A novel FPGA-based system for Tumor Growth Prediction

Konstantinos Malavazos
Electrical and Computer Engineering
Technical University of Crete
Chania, Greece
kmalavazos@mhl.tuc.gr

Maria Papadogiorgaki
Electrical and Computer Engineering
Technical University of Crete
Chania, Greece
mpapadogiorgaki@mhl.tuc.gr

Pavlos Malakonakis
Electrical and Computer Engineering
Technical University of Crete
Chania, Greece
pmalakonakis@mhl.tuc.gr

Ioannis Papaefstathiou
Electrical and Computer Engineering
Aristotle University of Thessaloniki
Thessaloniki, Greece
ygp@ece.auth.gr

***Abstract*—An emerging trend in the biomedical community is to create models that take advantage of the increasing available computational power, in order to manage and analyze new biological data as well as to model complex biological processes. Such biomedical software applications require significant computational resources since they process and analyze large amounts of data, such as medical image sequences. This paper presents a novel FPGA-based system that implements a novel model for the prediction of the spatio-temporal evolution of glioma. Glioma is a rapidly evolving type of brain cancer, well known for its aggressive and diffusive behavior. The developed system simulates the glioma tumor growth in the brain tissue, which consists of different anatomic structures, by utilizing individual MRI slices. The presented innovative hardware system is more than 60% faster than a high-end server consisting of 20 physical cores (and 40 virtual ones) and more than 28x more energy efficient.**

***Index Terms*—High performance computing, Field programmable gate arrays, High level synthesis, Magnetic resonance imaging**

I. Introduction

Recently, the biomedical community develops more and more applications that require enormous computational power, which model complex real-world biological processes. At the same time, according to the World Health Organization (WHO), cancer is the second leading cause of death in developed countries and is among the three leading causes of death for adults in developing countries. The identification of cancer characteristics and the prediction of tumor growth can lead to useful insight into the disease dynamics, which can improve clinical outcomes. In particular, glioma is a rapidly evolving type of brain cancer, well known for its aggressive and diffusive behavior [1] that has lead several research efforts to explore its progression. Along this direction, several mathematical models of glioma growth have been developed, attempting to identify the interactions between the cancer cells and the tissue micro-environment, with the aim of predicting tumor's future spatial and temporal evolution [2].

Current modeling techniques follow either a discrete cell-based approach, or they are based on a continuum framework that covers the evolution of local cellular densities. Discrete models, such as the one in [3], simulate individual cell behavior but they are prohibitively computationally intensive and thus they are limited to the exploration of small tumors, below the level of clinical detection. On the other hand, continuum models, that describe tumors at macroscopic level as spatial distributions of cell densities, represent a promising, set of alternative approaches in exploring tumor progression dynamics [4], while additionally their computational requirements are substantially lower than those of the discrete models. Recently, hybrid models have also been developed so as to overcome certain limitations of the continuum and discrete approaches [5]; however, they are still constrained by significant computational costs. Alternatively, multi-compartmental continuum models can cover also heterogeneous populations and appear to be very effective in describing how sub-populations of various cell types proliferate and diffuse, while they overcome the computational deficiencies of the discrete models [6], [7]. According to those approaches, cells are grouped within each compartment based on their phenotype and/or their access to the necessary nutrients [8].

Cancer modeling software applications are very computationally intensive when trying to estimate in detail the future tumor growth, especially when they process and analyze large amounts of medical imaging data, such as MRI sequences. As an example, modeling applications may need, depending on the processing units utilized, over a week of processing time in order the simulate the tumor growth for some months; the acceleration of those predictions is very beneficial for both research and clinical purposes. From a research aspect, the model estimation should be available within a reasonable amount of time so that the researcher can directly inspect and verify the related results, in order to accordingly adapt the model, e.g. by adjusting parameter values, incorporating additional factors, etc. On the other hand, when the model is exploited in clinical practice for providing an assessment to the medical expert, the quick application's correspondence is of

978-3-9819263-4-7/DATE20/©2020 EDAA

great importance, since the clinician may simulate different therapy strategies simultaneously, so as to select the best treatment, or, depending on the results, the medical expert may decide that the patient should arrange the next follow-up examinations sooner than the predetermined time.

In this paper, a novel hardware system, implemented in Reconfigurable Logic, is proposed, which accelerates the execution of a state of the art multi-compartmental continuum model. This model, presented and analyzed in [7], simulates the spatio-temporal evolution of glioma tumors and it has already been sufficiently tested and evaluated. Due to the nature of the algorithm, its software implementation requires extensive computational resources so as to report detailed prediction results for the growth of the tumor for a period of a few months. Our hardware system accelerates the processing, over a high-end many-core server, by more than 60% while it consumes 28 times less energy.

II. Related Work

It should be stressed that no similar reconfigurable and/or parallel system to the one proposed in this paper (i.e. for predicting the tumor growth in the case of glioma), has been presented so far. So in this section we list highly parallel and FPGA-based systems, which implement similar biomedical tasks/sub-tasks and involve medical image processing (e.g. Magnetic Resonance Imaging (MRI)); obviously, there is no direct comparison that can be made between our system and those but we believe that mainly the FPGA-related results further demonstrate that FPGAs are a very promising approach for such biomedical systems.

The only known cancer prediction model implemented on a highly parallel system is the one presented by Adrian Kusek et al. [9]. it is a multi-GPU implementation of a three-dimensional (3-D) cancer prediction model which allows for the simulation of both the growth and the treatment phases of the tumor. Their application has been implemented in CUDA. The simulations were performed on computational nodes that consisted of two Intel Xeon 6 core CPUs, and were associated with 8 Nvidia TeslaTM GPGPU boards (each GPGPU utilizes 512 cores). Their simulations also run on another lower-spec system, which consists of two simple Nvidia Tesla K40 XL GPU cards and two Intel Xeon E5-2680v3 CPUs (consisting of 12 physical cores, 24 threads). The results showed that the total computational time for a typical simulation of the tumor growth with an initial diameter equal to $5mm$ ($250 \times 250 \times 250$, 2×10^4 timesteps), when utilizing 4 GPU boards, was approximately 15 min. For the simulation of a tumor of size $1.3cm$ ($750 \times 750 \times 750$) and $4x10^4$ timesteps the computational time needed was 5 hours on the high-spec system utilizing all 8 GPU boards and about 6 hours on the lower-spec processing system. No comparison can be made with our system since the biomedical approach, as well as the algorithms utilized are totally different, while our system is expected to be significantly more precise, due to the algorithm used.

There are also FPGA-based systems that perform real-time analysis of image-data; Richard M. Jiang and Danny Crookes [10] present a system that mainly implements a novel area-efficient high-throughput 3D discrete wavelet transform (DWT) architecture on an FPGA; the presented system has low silicon cost and it can perform a five-level 3D wavelet analysis for seven $128 \times 128 \times 128$ images per second.

Three-dimensional ultrasound computer tomography (3D USCT) promises high-quality breast images, but it is currently limited by a time-consuming synthetic aperture focusing technique for the image reconstruction; the basic idea is to accumulate many low-quality images, recorded by transducers at different geometric positions, in order to create one high resolution image. The system proposed by M. Birk et al. [11] accelerates this image reconstruction by utilizing a GPU and an embedded FPGA. The CPU-based implementation used an Intel Core i7-920 @2.67 GHz. The CUDA implementation was tested on a Nvidia Geforce GTX 580 and the hardware implementation on a Custom Data Acquisition (DAQ) system with 81 Altera Cyclone II FPGAs. The final results showed that the multi-threaded software, on the Intel Core i7 machine, using 8 threads and Hyper-threading, achieved 1.02 GVoxel/s. The FPGA achieved 1.6 GVoxels/s and the GPU up to 8.2 GVoxels/s.

In the work of Pagliari et al. [12] two alternative methods for breast cancer prediction were implemented on an ASIC and on an FPGA, namely the MIST Beamforming technique, which requires the execution of a set of simple operations on a massive amount of input data and the MUSIC-Inspired (MUSIC-I) method that processes a relatively smaller amount of data but requires the execution of significantly more complex calculations. The performance of their FPGA implementations has been compared with that of the optimized software running on an ARM Cortex-A9 CPU at 667 MHz while their ASIC results were evaluated against the execution time of the optimized software on an ARM Cortex-A9 implemented on a 32-nm CMOS technology. Their FPGA accelerators are faster than the ARM CPU by up to 25x, while for the ASIC the speedup is 160x for the MIST and 2000x for MUSIC-I.

Finally, the usage of MRI has intrigued the scientific community, as it is a non-invasive procedure. A lot of work has been done in order to extract edge-defined and noise-free MRI scans. In [13] an FPGA-based implementation of a generalized parallel 2-D MRI filtering algorithm is proposed. The proposed architecture working at a frequency of 230MHz has been implemented in two Xilinx Virtex-6 FPGA boards.

III. Model Description

The implemented model falls in the category of multicompartmental continuum approaches and focuses on predicting the spatio-temporal evolution of glioma tumors in an isotropic and heterogeneous brain tissue, which consists of different anatomic structures. The brain tissue heterogeneity is incorporated through the variable diffusion coefficient according to the different expansion rates depending on the invaded area. Looking at the process of tumor growth, in the primary stage, its size and density are small and all cells are sufficiently supplied with nutrients, namely oxygen and glucose [14]. However, as time proceeds, the tumor radius increases and the concentrations of nutrients decrease in its central part until they fall below critical levels that are not sufficient to meet the needs of all cells. Then, hypoxic and/or hypoglycemic regions appear in the core of the tumor as analytically described in [7]. When nutrients are depleted, hypoxic/hypoglycemic cells start dying and turn to necrotic, forming a necrotic region at the tumor core. Eventually tumor consists of different regions, namely the necrotic core, the hypoxic and/or hypoglycemic zones around it and the outer proliferating area.

In our modelling approach, the different cell and chemical compartments (cell densities and concentrations of chemical ingredients), which constitute the building blocks of the model, employ continuum variables and express spatial concentration changes subjected to diffusion-reaction equations, i.e. second order partial differential equations that can be solved using numerical methods. In particular, tumor growth is based on diffusion and proliferation, depending on the nutrients availability, i.e. oxygen along with glucose. Four different cancer cells phenotypes are involved (i.e. the proliferative, the hypoxic, the hypoglycemic and the necrotic) with different metabolic profiles, as well as proliferation and invasion properties. In the implemented model, the proliferative cells are converted to hypoxic and/or hypoglycemic under oxygen and/or glucose inadequacy respectively, which are then turned to necrotic due to the nutrients depletion. The invasions of proliferative hypoxic and hypoglycemic cancer cells are modeled using the diffusion-reaction equations that are described in detail in [7], while the change in the necrotic cells-density is expressed by a simple ordinary differential equation [7]. The concentration-changes of nutrients (oxygen and glucose) are similarly modelled by diffusion equations.

The formed linear system is expanded in two spatial dimensions and the differential equations are numerically approximated by Finite Differences, while the implicit scheme of Forward Euler is used in order to derive the direct solutions, which are estimated by iteratively calculating the next-time approximation of spatial concentrations (at each tumor point and time instance) [7].

IV. System Architecture and Design Optimizations

Our model is applied on individual MRI slices of a real glioma tumor, that has been segmented from a MRI sequence taken from the RIDER-NEURO database [15] and is then embedded in the brain anatomic structure derived from the SRI24 atlas [16] (i.e. an MRI-based atlas of normal human brain anatomy [8]). Each slice of the final MRI sequence has a size of 240 x 240 pixels and a resolution of $1mm/pixel$. The initialization of the model's variables has been performed using literature-based reference values taken from the model design presented in [7].

The modeling process consists of three stages. First, a single slice of an MRI is read and pre-processed. In this prepossessing stage nine different Read/Write buffers, with sizes equal to that of the input image, are utilized; each of these buffers contains a set of attributes of the brain structure, such as the proliferative and hypoxic cells density, the oxygen and glucose concentration, etc. In the second stage, the contents of those buffers are sent to the model core and the simulation of the tumor growth prediction is executed. Finally, in the third stage, the results of the model, which are placed on the output buffers, are used to reconstruct the final image. In order to allow for full pipelining, we use double-buffering so in each time-step the Read/Write buffers are swapped. In more detail, the eight buffers, out of the nine, are double-buffered while one buffer is read-only. Based on an analysis of the execution time of each step we decided that the pre-processing and the image reconstruction tasks (i.e. steps 1 and 3) are executed on a CPU, while the computationally intensive model calculations are implemented in reconfigurable hardware. Thus a software application was developed, using the Xilinx SDK tool (Integrated Design Environment for creating embedded applications on any of Xilinx's microprocessors) for the pre-processing and the post-processing. The same application feeds the data to the DDR memory of the platform and after that the application starts the Hardware Accelerator. In the final stage, the same application reconstructs the final image by reading the data resulted by the model's calculations.

Thus, after the pre-processing task, performed in the CPU,

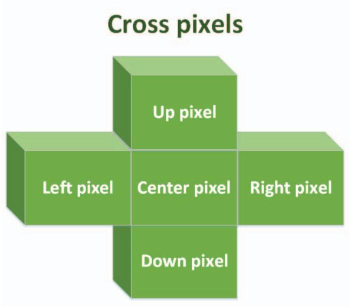

Fig. 1. 2D cross pixels positions.

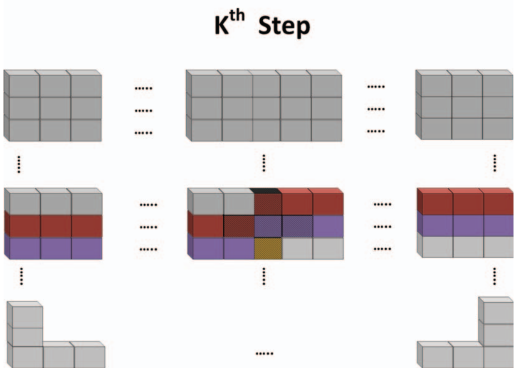

Fig. 2. Line Buffering and Streaming procedures. The red and purple are line buffers stored in the on-chip memories, the dark yellow is the streamed pixel.

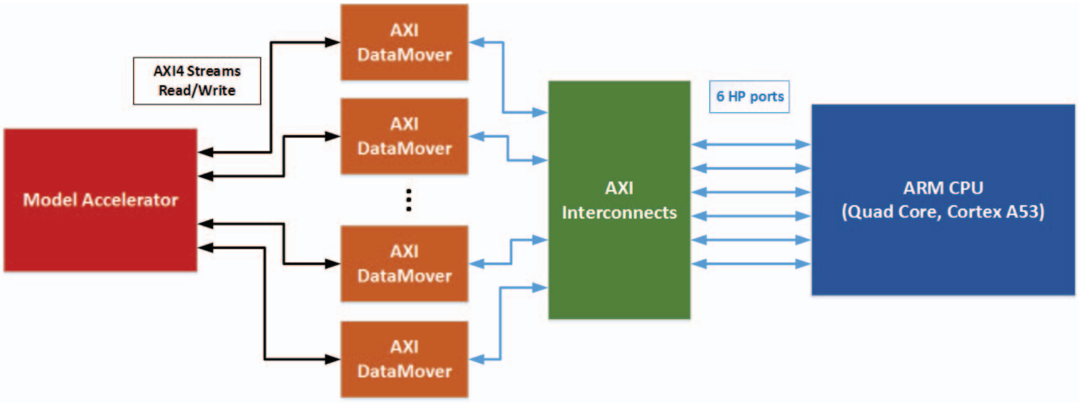

Fig. 3. System Architecture.

the data have to be transferred to the hardware accelerators. In order to get the maximum performance from the underlying hardware, the data are streamed by using the AMBA AXI4-Stream protocol from the platform's DDR memory, while the double buffering technique utilized allows for the full interleaving of the data transfers and the actual model processing.

For every output pixel, the actual model calculations need the data from five different pixels from the corresponding input buffer. Specifically these pixels are the ”2D cross-pixels”, shown in Fig. 1. As a result, in order to increase the locality and thus the performance of our processing core we employ ”line buffering”; we store in the on-chip low latency memories two lines of pixels, at a time. From the information needed, in order for the output pixel to be calculated, the four pixels are, at any given time, found in those memories, while the last pixel is transferred by the corresponding input stream. At the end of the calculations, every line buffer is updated and ”moves” through the entire corresponding input, as shown in Fig. 2. In this figure the red and purple pixels are stored in the two different on-chip line buffers while the dark yellow pixel is transferred through the AXI4 Stream in each cycle.

Moreover, an extra on-chip memory was utilized, in order to reduce the latency of the pipeline stages to one clock cycle. Each pipeline stage required two clock cycles, since each line buffer is read two times and written once, in every iteration, and the on-chip memories have only two ports. Every line buffer is copied to an additional on-chip memory and a duplicate of the buffer is created; those extra line buffers provide enough ports for the requiered I/O and the whole system is clocked at 150 MHz. The Post-Place and Route resources utilization are shown in Table I.

The high-level system architecture, shown in Fig. 3, consists of the Model Accelerator, nine AXI DataMovers, the AXI Interconnect and the ARM Cortex A53 Processing System. The Accelerator (red color) is connected to the nine DataMovers (orange color) by utilizing AXI4 Streams. In particular, each DataMover is connected with four AXI4 Streams to the Accelerator, two streams for Read and Write transactions and two streams for the commands. The DataMovers, also, drive the six slave High Performance Ports of the ARM CPU.

TABLE I
RESOURCE UTILIZATION REPORT FOR THE ENTIRE SYSTEM.

	Block RAM Tile	DSP48E	FF	LUT
Available	912	2520	548160	274080
Utilized	155.5	1421	169994	120651
Utilization (%)	17.05	56.39	31.01	44.02

V. EVALUATION AND VALIDATION

The performance and the energy efficiency of our hardware system are compared to those of a High-end Server. The reference Server is a Dell PowerEdge R530 with two Intel Xeon E5-2630v4 CPUs running Centos 7. The CPUs are clocked at 2.2GHz, have 25 MB Cache and Max Turbo Boost frequency of 3.1 GHz. Each CPU has 10 physical cores and 20 virtual ones/threads (i.e. Intel Hyper-threading). The Server includes the iDRAC v8 EE module, which is used to accurately measure its power consumption [17]. The initial code was a single-threaded C software, but also an optimized multi-threaded version of the reference software was developed using OpenMP.

Our novel system was implemented on Trenz's TE0808 UltraSOM platform which is built around the Zynq UltraScale+ *xczu9eg-ffvc900-1-i-es1* FPGA. The tool used for the implementation of the entire system is Xilinx Vivado 2016.2, and we developed our accelerator using Vivado HLS. Several experiments with different tumor growth simulation times have been conducted, to evaluate the functionality and the overall performance and energy consumption of our system. The power consumption has been measured using the on board monitoring circuitry that provides information of the entire platform, in real-time.

In order to be as precise as possible, we measure the overall energy consumption for the complete tumor growth simulation as well as for each iteration (i.e. time-step) of our simulations. Then the energy efficiency of the FPGA vs the Server is calculated by dividing the energy consumption, per timestep, of the FPGA-board over the energy consumption of the Server.

TABLE II
Hardware and Software Execution Time in seconds, Runtime Speed Up, Power in Watt and Energy Consumption in Kilojoule for 30 days simulation for different number of threads

SW Threads	SW/HW Execution Time (s)	HW Speed Up	SW/HW Power (W)	SW/HW Energy (kJ)
1	183.653	18.54	131	24.05
2	107.982	10.90	139	15.01
4	66.366	6.70	149	9.88
8	38.128	3.85	161	6.13
10	31.823	3.21	169	5.37
12	28.058	2.83	178	4.99
20	19.622	1.98	208	4.08
24	19.025	1.92	210	3.99
40	**16.254**	**1.64**	**215**	**3.49**
48	18.863	1.90	213	4.01
HW	9.9	-	25	0.24

TABLE III
Hardware Accelerator execution times vs. Software, in different model simulation times

Sim. Days	Iterations	Execution Time (seconds)		
		HW	SW(8-threads)	SW(40-threads)
30 days	30000	9.9	38.1	16.3
90 days	90000	29.4	113.2	48.2
180 days	180000	59.5	229.1	97.6
365 days	365000	120.3	463.2	197.3

A. Software and FPGA-based System Performance

The software experiments were conducted with different numbers of threads and tumor growth simulated times. The time-step selected is 0.001 days (a typical value for such simulations triggering accurate results) and the spatial grid dimensions are $dx = dy = 1mm$. The software was benchmarked with the simulation of 30 days of tumor growth (30000 time-step iterations), which requires around 3 minutes of processing time in the Server. Then we increase the tumor growth simulation time and we realize, as expected, that the execution time grows linearly to the increase of the simulation time. We then parallelize our software, so as to utilize multiple threads. The results in terms of the execution time and energy consumption for different number of threads, are shown in Table II.

As clearly demonstrated, the most energy efficient point is when we have 40 threads. In this case, the Server uses all the available virtual cores and the overall execution time is the lowest; more threads lead to higher execution times at the same energy cost while less threads increase the execution time at a much higher rate than they reduce the power consumption. In the hardware experiments, the model settings are the same as those used in the software ones, in order to fully verify and evaluate our novel system. The hardware-accelerator execution time, is exported for different tumor-growth simulation times.

B. Performance and Energy Comparison

The tumor-growth simulation times ($tmax$) for both hardware and software are set to different numbers of days, where each day needs 1000 iterations. The execution times for both software and hardware implementations, for those sets of iterations, are shown in Table III; for the software we list the 8-threads and 40-threads numbers because they are representative for a high-end desktop (i.e 8-threads) and a high-end Server (i.e. 40-threads). This Table demonstrates that the execution time of the hardware accelerator, just like the software, scales up linearly with respect to the number of iterations and as a result to the tumor-growth simulated time.

TABLE IV
System Energy per iteration.

	Accelerator	Software 40 threads
Runtime - 1 Iteration (Seconds)	0.000165	0.0005418
Power (Watts)	25	215
Energy (Joules) / Iteration	0.0041	0.1164
Efficiency	**28.4**	-

In Table II we also demonstrate the speed up triggered by our hardware system when compared with the software execution in the server. The tumor-growth simulated time for all those measurements is 30 days (30000 iterations), and our FPGA-system needs **9.9 seconds** to complete the entire simulation. We would like to note here that the time spent for sending the data to the FPGA is not factored, since we do not measure this latency in the server either. Additionally, the amount of data (initial and final MRI) is rather small making the transfer overhead through the network negligible (milliseconds). Finally, the pre- and post-processing tasks are handled by the ARM A53 processor. As expected the lowest speedup is triggered when the Software is executed in 40 simultaneous threads, and this is **1.64x**.

As analyzed at the beginning of this section, each platform is evaluated by its energy efficiency. Table IV shows the energy efficiency of our novel hardware system against the 40-threaded software. **The final energy efficiency calculated is 28.4x over the High-end Server**. The performance speedup and the energy efficiency for different number of threads are also shown in Fig 4. Note that in all the cases we measure the overall power consumption of the two platforms including the CPUs, the FPGAs, the DDR4 Memories, etc.

C. Validation

In order to fully validate the results produced by our hardware system, multiple experiments are conducted and the results are compared to those triggered by the underlying software. In all the cases, the generated picture by our hardware system is identical, based on a pixel-by-pixel comparison, to that produced by the reference software implementation.

Fig. 5 shows the results from the simulation of an actual tumor. As it is demonstrated, the first column (initial tumor) contains only proliferative cells (white), while after one month (second column) a hypoglycemic cell-population appears in the central part of the tumor (dark gray), surrounded by a hypoxic region (gray). Three months after the detection (third column), a necrotic core (darker gray) has already been developed surrounded by a hypoxic along with a hypoglycemic zone and the outward proliferative region. Six months after the

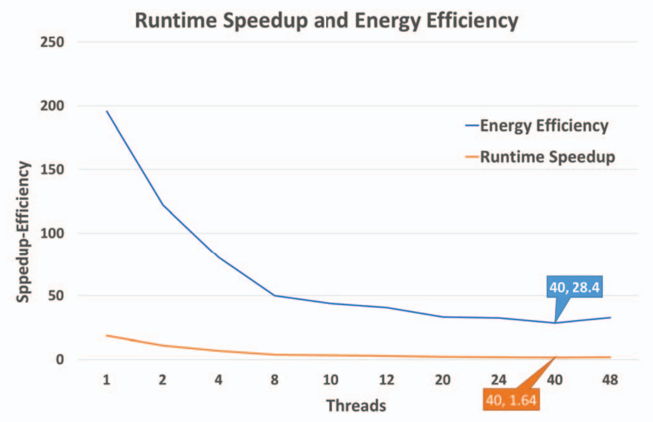

Fig. 4. Performance Speedup and Energy Efficiency of Reconfigurable hardware system vs. High-end server.

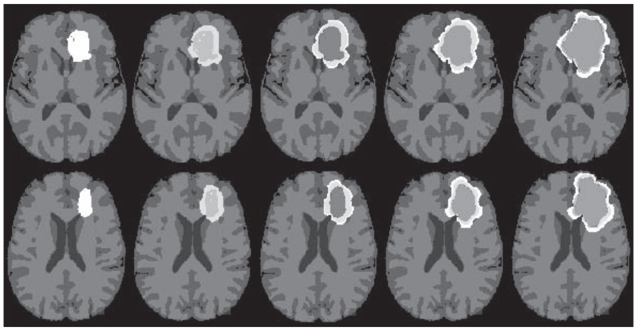

Fig. 5. Tumor growth simulation results on a medium-diffusion / medium proliferation tumor. Lines: two indicative frames. Columns: Initial tumor (1st), evolution after: 1 (2nd), 3 (3rd), 6 (4th) and 12 (5th) months.

assumed detection (fourth column) the tumor has significantly expanded, while the discrete zones are depicted. Finally, after one year (fifth column), the tumor reaches a steady state, where the necrotic region occupies most of the tumor body.

VI. Conclusions

Modeling and prediction of tumor's growth represents a very important, CPU demanding class of biomedical applications, which utilize initial tumors' MRI scans and tumors' evolution is depicted on the output images. In this paper, we demonstrate a novel FPGA-based system executing the complete application, which utilizes a relatively small FPGA with an embedded ARM CPU. As our real-world experiments show, our low cost system is 28 times more energy efficient than a many-core server executing the same application, while also being at least 60% faster.

The modeling approach implemented in the proposed FPGA-based system is a rather simple 2D one, dealing with individual and relatively low-resolution MRI slices. Thus the execution times demonstrated are indicative of the acceleration triggered by implementing such an approach in reconfigurable hardware (e.g. a 19x speedup vs. single-thread execution). The accelaration and energy reduction become more meaningful and significant when the execution times are considerably higher, namely for larger prediction intervals and/or for more complex models incorporating more chemical factors, higher images' resolution and model's dimensionality, i.e. whether it handles 2D (single slices), or 3D (entire sequences), as well as for large number of patients. Hence, the respective 3D modeling applications that include additional influence factors and/or the effect of different therapeutic approaches (which utilize the same basic algorithm but with many more interdependent variables and matrices) are extremely slow when executed on software, even on high-end servers, requiring several hours, even days for a single tumor growth prediction. As a result, the acceleration of such an application is certainly a priority for both clinicians and researchers in this field.

References

[1] M.D. Szeto, G. Chakraborty, and J. Hadley, "Quantitative metrics of net proliferation and invasion link biological aggressiveness assessed by MRI with hypoxia assessed by fmiso-pet in newly diagnosed glioblastomas," Cancer Res., vol. 69, no. 10, pp. 4502–4509, 2009.

[2] H. L. P. Harpold, E. C. Alvord, and K. R. Swanson, "The evolution of mathematical modeling of glioma proliferation and invasion," J. Neuropathol. Exp. Neurol., vol. 66, no. 1, pp. 1–9, 2007.

[3] H. Hatzikirou, and A. Deutsch, "Cellular automata as microscopic models of cell migration in heterogeneous environments," Curr. Top. Dev. Biol., vol. 81, pp. 401–434, 2008.

[4] K. R. Swanson, C. Bridgea, J. D. Murray, and E. C. Alvord, "Virtual and real brain tumors: using mathematical modeling to quantify glioma growth and invasion," J. Neurol. Sc., vol. 216, pp. 1–10, 2003.

[5] A. R. A. Anderson, "A hybrid mathematical model of solid tumour invasion: the importance of cell adhesion," Math. Med. Biol., vol. 22, no. 2, pp. 163–186, 2005.

[6] K. R. Swanson, R. C. Rockne, J. Claridge, M. A. Chaplain, E. C. A. Jr, and A. R. A. Anderson, "Quantifying the role of angiogenesis in malignant progression of gliomas: In silico modeling integrates imaging and histology," Int. Sys. Tech. Math. Onc. Cancer Res., vol. 71, no. 24, pp. 7366–7375, December 2011.

[7] M. Papadogiorgaki, P. Koliou, X. Kotsiakis, and M. E. Zervakis, "Mathematical modelling of spatio-temporal glioma evolution," Theor. Biol. Med. Model., vol. 10, no. 1, pp. 47–83, July 2013.

[8] M. Papadogiorgaki, P. Koliou, and M. E. Zervakis, "Glioma Growth Modeling based on the Effect of Vital Nutrients and Metabolic Products," Springer Med. Biol. Eng. Comput. pp. 1–15, March 2018.

[9] Kusek, A., o, M., Paszyski, M., & Dzwinel, W. (2018). Efficient model of tumor dynamics simulated in multi-GPU environment. The International Journal of High Performance Computing Applications. https://doi.org/10.1177/1094342018816772

[10] R. M. Jiang and D. Crookes, "FPGA implementation of 3D discrete wavelet transform for real-time medical imaging," 2007 18th European Conference on Circuit Theory and Design, Seville, 2007, pp. 519-522.

[11] M. Birk et al., "Acceleration of image reconstruction in 3D ultrasound computer tomography: An evaluation of CPU, GPU and FPGA computing," Proceedings of the 2011 Conference on Design & Architectures for Signal & Image Processing (DASIP), Tampere, 2011, pp. 1-8.

[12] Daniele Jahier Pagliari, Mario R. Casu, and Luca P. Carloni. 2017. Accelerators for Breast Cancer Detection. ACM Trans. Embed. Comput. Syst. 16, 3, Article 80 (March 2017), 25 pages.

[13] S. Hasan, S. Boussakta, and A. Yakovlev, Fpga-based architecture for a generalized parallel 2-d mri filtering algorithm, American Journal of Engineering and Applied Sciences, vol. 4, no. 4, pp. 566575, 2011.

[14] K. A. Rejniak, and A. R. Anderson, "Hybrid models of tumor growth," Wiley Inter. Rev., vol. 3, no. 1 pp. 115–125, 2010.

[15] The cancer imaging archive - rider neuro mri, https://wiki.cancerimagingarchive.net/display/Public/RIDER+NEURO+MRI

[16] Sri24 atlas: Normal adult anatomy, https://www.nitrc.org/projects/sri24

[17] Dell Remote Access Controller and iDRAC, https://en.wikipedia.org/wiki/Dell_DRAC