Functional analysis of DSP blocks in FPGA chips for application in TESLA LLRF system

Krzysztof T. Pozniak, Tomasz Czarski, Ryszard S. Romaniuk
Institute of Electronic Systems, WUT, Nowowiejska 15/19, 00-665 Warsaw, Poland

ABSTRACT

The paper contains the analysis of the application possibilities offered by the new generation of the FPGA chips. The new generation of the FPGA chips contain DSP blocks. The new functionalities are well suited for the application in the TESLA LLRF cavity simulation and control system (SIMCON). A debate on the programming methods of the new chips and the algorithm parameterization was presented. The aim of the, FPGA chip based, system analysis is the optimal chip usage to increase the maximum frequency at which the system can work efficiently, and the optimal usage of the accessible chip resources (DSP blocks). The exemplary results for a few practical calculated implementations were presented and analyzed. The implementations included some basic DSP operations performed in the FPGA chips of Altera and Xilinx. There were compared the results for a few different chips. The TESLA superconducting cavity simulator was efficiently implemented. The results were presented for the first time, for the pure FPGA/VHDL solution. The realization costs were debated in the dependence of given system parameters and the applied type of the FPGA chip

Keywords: SC accelerating cavity simulation and control, TESLA, FPGA, DSP, VHDL, Altera, Xilinx, Virtex, Stratix.

1. INTRODUCTION

The calculation resources needed to realize the real-time LLRF control system for the resonant, superconducting cavity modules of the TESLA accelerator are quite big [2]. They increase quickly with the need to introduce the new control system with some additional functionalities like the efficient exception handling. Fig.1 presents a general functional diagram of the TESLA LLRF control system, as it is understood today. The diagram includes the digital blocks, which

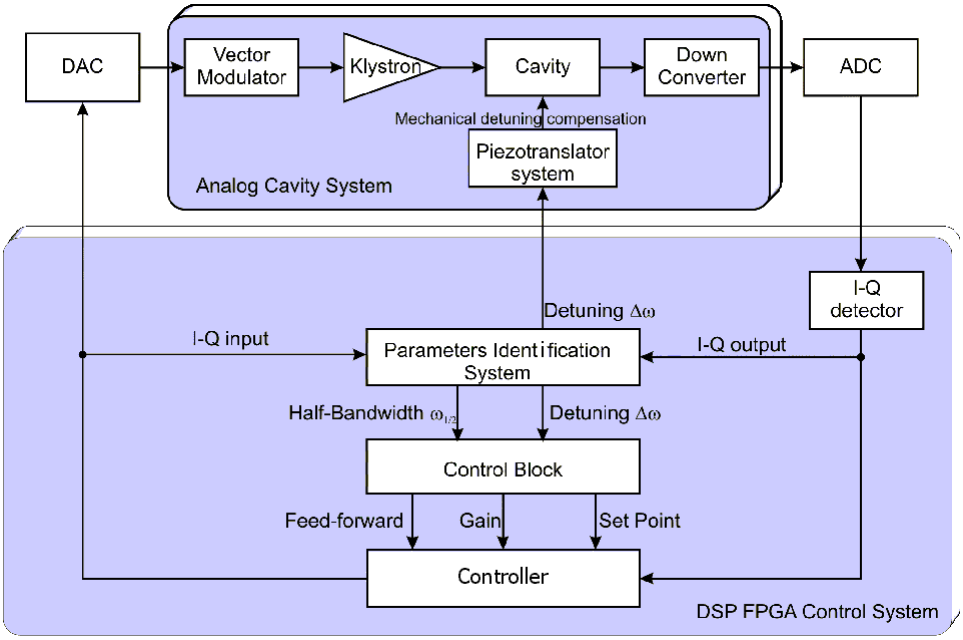
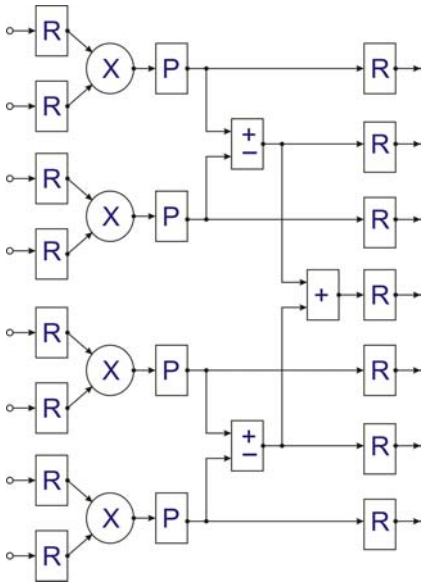


Fig. 1. Functional block diagram of the TESLA LLRF Cavity Simulation and Control System (SIMCON).

realization is performed in the FPGA chips equipped in the DSP capabilities [1]. The blocks are realized in the form of the complex numerical algorithms. The results of the analysis, presented in this paper, show that the frequency, realized for the digital algorithms in the FPGA chips of the new generation, allow for the real time processaction of the TESLA cavity simulator and controller. The consequences and further possibilities to use these new, FPGA-based, DSP

capabilities are precisely described in this work. The new series of the FPGA chips, that have recently been provided to the market by the major vendors, include more and more DSP resources. This seems to be a constant trend, thus, promising for the increased digital DSP system availability, increased reliability and calculation resources, decreased prices, requested system redundancy, etc. [4,5].



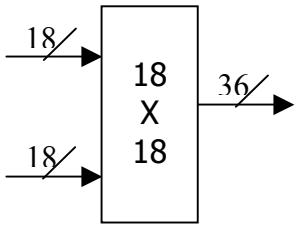
Stratix Device	DSP Blocks	9 ×9 Multipliers	18 ×18 Multipliers	36 ×36 Multipliers
EP1S10	6	48	24	6
EP1S20	10	80	40	10
EP1S25	10	80	40	10
EP1S30	12	96	48	12
EP1S40	14	112	56	14
EP1S60	18	144	72	18
StratixGX Device	**DSP Blocks**	**9 ×9 Multipliers**	**18 ×18 Multipliers**	**36 ×36 Multipliers**
EP1SGX10C	6	48	24	6
EP1SGX10D	6	48	24	6
EP1SGX25C	10	80	40	10
EP1SGX25D	10	80	40	10
EP1SGX25F	10	80	40	10
EP1SGX40D	14	112	56	14
EP1SGX40G	14	112	56	14

Fig 2. The structure of the DSP blocks realized by the Altera inside the Stratix and StratixGX series of FPGA chips. Symbols: R-optional D-registers layer, P-optional layer of pipeline registers. The basic multiplying circuits, arranged in a 9x9 matrix, are signed as X, summing circuits signed by +, summing and subtracting circuits signed by ±. The table contains the available resources as function of the kind of used FPGA chip.

The Altera has recently started to offer the chips of the Stratix and StratixGX series with the internal specialized DSP blocks. These blocks have the structure shown in fig. 2 [4]. All components of the blocks have been realized in a form of the dedicated hardware structures. The structures of the intermediate registers have been applied. It enables realization of the pipeline processes of the throughput up to the 300 MIPS.

The Xilinx has recently released to the market the following series of the FPGA chips: Virtex II-Pro, Virtex II and Spartan III. These chips are equipped in the 18x18 bits multiplying circuits. Fig 3. contains a description and the available DSP resources of these chips [5].

Virtex II - Pro Device	18 ×18 Multipliers
XC2VPX20	88
XC2VPX70	308



Virtex II Device	18 ×18 Multipliers
XC2V40	4
XC2V80	8
XC2V250	24
XC2V500	32
	40
XC2V1500	48
XC2V2000	56
XC2V3000	96
XC2V4000	120
XC2V6000	144
XC2V8000	168

Spartan III Device	18 ×18 Multipliers
XC3S50	4
XC3S200	12
XC3S400	16
XC3S1000	24
XC3S1500	32
XC3S2000	40
XC3S4000	96
XC3S5000	104

Fig. 3. The structure of the DSP blocks realized by Xilinx in the FPGA series Virtex II-Pro, Vitrex II and Spartan III. It bases on the 18x18 bits multiplying components. The tables contain available resources as function of the kind of the FPGA chip.

The elementary multiplying circuits may be combined into the groups, in order to perform any operation on the larger number of bits. For example, the 32-bit words require the usage of four elementary multiplying blocks. The appropriate construction of the cascade cross-connections inside the FPGA chips allows to obtain sufficiently fast implementations.

2. PARAMETERIZED FUNCTIONAL STRUCTURE

The minimization of the usage of the DSP resources by the structurally complex calculation algorithm, for a defined system speed, requires to choose the proper scaling conditions for the involved mathematical operations. The optimization process should take into consideration the possibility to use the parameterized functional structure of the basic mathematical operators. This situation was illustrated in fig. 4

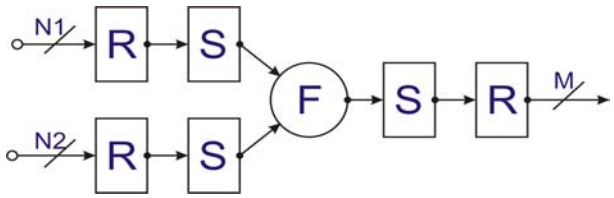


Fig.4 General functional structure of the parameterized two-argument arithmetic operator. Symbols: „F" –the operators, respectively, of summation, subtraction, and multiplication, R- optional D-registers, S- optional scaling blocks, „N1", „N2", „M" – the widths of data buses expressed in bits;

3	011
2	010
1	001
0	000
-1	111
-2	110
-3	101
-4	100

Fig. 5. U2 type coding for N=3

The parameterization conditions include the possibility to choose the proper input and output resolutions, as well as the position of the fixed point. The writing of the integer values, for the U2 coding, and for N-bits is confined to the following range of the values: from -2^{N-1} to $2^{N-1}-1$. The format of the writing bases on the method of coding, what is shown exemplarily in fig. 5, for N=3. It is a natural binary code (NB) shifted and convoluted against the value of "0". The arithmetic operations are performed only on the integer values, from the point of view of the implementation.

The definition of the situation of the point is relative. It requires the introduction of the additional scaling blocks, respectively, before and after the realization of a particular function. The operation of the summation (or subtraction) with a fixed point, which is defined by the parameter *P1* for the integer number *A* and by the parameter *P2* for the integer number *B,* is expressed as follows:

$$A_{(P1)} \pm B_{(P2)} \underset{df}{=} A * 2^{-p1} \pm B * 2^{-p2}$$

The process of reducing the problem to the operations carried out only on the integer numbers requires the unification of the position of point through indication of the bigger value from *P=max(P1, P2)*. Such a kind of the "upward" unification does not lead to the introduction of the numerical errors. These operations are realized by the scaling blocks of S_A and S_B and agree with the following relations:

$$A * 2^{-p1} \pm B * 2^{-p2} = A * 2^{-p1+p-p} \pm B * 2^{-p2+p-p} = (A * 2^{-p1+p} \pm B * 2^{-p2+p}) * 2^{-p} =$$

$$= (A * S_A \pm B * S_B) * 2^{-p}$$

It is to note, that for such a defined *P* parameter, only a single scaling block will be activated. The scaling blocks are not necessary only in the case of the fulfilled condition *P1=P2*. The loss-less scaling requires the increase in the number of bits of the arithmetical operations by the factor of $|P_1 - P_2|$. The aim of the scaling, performed after the arithmetical operations, is the eventual setting of the position of the point. One has to take into account the possibility of appearing of two kinds of the errors: 1) the resolution error – by the scaling of the values, which are less than the unity, and 2) the saturation error – when the scaling operation requires the multiplication by the values greater than the unity, and the given number of bits *M* does not cover the range of the operation result.

The input scaling procedure is not required for the operation of multiplication, what is shown below:

$$A_{(P1)} * B_{(P2)} = A * 2^{-p1} * B * 2^{-p2} = (A * B) * 2^{-p1-p2}$$

However, for most of the practical cases it is necessary to perform cautiously the input scaling, especially when the realized algorithm keeps the position of the point in the same place, all the time and for all the mathematical operations.

A completely new system solution for the parameterized DSP operator was suggested in this work. The solution, written in the VHDL, in the form of a behavioral description, takes into account the "R" registers and the scaling blocks "S". This enables the possibility to implement the DSP operators in the various FPGA chips. Additionally, the analysis of the cost of implementation was performed as well as the studies on the maximum available work frequency for the synchronous regime. The investigations have been carried out with the „*Leonardo Spectrum 2002d*" compiler [6] and the packets of *ISE 5.1*. [5] and *Quartus 3* [4].

The exemplary results of the analysis, for the operations of the summation and multiplication were presented in tables 1 and 2. Two comparable, big (in the sense of the number of the gates and the number of the DSP blocks) series of the FPGA chips have been subject to the analysis. As above, the analysis concerned two vendors, Altera (table 1) and Xilix (table 2). A simplifying assumption was taken of the identical number of the bits in the processed input and output words. The latter value is expressed by the DATA WIDTH parameter and assumes the range of: from 8 to 64 bits.

ALTERA	data width	data registers	SUMATOR		MULTIPLIER		
			LCELL number	fmax [MHz]	LCELL number	DSP number	fmax [MHz]
STRATIX EP1S10B672C-6	8	16	32	331	14	1	152
STRATIX EP1S10B672C-6	16	32	64	296	28	1	124
STRATIX EP1S10B672C-6	24	48	96	283	40	4	89
STRATIX EP1S10B672C-6	32	64	128	248	54	4	87
STRATIX EP1S10B672C-6	40	80	160	235	308	9	70
STRATIX EP1S10B672C-6	48	96	192	229	385	9	64
STRATIX EP1S10B672C-6	56	112	224	215	650	16	49
STRATIX EP1S10B672C-6	64	128	256	205	787	16	45

Table 1. Summary of the implementation result parameters for the summation and multiplication operations for the ALTERA STRATIX EP1S10B672C-6 FPGA chip.

XILINX	data width	data registers	SUMATOR		MULTIPLIER		
			LCELL number	fmax [MHz]	LCELL number	DSP number	fmax [MHz]
VIRTEX-II 2V250fg456-5	8	16	34	149	31	1	108
VIRTEX-II 2V250fg456-5	16	32	66	138	59	1	85
VIRTEX-II 2V250fg456-5	24	48	98	130	157	4	47
VIRTEX-II 2V250fg456-5	32	64	130	113	185	4	42
VIRTEX-II 2V250fg456-5	40	80	162	113	335	9	35
VIRTEX-II 2V250fg456-5	48	96	194	103	491	9	34
VIRTEX-II 2V250fg456-5	56	112	226	100	733	16	28
VIRTEX-II 2V250fg456-5	64	128	258	94	834	16	28

Table 2. Summary of the implementation result parameters for the summation and multiplication operations for the XILINX VIRTEX-II 2V250fg456-5 FPGA chips.

Fig. 6 presents the direct comparison of implementation of the summation operation for the both analyzed chips. The analysis was done as the function of the number of bits in the operation. The dependence of the number of the LCELL elements as a function of the number of operation bits is linear and very similar for the both compared, DSP block equipped, FPGA chips (left panel of fig.6). However, the fundamental logical structures of the FPGA chips from the Altera and the Xilinx are essentially different. Thus, the presented results reflect the reality only in an approximated manner.

The usage of the LCELL elements is relatively small in reference to the available resources. The conclusion is that the cost of realization of a single summation (or subtraction) operation is very low in these chips. The analysis of the work frequency, for the compared chips, (right panel of fig.6) shows that, even for the very large numbers (in the sense of bits), for example for N=64, the overall speed is very high. There are, however, much bigger differences in this case, than previously for the number of the LCELL elements. The maximum frequency is around 100MHz. It is important, that in the both cases, it is bigger than 60MHz, what is more than satisfactory, for the solution realized now of the TESLA LLRF simulation and control (SIMCON) system. It is to note, however, that in the carried out performance tests for the maximum frequency, over two times better results were obtained for the STRATIX FPGA chips over the VIRTEX FPGA chips.

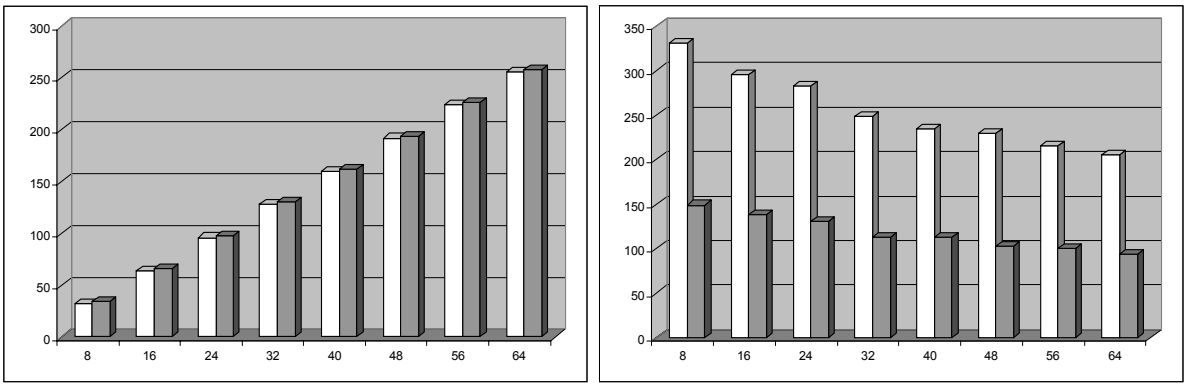


Fig. 6. Comparison between the number of the used logical cells (a diagram in the left panel) and maximal work frequency in MHz (right panel) for the operation of summation, as the function of the number of operation bits. White bars are for the STRATIX EP1S10B672C-6 FPGA chip. Grey bars are for the VIRTEX-II 2V250fg456-5 FPGA chip.

Fig. 7 contains the comparison of implementations for the operation of multiplication as a function of the number of operation bits. There is a considerable difference in the number of the required LCELL components at the multiplication operation for the compared FPGA chips of Stratix (Altera) and Virtex (Xilinx). The DSP blocks, in the Stratix chip create essentially an autonomous and configurable hardware unity. They do not require additional chip resources for the configuration purposes. On the contrary, the multiplication circuits of the Virtex chip are combined in the larger groups with the aid of the internal basic logical blocks. Thus, the requirement of the Virtex chip for the

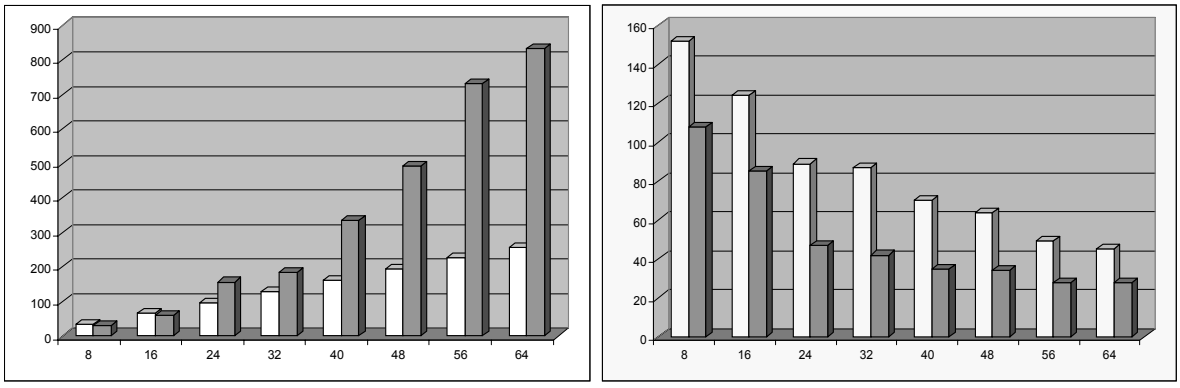


Fig.7. Comparizon between the number of used logical components LCELL (left diagram) and maximal work frequency in MHz (right panel) for the operation of multiplication as the function of the number of operation bits. White bars are for the STRATIX EP1S10B672C-6 FPGA chip and grey bars are for VIRTEX-II 2V250fg456-5 FPGA chip.

resources increases exponentially as the function of the number of the bits for the operation of multiplication. The elementary multiplication structures of the both chips differ essentially, thus, the direct comparison of the performance results and the presented numbers have only a confined (rough estimation) meaning. The analysis of the work frequency

shows, that even for the large operations, in the sense of the number of the used bits, i.e. for N=64, the Stratix may work faster than the Virtex. The confining frequency of the Stratix is more than 40MHz, while for the Virtex is approximately two times smaller. For N=64, the maximum frequency of the Virtex is around 25MHz. However, in both cases, the required bandwidth is sufficient to realize the assumed numerical algorithms of the digital models and hardware control of the TESLA cavity SIMCON (LLRF system).

3. IMPLEMENTATION OF THE DSP PIPELINE PROCESS

The analysis of the maximum reachable frequencies by the arithmetic operators for the addition and multiplication, presented in the previous chapter, shows that the most critical are implementations of the multiplications at the large number of bits. This paragraph shows the possibility to use the pipeline work regime for a very big multiplication circuit. The large multiplication circuit consists, in the pipeline case, of many elementary blocks of the pre-set dimension of *(K+1)* bits, where the additional bit over *K* is a bit of the sign. The dimensional factor is equal to *K=8* for the DSP block structure of the STRATIX chip. The dimensional factor is equal to *K=17* for the multiplier structure of the VIRTEX chip. For such assumed data, the multiplication process may be written, for the *L* elementary multiplying circuits, in the following form:

$$A * B = (K^{L-1}a_{L-1} + K^{L-2}a_{L-2} + a_0) * (K^{L-1}b_{L-1} + K^{L-2}b_{L-2} + b_0) = \sum_{i=1}^{L} \sum_{j=1}^{L} (K^{L-i}a_{L-i} * K^{L-j}a_{L-j})$$

where: the value of *L* is calculated as: *CELL(N-1)/K*.

As the result of the above calculations, a matrix of partial multiplications is generated. The system of the *K* multipliers indicate the respective shifts of the bits of the results. For example, table 3 shows the partition of the input data word to the four parts *(L=4)*, which leads to the usage of L^2 =16 operations:

	K^3*a_3	K^2*a_2	K^1*a_1	K^0*a_0
K^3*b_3	$K^6*a_3*b_3$	$K^5*a_2*b_3$	$K^4*a_1*b_3$	$K^3*a_0*b_3$
K^2*b_2	$K^5*a_3*b_2$	$K^4*a_2*b_2$	$K^3*a_1*b_2$	$K^2*a_0*b_2$
K^1*b_1	$K^4*a_3*b_1$	$K^3*a_2*b_1$	$K^2*a_1*b_1$	$K^1*a_0*b_1$
K^0*b_0	$K^3*a_3*b_0$	$K^2*a_2*b_0$	$K^1*a_1*b_0$	$K^0*a_0*b_0$

Table 3. Exemplary collection of the multiplying coefficients for the input data word partitioned to four parts *L=4*

Grouping of the multiplying coefficients, according to their powers, allows to calculate the partial sums and shift these sums accordingly to the proper positions determined by the values of K^X. This process was shown in table 4, for the same example of *L=4*, as in table 3.

K^6	a_3*b_3	a_3*b_2 $+a_2*b_3$					
K^5			$a_3*b_1+ a_2*b_2$ $+ a_1*b_3$				
K^4				$a_3*b_0+a_2*b_1$ $+a_1*b_2+a_0*b_3$			
K^3					$a_2*b_0+a_1*b_1$ $+ a_0*b_2$		
K^2						a_1*b_0 $+ a_0*b_1$	
K^1							a_0*b_0
K^0							

Table 4. The collection of the partial multiplying factors for the exemplary partition *L=4*

The partial calculations of the large multiplication process are as follows. The first summation groups the initial values of the partial sums for, respectively, odd and even powers of the *K* coefficient. On the basis of the result, the eventual sum is calculated. The absolute result of the multiplication is obtained. The sign is changed to the minus one, in the case of different signs of the input data words. The example of the pipeline multiplication process, as it proceeds in the time, is presented in fig.8. A number of the test implementations have been done, first of all for the VIRTEX chip, for which the multiplication process is slower. The obtained frequencies for the multiplication circuit, for the largest word width N=64 bits, was f=47MHz. This result, for the elementary multiplication, and for the same number of multiplication circuits, is nearly two times better than in the previous case (compare with table 2). The frequency of 29MHz is obtained for the increased number of the input bits to N=128.

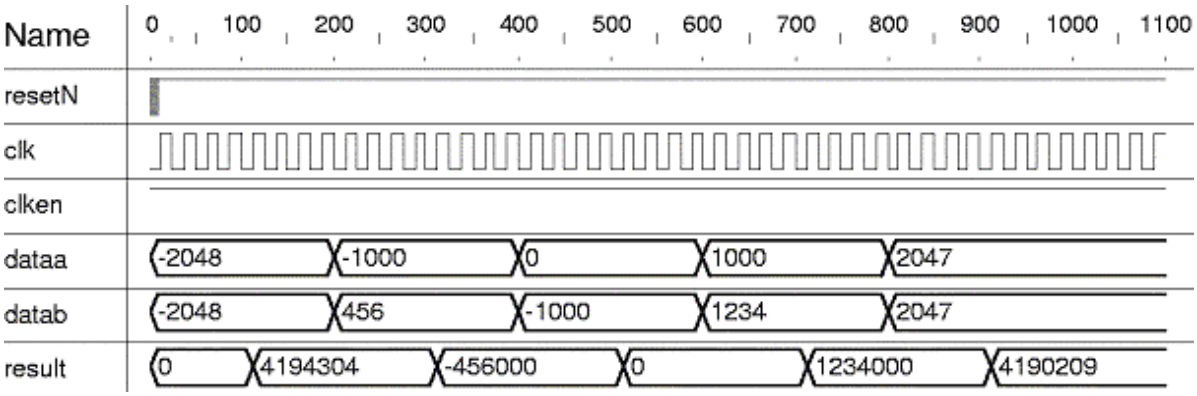


Fig. 8. Exemplary result of the activity of pipeline multiplication block, for the 24-bit word of the result and 5 clock periods of the system latency.

4. OPTIMISATION USAGE OF THE DSP RESOURCES

The optimal usage of the DSP resources may be realized additionally through the multiple employment of the same arithmetic components. This possibility concerns the multiplication circuits, which available resources were presented in the examples in figs.2 and 3. The application of such a model is possible due to the full synchronous realization of the mathematical models for the final processing algorithms. Assuming, for the sake of simplification, that there are N different mathematical resources, and there are T clock periods available to realize the operation for a single step of the algorithm, one obtains $M=N*T$ multiplexed mathematical resources. The exemplary solution of the pipeline summation of two vectors is presented in fig.9.

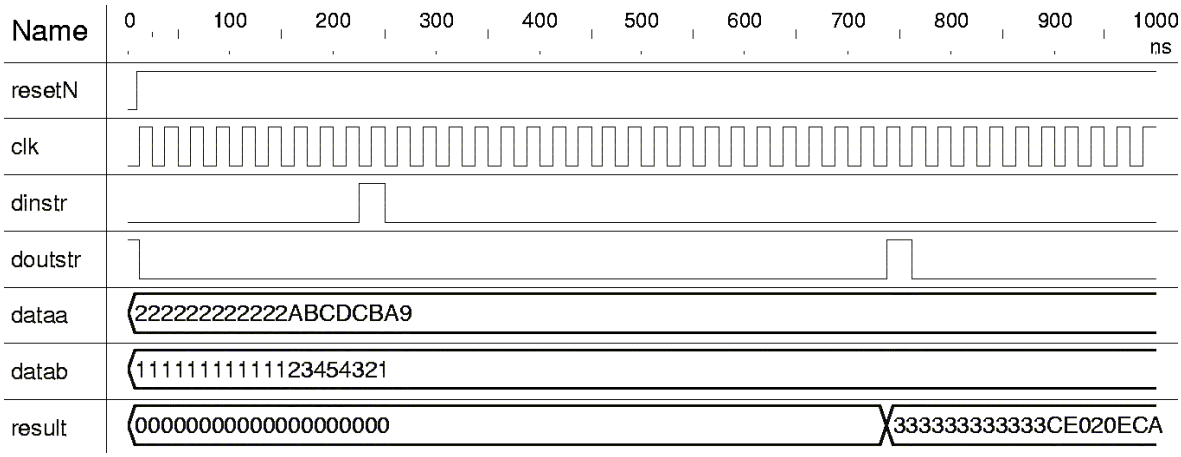


Fig. 9. An example of the realization of the pipeline summation of two vectors. Each vector contains 20 numbers, each number of 4-bits long. There was used only a single adding circuit. The circuit performs only a single adding operation per a single system clock period. The starting of the process is realized by the *dinstr* signal. The stopping of the process and outputting of the data is realized by the *doutstr* signal.

All the debated solutions use the strobe signals. Due to this, the pipeline calculation elements are triggered in the series, in the cascade structures. Additionally, in the parallel system, one may carry out the automatic synchronization of the parallel data streams reaching the block with the different latencies.

5. EXAMPLE OF IMPLEMENTATION OF TESLA CAVITY SIMULATOR USING PARAMETERISED DSP BLOCKS

This chapter presents an example of the usage of the FPGA/DSP technology to realize a behavioral algorithm for the superconducting resonant cavity of the TESLA linear accelerator. The discrete processing of the cavity algorithm has been developed for the digital implementation of the cavity model. The continuous algorithm of the cavity behavior is estimated by the arithmetic procedure realized iteratively in the finite number of steps. The general structure of the algorithm was presented in fig. 10, [1-3].

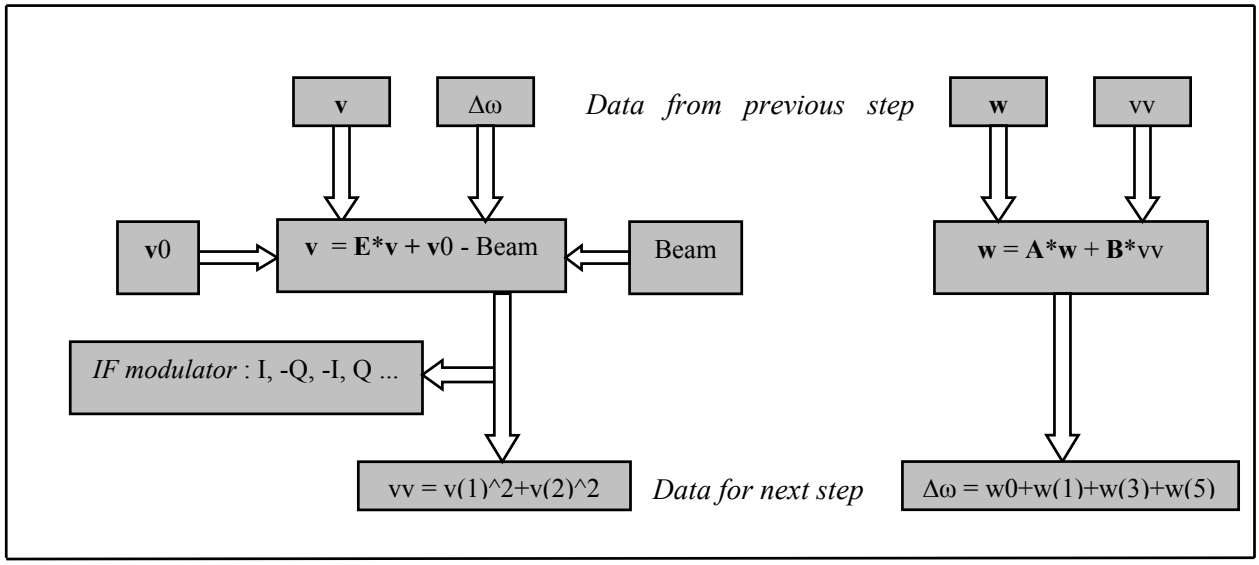


Fig. 10. Schematic diagram of the FPGA processing system for one step of the TESLA cavity algorithm.

The left side of the diagram, presented in fig.10, describes the discrete algorithm of the electrical model of the cavity. The right side of the diagram, presented in fig.10, presents the mechanical model of the resonant TESLA cavity. The discrete algorithm of the cavity was realized in the VHDL in a form of the fully parameterized behavioral description. This allows for a choice of the width of the DSP word and the position of the point. To realize the processing, there was applied a pipeline bus work regime.

The above algorithm was implemented in the described FPGA chips for three widths of the DSP word. The word widths were 18, 24 and 32 bits, with the position of the point on 8, 16 and 24 bit, respectively. The algorithm is realized during the time of the 4 periods of the system clock.

data width	ALTERA STRATIX	LCELL no	MULT. no	fmax [MHz]	time [ns]	XILINX VIRTEX	LCELL nr	MULT. no	fmax [MHz]	time [ns]
18	EP1S60F1508C	1414	18	50	80	2V6000ff1517	1403	18	28	143
24	EP1S60F1508C	1832	80	33	121	2V6000ff1517	3247	72	22	182
32	EP1S60F1508C	2396	80	28	143	2V6000ff1517	4468	72	21	190

Table 5. Collection of the TESLA cavity simulator implementation results for the three chosen widths of the DSP word and for the STRATIX EP1S60F1508C-6 and VIRTEX 2V6000ff1517 FPGA/DSP chips. The time of realization of a single step of the algorithm was given in [ns].

There are some similarities in the occupancy of the hardware multiplication components to the previous predictions, presented in figs. 6 and 7. The number of the occupied multipliers depends in a step-like way on the number of the bits of the DSP word. The most optimal system usage is at the width of 18 bits of the DSP word, because in such a case there is implemented a single multiplier for a single operation. The realization of the 36-bit word operation is performed with the usage of as much as a group of four multipliers, per a single operation.

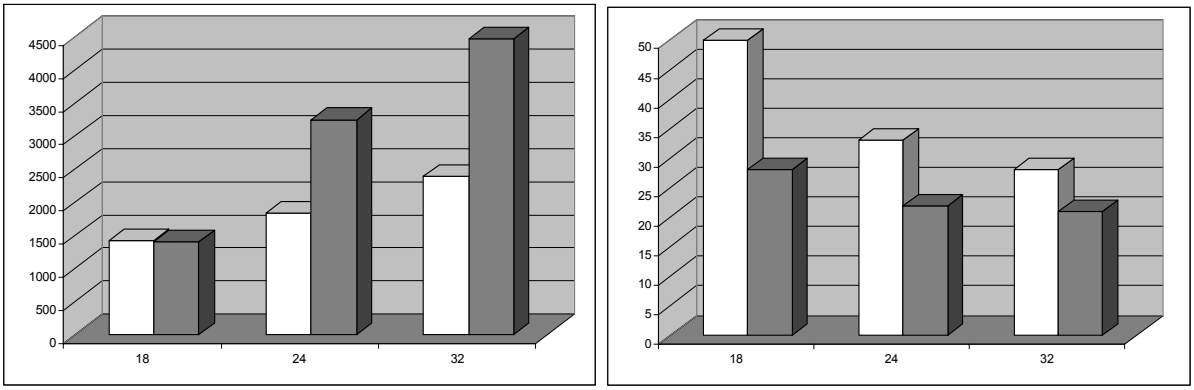


Fig. 3. Comparison between the numbers of used logical cells (left diagram) and maximal work frequency in MHz (right diagram) for the operation of adding as a function of the number of the operation bits. White bars represent the results for the STRATIX EP1S10B672C-6 FPGA/DSP chip. Grey bars represent the data for the VIRTEX-II 2V250fg456-5 FPGA/DSP chip.

Fig.3 presents the comparison between two implementations of the TESLA Cavity simulator in the function of the number of DSP bits. The more saving realization, in respect to the FPGA/DSP chip resources, like the number of the LCELL, is provided by the STRATIX. This stems from the existence of the internal connections between the DSP block structures into a bigger entities. Contrary, the VIRTEX has these connections realized by the FPGA logical blocks (description to fig. 6.). The STRATIX chip provides also a bigger work frequency of the test algorithm of the TESLA cavity simulator.

6. CONCLUSIONS

This work describes, in a concise way, the chosen examples of practical usages, of the newest versions, of FPGA chips, equipped with the DSP blocks. The implementation analysis univocally indicates that these chips have big calculation efficiency for the considered applications in the LLRF systems. It was confirmed in the precisely analyzed example of a discrete algorithm of the TESLA cavity simulator. The time needed to realize a single step of the TESLA cavity simulation changes in the range of 80 – 200 ns. The exact value of the time depends on the particular kind of the used FPGA chip and on the width of the DSP word.

REFERENCES

1. T.Czarski, K.T.Pozniak, R.Romaniuk, S.Simrock: “TESLA Cavity Modeling and Digital Implementation with FPGA Technology Solution For Control System Purpose”, TESLA Report, 2003-28;
2. T.Czarski, R.S.Romaniuk, K.T.Pozniak S.Simrock “Cavity Control System Essential Modeling For TESLA Linear Accelerator”, TESLA Technical Note, 2003-08;
3. T.Czarski, R.S.Romaniuk, K.T. Pozniak “Cavity Control System, Models Simulations For TESLA Linear Accelerator ”, TESLA Technical Note, 2003-09;
4. http://www.altera.com/ [Altera Homepage]
5. http://www.xilinx.com/ [Xilinx Homepage]
6. http://www.mentor.com/ [Mentor Graphics Homepage]