2014 IEEE International Conference on Control System, Computing and Engineering, 28 - 30 November 2014, Penang, Malaysia

Cardiac Excitation Modeling: HDL Coder Optimization towards FPGA stand-alone Implementation

Norliza Othman[1,2,4], Farhanahani Mahmud[1,2,5], Abd Kadir Mahamad[1,6], M. Hairol Jabbar[1,3,7] and Nur Atiqah Adon[1,2,8]

[1]Faculty of Electrical and Electronic Engineering, University Tun Hussein Onn Malaysia (UTHM), Batu Pahat, Johor, Malaysia.
[2]Cardiology and Physiome Analysis Research Laboratory
[3]Manycore System on Chip (MCSoC) Research Laboratory
Microelectronics and Nanotechnology – Shamsuddin Research Centre (MiNT-SRC), UTHM, Batu Pahat, Johor.
{[4]ge130062, [8]ge130013}@siswa.uthm.edu.my {[5]farhanah, [6]kadir, [7]hairol} @uthm.edu.my

***Abstract*— The aim of this paper is to discuss the optimization of the hardware description language (HDL) design using fixed-point optimization and speed optimization through a pipelining method. This optimization is very crucial to achieve the best performance in terms of speed, area and power consumption of the generated HDL code before deploying the field programmable gate array (FPGA) stand-alone implementation. As computational mathematical modeling needs immense amounts of simulation time, FPGA could bring the solutions as it provides high performance, and able to perform real-time simulations and compute in parallel mode operation. In this study, in order to ease verification, prototyping, and implementation FPGA, rapid prototyping model-based design approach of HDL Coder from MathWorks has been used to automate HDL codes generation from a designed MATLAB Simulink blocks of Luo-Rudy Phase I (LR-I) model towards FPGA hardware-implemented for numerical solutions of ordinary differential equations (ODEs) responsible in generating the action potential (AP) waveform of mammalian cardiac ventricle cell. By using HDL Coder, the model is successfully converted into an optimal fixed-point VHDL design and the operating frequency is increased from 9.819 MHz to 23. 613MHz by pipelining optimization.**

***Index Terms*— FPGA hardware-implemented cardiac excitation modeling, Luo-Rudy Phase-I model, HDL Coder, fixed-point optimization, pipelining optimization.**

I. Introduction

The mathematical modeling computation by using hardware was used to generate action potential in order to study the underlying mechanism of the human ventricular cell since it is not associated with experimental problems [1]. However, the focus on developing minimum power consumption in hardware prototypes has been shown to boost up most. Therefore, one way to achieve reduction on the power consumption as well as size and to develop high performance hardware is to implement the design using Very Large Scale Integration (VLSI) technology. Meanwhile, with the reliability requirements of biomedical instruments, field programmable gate array (FPGA) and embedded system development showing a trend of growth. FPGA technology is now considered very useful by an increasing number of designers in various fields of application due to confidentiality of the algorithm and architecture, capable to meet many constraints for space applications, and it can be adapted to any changes in design by dynamic reconfiguration [2]. However, the traditional workflow of designing the hardware by using FPGA which need to hand-written the code was very time consuming, tedious and error prone [3]. Besides, the mathematical modeling of heart model that contains ordinary differential equations (ODEs) with hundred or millions of lines of register transfer level (RTL) code is very complex to handle when using manual coding method. Whereas, automatic code generation lets designers to make changes in the system level model and produce an updated Hardware Description Language (HDL) in a short time.

Therefore, model-based design such as HDL Coder from the MATLAB Simulink offers great opportunities in effectively improve system reliability and also reduce the development time by 33% as compared to manual coding [4] and cost to 1/5 up to 1/2 of that traditional methods [5]. In addition, MATLAB model-based design facilitates creation of FPGA-based prototypes and automates HDL code verification by co-simulating with Simulink and optimizes the models to meet speed, area and power consumption on FPGA. In brief, by using the model-based design, designers can spend more time on fine tuning algorithms such as model modification, hardware and software co-simulation for verifications, and experimentation and less time on learning and writing HDL code.

However, HDL Coder does not support floating-point data types, therefore, it is necessary to convert the floating-point model into fixed-point model to reduce hardware resources [4]. Moreover, floating-point data types are inefficient for hardware realization [6]. The HDL coder provides an automated workflow for floating-point to fixed-point conversion by using Fixed-point Advisor method which able to simplify and accelerate the conversion process as the conversion using a conventional method can be very challenging and time-consuming, typically demands 25 to 50 percent of the total design and implementation time [4]. Fixed-point optimization can also be done by choosing suitable word length (WL) and

fraction length (FL) to achieve the optimum level of the VHDL code design. Moreover, HDL coder also offer speed optimizations such as pipelining methods. The HDL Workflow Advisor highlights the critical path timing in MATLAB Simulink to help identify speed bottlenecks that can cause wide-range response time variations and limit the overall system performance [7] and to improve the performance of the design. Through pipelining method a higher maximum clock rate can be achieved by inserting multiple inputs and output pipeline registers at strategic points at various stages of the designed model to break the critical path.

Optimization through HDL Coder introduces several optimization such as RAM Mapping, Pipelining, Pipelining Distribution, Resource Sharing, and Loop Streaming to achieve optimum performance according to design purpose, where the applicable optimization process for this research are fixed-point and pipelining method. Therefore, in this paper, fixed-point and speed optimization through HDL Coder are presented to achieve optimum performance of FPGA implementation for Luo-Rudy Phase-I cardiac modeling.

The structure of this paper is as follows. Methods used for the HDL coder model-based design of cardiac excitation modeling are presented in section II. Research findings and discussions on fixed-point and pipelining optimization, and stand-alone FPGA hardware implementation are presented on the section III. Lastly, concluding remarks are given in the last section summarizing the contribution of this paper.

II. HDL Coder Model-Based Design of Cardiac Excitation Modeling

The design of software simulations of LR-I mathematical modeling by using MATLAB Simulink is made mainly to solve a set of nonlinear ODEs as in Eq. 1, Eq. 2, Eq. 3, Eq. 4, Eq. 5, Eq. 6, Eq. 7, and Eq. 8 to generate action potential (AP) in a single mammalian cardiac ventricular cell. Here, the MATLAB Simulink blocks are designed to represent six ion currents of sodium, potassium, calcium and chloride as in Eq. 9, in which the inflow and outflow of these currents will cause changes to the membrane voltage, V_m and the AP will be generated as an external stimulation current, I_{ext} is applied to the cell [1, 2, 8].

$$\frac{dV_m}{dt} = -\frac{1}{C_m}(I_{ext} + I_{ion}) \quad (1)$$

$$\frac{dm}{dt} = \alpha_m(1-m) - \beta_m(m) \quad (2)$$

$$\frac{dh}{dt} = \alpha_h(1-h) - \beta_h(h) \quad (3)$$

$$\frac{dj}{dt} = \alpha_j(1-j) - \beta_j(j) \quad (4)$$

$$\frac{dd}{dt} = \alpha_d(1-d) - \beta_d(d) \quad (5)$$

$$\frac{df}{dt} = \alpha_f(1-f) - \beta_f(f) \quad (6)$$

$$\frac{dx}{dt} = \alpha_x(1-x) - \beta_x(x) \quad (7)$$

$$\frac{d[Ca]i}{dt} = -10^{-4} \times I_{si} + 0.007(10^{-4} - [Ca]i) \quad (8)$$

$$I_{ion} = I_{Na} + I_{si} + I_K + I_{K1} + I_{Kp} + I_b \quad (9)$$

V_m : cardiac cell membrane voltage
C_m : membrane capacitance
I_{ion} : ionic current consist of six kind of ion currents
m : activation gate of fast sodium current, I_{Na}
h : inactivation gate of fast sodium current, I_{Na}
j : slow inactivation gate of fast sodium current, I_{Na}
d : activation gate of slow inward current, I_{si}
f : in activation gate of slow inward current, I_{si}
x : activation gate of time-dependent potassium current, I_K
Ca_i : Calcium uptake

In implementing the MATLAB HDL Coder for FPGA hardware implementation of cardiac excitation modeling, the MATLAB Simulink blocks need to be modified to match conditions required by the HDL Coder. Firstly, conversion from the continuous-time to a discrete-time modeling must be done and all blocks used in the designed model must be from the HDL supported blocks which known as *hdlsupported* library. Floating-point data type to fixed-point data type conversion is also needed as stated in the previous works [9, 10, 11] in which the fixed-point data type will be set using a suitable WL and FL. Here, the WL and FL can be defined as the number of bits in the representation of signals and the scaling or binary point location represents the number of bits of the integer part, respectively. Besides, exponential and logarithmic functions in the model must be presented in lookup tables to avoid bits overflow in the fixed-point.

Secondly, fixed-point optimization need to be accomplished in order to achieve good results in term of number of slice registers, maximum operating frequency, and power consumption. This optimization process can be performed by using manual setting or automatic tool provided by MathWorks which known as Fixed-point Advisor tool. The fixed-point optimization is a process of achieving optimum level of the design by allocating a suitable WL and FL where in a fixed-point domain, a pair of the WL and FL is considered for each of the parameter in algorithms. Larger WL and FL will result in better performance and lower Bit Error Rate (BER) but the design will consumes larger resources and thus requires more expensive FPGA boards [14]. On the other hand, smaller WL and FL will result in larger BER but smaller footprint. Therefore, the optimization of the WL and FL is very crucial issue for FPGA hardware implementations. During fixed-point optimization process, by using Fixed-point Advisor from the HDL Coder, appropriate FL and WL applicable to the design model will be proposed. In addition, the WL and FL values proposed by the Fixed-Point Advisor also could act as a reference to improve the optimization further by manually reduce the proposed WL and FL while the accuracy of the simulation results are still maintained.

Lastly, speed optimization through HDL Coder can be performed by using pipelining method [15, 16]. To accomplish this optimization, a communication link between ISE Design

suite 14.6 software and MATLAB must be done through hdlsetuptoolpath. Then, through HDL Workflow Advisor, FPGA is selected as the target device workflow and Xilinx ISE as the synthesis tool in order to complete the pipelining optimization process. Since the pipelining optimization is to achieve the best maximum frequency [17], therefore the input pipelining registers and output pipelining registers is inserted to the critical path which is highlighted in light blue color as shown in Fig. 3, where the same process is repeated until a maximum frequency is obtained.

III. Results and Discussion

A. Floating-point to Fixed-point Conversion

Figure 1 shows the simulation result of action potential wave generation after the fixed-point conversion and optimization of the LR-I model, which is comparable to the results from previous studies [1, 2, 7, 12, and 13].

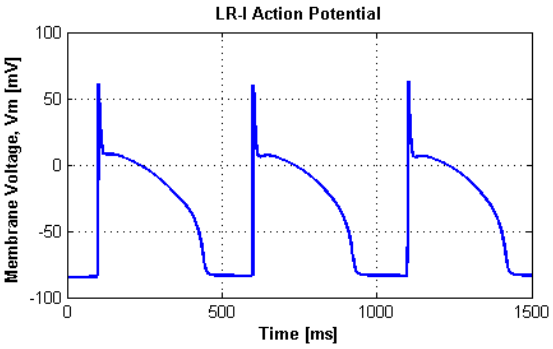

Fig.1: Action potential of Luo Rudy Phase-I model

Figure 2 shows the comparison of the LR-I action potential waveform between the simulation using the floating point and the fixed-point with the WL and FL values of 36 and 22, respectively. Panels (a), (b), and (c) indicate the floating point, fixed-point and differences of the floating point and the fixed-point, respectively. According to Fig. 2 (c), there are only very small differences between the results from the floating point and the fixed-point with the maximum value of 7.87 mV. This shows that the LR-I algorithm designed for the FPGA implementation is significant which is able to emulate comparable results. The MATLAB Simulink blocks of the LR-I HDL design model is depicted in Fig. 3.

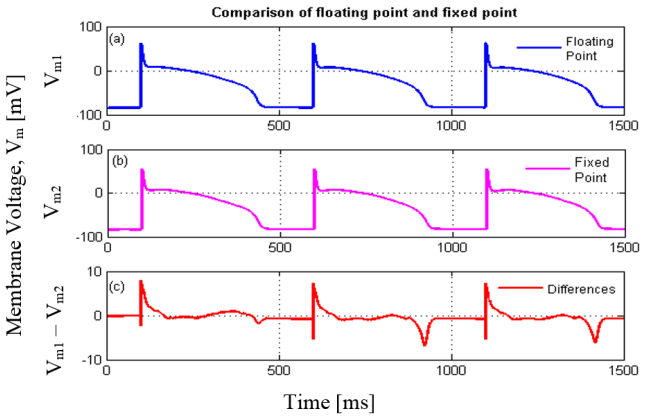

Fig. 2: Comparison of floating point and fixed-point of LR-I model. Panels (a), (b), and (c) indicate the simulation results of action potential wave generation in the floating point, in the fixed-point, and the differences of floating point and fixed-point, respectively.

B. Fixed-point Optimization

From the optimization process, three optimum values of fixed-point represented as (WL, FL), which are (60, 40), (50, 30) and (36, 22) are obtained. These values indicate the maximum values that are being used for all signals in the VHDL codes. According to each of these values of fixed-point, three optimized VHDL codes are generated through HDL Workflow Advisor in HDL Coder to analyze the area consumption in terms of slice registers and slice LUTs, maximum frequency and power consumption.

Table I shows the analysis results of the HDL design model performance based on a target board of FPGA Virtex-6 XC6VLX240T ML605 Development board that have 301440 of slices registers (flip flops) and 150720 of slice LUTs. By referring to Table I, the operating frequency is basically increased with the reduction in WL and FL of fixed-point optimization. For the fixed-point of (60, 40), the maximum frequency indicates 5.946 MHz, and when it is reduced to (50, 30), the maximum frequency increase to 5.984 MHz and the frequency goes up to 9.819 MHz when it is reduced further to (36, 22). This gives 39% for overall increment of the operation frequency which indicates the best optimization for the cardiac model. Meanwhile, the value of power decreases gradually from 40 mW for fixed-point of (60, 40) to 30 mW for fixed-point of (50, 30) and then to 29 mW for fixed-point of (36, 22). Number of slice LUTs also is influenced by the reduction of WL and FL, where the number of slice LUTs with the fixed-point of (60, 40), (50, 30) and (36, 22) is 14.38%, 10.25% and 5.35%, respectively. Whereas, the slice registers for previous optimizations also gradually reduces from 0.49%, 0.32% and 0.27%, respectively.

From this results, , it can be deduced that the fixed-point optimization reduces the area consumption in terms of slice registers and slice LUTs which leads to the low power consumption and maximizes operating frequency. Maximizing the operating speed is a crucial step in the design in order to run with the real-time reconfigurable hardware [4].

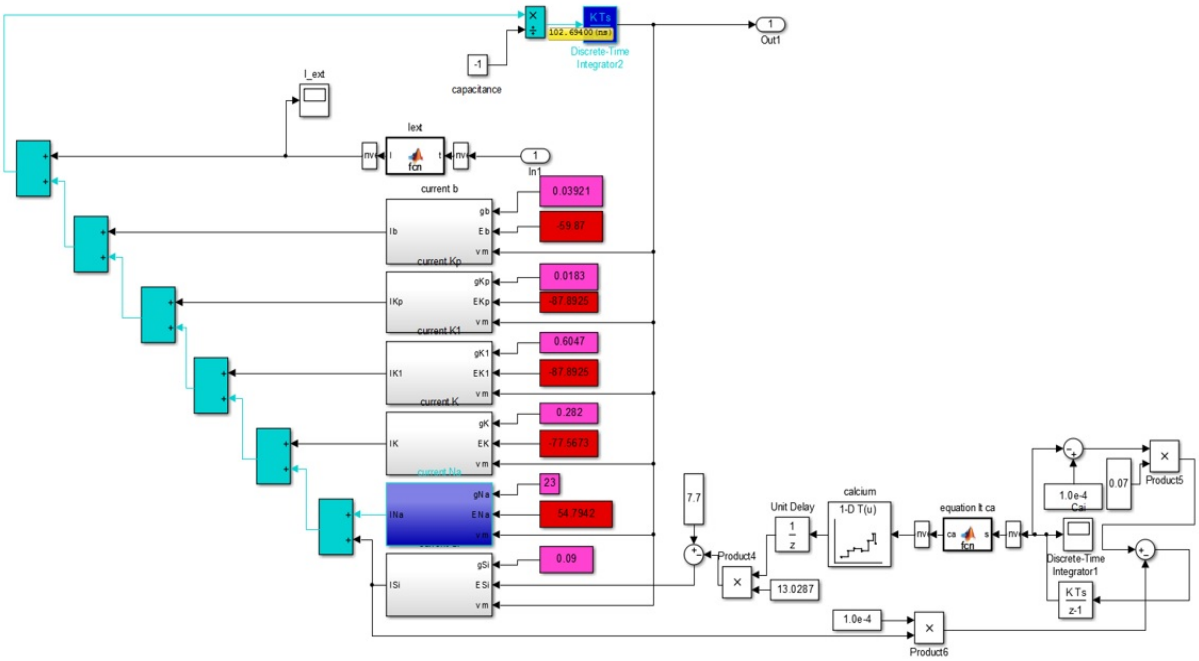

Fig. 3: Luo-Rudy Phase-I model by using MATLAB Simulink

TABLE I. COMPARISON OF AREA, SPEED AND POWER CONSUMPTION ON VIRTEX-6 FPGA BOARD VARIOUS OF WORD LENGTH AND FRACTION LENGTH

Board	Xilinx Virtex-6 ML605 Development board		
Fixdt(1,WL,FL)	**Fixdt(1,60,40)**	**Fixdt(1,50,30)**	**Fixdt(1,36,22)**
Number of Slice Registers	1, 479 (0.49%)	967 (0.32%)	810 (0.27%)
Number of Slice LUTs	21, 672 (14.38%)	15, 452 (10.25%)	8, 063 (5.35%)
Operating frequency (MHz)	5.946	5.984	9.819
Power (mW)	40	30	29

C. Pipelining Optimization

Table shows the results of pipelining optimization According to the results, the maximum frequency before inserting any pipelining is 9.819 MHz with first critical path of the divisions block given as 102.694 ns. Through repetition process of inserting pipelining registers on input and output of critical path blocks, the maximum frequency gradually increased until it reaches 3.613 MHz maximum speed with the reduction of the critical path on block division to 47.249 ns.

Based on Table II, the maximum operating frequency and the power is increased by 140% and 167%, respectively after the pipelining optimization process. This shows that higher maximum frequency will expand the chip area and power. Therefore, a compromise between frequency and power is important to achieve performance targets based on design requirements. Meanwhile, the number of slice registers increase from 0.27% up to 0.80% when the pipelining is inserted. The number of slice LUTs also increased from 5.35% up to 5.86%. This shows that the pipelining expand the area of the design model.

TABLE II. COMPARISON OF AREA, SPEED AND POWER CONSUMPTION ON VIRTEX-6 FPGA BOARD AFTER PIPELINING OPTIMIZATION

Board	Xilinx Virtex-6 ML605 Development Board	
Task	**Without pipelining**	**With pipelining**
Fixdt(1, WL, FL)	**Fixdt(1, 36,22)**	**Fixdt(1, 36,22)**
Number of Slice Register	810 (0.27%)	2, 415 (0.80%)
Number of Slice LUTs	8, 063 (5.35%)	8, 836 (5.86%)
Operating Frequency (MHz)	9.819	23. 613
Power (mW)	29	80

D. FPGA Stand-alone Implementation

Figure 4 illustrates the block diagram of FPGA stand-alone implementation which functions to model the LR-I model in real-time system using the Virtex-6 XC6VLX240T FPGA in the ML605 Development board. The switch represents a single bit of input. Besides, 15 bits of a pulse stimulus current will be applied in this system to start the simulation controlled by an enable switch and the action potential waveform that represent in 16 bits as the output will be displayed by using a data logger. Here, dual channels 16 bits Digital-to-Analog Converter (DAC) FPGA Mezzanine Card (FMC) supported by ML605 FPGA development board will be used to convert the output digital signal into an analog signal with maximum voltage of 2.5 V from the FPGA board which is connected using VITA 57.1 FMC connector. The FMC I/O standard gives a new

opportunity for FPGAs to be used as a communication platform as it provides high-speed D/A converter I/O directly to the data logger [19].

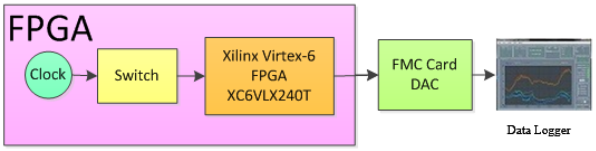

Fig.4: Layout FPGA stand-alone implementation

IV. Conclusions

In conclusion, Luo-Rudy mathematical model has successfully been designed using MATLAB Simulink and then converted to HDL codes for FPGA implementation. Compared with conventional method of manually write HDL codes, this approach saves much time to develop and optimize fixed-point implementation for complex designs.. Here, the fixed-point optimization of the HDL design model has been done to obtain an optimum value of the FL and WL which eventually improves FPGA implementation speed suitable for real time simulation. The speed optimization of pipelining methods also has been done to increase the operating frequency but has caused increments in the power and the area consumption. Therefore, a tradeoff between the operating frequency and the power is crucial in order to achieve the optimum speed while maintaining reasonable power consumption. Moreover, the optimization of the system can also be done manually through the ISE software to get a better performance of the design model. According to this optimized LR-I HDL design model, a stand-alone FPGA hardware implementation will be conducted towards real-time simulations of cardiac excitation for analysis of electrophysiological mechanism.

Acknowledgment

The authors gratefully acknowledge the support by the Fundamental Research Grant Scheme (FRGS) (vote no. 1053) under Ministry of Higher Education Malaysia.

References

[1] J. Smith, M. J. Jones, l. Houghton et al., “Future of health insurance,” N Engl J Med, vol. 965, pp. 325–329, 1999.

[2] F. Mahmud, S. Naruhiro, M. Masaaki, and N. Taishin, “Reentrant excitation in an analog-digital hybrid circuit model of cardiac tissue,” American Institute of Physics, Chaos, vol. 21, pp. 1-14, 2011.

[3] E. Monmasson, Marcian N. Cirstea, “FPGA design methodology for industrial control systems – A review”, IEEE Trans. Industrial Electronics, vol 54, no.4, pp. 1824-1842, August 2007.

[4] P. Y. Siwakoti, E. T. Graham, “Design of FPGA-controlled power electronics and drives using MATLAB Simulink,” Macquarie University, Australia, pp. 571- 577, 2013.

[5] Stephan Van Beek and Sudhir Sharma, Mathworks, “Four Best Practices for protyping MATLAB and Simulink Algorithms on FPGAs”, Verification Horizons, pp. 49-53, August 2011.

[6] K. Kintali, “Converting MATLAB algorithms into serialized designs for HDL code generation,” MATLAB Digest, pp. 1-4, 2011.

[7] Q. Wang, et al, “Detecting transient bottlenecks in n-tier application through fine-grained analysis”, IEEE 33rd International Conference on Distributed Computing Systems, 2013.

[8] C.H. Luo and Y. Rudy, “A model of the ventricular cardiac action potential. Depolarization, repolarization, and their interaction,” Circulation Research vol. 68, No 6: pp. 1501-1526, 1999.

[9] N. Othman, M.H. Jabbar, A.K. Mahamad and F. Mahmud, Luo Rudy Phase I excitation modeling towards HDL Coder Implementation for Real-time Simulation. The 5th International Conference on Intelligent & Advanced Systems, A Conference of World Engineering, Science and Technology Congress. pp. 144-149, 2014.

[10] N. Othman, F. Mahmud, A.K. Mahamad, and M.H. Jabbar, FPGA-in-the-Loop simulation of cardiac excitation modeling towards real-time simulation, IFMBE Proceedings, pp. 266-269. 2014.

[11] J. Liu, "Model based design workflow for FPGA compliance with DO-254 standard," Information Technology in Medicine and Education (ITME), 2012 International Symposium on, vol.2, pp.1026-1030, 3-5 Aug. 2012.

[12] F. Mahmud, T. Sakuhana, N. Shiozawa, and T. Nomura, An analog-digital hybrid model of electrical excitation in a cardiac ventricular cell. Trans JPN Soc Med Biol Eng., vol. 47, pp. 428-435, 2009.

[13] F. Mahmud, Real-time simulations for resetting and annihilation of reentrant activity using hardware-implemented cardiac excitation modeling. IEEE EMBS International Conference on Biomedical Engineering and Sciences. 978 (2012) 321-325.

[14] Z. Zhang, V. Anantharam, Wainwright, M.J.; Nikolic, B., "An Efficient 10GBASE-T Ethernet LDPC Decoder Design With Low Error Floors," Solid-State Circuits, IEEE Journal of Engineering , vol.45, no.4, pp.843,855, April 2010.

[15] F. Xu, H. Chen, W. Jin, Y. Xu, "FPGA implementation of nonlinear model predictive control," Control and Decision Conference (2014 CCDC), The 26th Chinese, pp.108-113, May 31 - June 2, 2014.

[16] H. Sven, f. Winkler, B. Scheuermann, K. Reinhardt, "Building Optimized Packet Filters with COFFi," Field-Programmable Custom Computing Machines (FCCM), IEEE 22nd Annual International Symposium pp.105, 11-13 May 2014.

[17] M. Schwall, F. K. Jondral, "High-Speed Turbo Equalization for GPP-Based Software Defined Radios," IEEE Military Communications Conference, pp.1592-1596, 18-20 Nov. 2013.

[18] C. amarreno-Ramos, A. Linares-Barranco, T. Serrano-Gotarredona, Linares-Barranco, B., "Multicasting Mesh AER: A Scalable Assembly Approach for Reconfigurable Neuromorphic Structured AER Systems. Application to ConvNets," Biomedical Circuits and Systems, IEEE Transactions on biomedical engineering, vol.7, no.1, pp.82,102, Feb. 2013.

[19] M. Bouain, V. Viswanathan, R. Ben Atitallah, J. L. Dekeyser, "Communication-centric design for FMC based I/O system," Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC), 2014 9th International Symposium, pp.1-8, 26-28 May 2014.