGAs has opened a new field of application: the design of single-chip systems with an embedded custom DSPs (CDSPs). The advantages of this approach are multiple: extra component are avoided, the off-chip connections are reduced and finally, the DSP core can be simplified and optimized for the application, considering aspects like the required data rate and precision, or the bit-level peculiarities of the coefficient. In several cases it is senseless to use conventional bit-parallel circuits: their implementations 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 implement 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.

The organization of this paper is as follow. In the next section the digit-serial architectures are briefly exposed; the basic digit-serial adder cell is explained and some rudiments of the digit-serial processors are commented upon. In section 3, the multipliers used to perform the filters are presented and the way they give the results of the computation is depicted. In section 4, the design of a set of FIR filter are detailed; emphasis on the design of the digit-serial full precision accumulator and in the multiple-to-single-precision circuitry is done. In section 5, the results of the implementation of the filter in EPF10K50 are given and finally, the conclusions are presented.

2.- 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 least significant digit first. A complete word is processed in P=W/N clock cycles and consecutive words follow each other without a break. 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 when one word ends and the next word begins. A more detailed explanation of this kind of architectures can be found in [1], [2], [3], [4], [5].

A family of digit-serial architectures can be designed by using different digit-sizes. Each element of this family will have a distinct size and throughput. Thus, it is possible to choose a digit-size to achieve the speed of the application with at minimum cost in terms of area.

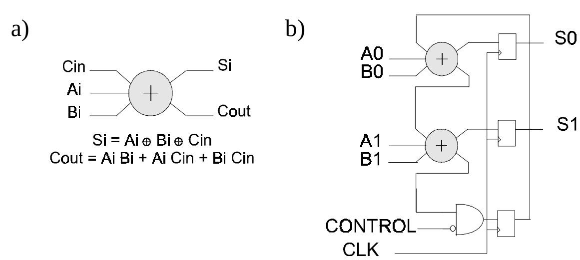

Fig.1: a) Full adder; b) digit-serial adder with N=2.

2.1- Digit-Serial Adder

In Fig. 1.b. the digit-serial adder with digit-size N=2 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 digit-adder. The carry-in to the least significant is a zero. A ripple carry addition of the two bits is computed and both the two least significant bits of the addition and a carry-out are produced. The carry-out is delayed one clock cycle to be added to the next digit. In the fourth clock cycle the last (or more significant) digit of the two words are presented to the digit-adder 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 word.

2.2.- Digit-serial processors

The generic scheme of a digit-serial processor is shown in Fig.2. It needs a parallel-serial and serial-parallel converters to process in digit format and to give the result in parallel format. To implement a digit-serial processor, the digit-serial operators are connected together following the data-flow algorithm of the application. Every digit-serial operator has a latency, so that no two consecutive digit-serial operators can be cascaded together in the same clock cycle. To synchronise the digit-serial operators, a number of P different control signals can be required. These control signals are the P delayed versions of the CONTROL signal. Each operator can use one or more of these control signals.

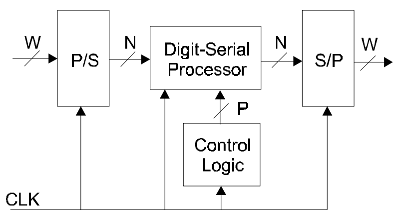

Fig. 2. Generic scheme of a digit-serial processor

3.- 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. The design of this kind of digit-serial circuits has been highly detailed in [5], [2]. It is shown in Fig.3, for 4 bits coefficient-size and digit-size N=2.

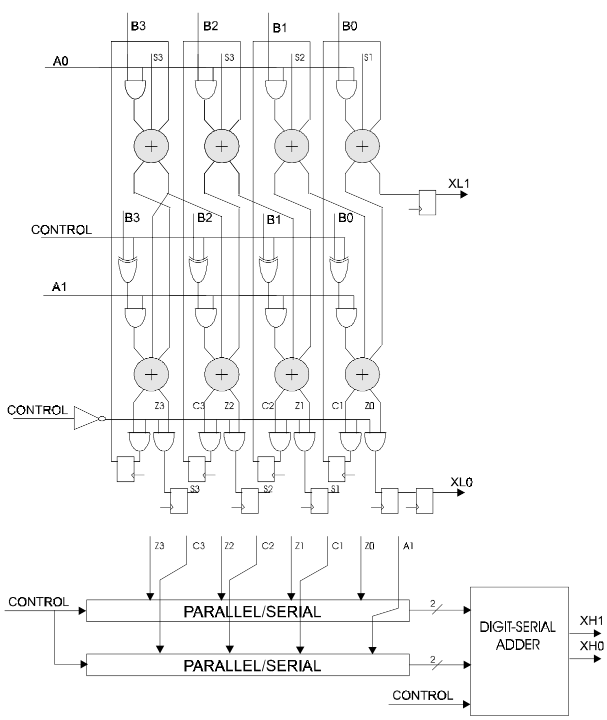

Fig.3. Digit-serial array multiplier with digit-size N=2.

This multiplier computes 2xAxB (by inserting a 0 in the least significant bit position and ignoring the most significant bit) and it produces an standard precision result [6]. If data A and coefficient B are W bit-word size, the resulting low word of the product (W bit size) is outputted with a latency of 1 with respect to A; meanwhile, the high word (W bit size) is produced from the digit-serial adder with a latency of W+1 with respect to A, both in digit-serial format. This multiplier computes at the same time, the low order product word of a input-word and the high order product word of the previous input-word.

4.- Digit-Serial FIR filter

A family of digit-serial FIR filters with programmable coefficients has been designed. The transposed direct form structure has been chosen (Fig.4). Each filter is 8th-order and keeps full precision along the whole circuit. The size of the input data and coefficient are W=8 bits, and resulting output is truncated in order to pass it to single precision. The digit-serial multiplier of Fig.3 has been used to perform the multiply-accumulate operator.

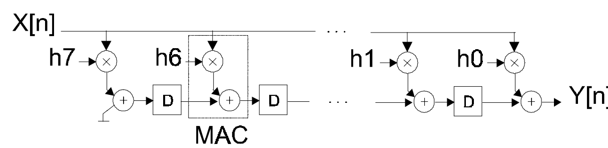

Fig.4. Transposed direct form FIR filter structure.

4.1.- Multiple precision digit-serial MAC

In this FIR filters family, a full precision is guaranteed if 19 bits word length is adopted ($W_{in\text{-}data}+W_{coefficient}+log_2M=19$, where M is the number of stages of the filter).

In order to design the accumulator several modifications of the digit-serial adder cell are necessary. The digit-serial accumulator cell allows the carry-out to propagate to the next cell as well as to feed the carry-out from the previous cell into carry-in of this cell. During the fist digit of every word the CONTROL signal is high and the latched carry-out of the previous cell is fed to the least significant full adder of the cell. This new cell is depicted in Fig. 5. This operator has a latency of 1.

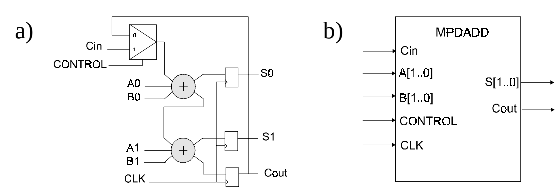

Fig.5: Basic digit-serial accumulator cell: a) scheme, b) symbol of the cell.

The multiple precision accumulator is constructed by connecting several of these cells and adding some circuitry which allows it to keep the sign extension to higher order bits. In our case, to guarantee 19 bits of precision, only three cells are needed. The triple precision digit-serial accumulator is shown in Fig.6. 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 extension is kept by latching the sign bit of the double precision word and feeding it to the input of one of the digits inputs. It happens during the last digit of every word, when CONTR_2 signal is high.

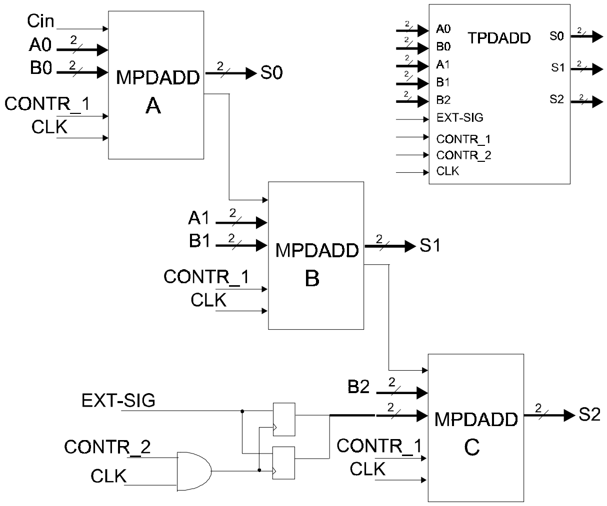

Fig.6. Triple precision digit-serial accumulator with digit-size N=2: scheme and symbol.

The multiply-and-accumulate cell is illustrated in Fig.7. The multiplier low-word output (XL) is connected to a digit input of the least significant cell of the digit-serial accumulator. The multiplier high-word output (XH) is connected to a digit input of the cell B. The most significant bit of XH is connected to the EXT_SIGN input of cell C. The other digit inputs are connected to the one-sample-delayed outputs of the previous digit-serial MAC.

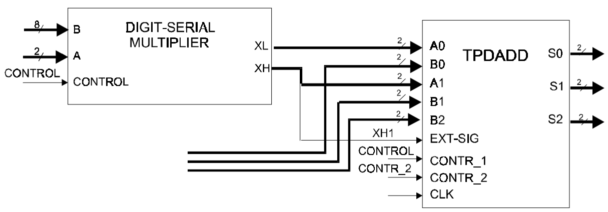

Fig.7. Digit-serial MAC

4.2.- Multiple precision to single precision block

The full precision data has to be truncated to single precision before going into the final serial-parallel converter. The triple precision digit-serial accumulator is able to generate 24 bits of precision. Nevertheless, the only useful bits for us are from S11 bit (least significant bit) to S19 (most significant bit). Therefore, to feed these 8 bit into the serial-parallel converter they have to arrive with the right format. The inputs of this block are the 3xN wires which come from the previous MAC and the outputs are N wires to be connected to the serial-parallel converter.

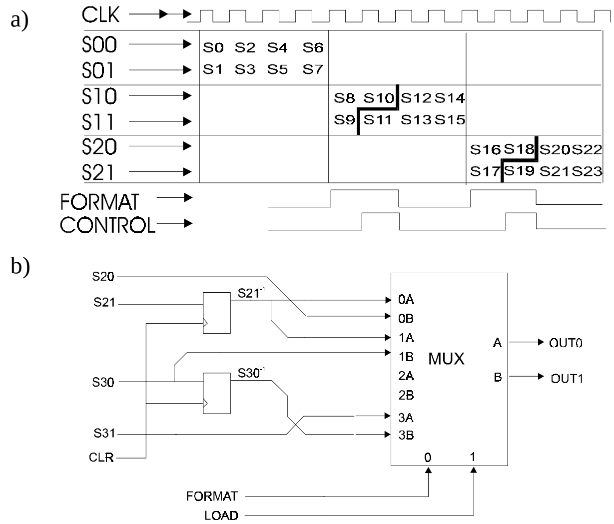

Fig.8. Format 3 to 1 block: a) digit input sequence; b)circuit.

Considering the case of a digit size of N=2, the digit input sequence to the format 3 to 1 block and the circuit to format the single precision output is shown in Fig.8. Two control signals are needed to format the single precision output: first, the CONTROL signal, that signal which is high during the first digit of every word, and second, a new signal called FORMAT which is high during the last and first digit of every word. The format block circuit is composed by a four two-input multiplexor whose selection sequence is 0, 0, 1, 2. The digit-serial output sequence ([OUT1, OUT0]) will be [S20,S21^{-1}], [S20,S21^{-1}], [S30,S21^{-1}], [S30^{-1},S31].

4.3.- Scheduling considerations

The timing of signals through the filter is shown in Fig.9. Each of the digit-serial operators considered in this paper has a latency of 1. 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 W/N-1 cascade block of registers, each one with N registers in parallel.

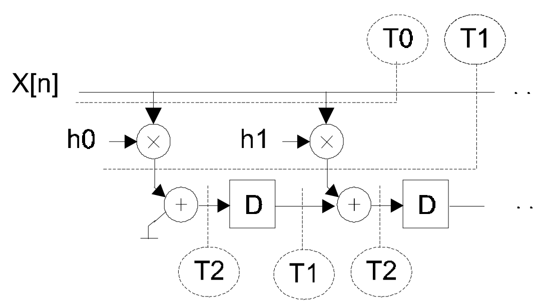

Fig.9. The timing of signals through the filter.

5.- FPGA implementation

A family of digit-serial FIR filters designed has been implemented in ALTERA FPGA. The device used is the EPF10K50GC403-3, [7]. The automatic placement and routing compilation option has been used.

TABLE I: OCCUPATION OF THE DIGIT-SERIAL OPERATORS IN LEs

Digit-Serial Operators	Digit size (N)			
	1	2	4	8
Serial-Parallel Converter	8	8	8	8
Parallel-Serial Converter	8	8	8	8
Array Multiplier	64	80	112	140
Triple precision Accumulator	11	15	27	43
Formatter 3 to 1	1	5	4	8
Delay	7	6	4	0

The occupation of each operator (in the number of LEs) is shown in Table I. With this data, the occupation of any digital filter in chip selected can be estimated. Fig. 10 shows the results of that estimation to FIR filters whose order is up to 20. The horizontal lines represent the maximum capacity of each chip in FLEX 10K family.

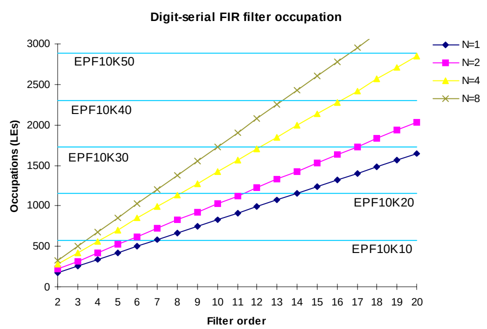

Fig. 10. Occupation of the digit-serial FIR filters

The speed and occupation of each digit-serial 8th-order digit-serial FIR filter is represented in Fig.11 and its efficiency (area x time) is shown in Fig.12. The throughput range of the different versions of the filter is from 5 to 22 MHz. From the bit-serial circuit to the bit-parallel one an increment in speed close to 4 times higher is achieved with only doubling the size of the filter. The circuits with digit-size N=2 and 4 have similar area-time product.

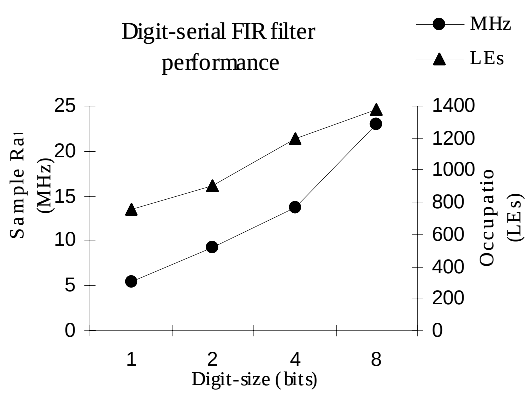

Fig.11. Sample rate and occupation of the family of digit-serial 8th-order FIR filters.

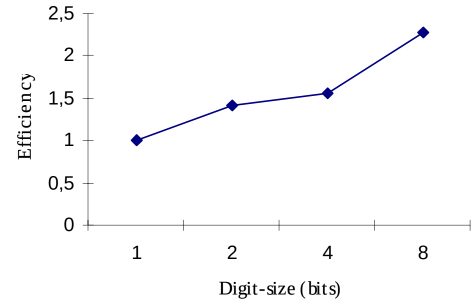

Fig. 12. Efficiency of the family of digit-serial 8th-order FIR filters.

6.- Conclusions

The design of digit-serial 8th-order FIR filters with programmable coefficients has been presented. Every filter of the family allows the computation with full precision of the intermediate data. The output data is truncated to 8 bits. The design of both, the digit-serial multiple precision multiply-and-accumulate and the digit-serial multiple-to-single-precision converter, has been detailed. Each filter has been implemented with a latency of 3P and with only three control lines to synchronises the whole circuit.

The throughput achieved by the filters lets them be used in applications where the sample rate range goes from 5 to 22 MHz. The occupation obtained allows the designer to choose different chips for the implementation; the EPF10K20 would be enough to fit the two lower digit-size versions of the filter. The circuits with digit-size N=2 and 4 have similar area-time product.

REFERENCES

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

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

[3] 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.

[4] 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.

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

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

[7] ALTERA, "*Data Book*", 1996.