A New FPGA Based Green Controller Using Modeling Language

Kenichi Morimoto*, Yuichiro Shibata*, Yudai Shirakura*, Hidenori Maruta*, Fujio Kurokawa*, Masanori Nobe‡, Masaharu Tanaka‡‡

*Nagasaki University, 1-14, Bunkyo-machi, Nagasaki, 852-8521 Japan

‡ Mitsubishi Hitachi Power Systems, LTD., 3-3-1 Minatomirai, Naka-ku, Yokohama, 220-8401 Japan

‡ ‡ Mitsubishi Heavy Industries, LTD., 2-16-5 Konan, Minato-ku, Tokyo 108-8225 Japan

(bb52215205@cc.nagasaki-u.ac.jp, shibata@cis.nagasaki-u.ac.jp, {hmarta,fkurokaw}@nagasaki-u.ac.jp)

‡Corresponding Author; Kenichi Morimoto 1-14, Bunkyo-machi, Nagasaki, 852-8521 Japan

bb52215205@cc.nagasaki-u.ac.jp

Received: June.15.2016 Accepted: June.17.2016

Abstract- In recent years, the renewable energy system requires the modeling language based control and the high-performance CPU. As for CPU, the improvement of the CPU performance reaches its limitation caused from its clock frequency. Therefore, the multi-core and the virtualization are mainstream strategies for its improvement and such mechanisms have some issues including complicated algorithm of controller and inheritance of design on CPU architecture generation. For such reasons, Field Programmable Gate Array (FPGA) is expected to replace the role of CPU since it has several advantages such as long period supply, inheritance and power-saving. It is applied to the small machinery control so far, and there are few researches to apply it to the power plant control. For such purpose, a flexible control with the modeling language like CPU and on-line modification of the control logics are necessary. In this paper, to fulfill such requirements on the control system for the infrastructure like the power system, a new FPGA based controller named "green controller" is proposed. An architecture to realize it and required conditions based on FPGA are clarified. Furthermore, performance and FPGA resource usage are measured in the experiment with the proposed architecture for evaluation.

Keywords FPGA, modeling language, HLS, function block

1. Introduction

The large scale renewable energy grid system including several types of renewable energy resources needs a modeling based optimal control which is combined with power generator and battery models [1]. In recent controller, high-performance CPU is applied to realize complicated control algorithms. From the past, it has been applied in thermal power energy plants, e.g., gas-turbine, coal-firing boiler and so forth, since such conventional power plants require complicated control algorithms recently [2]. Due to the scale expansion of the renewable energy network, such high-performance CPU has also been used in the controllers of such systems.

A simple control algorithm has been adopted to the conventional renewable energy equipment [3]. The control has been realized by PLC (Programmable Logic Controller) which can only calculate the ladder logics with blocks and it has consisted of customized ASICs from the past [4],[5]. Therefore, there is no issue about a long term supply and maintenance for such systems.

However, advanced control algorithms for the renewable energy system become complicated ones to satisfy several requirements. For example, an off-shore wind turbine requires to control its pitch and its transmission physical modelling with modelling languages (e.g. MATLAB/Simulink) since the size of it increases [6]. For such reasons, high-performance CPU is required and indispensable for the energy plants.

On the other hand, there are several problems concerned with the realization of the control system by CPU. The performance of recent CPU depends on a number of multi-

cores rather than its clock frequency [7],[8] and the control system needs a complex middle-ware such as a multi-core real-time OS [8] or a virtualization OS [9],[10]. The difference of multi-core CPU architectures is large against the generation change and it is difficult to keep the system design since it does not consider about inheritance.

To address such problems, FPGA focuses attentions in the industrial field since it has advantages such as simple structure and long term inheritance. FPGA has been applied to a small scale machinery control [11]-[13] and a software product was realized to translate the modeling language to the FPGA HDL [14]. However, FPGA is to be applied to the infrastructure such as the large scale power system and the large scale renewable energy system so far since there are some problems.

It is necessary to tune control parameters and to modify control circuitry which affect the performance of power generation even when the power plant is running [15]. Moreover, similar approaches in CPU based development such as softcore and general processing unit cannot make use of advantages of FPGA since the most basic feature of FPGA is its parallel processing capacity [11]. Among them, the most important problem to adopt FPGA is to utilize software and design resources developed so far into advanced renewable energy systems. This problem is not discussed and published yet as far as our knowledge.

Therefore, a novel controller architecture, which can suit to features of FPGA, is required to adopt the FPGA based control with modelling languages to renewable energy system. We name the proposed controller as "green controller". By using FPGA processing, the green controller can realize complicated control methods with various inputs and outputs and on-line parameter modification. Also, it can address problems of the inheritance about architecture and the long term supply.

This paper is organized as follows. In Sec. 2, hardware implementation of control systems are discussed from a point of view of shifting the control architecture to the FPGA based one for energy systems. The proposed architecture is evaluated by the processing performance and the occupied FPGA resource of the test model in Sec. 3. Conclusions are given in Sec. 4.

.

2. Requirement of Control System For Energy System

2.1. Conventional hardware implementation of control systems

The control application is required to have flexibility in its designing. It is also required to realize a sufficient calculation performance against the large scale control logic of the power plant with limited computation resources. Actual control algorithms of the power plant control system are complicated since they consist of multi-input and multi-output systems which are mutually related with each other.

The scale of the power plant system can be estimated by the total number of function blocks. It means that the scale is estimated by using the number of the PI control loop based on the number of the output signals. The single-input single-output PI control loop can be used as a unit in order to evaluate the performance and the required computation resource and it is named "*1-loop*" logic. By using *1-loop* logic, the complicated control system, for example, gas-turbine, coal firing boiler, and so forth, which is not *1-loop* logic control, it is possible to convert and estimate the logic volume by using *1-loop* logic. The logic volume is about 1,000 loops or 2,000 loops in real cases. In such systems, a process cycle of the controller is 50 ms generally.

From a point of view of the next generation of power plants, the control system is required to fulfill a 10 times higher performance than the current processing performance caused from increasing calculation volume in future. Therefore, the required performance is about 2.0×10^4 *1-loop* logics in 50 ms.

To tune the plant control parameter to environments and energy sources, it is required to maximize the performance using a long term test-run process [16]. For example, coal firing power plants need several months test-run for the optimization of the control parameters to check a variation of coal [17]. The gain or time constant parameters of temperature control of them are tuned because the dynamic characteristic of the combustion system changes by calorie and water content of the coal which is actually used in the field.

Recent renewable energy systems, e.g., off-shore wind turbine need modern optimum control algorithms with physical machinery models [6]. Those control algorithms are described by the modeling language like MATLAB/Simulink, and the logics of the control system are constructed after the model simulation. For such situations, a traditional methodology based on ASIC is not appropriate and the high-performance CPU is appropriate for recent controller architecture [18], [19]. CPU based controllers also have an advantage from the view points of the parameter tuning, online monitoring and modification. It is easy for the high performance CPU to realize the modification of control logics parameters in the process cycle time.

However, it arises other problems to adopt it to recent control systems. The performance of CPU has been improved by the clock frequency. The upper limit of the clock frequency of recent CPUs per core is around 2GHz and the performance improvement is realized by increasing the number of multi-cores [7], [8]. The multicore architectures are complex and are different depending on CPU products. Addition to such software problems, a hardware problem about heat increasing also arises. The computation performance of CPU consequently brings additional cost for cooling equipment [20]-[23]. From above reasons, the strategy to shift of the CPU performance improvement should have serious risk and cost for the infrastructure of the power system.

2.2. FPGA realization of control systems

To address the problems of CPU based controllers, FPGA is focused and expected in the industrial field [24]-[26] since the computation performance rapidly increased and the power consumption and the cost per unit become lower. In the FPGA development, HLS (High-Level Synthesis) [27] can be widely applicable to convert traditional program codes for CPU to codes for FPGA like HDL. From the view point of inheritance, FPGA has a great advantage that HDL codes can keep their compatibility against the FPGA generation change. This feature is very expected one for the industrial use which requires the long term operation and maintenance. Furthermore, the internal structure of FPGA is quite simple. When a trouble happens, it is easy to investigate and to prevent since it is possible to reconfirm the behavior by the simulation.

There are two approaches to apply FPGA to the controller. One is to implement the controller on FPGA using HLS directly. This approach can be easily implemented widely provided as a product solution. Another is an architecture which can realize the control system in a small resource consumption. To provide a solution to develop the FPGA based architecture for the power plant applications, the novel green controller is proposed in this paper.

3. Proposed Concept and Architecture

3.1. Test Model and Control Logic

The performance and the resource consumption of the proposed FPGA based architecture are evaluated by the model simulation. A tank model, which is the most basic model for the plant modeling, is selected as a test model. Its model and control logic are shown in Fig. 1. As shown in this figure, the tank model has a water supply line via a flow control valve and a drain line. The drain water flow is a constant value by a flow limit valve for simplification. This tank model is realized on another FPGA resource. The tank water level is controlled by the PI controller.

In the power plants, special function block diagram (FBD) languages have been adopted as the modeling language to describe the control algorithm from the past. Therefore, MHI-IDOL, one of the FBD languages developed by Mitsubishi Heavy Industries [28], is adopted as a case study since it is widely used in existing power plants. It is noted that the proposed concept can be applicable to any other modeling languages which are realized with C code based function blocks. Figure 2 shows the PI controller described in the FBD language used as a test study. The closed transfer function H(s) of this model is described as Eq. (1);

$$H(s) = \frac{G_C(s)G_P(s)}{1 - G_C(s)G_P(s)} = \frac{\alpha k_p s + k_i}{s^2 + \alpha k_p s + k_i} \qquad (1)$$

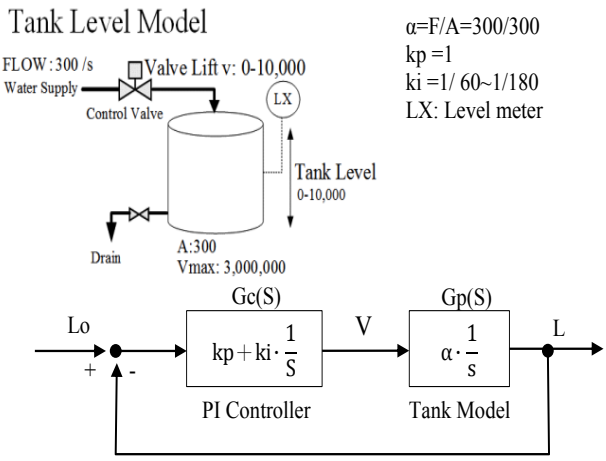

Fig. 1. Tank model and the transfer function with the feedback.

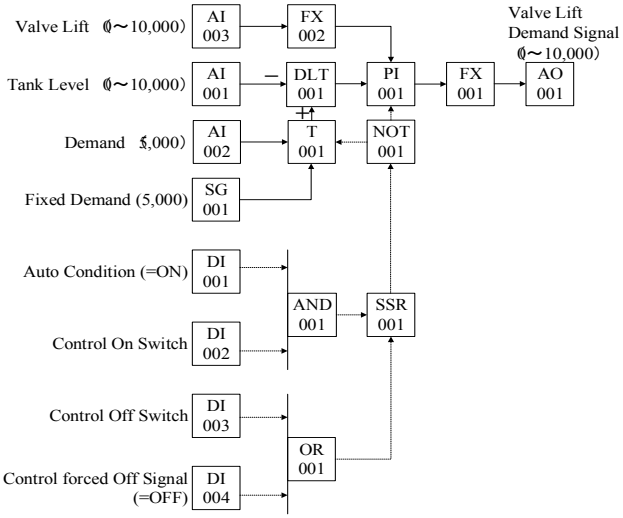

Fig. 2. PI Controller described by FBD.

3.2. Direct Synthesis with HLS

To realize the green controller, the modelling language for arithmetic operation is used to describe control applications. In the modeling language, arithmetic operations are realized as function blocks and the control algorithm is described the connected relation of those blocks. Connection lines represent input-output relations of signals. Such function blocks are executed as C coded subroutines in the traditional CPU based control system.

For the shift from the CPU based control architecture to the FPGA based one, it is needed to confirm that function blocks described with C language to FPGA circuit data with HLS. Table 1 shows converted results from the function blocks with C code subroutines in MHI-IDOL to FPGA circuit data. From this result, it is confirmed that 252 types of subroutines are converted into the FPGA data without any difficulties with HLS.

Table 1. HLS Results of function blocks.

No.	ID Name	Description	LATENCY			DELAY	ASSIGNED RESOURCES			
			MIN	AVE	MAX	(ns)	FF	LUT	BRAM	DSP
1	AND	And	0	0	0	1.68	23	30	0	0
2	OR	Or	0	0	0	1.68	23	30	0	0
3	NOT	Not	0	0	0	0.85	2	5	0	0
4	XOR	Xor	0	0	0	0.86	4	7	0	0
5	SSR	Set/Reset	0	0	0	1.82	7	14	0	0
6	SRR	Set/Reset	0	0	0	1.82	7	13	0	0
7	OND	On delay	4	10	27	8.30	1049	1647	0	11
8	OFD	Off delay	4	10	27	8.61	1049	1634	0	11
9	OSP	One shot	5	11	27	8.41	1054	1590	0	11
		No 10~14 are omitted								
15	HIM	High Monitor	2	2	2	2.57	133	86	0	0
16	LOM	Low Monitor	2	2	2	2.64	133	86	0	0
17	HLM	H/L Monitor	3	4	4	2.57	235	189	0	0
18	HMH	Hysteresis	3	6	10	6.55	407	422	0	2
19	LMH	Hysteresis	3	6	10	6.84	405	417	0	2
		20~106 are omitted								
107	DI	D input	2	2	2	1.88	35	24	0	0
108	AI	A input	0	0	0	NA	0	1	0	0
109	PU	Pulse input	0	0	0	0.54	0	0	0	0
110	DO	D output	2	2	2	3.13	109	143	0	0
111	AO	A output	1	20	39	8.94	3829	8663	0	14

Figure 3 shows the control architecture realization on FPGA with HLS directly. As shown this figure, the circuit layout and the order of the calculation sequence needed to proceed the control algorithm in direct implementation with HLS.

This architecture is evaluated by the computation performance and resource consumption. It is seen that the computation performance is 0.88 μs per *1-loop* and the resource consumption is 783 slices per *1-loop*. From results, especially, it has a drawback in the resource consumption to realize the control system in real use since the most advanced FPGA (Virtex-7 XC7V200T) has 305400 slices which can only realize 390 *1-loop*. This resource consumption cannot satisfy the required condition for the green controller.

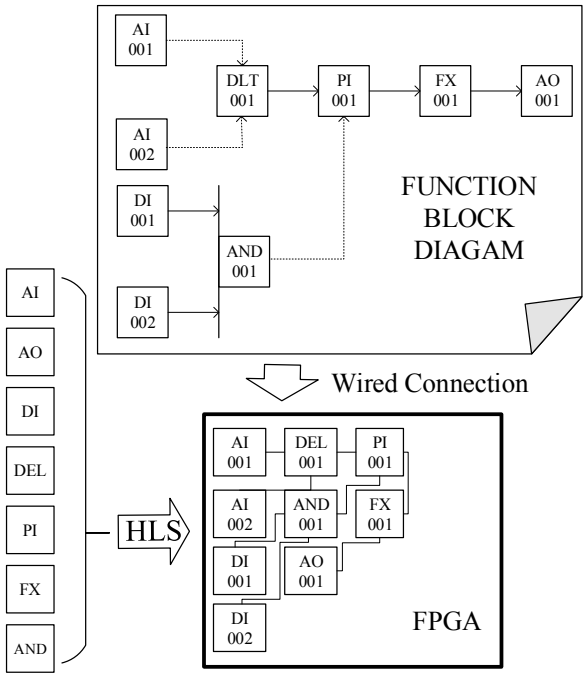

Fig. 3. Example of FPGA realization with HLS directly.

3.3. *Two Dimensional Bus(TDB) Architecture*

TDB is another architecture to realize the control system as following. TDB has three characteristic structures; (1) the function blocks compiled by HLS are located in FPGA as processing element (PE), (2) PE memorizes the calculation sequence in its memory part and starts its own task by the synchronized signal (3) each PE has the address array which is generated from the connection information of the FBD logics. Each PE has its own identifier and it calculates own result along the specific sequence. The calculated output results are transferred to the other PE as the input signal. Each PE has its own address-array R which stores the identifier which the calculation results are transferred. The connections between function blocks of FBD are realized as the logical communication lines between PEs by each R. The calculation sequence and the circuitry connection of this architecture are independent from the internal wiring of FPGA. Those are configurable parameters without the re-configuration of FPGA. TDB is to construct the control algorithm by arranging FBD function blocks on FPGA as PEs and to calculate by ordered PE sequence. Figure 4 shows a structure of TDB. The PEs are arranged in two dimensional matrix and each PE connects via vertical buses and horizontal buses.

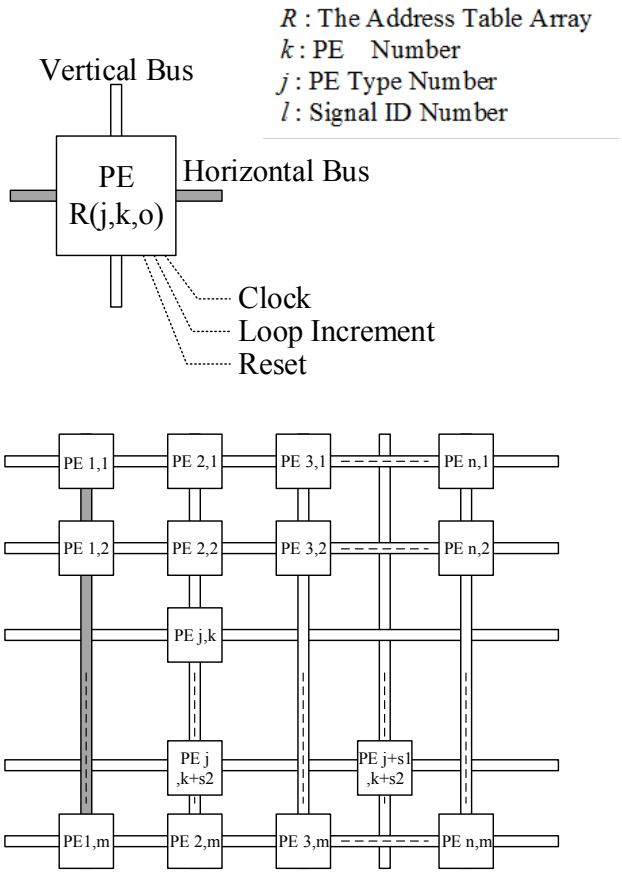

Fig. 4. Bus communication architecture.

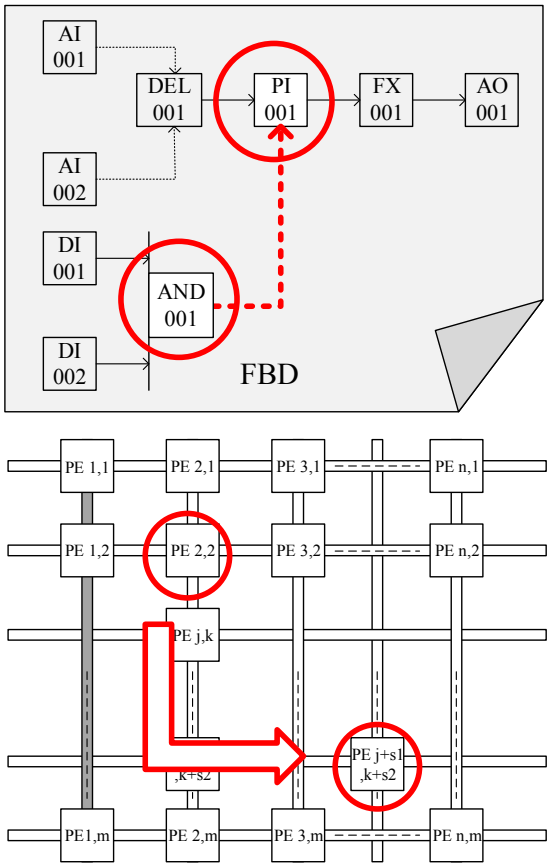

Fig. 5. Transition example of signal in the proposed architecture.

In Fig. 4, suffix k denotes the PE identifier number. The same type of PE is arranged in a vertical order and suffix j denotes the type of PE. Every PE is identified with j, k. Also, suffix l denotes the input signal identifier number of each PE. The input signals of the function block described by FBD is identified these suffixes. It is noted that the maximum numbers of j and k are decided by FPGA hardware limitation.

Each PE starts its calculation by the synchronized signal. The calculation result is transferred to the vertical bus just after the calculation finished. PE(j,k) receives the data only from PE(j,*) which has the same identifier j. The PE just transfers the data through the horizontal bus when the destination address is not same as its own address.

When PE(j,k) which receive data whose destination address has type and identifiers "j,k", PE(j,k) uses the data as its input signal and addresses with l. Figure 5 shows a transition example of the signal in the proposed architecture implemented in FPGA. Figure 6 shows an implemented example by TDB architecture. Since the connection information of function blocks is realized by the array R as the communication address table, the modification of R is only needed when the control circuit is changed. R of TDB can be extended to more than *1-loop*. It is also worth to say that the resource of the FPGA is saved since PEs are shared among in multiple *1-loop*s. Each PE has the function to proc ess the data packet routing.

It is necessary to evaluate how the overhead of the communication function affects to calculation performance and whether it affects to realization of control system for the actual power system.

Figure 7 shows the result of controller in this test study. This control regulates the tank level to be 5000 in Fig. 1. To converge the tank level, it takes about 3000 cycles. Since an overshoot is observed in the simulated tank level, it is confirmed that TDB performs the PI control well.

TDB is also evaluated by the computation performance and resource consumption. It is seen that the computation performance is 2.36 μs per *1-loop* and the resource consumption is 4,402 slices per *1-loop*. The exchange time from one control logic to other control logic by a replacement of R is 10 ns. Therefore, the mean processing time for *1-loop* is 2.37μs. From this result, it is estimated that TDB can process 2.1×10^4 loops per 50 ms. Additionally, the FPGA resource for 2.1×10^4 loops is also 4,402 slices which is equal to the resource for *1-loop* calculation since the same arithmetic resources utilized for different loops are recycled in a time sharing manner by switching the connection topologies stored in R.

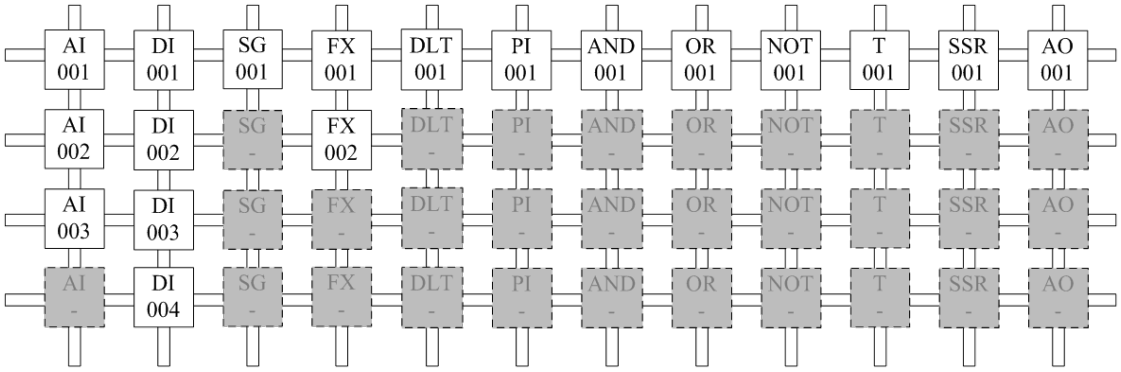

Fig. 6. Implemented example with TDB.

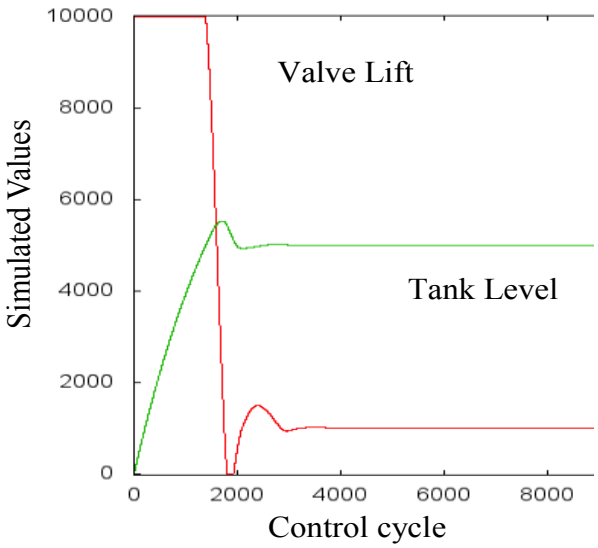

Fig. 7. Step signal response of this test study.

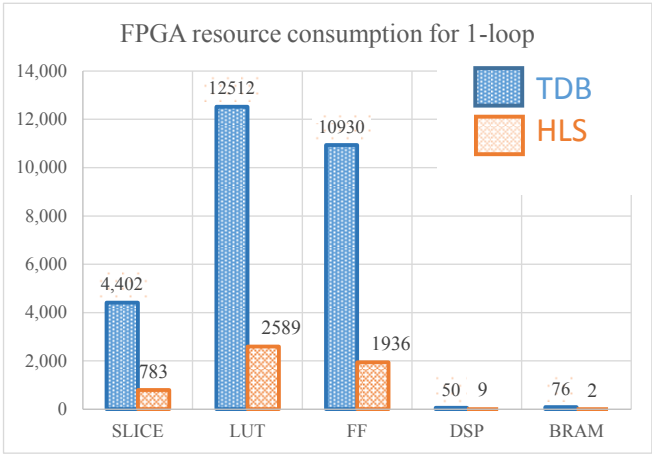

Fig. 8. FPGA resource consumption for *1-loop*.

Figures 8 and 9 show FPGA resource consumption using the direct HLS implementation and TDB. From Fig. 8, it is seen that HLS has a superior resource consumption when only one *1-loop* is implemented. On the other hand, it is seen from Fig. 9 that TDB has a superior one when 4096 *1-loop*s are implemented since it realizes the resource sharing in FPGA.

Figure 10 also shows an execution time comparison between the direct HLS implementation and TDB. From this result, it is revealed that TDB needs about a three times longer time compared to the HLS since TDB has the communication overhead between PEs. However, both approaches satisfy the required condition about the execution of *1-loop*.

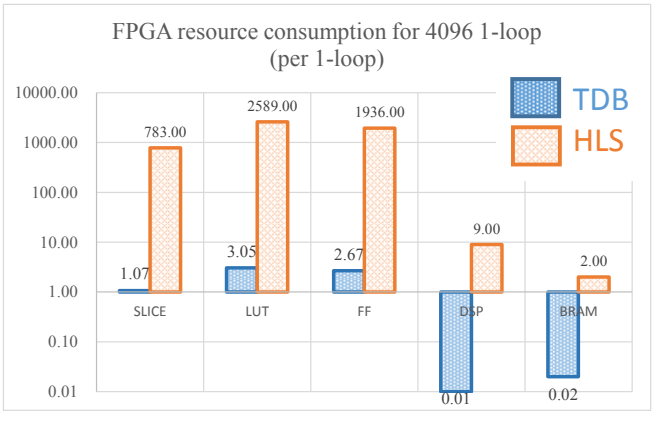

Fig. 9. FPGA resource consumption for 4096 *1-loop*s (per *1-loop*). The vertical axis is logarithmic representation.

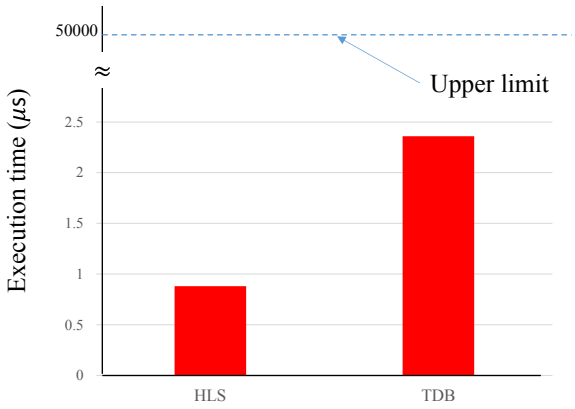

Fig. 10. Execution time comparison between HLS and TDB.

4. Conclusion

In this paper, the new FPGA based control architecture, name "green controller", is proposed for the next generation power plant for renewable energy power network. The performance and the resource consumption of the proposed architecture are evaluated with the simple tank level mode simulation. It is clarified quantitatively that the architecture has enough calculation performance to control the large scale energy system. The calculation performance is improved by the parallelization in FPGA. Furthermore, it is demonstrated that the architecture realizes the on-line modification of the control logic without re-synthesis.

As a result, it is proved that the proposed architecture fulfills all of the requirement for the green controller. It is expected that this result contributes to evolution and expansion of the future renewable energy which requires the complex model based control.

References

[1] V. Prema, S. Dutta, K. U. Rao, S. Shekhar, B. S. Kariyappa, "Effective battery usage strategies for hybrid power management," in Proc. International Conference on Power and Advanced Control Engineering (ICPACE), pp. 95-98, Aug. 2015.

[2] C. Wang, W. Li, J. Belanger, "Real-time and faster-than-real-time simulation of Modular Multilevel Converters using standard multi-core CPU and FPGA chips," in Proc. Industrial Electronics Society (IECON), pp. 5405-5411, Nov.2013.

[3] N. Luo, Y. Vidal, L. Acho, Wind turbine control and monitoring, Springer International, 2014.

[4] IEC, "61131-1 Programmable controllers - Part 3: Programming languages Ed.3," Feb. 2013.

[5] K. H. John, M. Tiegelkamp, "Programming industrial automation systems," Springer International, 2001.

[6] M. K. Paul, B. Hamane, M. L. Doumbia, A. Cheriti, "Pitch control of a wind energy conversion system based on permanent magnet synchronous generator (PMSG)," in Proc. International Conference on Ecological Vehicles and Renewable Energies (EVER), pp. 1-7, Mar. 2015.

[7] D. Patterson, J. L. Hennessy, Computer Organization and Design, 5th Edition, Elsevier, Morgan Kaufmann, 2013.

[8] J. Cownie, "Multicore: the software view," in Proc. 12th EMEA ACADEMIC FORUM, Intel Corporation, 2007.

[9] H. Kim, A. K. Lu, R. Rajkumar, "A coordinated approach for practical OS-level cache management in multi-core real-time systems," in Proc. Euromicro Conference on Real-Time Systems, pp. 80-89, Jul. 2013.

[10] K. Gilles, S. Groesbrink, D. Baldin, T. Kerstan, "Proteus hypervisor: full virtualization and paravirtualization for multi-core embedded systems, embedded systems: design, analysis and verification," in Proc. IFIP International Embedded Systems Symposium (IESS), pp. 293-305, Jun. 2013.

[11] Y. Y. Tzou, T. S. Kuo, "Design and implementation of all FPGA-based motor control IC for permanent magnet AC servo motors," in Proc. International Conference on Industrial Electronics, Control and Instrumentation (IECON), vol.2, pp. 943-947, Nov. 1997.

[12] J. T. Welch, J. Carletta, "A direct mapping FPGA architecture for industrial process control application," in Proc. International Conference on Computer Design (ICCD), pp. 595-598, Sep. 2000.

[13] D. A. Lee, E. S. Kim, J. Yoo, J. S. Lee, J. G. Choi, "FBD to Verilog 2.0: an automatic translation of FBD into Verilog to develop FPGA," in Proc. International Conference on Information Science and Application (IcISA), pp. 1-4, May 2014.

[14] M. Kocur, S. Kozak, B. Dvorscak, "Design and implementation of FPGA - digital based PID controller," in Proc. International Carpathian Control Conference (ICCCC), pp. 233-236, May 2014.

[15] J. Khalifat, A. Ebrahim, A. Adetomi, T. Arslan, "A dynamic partial reconfiguration design for camera systems," in Proc. Adaptive Hardware and Systems (AHS), pp. 1-7, Jun. 2015.

[16] Y. Si, H. R. Karimi, H. Gao, "Parameter tuning for nacelle-based passive structural control of a spar-type floating wind turbine," in Proc. Industrial Electronics Society (IECON), pp. 7687-7691, Nov. 2013.

[17] S. Li, H. Liu, W. J. Cai, Y. C. Soh, Li. H. Xie, "A new coordinated control strategy for boiler-turbine system of coal-fired power plant," IEEE Transactions on Control Systems Technology, vol. 13, no. 6, pp. 943-954, Nov. 2005.

[18] I. Yousefi, M. Yari, M. A. Shoorehdeli, "Modeling, identification and control of a heavy duty industrial gas turbine," in Proc. International Conference on Mechatronics and Automation, pp. 611-615, Aug. 2013.

[19] G. P. Lim, H. H. Lee, "Development and application of boiler combustion air flow control algorithm for coal-fired power plant," in Proc. International Conference on Industrial Electronics and Applications (ICIEA), pp. 662-667, Jun. 2013.

[20] H. Chen, P. C. Xiong, K. Schwan, A. Gavrilovska, C. Z. Xu, "A cyber-physical integrated system for application performance and energy management in data centers," in Proc. International Conference on Green Computing Conference (IGCC), pp. 1-10, Jun. 2012.

[21] Y. Chen, D. Gmach, C. Hyser, W. Zhikui, C. Bash, C. Hoover, S. Singhal, "Integrated management of application performance, power and cooling in data centers," in Proc. Network Operations and Management Symposium (NOMS), pp. 615-622, Apr. 2010.

[22] P. Paschke, M. Plonczak, P. Klis, M. Grunt, "Perspectives of development of integrated monitoring system of power supply and air conditioning equipment towards technical environment equipment monitoring system of the operator," in Proc. International Telecommunications Energy Conference (INTELEC), pp. 1-6, Sept. 2008.

[23] H. F. Hamann, V. Lopez, A. Stepanchuk, "Thermal zones for more efficient data center energy management," in Proc. Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm), pp. 1-6, Jun. 2010.

[24] E. Monmasson, M. N. Cirstea, "FPGA design methodology for industrial control systems-a review," IEEE Transactions on Industrial Electronics, vol. 54, no. 4, pp. 1824-1842, Aug. 2007.

[25] G. Singh, S. S. Gill, "Design and Implementation of process controller for direct digital control on FPGA," in Proc. International Conference on Machine Intelligence and Research Advancement (IC-MIRA), pp. 326-330, Dec. 2013.

[26] S. W. A. Hashmi, M. Rehan, M. Aamir, H. Kumar, F. Liaquat, "Distributed process monitoring and control using FPGA," in Proc. International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronics System (VITAE), pp. 1-5, May 2014.

[27] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, Z. Zhang, "High-level synthesis for FPGAs: from prototyping to deployment," IEEE Transactions on Computer-Aided Design of Integrated Circuits and System, vol. 30, no. 4, pp. 473-491, Apr. 2011.

[28] Mitsubishi Hitachi Power Systems, LTD.,"DIASYS Netmation4S: control system," https://www.mhps.com/en/products/category/diasys_pro_netmation.html