International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 1, February 2021, pp. 249~256
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i1.pp249-256

FPGA based synchronous multi-channel PWM generator for humanoid robot

Adharul Muttaqin[1], Stefanus Dion Finnadi[2], Zainul Abidin[3], Kakeru Araki[4]
[1,2,3]Department of Electrical Engineering, Universitas Brawijaya, Indonesia
[4]Department of Electrical and System Engineering, University of Miyazaki, Japan

Article Info

Article history:

Received Nov 21, 2019
Revised Jun 22, 2020
Accepted Jul 9, 2020

Keywords:

Field programmable gate array
Humanoid robot
Pulse width modulation
Serial communication
Servo motor

ABSTRACT

In this paper, synchronous multi-channel pulse width modulation (PWM) generator for driving servo motors of humanoid robot was proposed. In an application, the humanoid robot requires smooth and beautiful movement, therefore the PWM signal for each servo motor must be synchronized. Since microcontroller (slave) has no enough channels to generate synchronous PWMs for 32 servo motors, field programmable gate array (FPGA) was used as slave for the humanoid robot. The FPGA was controlled by microcontroller (master) using serial communication. Simulation results show the system can perform serial communication, synchronize, and convert data well. The system can also generate PWM simultaneously with accurate duty cycle and fix period of 20ms.

Corresponding Author:

Adharul Muttaqin,
Department of Electrical Engineering,
Universitas Brawijaya,
Jl. MT Haryono 167, Malang, 65145, Indonesia.
Email: adharul@ub.ac.id

1. INTRODUCTION

Humanoid robot and its controller have been developed rapidly for several purposes [1-5]. In some particular event, the humanoid robot is designed for dance robot [6-9]. In order to design the dance robot, a number of DC servo motors must be used [8]. In this case, synchronous and simultaneous PWM to control the DC servo motors is required for the dance robot to move smoothly and beautifully. Microcontroller is usually used to control the DC servo motors [10]. In the case of microcontroller STM32F407VG, the PWM signal is generated with timer of general purpose input output (GPIO) which has only 4 channels [11]. Since the designed dance robot needs more than 30 DC servo motors, the STM32F407VG can not handle it. field programmable gate array (FPGA) is a semiconductor device which after manufacture can be programmed to perform a specific design of application. The FPGA is a system which typically specified in digital logic [12, 13]. The FPGA has successfully applied to many applications especially in the fields of industrial control, microelectronics, internet of things (IoT), and robotics [14-16]. More specifically, the FPGA can support automated test equipment [17], Voltage Source Inverter (VSI) [18], design of stepper motor movement [19], and speed control of induction motors [20]. The FPGA has capability to handle parallel processing and can make higher data processing in real time [21-26]. The capability of parallel processing is expected to make the FPGA as multi-channel PWM generator [27].

In this paper, the FPGA is designed as slave to generate synchronous and simultaneous multi-channel PWM to control DC servo motors. The FPGA is controlled by microcontroller (master) using serial communication. Furthermore, this paper is organized as follows. Section 2 describes method used in this research. Performance is presented by simulation result and its analysis which is stated in section 3. Finally, section 4 concludes this research.

2. RESEARCH METHOD

This research is based on an applicative problem. FPGA was chosen because of its high flexibility and can be adjusted to the designer needs. In addition, the system inside the FPGA can be changed using a program so that the modules designed can be developed for further research. Block diagram of DC servo motors controller can be seen in Figure 1. As a master, the microcontroller can control the FPGA (slave) via serial communication. Block diagram of the overall FPGA system is shown in Figure 2. It has 4 main blocks, i.e. serial receiver, data holder, data synchronization, and PWM generator. Design of the FPGA system has specifications as shown in Table 1. The detail main block (module) designs are described as follow.

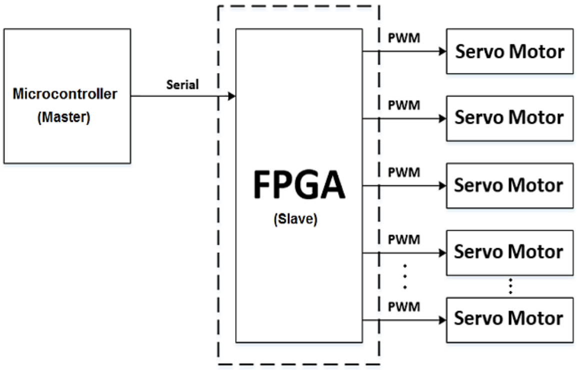

Figure 1. Block diagram of DC servo motors controller

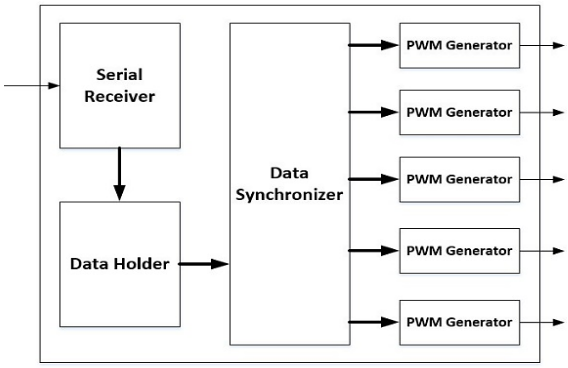

Figure 2. Block diagram of FPGA system

Table 1. Specifications of the FPGA system

Specifications	
PWM period	20 ms
Duty cycle range	1 - 2 ms
Data_angle	8 bits
Data_ID servo motor	8 bits
Minimum baud rate	38400
Stop bit	1
Parity bit	-
Number of Output	32

2.1. Serial receiver module

Serial receiver module is designed to receive serial data from microcontroller (master). The Serial receiver is a modification of the universal asynchronous receiver transmitter (UART) by eliminating its transmitter function. The transmitter function is not needed because the FPGA only accepts data from the microcontroller. Baud rate used in the serial receiver is 57600, stop bit 1, and without parity bit. The serial receiver module is designed in 3 blocks, i.e. FIFO buffer, baud generator, and receiver as shown in Figure 3. The baud generator is used to generate pulse signal in accordance with determined baud rate. The receiver is used to receive serial data from pin rs_232_rx and convert to parallel data.

2.2. Data holder module

Data holder module is designed to unite two 8 bits data into 16 bits data. The first 8 bits data is the servo angle data and the second 8 bits data is the servo ID data. The 16 bits data is then sent to data synchronizer. The data holder module can be seen in Figure 4.

2.3. Data synchronizer module

Data synchronizer is designed to process the 16 bits data received from the data holder. The 16 bit data will be separated into servo angle data and servo ID. The angle data will be converted to 20 bit data then sent to PWM generator. The data synchronizer also has role as a demultiplexer to send angle data to the PWM generator with the appropriate ID. The update signal is sent to all PWM generators for the simultaneous PWM signals. Module of data synchronizer is designed as shown in Figure 5. It consists of data read, comparator, multiplier, adder, latency, and demultiplexer. The data read is used to receive and distribute the data. The comparator is used to send update signal to all PWM generator if the received data is the same as the saved one in comparator. Multiplier and adder are used to convert angle data to 20 bit data. Latency module is useful for delaying data. While the demultiplexer is used to select which PWM generator will receive the angle data.

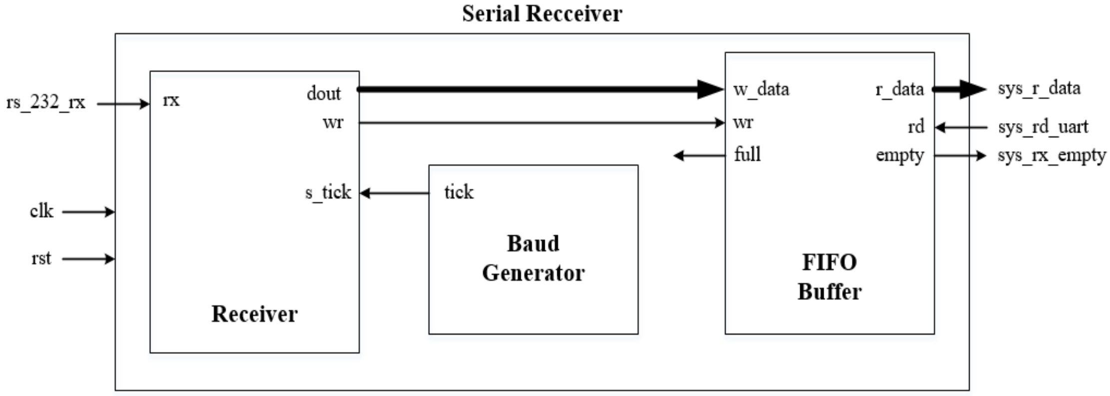

Figure 3. Block diagram of serial receiver module

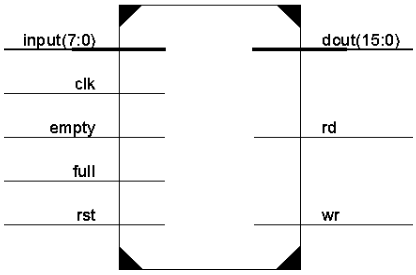

Figure 4. Module of data holder

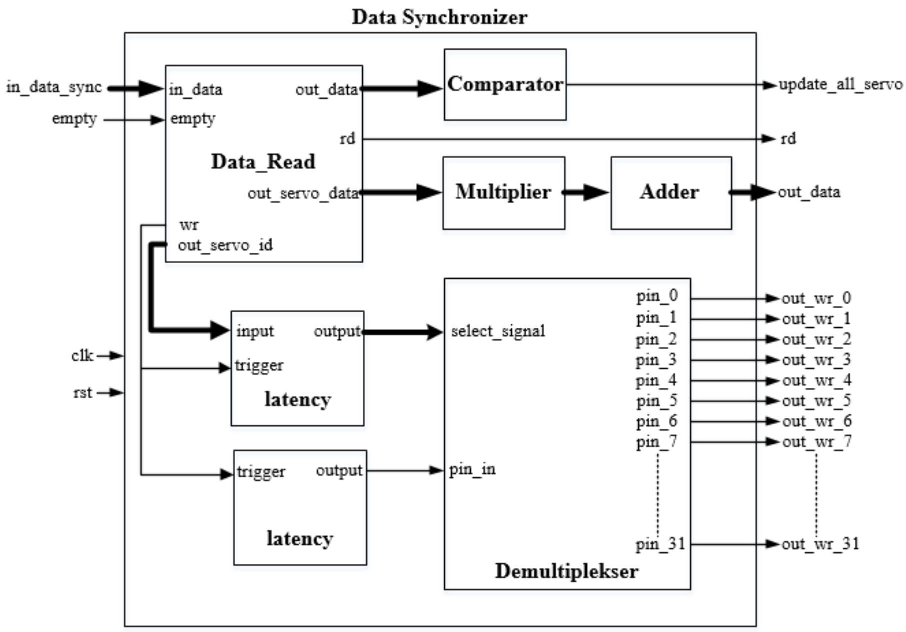

Figure 5. Block diagram of data synchronizer module

2.4. PWM generator module

PWM generator module is shown in Figure 6. The PWM generator is used to generate PWM signal based on the received angle data. Duty cycle is defined by using (1).

$$\text{Duty Cycle} = \left[\frac{\text{Angle Data}}{250}\right] + 1 \quad (1)$$

The 20-bit angle data received from the data synchronizer is converted to a PWM signal on the PWM generator. The block diagram of the PWM generator module can be seen in Figure 7. The working principle of PWM generator is to use counter. There are 2 registers, one is used to save duty cycle and the other one is used save period. When the counter starts counting, the PWM signal will be high. Then, if the counter reaches the same value as in duty cycle register, the PWM signal will be low. The counter will keep run until the value is as same as in period register. When the counter value is the same as the period register value, the counter will be reset and start counting again from the beginning. The period register value is fixed, so that the generated PWM period remains. While the duty cycle value varies according to the data sent by the data synchronizer. If the PWM generator module does not receive an update signal, then new angle data will not be entered in the duty cycle register. The PWM generator module will produce the same PWM signal continuously until it receives new data and an updated signal.

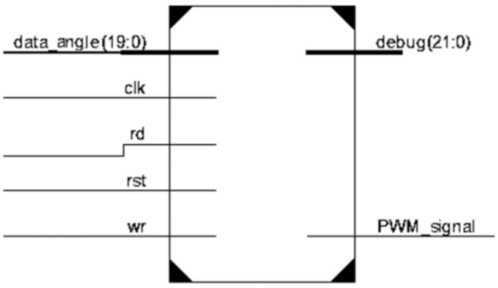

Figure 6. Module of PWM generator

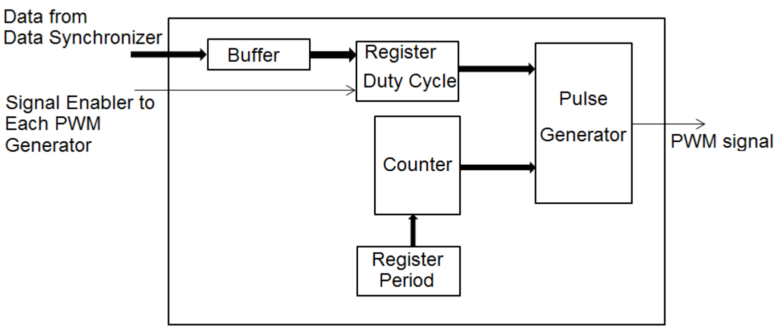

Figure 7. Module of PWM generator

3. RESULTS AND ANALYSIS

In this research, performance of the FPGA system are confirmed by simulations. The simulations were conducted according to serial receiver, data holder, data synchronizer, PWM generator, and overall FPGA system. The simulation results are as follow.

3.1. Serial receiver testing

In the serial receiver module, the internal module is set up with a baud rate of 38400 at 200 MHz clock. Simulation result is shown in Figure 8. Testing for sending data on the first time frame is successful, this is indicated by the change in the value of the data bus sys_r_data 00000000 to 11001000. Testing on the second time frame is also succeeded in accordance with the change in value of 11001000 to 00000001 when the pulse signal is given to sys_rd_uart, which is indicated by circle A.

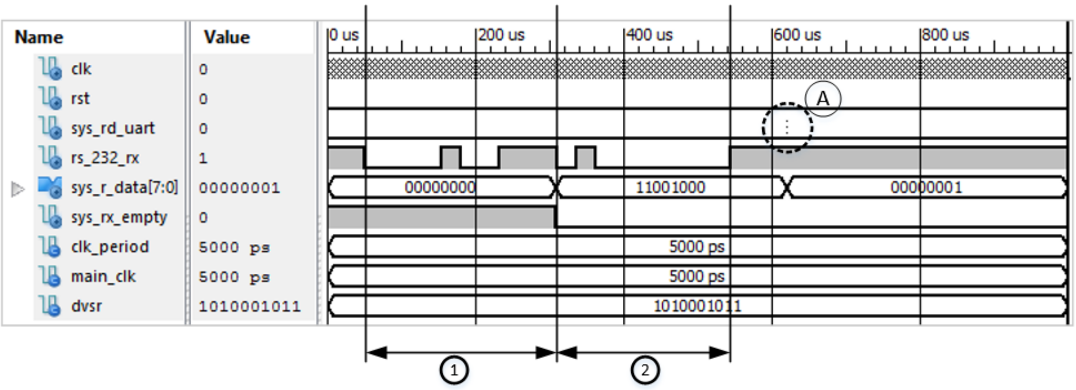

Figure 8. Simulation result of serial receiver module

3.2. Data holder testing

Simulation result of the data holder module can be seen in Figure 9. During simulation, the module is observed when the state is idle, one_data, wait_data, and two_data. The simulation is done by providing data input of 11001000 and 00000001 with 2 clock delay and providing an empty low signal so that the data can be read. At the first time frame, the data holder module is in an idle state, that is waiting for an empty signal with a low logic. At the second time frame, there is state one_data, marked by the input signal empty low logic and the first 8 bits data is 11001000. The read signal will be logic high and the output changes from 0000000000000000 to 0000000011001000. Then the read signal becomes low again and the state moves to the state wait_data as shown the third time frame. At the fourth time frame, there is state two_data, marked by the input signal empty low logic and the second 8 bits data is 00000001. The read signal will be logic high and the output changes from 0000000011001000 to 0000000111001000. Then the read signal becomes low again, the logic signal is high and state back to be idle.

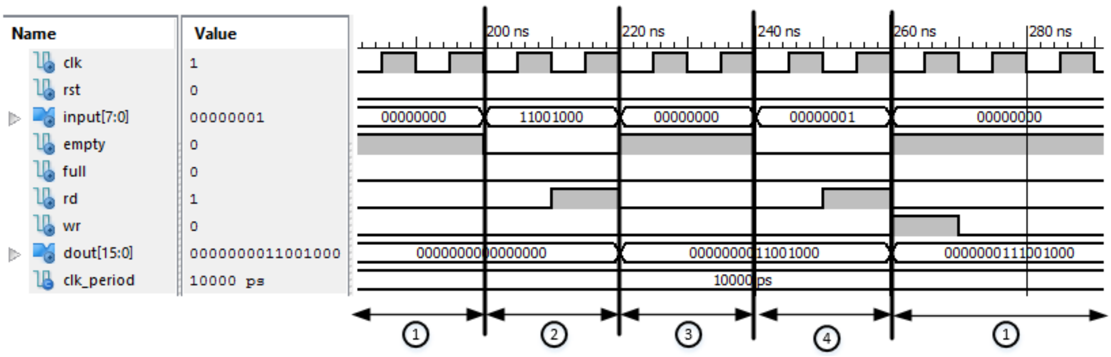

Figure 9. Simulation result of data holder module

3.3. Data synchronizer testing

The simulation results of the data synchronizer module can be seen in Figure 10, where the simulation is done by receiving 16 bits data 0000000111001000 in the first time frame, 0000001011111010 in the second time frame and 0000010001111101 in the third time frame. The simulation of data input and output are shown in decimal values to make it easier to check. The first 16 bits data is angle data 11001000 in decimal 200 and ID data 00000001 in decimal 1. Second 16 bit data is angle data 11111010 in decimal 250 and ID data 00000010 in decimal 2. Data 16 third bit is angle data 01111101 in decimal 125 and ID data 00000100 in decimal 4. Testing for data in the first time frame was successful, this is indicated by the change in the value of out_data to 360000 and out_wr_1 having high logic. Testing for data on the second time frame was successful, this is indicated by changes in the value of out_data to 400000 and out_wr_2 logic high. Testing for data on the third time frame was also successful, this is indicated by the change in the value of out_data to 300000 and out_wr_4 with high logic. The angular data conversion process is in accordance with the design, and the write output is also in accordance with the ID given.

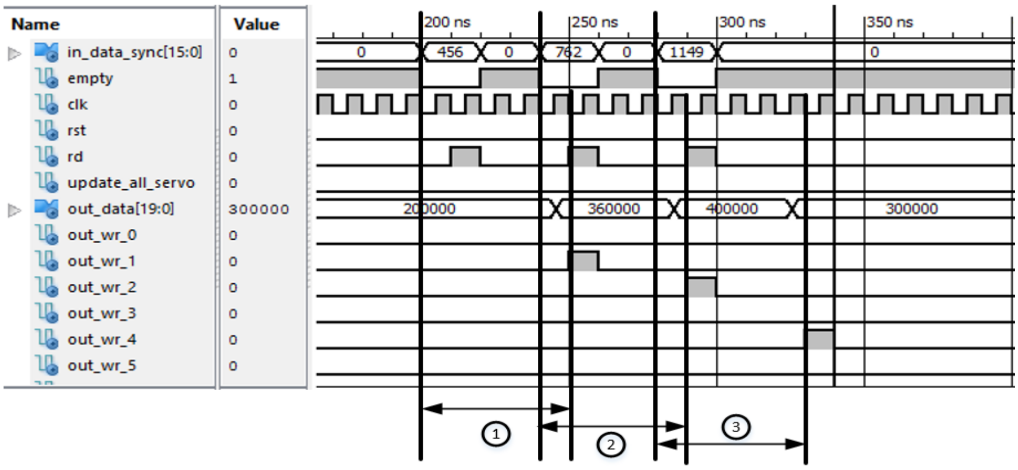

Figure 10. Simulation result of data synchronizer

3.4. PWM generator testing

The simulation results of the PWM Generator module can be seen in Figure 11. In the simulation, input provided is 360000, which is the result of conversion to 200 angular data. The output of the simulation shows the PWM signal generated has a duty cycle of 1.8 ms and a period of 20 ms. In the simulation results there is a delay of 95 ns before the PWM signal has high logic. The results of testing PWM generator modules with the lowest to highest angle data can be seen in Table 2. It can be seen that the error of each data angle is 0%. Duty cycle theory is the result of calculations using (1) in the design of PWM generators. While the duty cycle results are the results of testing using simulations. Tests on this simulations used ideal conditions, so that there was no errors generated.

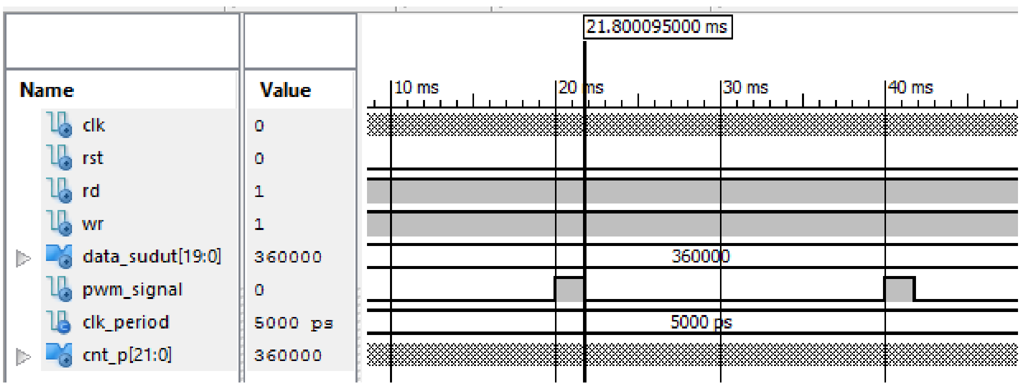

Figure 11. Simulation result of PWM generator

Table 2. Simulation results of PWM generator

Angle data	*Duty Cycle* (theory)	*Duty Cycle* (result)	*Error*
0	1 ms	1 ms	0 %
25	1.1 ms	1.1 ms	0 %
50	1.2 ms	1.2 ms	0 %
75	1.3 ms	1.3 ms	0 %
100	1.4 ms	1.4 ms	0 %
125	1.5 ms	1.5 ms	0 %
150	1.6 ms	1.6 ms	0 %
175	1.7 ms	1.7 ms	0 %
200	1.8 ms	1.8 ms	0 %
225	1.9 ms	1.9 ms	0 %
250	2 ms	2 ms	0 %

3.5. FPGA system testing

In PWM hardware systems, testing with three different baud rates is 38400, 57600, and 76800 at 200 MHz clock. The simulation results of each baud rate can be seen in Appendix 20. At 3800 baud rate the serial data packet can be received within 20.1 ms. The time generated by the 38400 baud rate does not meet the requirements because it is slower than the period in the specification, as a result the data update is too late. At the 76800 baud rate, serial data packets can be received in 7.4 ms. At the 76800 baud rate the duty cycle generated does not match the data sent. This is because the data sent is too fast and the system is not fast enough to receive data. Whereas at 57600 baud rate serial data packages can be received within 14.15 ms. At a baud rate of 57600 duty cycles are generated according to the sent data.

The results of the PWM hardware system simulation can be seen in Figure 12. At the first time frame a 1 ms duty cycle data packet is sent to all PWM generators. Then in the second time frame the duty cycle data packets are sent 2 ms, 1.8 ms, 1.6 ms, 1.4 ms 1.2 ms and 1 ms repeatedly for all PWM generators. On the third time frame, a 1.5 ms duty cycle data packet is sent to all PWM generators.

Circle A in Figure 12 shows that PWM signals can rise simultaneously with the same duty cycle of 1 ms. Then in circle B , PWM signals rise simultaneously with a duty cycle of 2 ms, 1.8 ms, 1.6 ms, 1.4 ms 1.2 ms and 1 ms repeatedly until the last PWM generator. In circle C, PWM signals can rise simultaneously with the same duty cycle of 1.5 ms. The system test was successful because the duty cycle of the generated PWM signal was in accordance with the data sent, and the period of the generated PWM signal was 20 ms.

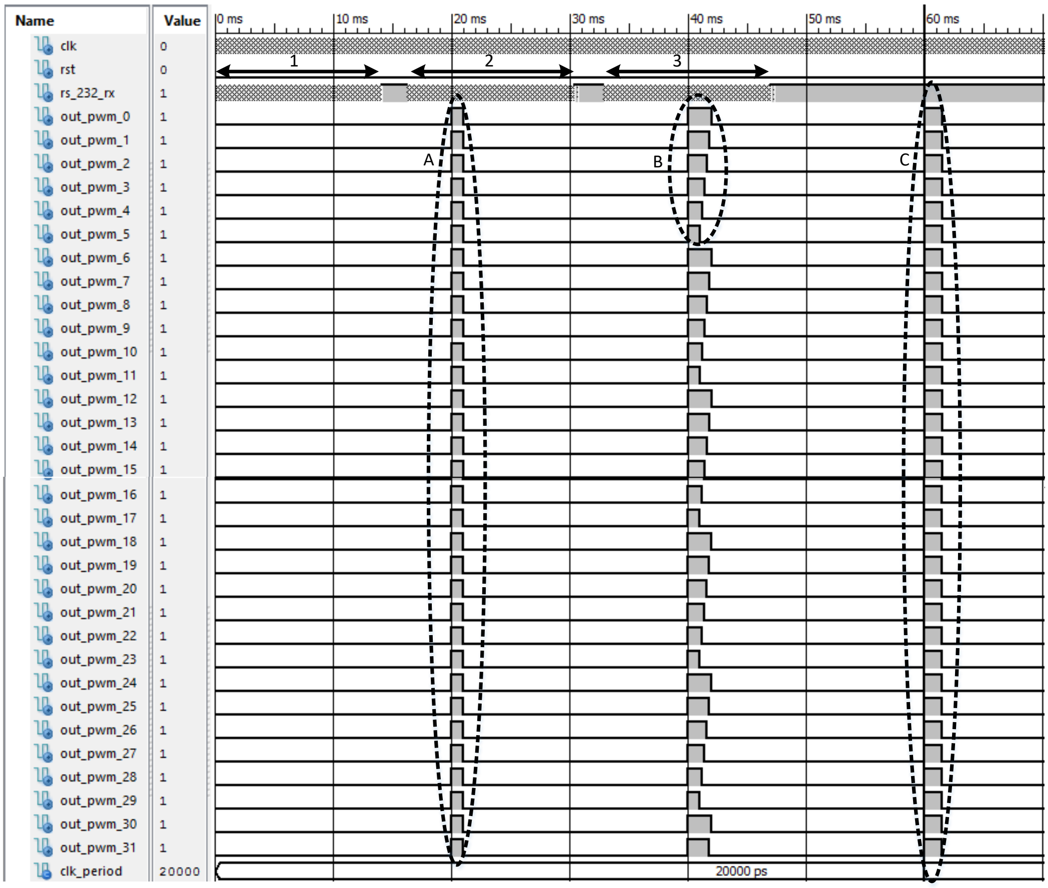

Figure 12. Simulation result of overall system

4. CONCLUSION

In this paper, synchronous multi-channel PWM generator using FPGA for driving servo motors of humanoid robot has been reported. The designed digital system has best perfomance at the baud rate of 57600. The digital system successfully received serial data and generate PWM signals simultaneously with accurate duty cycles and have a fixed period of 20 ms. Since the multi-channel PWM has been generated succesfully, in the future, actual implementation on 32 servo motors will be evaluated.

ACKNOWLEDGEMENTS

This work is supported by Lembaga Penelitian dan Pengabdian Masyarakat Universitas Brawijaya (LPPM - UB) through Hibah Peneliti Pemula (Award No. 436.5/UN10.C10/PN/2020).

REFERENCES

[1] M. A. Miskam, et al., "Humanoid robot NAO as a teaching tool of emotion recognition for children with autism using the Android app," *International Symposium on Micro-NanoMechatronics and Human Science (MHS),* pp. 1-5, 2014.
[2] S. Yi and D. D. Lee, "Dynamic heel-strike toe-off walking controller for full-size modular humanoid robots," *16th International Conference on Humanoid Robots (Humanoids),* pp. 395-400, 2016.
[3] S. Shamsuddin et al., "Initial response of autistic children in human-robot interaction therapy with humanoid robot NAO," *IEEE 8th International Colloquium on Signal Processing and its Applications,* pp. 188-193, 2012.
[4] S. Behnke, M. Schreiber, J. Stuckler, R. Renner and H. Strasdat, "See, walk, and kick: Humanoid robots start to play soccer," *6th IEEE-RAS International Conference on Humanoid Robots,* pp. 497-503, 2006.
[5] S. Rader, et al., "Design of a high-performance humanoid dual arm system with inner shoulder joints," *16th International Conference on Humanoid Robots (Humanoids),* pp. 523-529, 2016.
[6] B. Abror and D. Pramadihanto, "Dance motion pattern planning for K. Mei as dancing humanoid robot," *Int. Conference on Robotics, Biomimetics, and Intelligent Computational Systems (Robionetics),* pp. 6-11, 2017.
[7] D. F. P. Granados, J. Kinugawa, Y. Hirata and K. Kosuge, "Guiding human motions in physical human-robot interaction through COM motion control of a dance teaching robot," *16th International Conference on Humanoid Robots (Humanoids),* pp. 279-285, 2016.
[8] B. Abror, A. R. A. Besari, K. H. A. Subkhan and D. Pramadihanto, "Trajectory dancing modelling of humanoid robot dancing 33 degree of freedom," *International Electronics Symposium (IES),* pp. 340-344, 2016.
[9] J. Seo, J. Yang, J. Kim and D. Kwon, "Autonomous Humanoid Robot Dance Generation System based on real-time music input," *IEEE RO-MAN,* pp. 204-209, 2013.
[10] Y. Lai and Y. Hsu, "Dual Servo Motor Drives Control Using Single MCU," *IEEE Industry Applications Society Annual Meeting (IAS),* pp. 1-6, 2018.
[11] "Datasheet of STM32f407vg," [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f407vg.pdf.
[12] S. M. Trimberger, "Field-Programmable Gate Array Technology," *Norwell, MA, USA: Kluwer,* 1994.
[13] S. M. Trimberger, and J. J. Moore, "FPGA security: motivations, features, and applications," *Proceedings of the IEEE,* vol. 102, no. 8, pp. 1248-1265, 2014.
[14] E. Monmasson, et al., "FPGAs in industrial control applications," *IEEE Transactions on Industrial Informatics,* vol. 7, no. 2, pp. 224-243, 2011.
[15] J. J. Rodríguez-Andina, M. D. Valdés-Peña, M. J. Moure, "Advanced features and industrial applications of FPGAs-a review," *IEEE Transactions on Industrial Informatics,* vol. 11, no. 4, pp. 853-864, 2015.
[16] M. Elnawawy, et al., "Role of FPGA in internet of things applications," *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT),* pp. 1-6, 2019.
[17] A. Muttaqin, Z. Abidin, R. A. Setyawan, and Itsna Az Zahra, "Development of advanced automated test equipment for digital system by using FPGA," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS),* vol. 15, no. 2, pp. 661-670, 2019.
[18] S. S. Vashishtha and K. R. Rekha, "A survey: Space vector PWM (SVPWM) in 3φ voltage sourceinverter (VSI)," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 8, no. 1, pp. 11-18, 2018.
[19] F. A. Silaban, S. Budiyanto, W. K. Raharja, "Stepper motor movement design based on FPGA," *International Journal of Electrical and Computer Engineering (IJECE),* vol. 10, no. 1, pp. 151-159, 2020.
[20] R. Arulmozhiyal and K. Baskaran, "Implementation of a Fuzzy PI controller for speed control of induction motors using FPGA," *Journal of Power Electronics,* vol. 10, pp. 65-71, 2010.
[21] I. M. Popescu, et al., "Evaluation of parallel and real-time processing performance for some vibration signals using FPGA technology," *19th International Carpathian Control Conference,* pp. 365-370, 2018.
[22] Z. Kokosinski and B. Malus, "FPGA implementations of a parallel associative processor with multi–comparand multi–search operations," *International Symposium on Parallel and Distributed Computing,* pp. 444-448, 2008.
[23] N. Fujita, R. Kobayashi, Y. Yamaguchi and T. Boku, "Parallel processing on FPGA combining computation and communication in openCL programming," *IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW),* pp. 479-488, 2019.
[24] F. A. Alquaied, A. I. Almudaifer, M. A. AlShaya, "A novel high-speed parallel sorting algorithm based on FPGA," *Saudi International Electronics, Communications and Photonics Conference (SIECPC),* pp. 1-4, 2011.
[25] W. Chen, S. Ding, Z. Chai, D. He,W. Zhang, G. Zhang, Q Peng, W. Luo, "FPGA-based parallel implementation of SURF algorithm," *IEEE 22nd International Conference on Parallel and Distributed Systems,* pp. 308-315, 2016.
[26] Z. Zhou, et al., "FPGA based high-speed parallel transmission system design and implementation," *International Conference on Intelligent System Design and Engineering Application,* pp. 767-770, 2010.
[27] Z. Qingxianga, C. Litaoa, Y. Chenga, "Design of multi-channel dual-frequency digital receiver based on FPGA," *12th International Conference on Signal Processing (ICSP),* pp. 403-407, 2014.