Meas. Sci. Technol. **18** (2007) 2404–2412 doi:10.1088/0957-0233/18/8/015

SEU-tolerant IQ detection algorithm for LLRF accelerator system

M Grecki

Department of Microelectronics and Computer Science, Technical University of Lodz, Lodz, Poland

E-mail: grecki@dmcs.pl

Received 19 November 2006, in final form 16 March 2007
Published 6 July 2007
Online at stacks.iop.org/MST/18/2404

Abstract
High-energy accelerators use RF field to accelerate charged particles. Measurements of effective field parameters (amplitude and phase) are tasks of great importance in these facilities. The RF signal is downconverted in frequency but keeping the information about amplitude and phase and then sampled in ADC. One of the several tasks for LLRF control system is to estimate the amplitude and phase (or I and Q components) of the RF signal. These parameters are further used in the control algorithm. The XFEL accelerator will be built using a single-tunnel concept. Therefore electronic devices (including LLRF control system) will be exposed to ionizing radiation, particularly to a neutron flux generating SEUs in digital circuits. The algorithms implemented in FPGA/DSP should therefore be SEU-tolerant. This paper presents the application of the WCC method to obtain immunity of IQ detection algorithm to SEUs. The VHDL implementation of this algorithm in Xilinx Virtex II Pro FPGA is presented, together with results of simulation proving the algorithm suitability for systems operating in the presence of SEUs.

Keywords: IQ detection, SEU, SEU-tolerant, algorithm, FPGA

1. Introduction

The RF (radio frequency) field accelerating charged particles in accelerators has to be precisely stabilized in order to obtain the required beam parameters [1–5]. The requirement of a low energy spread of the beam demands a constant accelerating gradient that is stabilized by the complex LLRF (low level radio frequency) control system [6]. The very tight requirements on field stability in modern accelerator control systems require application of digital signal processing techniques based on FPGAs/DSPs (field programmable gate arrays/digital signal processors).

The XFEL (X-Ray Free Electron Laser) will be built in the next few years at DESY Hamburg [1]. The laser will be driven by an electron linear accelerator controlled by the LLRF system. The accelerator components will be placed in a single tunnel to save costs [1]. The control system must be designed in a robust fashion since the components placed in the tunnel are not accessible during accelerator operation. Additionally, they operate close to the accelerator beam pipe and hence are exposed to neutron and gamma radiation. The faults tolerance of the control system (implemented both in hardware and in software) helps to fulfil high availability and reliability requirements of the whole system.

The single RF station of XFEL will consist of four accelerating cryomodules (each composed of eight superconducting cavities) and a klystron supplying the RF power controlled by the LLRF system [1]. The RF system will have a similar architecture to that operated presently in FLASH (Free-Electron LASer in Hamburg) [2, 7]. The LLRF (figure 1) of the XFEL will be designed as a closed loop digital control (identical to the LLRF operated in FLASH). The state of RF electromagnetic field filling the accelerating cavities is measured by sensors determining electric field, forward power and reflected power signals (Vacc, Ainc, Aref—see figure 1). All of them are RF signals and their digital processing requires downconversion to intermediate frequency (f_{IF}) signals, preserving information about the amplitude and phase of the RF field. The IF (intermediate frequency) signals are sampled by ADCs (analogue-to-digital converters) with

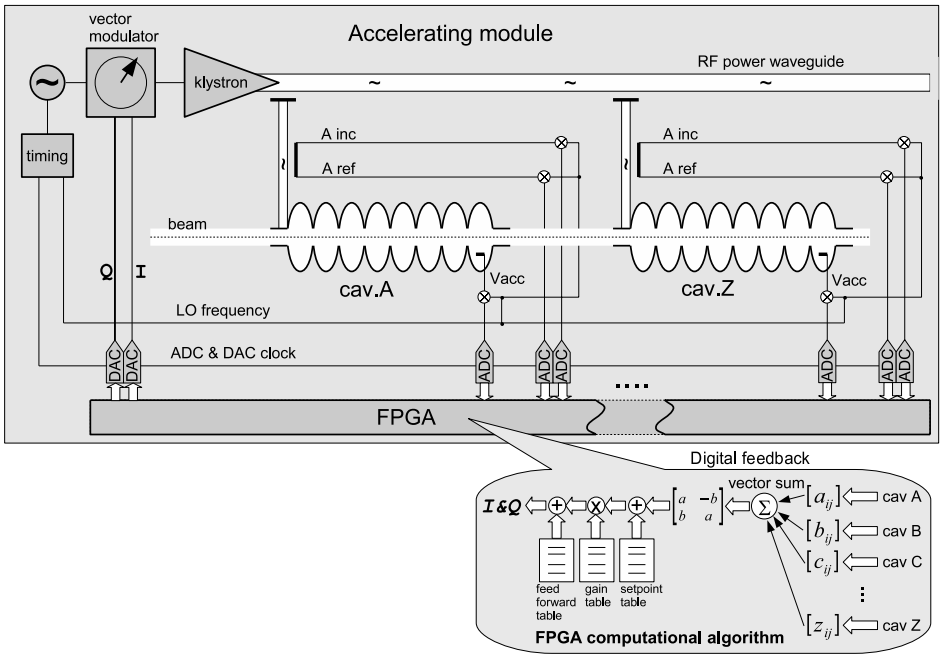

Figure 1. The LLRF control system architecture for XFEL linear accelerator [1] (simplified picture of single accelerating module, showing only the parts covered in the paper).

sampling rate f_S and the computation block has to calculate the I and Q (in phase and quadrature) components. Consequently, the I and Q components of the original RF signal will be determined. The I and Q for each cavity require calibration in amplitude and phase due to different cable lengths, variations in external Q of the probe and downconverter characteristics. The samples of accelerating voltages (Vacc—see figure 1) from all cavities driven by the same RF power station (klystron) are used to calculate the vector sum. The current value of the vector sum is compared to the setpoint and the error signal is driving a regulator that can be a simple PI controller or a much more complex filter [6, 8, 9]. Regulator outputs (I and Q) drive the vector modulator through DACs (digital-to-analogue converter), providing the required input signal to the klystron which drives the cavities. The control system has to keep stable amplitude and phase of the RF field in cavities in spite of beam influence, noises and drifts in the system and other disturbances.

2. IQ detection

The very first stage in the digital control loop (figure 1) is a field detection subsystem. The intermediate frequency analogue signal $x(t) = A\sin(2\pi f_{\mathrm{IF}}t + \varphi) = I\sin(2\pi f_{\mathrm{IF}}t) + Q\cos(2\pi f_{\mathrm{IF}}t)$ where A is an amplitude and φ is a phase related to the reference oscillator signal is sampled by ADC with a constant sampling rate f_S. The task is to calculate the I and Q components from these samples.

2.1. Direct IQ detection scheme

When one takes into account two successive samples of the IF signal then (1) can be written as

$$\begin{aligned} x_1 &= x(t_1) = A\sin(\omega t_1 + \varphi) + u_{\mathrm{off}} \\ &= I\sin(\omega t_1) + Q\cos(\omega t_1) + u_{\mathrm{off}} \\ x_2 &= x(t_2) = A\sin(\omega t_2 + \varphi) + u_{\mathrm{off}} \\ &= A\sin(\omega t_1 + \Delta\varphi + \varphi) + u_{\mathrm{off}} \\ &= I\sin(\omega t_1 + \Delta\varphi) + Q\cos(\omega t_1 + \Delta\varphi) + u_{\mathrm{off}} \end{aligned} \tag{1}$$

where $\omega = 2\pi f_{\mathrm{IF}}$; $\Delta\varphi = 2\pi f_{\mathrm{IF}}/f_S$ is a phase advance between successive samples and u_{off} is the sum of an ADC and an amplifier dc offset.

Solving (1) one obtains the I and Q components (2),

$$\begin{aligned} I &= -\frac{(x_1 - u_{\mathrm{off}})\cos(\omega t_1 + \Delta\varphi) - (x_2 - u_{\mathrm{off}})\cos\omega t_1}{\sin\Delta\varphi} \\ Q &= \frac{(x_1 - u_{\mathrm{off}})\sin(\omega t_1 + \Delta\varphi) - (x_2 - u_{\mathrm{off}})\sin\omega t_1}{\sin\Delta\varphi}. \end{aligned} \tag{2}$$

It should be noted that $\sin\Delta\varphi$ must not be equal to zero and therefore phase advance $\Delta\varphi$ cannot be a multiple of π. Hardware implementation of equations (2) requires two adders and three multipliers operating at full speed for each of I and Q (assuming keeping computed $(x_2 - u_{\mathrm{off}})$ in register, that becomes $(x_1 - u_{\mathrm{off}})$ in the next iteration). It is worth mentioning that offset u_{off} must be known precisely since the unbalanced offset is reflected in output IQ vector by an error vector rotating with the frequency f_{IF}. The I and Q components calculated according to (2) are also sensitive to ADC noise particularly for high f_S and moderate f_{IF}. The ADC noise is propagated to the I and Q values with factor $|2/\sin\Delta\varphi|$ (in the worst case) and when $|\sin\Delta\varphi| \ll 1$ the noise can be significantly amplified (e.g. for phase advance $\Delta\varphi = 11^\circ$ the noise in I and Q can be one order of magnitude higher than ADC noise).

The method of IQ detection based on (2) uses only two successive signal samples, thus giving the lowest possible

latency. However, due to the high utilization of FPGA resources (in total four adders and six multipliers) and offset dependence of I and Q the implementation of (2) in a FPGA is not advisable.

2.2. IQ detection scheme using samples from an integer number of IF periods

The direct IQ detection scheme calculates the I and Q components at the same rate as the sampling process. However, such a high rate is not usually needed since the other parts in the control loop (figure 1) limit the bandwidth [6]. Therefore, the I and Q components can be averaged during a given time (it should be short compared to the dynamics of the cavity signal), thus reducing the sampling rate of I and Q and lowering the noise level. In such a case the new method of I and Q detection (3) can be used [10],

$$I = \frac{2}{N}\sum_{i=0}^{N-1} x_i \sin(i \cdot \Delta\varphi)$$
$$Q = \frac{2}{N}\sum_{i=0}^{N-1} x_i \cos(i \cdot \Delta\varphi). \tag{3}$$

This method assumes perfect synchronization between the IF signal and the sampling process and uses N samples taken during M periods of the IF signal (M, N: integer numbers). Taking into account more samples than necessary leads to the statistical improvement of calculated I and Q values, but on the other hand increases the latency of the system.

The phase advance $\Delta\varphi = 2\pi f_{\mathrm{IF}}/f_S = 2\pi M/N$ being a multiple of π cannot be used since it leads to a linear dependence between successive samples [10]. The $2M/N$ consequently cannot be an integer and the lowest possible number of samples N is equal to 3 that makes latency slightly worse compared to the direct method of I and Q calculations using (2).

Equations (3) are easy to implement in FPGA and require only two multipliers and two accumulators working at full speed for both I and Q. The final scaling by $2/N$ can be done with an additional multiplier operating at the f_S/N frequency or it can be substituted by a wire shift (scaling by the power of 2) if the sin/cos tables are scaled accordingly in advance. The necessary scaling factor F can be calculated by decomposition of the number $2/N$ according to (4):

$$\frac{2}{N} = F \cdot 2^K \tag{4}$$

where $1 \leqslant F < 2$ and K is a negative integer number.

The sin/cos tables must be N entries long. They can be implemented in FPGA as a ROM memory (using e.g. lookup tables) or as a memory block provided for storing coefficients for multipliers (this is a better way—usually it gives higher performance).

Assuming the application of 14 bit ADCs, $f_{\mathrm{IF}} = 50$ MHz and I and Q calculations based on three successive samples f_S can be 75 MHz ($N = 3$, $M = 2$, $\Delta\varphi = 240^\circ$) giving the IQ detection latency equal to 40 ns. The $f_{\mathrm{IF}} = 1$ MHz while $f_S = 100$ MHz ($N = 100$, $M = 1$, $\Delta\varphi = 4.8^\circ$) gives the latency of 1 μs while noise due to averaging is reduced by a factor of 10 (square root of 100).

The important feature of this method is an insensitivity to the offset value v_{off}. Since the samples are taken from M whole periods of IF signal, the offset is cancelled due to the periodic nature of sin and cos functions. Another benefit comes from the possibility of trade-off between latency and noise reduction due to averaging.

2.3. Additional features of IQ detection algorithm

The control algorithm of the LLRF system (figure 1) has to take into account the differences in signal delay and attenuation in each cavity channel (caused by different cable lengths). Therefore the signal must be calibrated in amplitude and phase. The calibration is performed by the magnitude scaling of the IQ vector by factor g and phase shift by angle α (5) [6]:

$$\begin{bmatrix} I_{\mathrm{cal}} \\ Q_{\mathrm{cal}} \end{bmatrix} = \begin{bmatrix} a & -b \\ b & a \end{bmatrix} \begin{bmatrix} I \\ Q \end{bmatrix} \tag{5}$$

where $a = g\cos\alpha$, $b = g\sin\alpha$.

The hardware implementation of the calibration step according to (5) requires four multipliers and two adders. However, using slightly modified formulae (3) one can obtain the calibration 'for free' (6) while I and Q are calculated as

$$I_{\mathrm{cal}} = \frac{2}{N}\sum_{i=0}^{N-1} = x_i(g\sin(i\Delta\varphi - \alpha))$$
$$Q_{\mathrm{cal}} = \frac{2}{N}\sum_{i=0}^{N-1} = x_i(g\cos(i\Delta\varphi - \alpha)). \tag{6}$$

The only cost of parallel calibration while computing the IQ vector is the parametrization of sin/cos tables. Each input channel has to have its own sin/cos tables where the calibration parameters are taken into account. For hardware FPGA implementation it is not a big problem since usually multipliers are connected internally to the special memory blocks provided for coefficient storage. Utilization of those memory blocks gives optimal performance of the chip while using a single-memory block for all sin/cos tables reduces performance significantly.

3. SEU tolerance

The ionizing radiation has a negative influence on electronic devices. It reduces the lifetime of the circuit and also can cause temporary malfunction during circuit operation [11, 12].

The temporary effects (single event effects—SEE) are caused by charged particles energetic enough to ionize the semiconductor by generating excessive electron/hole pairs [12–15].

The single-event transients (SET) [16] caused by temporary ionization of the semiconductor resulting in current pulses can propagate through the circuit logic and generate faulty signals. These disturbances are only temporary and after a short period (usually below a nanosecond) the SET vanishes. Different situations occur when the SET is locked in a data storage component (flip-flop, register or memory). In such a case the SET is no longer temporary, but permanently changes information stored in the affected component (single

event upset) [17, 18], however restoring information to the previous state recovers normal operation of the circuit.

It is estimated that the SEU rate will become the most limiting factor of further integration scale growth [19, 20] since it rises faster than the integration scale foreseen by Moore's law.

It should be mentioned that in the case of SRAM-based FPGAs SEUs can occur not only in the working registers and memory but also in the configuration memory [21, 22]. The SEU in the configuration memory can change both the functionality of the FPGA cell and the programmable connections between them.

3.1. SEU mitigation techniques

The SEE countermeasures can be applied on various levels of circuit design. The fabrication process-based techniques allow obtaining radiation immunity using e.g. silicon-on-insulator (SOI) devices equipped with guarding rings for charge collection, high supply voltage and additional components (capacitors and resistors) increasing the energy level the particle has to deposit in the semiconductor in order to generate SEE [23–25]. These technological countermeasures usually make the circuit costs extremely high and also often significantly degrade the circuit performance. Additionally they may not be used on the system level design where the designer cannot choose the parameters of the technological manufacturing process and has to use commercially available components.

An SEU mitigation solution can be also obtained by using redundancy in information processed by the circuit. The well-known triple modular redundancy (TMR) together with the voting circuit allows us to detect and correct single errors induced by SEU. The data stored in registers and memory can be secured by using error-correcting codes like the Hamming or Reed–Solomon ones. Application of TRM and error-correcting codes give different results (in the sense of required overhead) depending on the kind and size of the protected functional block [27]. The SETs can be mitigated using time redundancy [26].

All these methods require utilization of more resources and/or more time to process signals in comparison with the circuit with no redundancy (in the case of TMR over three times more resources are needed). Therefore other methods that can reduce the resource utilization while keeping high processing speed are strongly awaited.

The methods using algorithm-based fault tolerance (ABFT) techniques [29, 30, 32–35] move the SEU tolerance mechanisms to a high level of abstraction and allow us to improve the reliability of the system with minimum hardware and processing time overhead.

4. SEU-tolerant IQ detection algorithm

The IQ detection algorithm given in (3) is susceptible to SEU that can occur in sin/cos tables, accumulator registers and output registers. Also the functionality of computational blocks can be changed when SEU occurs in the FPGA configuration memory. All these effects can result in erroneously computed I and Q values that affect the regulator output (figure 1), not only degrading the beam energy stability but also activating interlocks when the klystron is overdriven.

Evidently, it is possible to obtain SEU tolerance by application of TMR but since the FPGA resources are limited and there are other functions the FPGA must perform, a method for reduction of resource utilization is highly desirable.

It is possible to significantly reduce the resource overheads still holding the SEU tolerance by using a smart algorithm for data processing. The IQ detection algorithm (3) can be also written in a matrix form (7):

$$\begin{bmatrix} I \\ Q \end{bmatrix} = \frac{2}{N} \begin{bmatrix} \sin\alpha_1 & \sin\alpha_2 & \cdots & \sin\alpha_N \\ \cos\alpha_1 & \cos\alpha_2 & \cdots & \cos\alpha_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \boldsymbol{AX} \tag{7}$$

where $\boldsymbol{A}$ is the scaled sin/cos table ($\alpha_i = (i-1)\Delta\varphi$) and $\boldsymbol{X}$ is a vector of IF signal samples.

For matrix multiplication there are known methods allowing us to detect and correct errors made during computation. The application of the WCC (weighted checksum code) method [30] to (7) leads directly to (8):

$$\begin{bmatrix} I \\ Q \\ CS_1 \\ CS_2 \end{bmatrix} = \frac{2}{N} \begin{bmatrix} \sin\alpha_1 & \sin\alpha_2 & \cdots & \sin\alpha_N \\ \cos\alpha_1 & \cos\alpha_2 & \cdots & \cos\alpha_N \\ pcs_1 & pcs_2 & \cdots & pcs_N \\ qcs_1 & qcs_2 & \cdots & qcs_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \\ \boldsymbol{PCS} \\ \boldsymbol{QCS} \end{bmatrix} \boldsymbol{X} \tag{8}$$

where $\boldsymbol{PCS}$ and $\boldsymbol{QCS}$ are checksums calculated according to (9) using coding vectors $\boldsymbol{p}$ and $\boldsymbol{q}$.

$$\begin{aligned} \boldsymbol{PCS} &= \boldsymbol{pA} = [1\ 1]\boldsymbol{A} \\ \boldsymbol{QCS} &= \boldsymbol{qA} = [1\ 2]\boldsymbol{A}. \end{aligned} \tag{9}$$

The vectors $\boldsymbol{p}$ and $\boldsymbol{q}$ must be linearly independent for the WCC method to be effective [31]. Several formulae for $\boldsymbol{p}$ and $\boldsymbol{q}$ vectors were proposed for the general case of matrix multiplications (average and weighted average vectors, periodic vectors, normalized vectors, etc [31]). Since the matrix $\boldsymbol{A}$ in (7) consists of only two rows of coefficients, the simplest formulae (9) fulfil the requirements while leading to straightforward implementation in FPGA.

The error detectability and correctability of the WCC method depends on accuracy of computations. Numerical errors made during computation (truncation, overflow) can produce the same effects as SEU-driven errors. The WCC method is capable of detecting and correcting all single SEU-generated errors assuming there is neither number truncation nor overflow during computations. Therefore, the application of fixed-point arithmetics with long enough multipliers and accumulators is appropriate for FPGA implementation. For given $\boldsymbol{p}$ and $\boldsymbol{q}$ formulae (9) the minimum length of registers accumulating CS_1 and CS_2 is respectively 1 and 2 bits more than the minimum length of I and Q accumulators. The detailed analysis of the required length of registers is presented in section 5.

Table 1. Syndrome values and their meaning.

S_1	S_2	Meaning
0	0	No errors
$\neq 0$	0	Error in CS_1
0	$\neq 0$	Error in CS_2
$\neq 0$	$\neq 0$	Error in I or Q

After the cycle of operation (processing of N samples of IF signal) the accumulated values of I, Q, CS_1 and CS_2 are accessible and the syndromes S_1 and S_2 determining the status of computations have to be calculated according to (10):

$$\begin{aligned} S_1 &= I + Q - CS_1 \\ S_2 &= I + 2Q - CS_2. \end{aligned} \tag{10}$$

The meanings of various syndrome values are collected and explained in table 1. Non-zero values of syndromes S_1 and S_2 indicate that an error has occurred during computation.

If both syndromes S_1 and S_2 (the last row in table 1) are non-zero the correction of I or Q value is necessary. Depending on the difference between S_1 and S_2 either I or Q is corrected by S_1 value (11):

$$\begin{aligned} \text{if}(S_1 = S_2) &\Rightarrow I = I - S_1 \\ \text{if}(S_1 \neq S_2) &\Rightarrow Q = Q - S_1. \end{aligned} \tag{11}$$

In other cases (the first three rows in table 1) no correction is needed since there was no error in computation or error appeared in checksums and I or Q have correct values. For diagnostic purposes all detected errors should alert the supervising system. Errors happening only rarely indicate SEU in working registers. Such errors are corrected internally by algorithm and no further action is required. High frequency of detected errors points to SEU in coefficient tables or the FPGA configuration memory. In such a case it is necessary to read and compare the coefficient tables by the supervising system. If inconsistency in these tables is detected the tables should be reloaded clearing the source of errors. Otherwise if the contents of tables is correct the SEU appeared in the configuration memory of FPGA and an interrupt in circuit operation is required in order to reboot the FPGA.

5. VHDL implementation

The algorithm described in section 2.2 together with modifications described in section 4 have been implemented in Xilinx Virtex2 Pro FPGA (the XC2VP30 chip of SIMCON 3.1 board [36]). The algorithm and its implementation will be tested in the FLASH accelerator after the upgrade of the LLRF control in FLASH. A simplified block diagram of the circuit is presented in figure 2. The main part consists of four almost identical MAC (multiply and accumulate) units computing the multiplications of rows of $\mathbf{A}/PCS/QCS$ matrix and vector of signal samples $\mathbf{X}$ (8) using fixed-point arithmetic. The $\sin/\cos/pcs/qcs$ tables are stored in separate RAMBLOCKs [37] coupled with multipliers and can be downloaded through external VME interface.

The implementation was done for the IF signal ($f_{\text{IF}} = 1$ MHz) sampled with 14 bit ADC with sampling rate $f_S = 50$ MHz, therefore $N = 50$. The internal FPGA multipliers (signed 18×18 bits) were used in MACs. The samples of

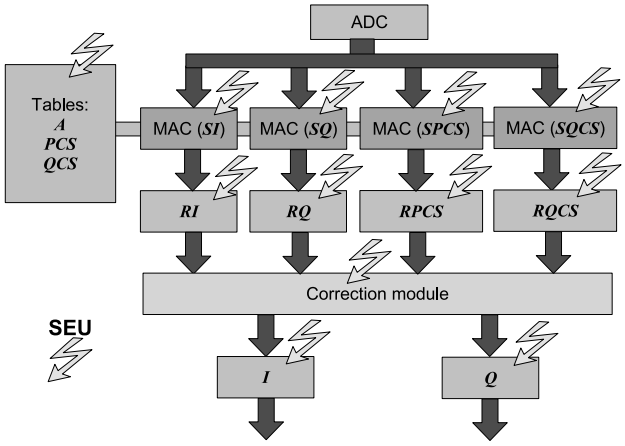

Figure 2. The block diagram of implementation of the IQ detection algorithm with SEU tolerance, Sxxx—accumulators in MACs (SI, SQ, SPCS, SQCS), Rxxx—result registers (RI, RQ, RPCS, RQCS).

the IF signal come from ADC as a 14 bits number coded in signed 2's complement format (with range $[-2^{13}, 2^{13}[$). They are sign-bit extended up to 18 bits and connected to one of the inputs of the MAC. The second input of the MAC is connected to the memory blocks containing tables of $\sin/\cos/pcs/qcs$ coefficients (written in 2's complement fixed point format ± 2.15 bits, i.e. scaled by 2^{15} and written in 18 bit long words). The final scaling by $2/N$ is realized by 'wire shift' and therefore $\sin/\cos/pcs/qcs$ coefficients tables require additional scaling. The factor $2/N = 2/50 = 0.04$ was decomposed into $F \cdot 2^K = 1.28 \times 2^{-5}$ and $\sin/\cos/pcs/qcs$ are multiplied by $F = 1.28$. The scaled $\sin/\cos/pcs/qcs$ coefficients fit well with fixed-point format ± 2.15 since even for qcs (the biggest magnitude of coefficients) the $F \cdot \max(\text{abs}(qcs)) = 2.86$ and the range of numbers that can be written in ± 2.15 format is $[-4, 4[$. This means there is even some more room ($\approx$40% of the range) for signal calibration (section 2.3).

The results of multiplications are accumulated in Sxxx accumulators without truncating less significant bits. The required length of accumulators can be calculated assuming no overflow is possible for any input signal. Taking that into account the required length of SI accumulator is $\log_2(F \cdot 2^{15} \cdot N \cdot 2/\pi \cdot 2^{13}) = 33.3$ so 34 bits are needed plus 1 bit for sign. In total 35 bits are needed in SI/SQ registers and 36 and 37 bits are needed for SPCS and SQCS registers respectively.

After the cycle of operation the results from MAC accumulators (Sxxx—figure 2) are transferred to result registers (Rxxx). The Rxxx registers are pipelines allowing error correction while the next cycle of partial sum calculation is executed in MAC units. The correction module is purely combinational and implements equations (11).

In order to perform final scaling of the resulting I and Q values the contents of RI and RQ registers are shifted right by 17 bits giving results written in fixed-point format ± 1.16 (scaled by 2^{16}) in 18 bit long word.

Table 2 presents the XC2VP30 resource utilization for the 8-channel IQ detector with implemented calibration, SEU tolerance and VME interface. The comparison of resource utilization of the IQ detector with and without SEU tolerance (table 2 and table 3) shows that the resource overhead for SEU

Table 2. FPGA resource utilization for the 8-channel SEU-tolerant IQ detector.

Resource	Utilization
Number of slices	2391 out of 13 696, 17%
Number of slice flip flops	2026 out of 27 392, 7%
Number of four input LUTs	4355 out of 27 392, 15%
Number of BRAMs	32 out of 136, 23%
Number of MULT18X18s	32 out of 136, 23%
Number of GCLKs	1 out of 16, 6%
Design statistics	
Minimum period: 8.532 ns (maximum frequency: 117.206 MHz)	
Minimum input arrival time before clock: 8.299 ns	
Maximum output required time after clock: 10.388 ns	
Maximum combinational path delay: 16.754 ns	

Table 3. FPGA resource utilization for the 8-channel IQ detector (without SEU tolerance).

Resource	Utilization
Number of slices	1277 out of 13 696, 9%
Number of slice flip flops	1104 out of 27 392, 4%
Number of four input LUTs	2137 out of 27 392, 8%
Number of BRAMs	16 out of 136, 12%
Number of MULT18X18s	16 out of 136, 12%
Number of GCLKs	1 out of 16, 6%

tolerance is considerably less than the overhead generated by application of other methods (e.g. TMR requires more than triple the amount of resources needed for the version without SEU tolerance).

The SEU occurrence may happen in all registers and memory blocks of the circuit. These places are indicated by small lightning symbols in figure 2. All of these SEU-generated errors (except errors in output registers) can be detected and amended in the correction module assuming that during the operating cycle only a single SEU can happen. The errors in the output registers should be treated separately and corrected by the application of error-correcting codes (e.g. Hamming codes [27]).

Additionally, SEUs can occur in the configuration memory of the FPGA changing the interconnections and/or functionality of components. One of many possible examples of error is a short to '1' or '0' of one of the bits in the SI register due to SEU in the routing part of configuration memory. In such a case every time this bit produces an error in the result value (it may not always happen, e.g. permanent '0' in the most significant bit and result value low enough to not set this bit), the error is detected and amended in the correction module. The detected errors should be monitored and analysed by the monitoring system. When the symptoms indicate the possibility of SEU in the FPGA configuration memory the FPGA should be reconfigured by external circuit.

The only errors that can be undetected (and uncorrected) are multiple errors or errors in the correction module. The consequences of such errors will be the miscalculation of the field in the cavity and later generation of wrong controller output signal (figure 1). The quality of field regulation in the cavity will be temporarily degraded. The wrong controller output signal can also overload the klystron and activate the interlock system, thus interrupting machine operation. In general, the performance and accessibility of the machine can be affected.

Multiple errors happen very rarely and even then most of the multiple errors can be detected (but probably not corrected) by the algorithm. The only case when multiple errors will not be detected is when they modify several registers but keep syndromes in (10) equal to 0 (as it is when there are no errors). Such a situation can occur e.g. when SI, SPCS and SQCS registers are simultaneously affected by bit flips in the same positions. The probability of such an event is very low and in all other cases the SEU will be detected, however after detection of the error the successive correction action will not bring the expected results. In the case of multiple errors the correction may not only not amend the error but even make it worse. The particular scenario depends on which registers are affected by errors. On the other hand, multiple errors occurring in the same register will be corrected by the algorithm; there is no difference between single and multiple errors in the affected register.

The correction module does not contain any registers (it is a pure combinational circuit) and therefore SEU can only happen in its configuration memory leading in the worst case to a malfunction of the circuit even when all intermediate results are calculated correctly. The probability of SEU in the configuration memory is roughly proportional to the surface occupied by the circuit. Since the correction module is relatively simple and small (it consists of only two adders, comparator and multiplexer) in comparison to the other parts of the IQ detector (coefficient tables, MAC units) the probability of SEU occurrence here is relatively small compared to the rest of the circuit. The application of the proposed algorithm therefore improves the effective reliability of the system; however the exact performance figures require further experimental investigation. One can estimate the reliability increase comparing the circuit areas sensitive to SEUs. If the IQ detector without SEU tolerance occupies area S_{SS} of the FPGA surface (the whole area S_{SS} is SEU sensitive) then the SEU-tolerant version of the IQ detector occupies area $S_{ST} \cong 2 \cdot S_{SS} + S_{CM}$ and only the area occupied by the correction module S_{CM} is SEU sensitive. Therefore the reliability boost (assuming the area occupied by the correction module is about 10% of the whole SEU-tolerant IQ detector) is $S_{SS}/S_{CM} \cong 5$ while the resource consumption increases roughly by a factor of 2.

Additionally, the external system should frequently check the configuration memory by configuration readout. Unfortunately, the Xilinx chips require stopping FPGA operation before readout [37]. In the case of the LLRF system

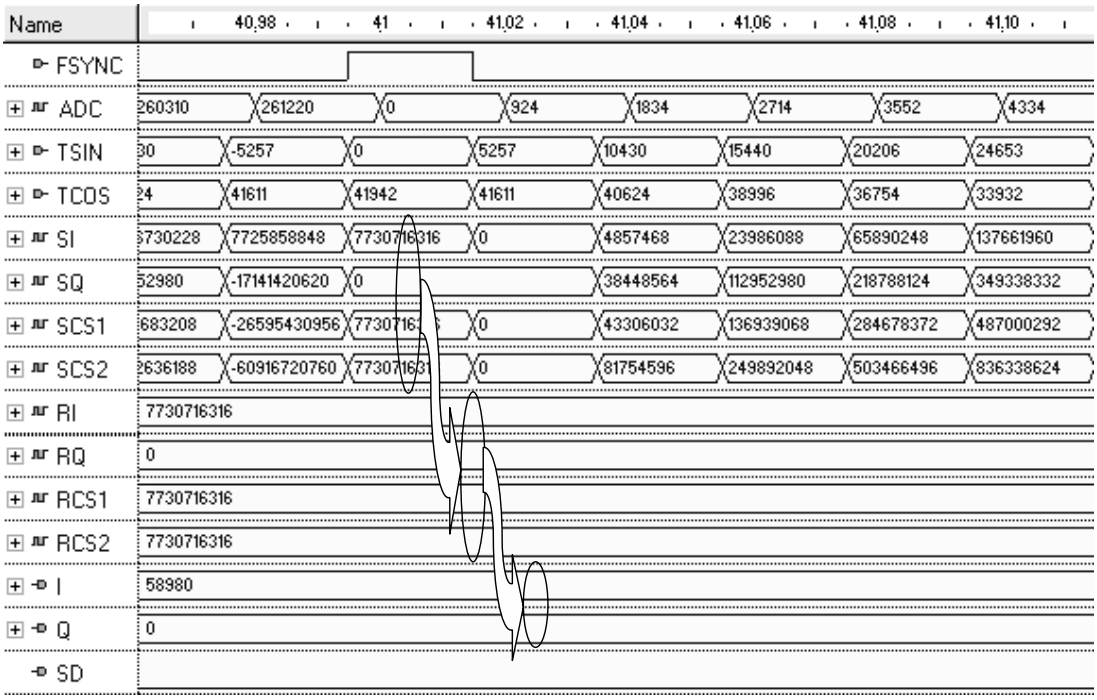

Figure 3. The simulation result of circuit operation without the presence of SEUs (time scale in μs). The data-flow in working registers is pointed out.

for the linear accelerator working in the pulse mode (like XFEL [1]) this is not a problem since there is enough time between pulses to perform such a test.

6. Simulation of circuit operation

In order to prove the SEU tolerance of the algorithm and its implementation the VHDL simulation of the circuit operation has been performed. The simulation method applied here was verified earlier by simulation of SEU appearance in the SRAM memory operated in radiation environment [38]. In order to complete simulations of the circuit in a reasonable time the SEU rate was increased by ten orders of magnitude compared to the measured value obtained for SRAM memories [28]. Therefore, the time scale of the presented results must not be treated literally.

The IF input signal for all the simulations is sinusoidal with frequency $f_{\mathrm{IF}} = 1$ MHz and amplitude $A = 0.9$ of ADC input range. The starting phase of the IF signal is equal to 0°. The ADC sampling rate is $f_S = 50$ MHz and therefore one cycle of algorithm takes 1 μs with the number of processed signal samples $N = 50$.

The expected output from the IQ detector is therefore $I = 0.9$ (in the fixed-point format ± 1.16 this is equal to $I = 58\,982$) and $Q = 0$. Figures 3–5 present the time waveform of chosen internal signals of the IQ detector during operation. The contents of all working registers Sxxx, Rxxx and output registers I and Q are presented. Additional signal SD (SEU detected) active high shows the status of circuit operation. When after counting N signal samples the trigger signal FSYNC appears (figure 3), the contents of Sxxx accumulators are transferred into Rxxx registers and the accumulators are cleared. Using the Rxxx registers (figure 2) the final I and Q outputs are calculated and optionally corrected (in the case of detected error).

In the absence of SEU-generated errors the waveforms in Rxxx and output I and Q registers are constant (figure 3), since in every cycle of algorithm the same value is latched in the registers (for the same input signal the intermediate results and output I and Q values do not change). Only the contents of accumulators Sxxx changes with every ADC input sample corresponding to the accumulated result.

SEU-generated errors affect the circuit operation by changes in working registers (figure 4). Each lightning symbol in figure 4 pinpoints one SEU occurrence. Only a few such occurrences are shown out of the many simulated here. The lightning symbol identifies which register was affected by the SEU and at what time this happened. Not all SEUs have direct impact on the calculation results. If the register contents after SEU occurrence are immediately overwritten by a new value the SEU is ignored. Therefore not all SEUs generated in the circuit were detected (SEU detection is indicated by a high level of signal SD). It can also be noted in figure 4 that no SEU-generated errors affect the contents of I and Q registers (confirming the proper operation of the circuit), except the one SEU occurring directly in the Q register at about $t = 580$ μs. The output registers I and Q are beyond the control of the computational algorithm and therefore SEUs occurring in these registers cannot be corrected without additional tools. However such errors can be easily detected and corrected by applications of error-detecting codes [27] in these registers.

The SEU in configuration memory may lead to various effects in circuit functionality and FPGA routing. However, they can be detected by the algorithm if the changes result in numerical errors during computation. Figure 5 presents the circuit operation in case of SEU in FPGA configuration

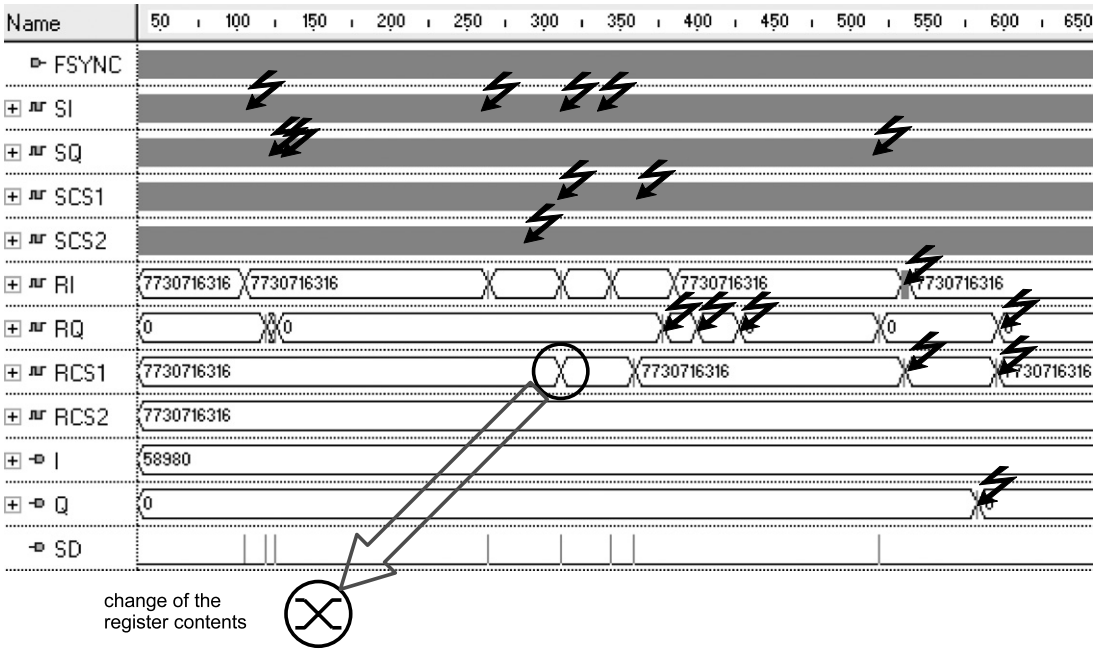

Figure 4. The simulation result of circuit operation in the presence of SEUs in working registers (time scale in μs).

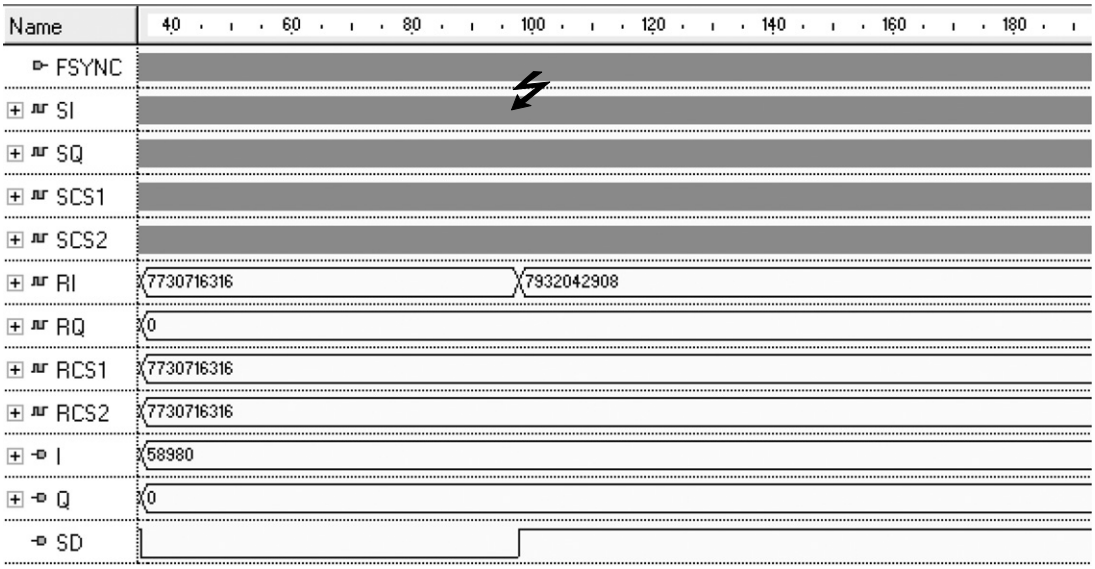

Figure 5. The simulation result of circuit operation with SEU appearing at $t = 97$ μs in the FPGA configuration memory causing a stuck '1' in the 23rd bit of the SI accumulator.

memory resulting in a stuck '1' in the 23rd bit of the SI accumulator. In this case every processing cycle leads to error detection (SD signal is permanently high). However, it is still possible to compute proper output I and Q values. Similar effects will appear when SEU occurs in $\sin/\cos/pcs/qcs$ tables. The supervising system will be alerted by high frequency of SEU detection and should check and restore the data.

7. Conclusion

Electronic equipment operating in a radiation environment is exposed to disturbances coming from ionizing radiation. With technology advances the temporary effects (SEU) become one of the main limitations in expansion of circuit complexity. Therefore SEU countermeasures are very desirable, especially in particle accelerator applications. This paper presents a high level algorithm and its FPGA implementation performing IQ detection of sinusoidal signals. The FPGA implementation issues are discussed and the simulation results of the circuit operation in the presence of neutron radiation are presented. All single SEU occurrences in working registers and coefficient tables were detected and corrected by the algorithm except those occurring directly in output registers (they must be treated by other methods—this is a reason to secure output registers by error-correcting codes [27]). All other SEUs were corrected properly while using a significantly

lower amount of FPGA resources compared to the TMR method.

Acknowledgments

I acknowledge the support of the European Community-Research Infrastructure Activity under the FP6 'Structuring the European Research Area' program (CARE, contract number RII3-CT-2003-506395), Polish National Science Council Grant '138/E-370/SPB/6.PR UE/DIE 354/2004-2007' and Polish National Science Council Grant 'N515 07231/3756'.

References

[1] DESY XFEL Project Group, European XFEL Project Team 2006 *The European X-Ray Free-Electron Laser Technical Design Report* (Deutsches Elektronen-Synchrotron DESY) pp 71–4
[2] Simrock S N 2004 State of the art in RF control *Proc. 2004 LINAC Conf. (Lübeck, Germany)* pp 523–5
[3] McIntosh P A, Beard C D, Dykes D M and Moss A J 2006 RF requirements for the 4GLS Linac systems *Proc. EPAC2006* pp 439–41
[4] McCarthy M 2001 LLRF requrements for APT *Proc. Particle Accelerator Conf. (Chicago, IL, 2001)* pp 788–90
[5] Ayvazyan V, Rehlich K and Simrock S N 2003 Requirements for RF control of TTF II FEL user facility *Proc. 2003 Particle Accelerator Conf.* pp 2342–4
[6] Schilcher T 1998 Vector sum control of pulsed accelerating fields in Lorentz force detuned superconducting cavities *PhD Thesis* University of Hamburg pp 50–64(a)
[7] Ayvazyan V, Petrosyan G, Rehlich K, Simrock S N and Vetrov P 2005 RF control system for the DESY VUV-FEL linac *Proc. Particle Accelerator Conf. (Knoxville, TN, 2005)* pp 2899–901
[8] Liepe M, Belomestnykh S, Dobbins J, Kaplan R and Strohman C 2003 A new digital control system for CESR-C and the Cornell ERL *Proc. 11th Workshop on RF Superconductivity (Travemünde/Lübeck, Germany, 8–12 Aug. 2003)* http://srf2003.desy.de/fap/paper/ThP30.pdf
[9] Simrock S N, Cichalewski W, Grecki M K, Jablonski G W and Jalmuzna W J 2006 Universal controller for digital RF control *Proc. EPAC 2006* pp 1459–61
[10] Grecki M, Jezynski T and Brandt A 2005 Estimation of IQ vector components of RF field—theory and implementation *12th Int. Conf. Mixed Design of Integrated Circuits and Systems (MIXDES 2005)* pp 783–8
[11] LaBel K A *et al* 2006 Radiation effects and analysis *Technical Report* NASA Goddard Space Flight Center http://radhome.gsfc.nasa.gov/top.htm
[12] Nguyen D N and Scheick L Z 2002 SEE and TID of emerging non-volatile memories *Radiation Effects Data Workshop (2002)* (Piscataway, NJ: IEEE) pp 62–6
[13] O'Bryan M V *et al* 1999 Recent radiation damage and single event effect results for microelectronics *Radiation Effects Data Workshop (1999)* pp 1–14 http://citeseer.ist.psu.edu/272532.html
[14] O'Bryan M V *et al* 1998 Single event effect and radiation damage results for candidate spacecraft electronics *Radiation Effects Data Workshop (1998)* pp 1–13 http://citeseer.ist.psu.edu/279834.html
[15] Baumann R C and Smith E B 2000 Neutron-induced boron fission as a major source of soft errors in deep submicron SRAM devices *Proc. 38th Int. Reliability Phys. Symp. 2000* pp 152–7
[16] Alexandrescu D, Anghel L and Nicolaidis M 2002 New methods for evaluating the impact of single event transients in VDSM ICs *Proc. 17th IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems DFT02* pp 99–107
[17] Dodd P E and Massengill W L 2003 Basic mechanisms and modelling of single-event upset in digital microelectronics *IEEE Trans. Nucl. Sci.* **50** 583–602
[18] Normand E 1996 Single event upset at ground level *IEEE Trans. Nucl. Sci.* **43** 2742–50
[19] Hareland S, Maiz J, Alavi M, Mistry K, Walsta S and Dai C 2001 Impact of CMOS process scaling and SOI on soft error rates of logic processors *Proc. Symp. VLSI Tech.* pp 73–4
[20] Baumann R 2002 The impact of technology scaling on soft error rate performance and limits to the efficacy of error correction *IEDM Tech. Dig.* pp 329–32
[21] Buchanan N J and Gingrich D M 2003 Proton radiation effects in XC4036XLA field programmable gate arrays *IEEE Trans. Nucl. Sci.* **50** 263–71
[22] Crain S H, Mazur J E, Katz R B, Koga R, Looper M D and Lorentzen K R 2001 Analog and digital single-event effects experiments in space *Trans. Nucl. Sci.* **48** 1941–848
[23] Baggio J, Ferlet-Cavrois V, Lambert D, Paillet P, Wrobel F, Hirose K, Saito H and Blackmore E W 2005 Neutron and proton-induced single event upsets in advanced commercial fully depleted SOI SRAMs *IEEE Trans. Nucl. Sci.* **52** 2319–25
[24] Velazco R, Bessot D, Duzellier S, Ecoffet R and Koga R 1994 Two CMOS memory cells suitable for the design of SEU-tolerant VLSI circuits *IEEE Trans. Nucl. Sci.* **41** 2229–34
[25] Agrawal G R, Massengill L W and Gulati K 1994 A proposed SEU tolerant dynamic random access memory (DRAM) cell *IEEE Trans. Nucl. Sci.* **41** 2035–42
[26] Majumdar A, Raghavendra C S and Breuer M A 1990 Fault tolerance in linear systolic arrays using time redundancy *IEEE Trans. Comput.* **39** 269–76
[27] Hentschke R, Marques F, Lima F, Carro L, Susin A and Reis R 2002 Analyzing area and performance penalty of protecting different digital modules with hamming code and triple modular redundancy *Proc. 15th Symp. on ICs and Systems Design, SBCCI02* pp 95–100
[28] Makowski D, Grecki M, Napieralski A, Simrock S and Mukherjee B 2006 A distributed system for radiation monitoring at linear accelerators *IEEE Trans. Nucl. Sci.* part 1 **53** 2008–15
[29] Vijay M and Mittal R 1997 Algorithm-based fault tolerance: a review *Microprocess. Microsyst.* **21** 151–61
[30] Jou J Y and Abraham J A 1986 Fault-tolerant matrix arithmetic and signal processing on highly concurrent computing structures *Proc. IEEE* **74** 732–41
[31] Nair V S S and Abraham J A 1990 Real-number codes for fault-tolerant matrix operations on processor arrays *IEEE Trans. Comput.* **39** 426–35
[32] Anfinson C J and Luk F T 1988 A linear algebraic model of algorithm-based fault tolerance *IEEE Trans. Comput.* **37** 1599–604
[33] Vinnakota B and Jha N K 1993 Synthesis of algorithm-based fault-tolerant systems from dependence graphs *IEEE Trans. Parallel Distrib. Syst.* **4** 864–74
[34] Patel J H and Fung L Y 1982 Concurrent error detection in ALUs by recomputing with shifted operands *IEEE Trans. Comput.* **31** 589–95
[35] Luk F T and Park H 1986 An analysis of algorithm-based tolerance techniques *Advanced Algorithms and Architectures for Signal Processing (SPIE* vol 696*)* pp 222–7
[36] Giergusiewicz W *et al* 2005 Low latency control board for LLRF system: SIMCON 3.1 *Proc. SPIE: Photonics Applications in Industry and Research IV* vol 5948 pp 710–5
[37] Xilinx Inc. 2005 *Virtex-II Pro and Virtex-II Pro X FPGA User Guide* http://direct.xilinx.com/bvdocs/userguides/ug012.pdf
[38] Grecki M 2006 VHDL simulation considering single event upsets (SEUs) *Proc. NSTI Nanotech 2006* vol 1 pp 717–20