Stepper motor movement design based on FPGA

Freddy Artadima Silaban[1], Setiyo Budiyanto[2], Wahyu Kusuma Raharja[3]
[1,2]Department of Electrical Engineering, Universitas Mercu Buana, Indonesia
[3]Postgraduate Department of Electrical Engineering, Gunadarma University, Indonesia

Article Info

Article history:

Received Feb 2, 2019
Revised Jul 11, 2019
Accepted Jul 17, 2019

Keywords:

CCW
CW
Full step
Half step
Stepper motor

ABSTRACT

A stepper motor is an electro mechanical device that can convert electrical pulses to the axis of movement. The finding problem in the movement of a stepper motor is cannot respond to the clock signal directly because the motor windings require a clock (sequence) in the correct order. If the control signal given is not correct, the motor is not moving according to the specified precision. To answer these problems, it is necessary to move the stepper motor with a clock signal that works in real time. The research method is done by designing and testing the stepper motor movement in full step and half step with the direction of Clock Wise (CW) and Counter Clock Wise (CCW) movement. These are simulated by using FPGA Isim and implementation using a stepper motor. The results of several experiments have been carried out the stepper motor movement degree according to the input value entered, responding timely movement, and the direction of movement stepper motor.

Corresponding Author:

Freddy Artadima Silaban,
Departement of Electrical Engineering,
Universitas Mercu Buana,
Meruya Selatan, Kembangan, Jakarta Barat 11650, Indonesia.
Email: freddy.artadima@mercubuana.ac.id

1. INTRODUCTION

Stepper motors are used in various industrial applications where low power is required and low speed, torque, fast dynamics and precise positioning are important factors, for example in medical applications for accurate medicament dosage with peristaltic pumps or pipettes and as motion control actuators in dialysis equipment [1]. The main utilization of stepper motors is positioning of stepper motor rotor with required precision. Stepper motor is suitable primarily for tasks where the precision is very important factor. Stepper motors are electromechanical incremental devices that convert electric pulses to shaft motion. The basic operation of the stepper motor allows the shaft to move a precise number of degrees each time a control pulse is sent to the motor. Stepper motor can be a good choice whenever controlled movement is required [2].

The problems found in the movement of the stepper motor are unable to respond to the clock signal directly because the motor winding requires a clock in the correct sequence, if the control signal given is not correct then the motor does not move or move is not in accordance with the precision of the degree of movement Already specified. However, to meet relative high performance applications, stepper motor performance needs to be further improved. Due to stepper motors' specific design, they can reach relatively high linearity even at open loop operation. The angular position error is the difference between the reference angular position and the angular position of the rotor [3]. Stepper motor is an electro-mechanical device that can convert electrical pulses to the movement shaft. The Parallel processing power of FPGA is utilized here to implement multiple stepper motor controls [4]. Due to the fact that the flexibility and high performance of Field programmable gate array (FPGA), it has been widely used on the hardware implementation of controller, some examples are about the design of the PID controller [5-7]. FPGA has a small error correction and is free to be implemented in various algorithms [8-10]. FPGA have higher processing speed, offer fast time to market, low design/manufacturing cost and risk, extremely high processing performance and programmability [11-15]. Xilinx Ise 14.5 in this research is used to create a program with VHDL (Very high speed integrated Hard Language Description Language) type and FPGA Isim is used to test and simulate the coded program that has been prepared [16-20]. The hardware description language(HDL) is used in FPGA to communicate with hardware which is very easily interfaced with them, Logic gates are used in FPGA as a memory blocks, these logic gates are interconnected with each other due to these feature it performs complex functions and compute fast response [21-25]. The trial is the first step in the process of producing the right motor movement and real time. Therefore, the researcher simulation and test the movement of stepper motor used simulinkIsim FPGA and application of stepper motor circuit.

2. RESEARCH METHOD

2.1. Stepper motor design movement

The basic concept before doing the program design (coding) and testing motor movements first determine the path systematically. The following are the software simulation stages. Figure 1 is a block diagram of the program implementation process that has been determined, the process of running the program in the process of synthesizing XST to check program errors, then looking at the RTL, implementing the design, producing a programming file, Configuring the Target Device, then running the program.

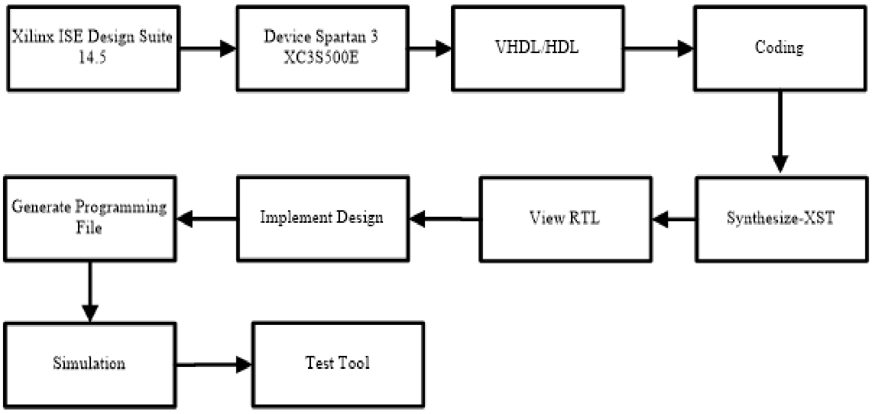
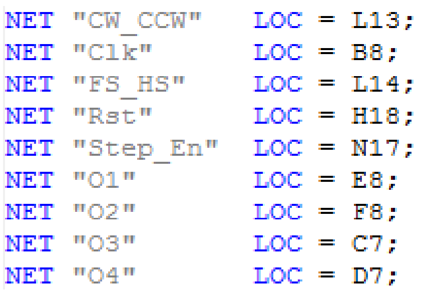

Xilinx ISE Design Suite 14.5
Device Spartan 3 XC3S500E
VHDL/HDL
Coding
Synthesize-XST
View RTL
Implement Design
Generate Programming File
Simulation
Test Tool

Figure 1. Block diagram before simulation and testing

2.2. UCF motor movement design stepper

The design of the stepper motor UCF program aims to declare input variables and output variables tailored to the LOC of the buttons and pins used on the Spartan 3E FPGA device. Here's a picture Figure 2 program listing. Figure 2 is a configuration of the use of pins and buttons between the Spartan 3E FPGA with stepper motor driver. Code net is used to define variables in the VHDL and code LOC programs to describe the input output pins on Spartan 3E FPGA device.



```
NET "CW_CCW"   LOC = L13;
NET "Clk"      LOC = B8;
NET "FS_HS"    LOC = L14;
NET "Rst"      LOC = H18;
NET "Step_En"  LOC = N17;
NET "O1"       LOC = E8;
NET "O2"       LOC = F8;
NET "O3"       LOC = C7;
NET "O4"       LOC = D7;
```

Figure 2. UCF motor stepper program design

2.3. Method

The device generally consists of several blocks: the input block, the process block, and the output block, the overall system block diagram can be shown in Figure 3. Figure 3 consists of several blocks: the input block, the process block, and the output block. The input block consists of several variables: FS (Full Step), HS (Half Step), CW (Clock Wise), CCW (Counter Clock Wise), Step_En (Step Enable), and Rst (Reset). The function of the variable to determine the input value to be sent on the process block. The interface block is a process block consisting of program encoding algorithm and for simulation using Isim FPGA to generate digit bit value then configure target device which function to flash program into FPGA. To block the output (output) is obtained from the stepper motor driver where the input value is received from the FPGA output. The output variables in stepper motors are D, C, B, and A.

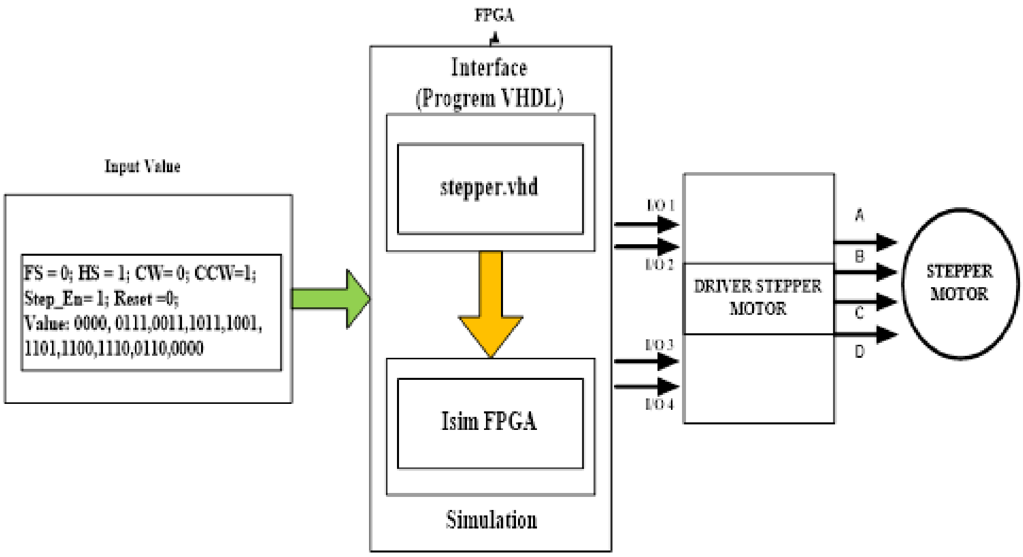


Figure 3. Block movement simulation results stepper

2.4. Synthesize –XST

The synthesize process aims to translate sequences of programs from the VHDL language into an IC-shaped schematic. Synthesize process is complete then View RTL Synthesize can be displayed, as shown below Figure 4. Figure 4 is composed of several schematic block form ICs equipped with the name of each pin. The blocks are translated from behavioral program blocks which consist of Clk, CW_CCW, FS_HS (Full Step / Half Step), Rst (Reset), and Step_En (Step Enable) input pins, while the output pins are O1, O2, O3, and O4. Step Enable serves to allow the driver to accept commands to move the motor. Clock functions to generate a pulse signal from input and output. Here is the schematic block stepper compiler.

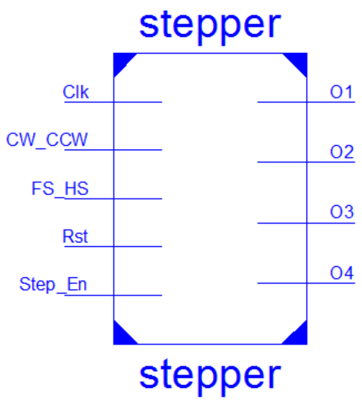


Figure 4. Results process synthesize program

3. RESULTS AND ANALYSIS

3.1. Implementation phase

The purpose of translation implemantation is to check, and translate the program into series in the form of IC. Figure 5 is a sequence of design composers simulating the movement of stepper motors with the direction of CW (Clock Wise) and CCW (Counter Clock Wise), composed of several logic gates and flip-flop circuit. The digital logic gates that make up the digital IC stepper are the AND logic gate, the logic gate of OR logic, the NOR logic gate, and the D Flip-flop. Flip-flop serves to store information as well as calculate the beat and synchronize variable time signal input for some of the time signal that is referenced. Arrangement of digital logic circuit ie 21 slot flip-flop gate, number 4 input LUTs 17, Occupied Slice 17, total 4 input LUTs counted 33, number BOBs IOBs as much as 9, number of BUFGUMXs 1.

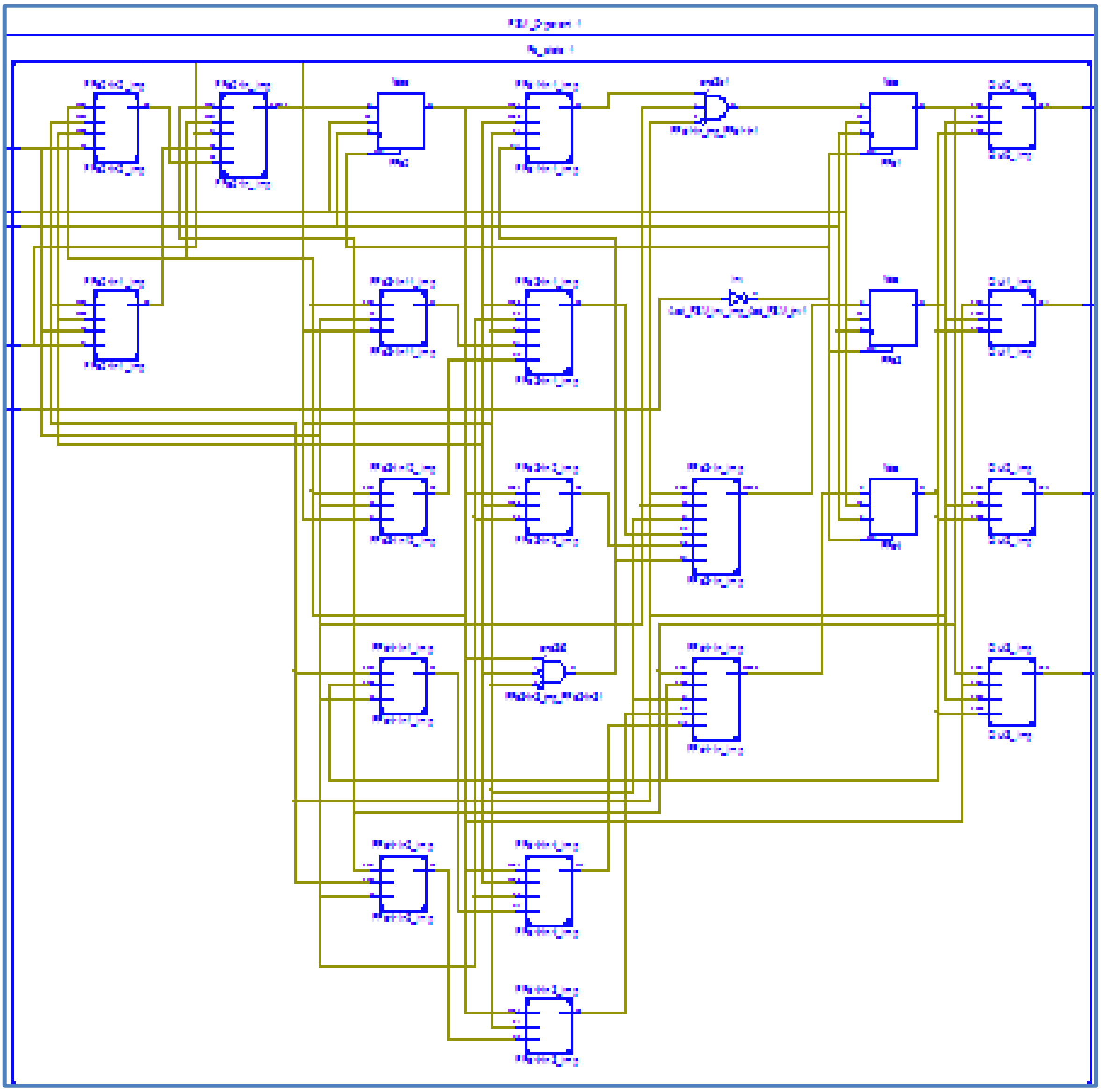


Figure 5. Block of stepper motor simulator composer

3.2. Test movement motor stepper status full1

Testing at this stage is a simulation where the value of Full1 (0111) serve as the input value to move the stepper motor and the result as shown in Figure 6. Figure 6 shows that the value used as input is the value of full1 0111. If the value fs_hs = 1 and cw_ccw = 0 the motor moves half step with the clockwise direction. The values at the output are o1 = 0, o2 = 0, o3 = 1, and o4 = 1.

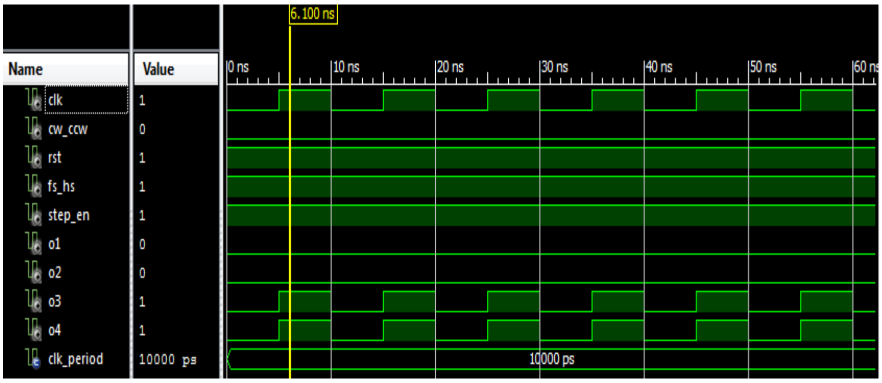


Figure 6. Movement simulation of half1 stepper motion

3.3. Test movement motor stepper status half1

Testing at this stage is a simulation where the value of Half1 (0011) serves as the input value to move the stepper motor and the result as shown in Figure 7 below. Figure 7 shows that the value used as input is the value of half1 is 0011. If the value of fs_hs = 0 and cw_ccw = 0 the motor moves full step with the clockwise direction.

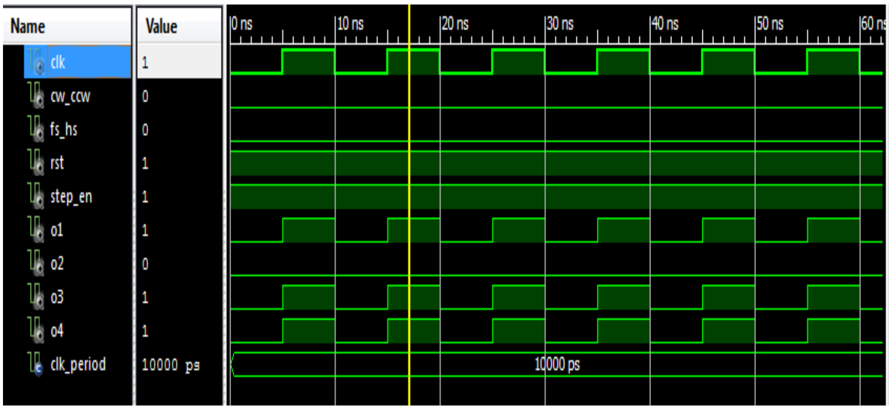


Figure 7. Movement simulation movement stepper full2

3.4. Test movement motor stepper status reset

Stepper motor step testing starts from the input reset status where the value of input = 0, if reset = 0 the motor is not moving and LED status off, as in Figure 8 below. Figure 8 shows that the motor can not move and all led A, B, C, and D status of when the reset state = 0.

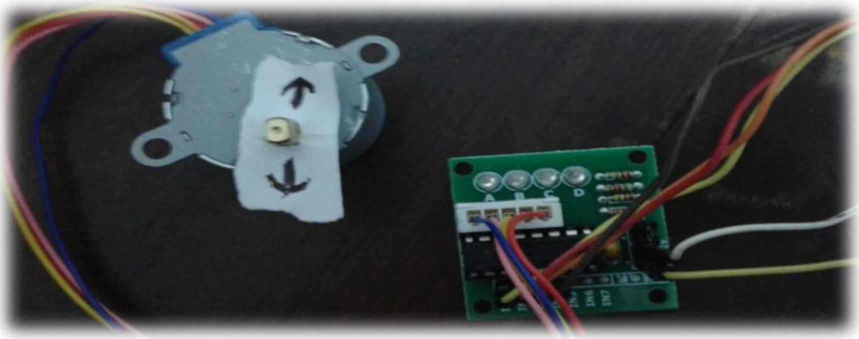


Figure 8. Status LED Off

3.5. Test movement motor stepper status full1

Next test stepper motor movement with the input case value is full1 = 0111 and the result as shown Figure 9. Figure 9 is a simulation result of movement of stepper motor with case full1. The result of movement is half step with the output value 0011 and led live is DC while BA status is off. The degree of producing movement is 45^0 with the clockwise direction.

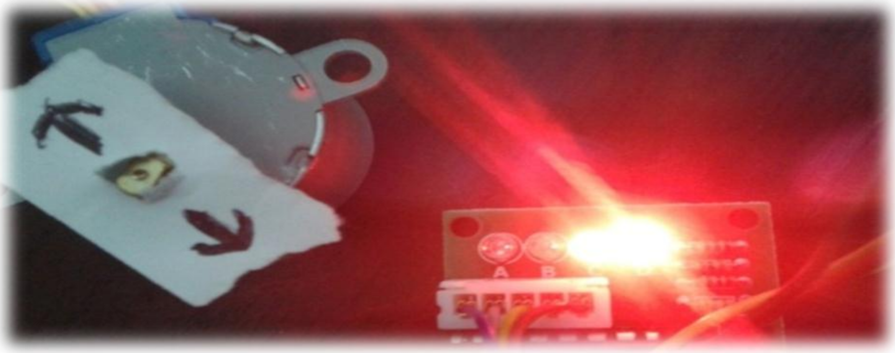


Figure 9. Status LED D and C On

3.6. Test movement motor stepper status half1

Next test stepper motor movement with the input case value is half1 = 0011 and the result as shown Figure 10. Figure 10 is a simulation result of movement of stepper motor with case half1. Result of movement resulted is Full step with value of output 1011 and led C is alive while ABD status is off. The degree of producing movement is 180^0 with the clockwise direction.

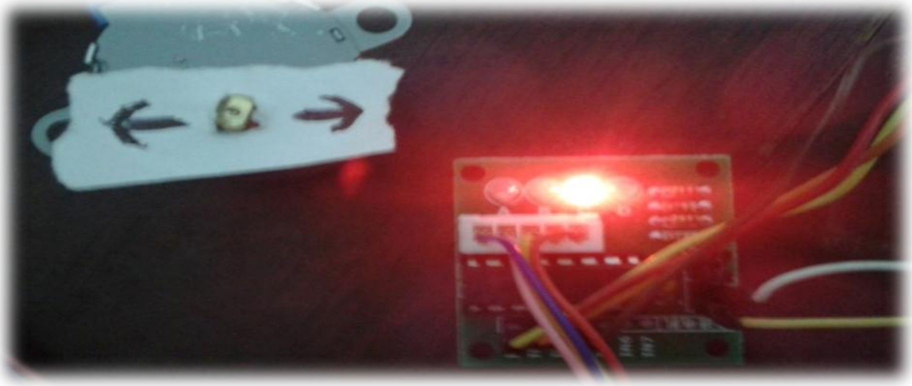


Figure 10. Status of LED C On

3.7. Test movement motor stepper status full4

The next step is the movement of the stepper motor with a case input value of Full4 = 1110 and the results as shown in Figure 11. Figure 11 is the simulation results of the stepper motor movement with the Full4 case. The result of movement is half step with the output value 0110 and led that live, namely AD while BC status off. The degree of producing movement is 360^0 in a clockwise direction. Furthermore, if the direction of motor movement is anticlockwise, the motion of the stepper motor becomes full1 with the output value of 0111 with LED D status on, as shown in Figure 12.

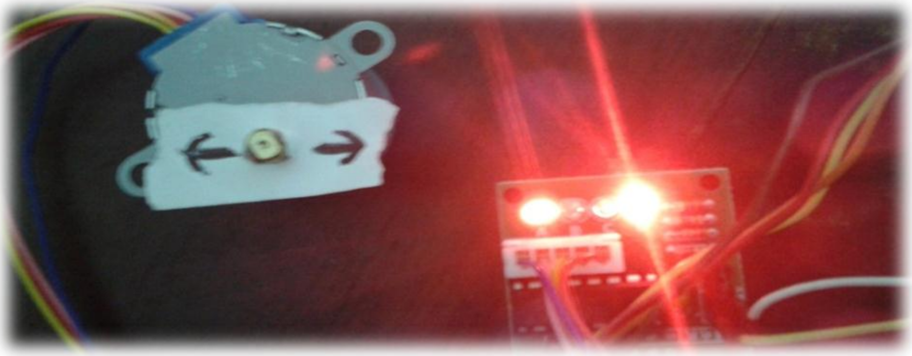


Figure 11. Status of LED A and B On

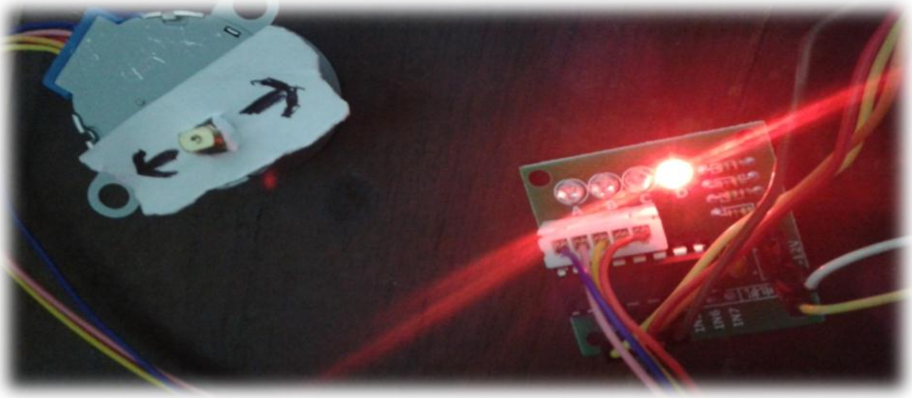


Figure 12. Status of LED D on

3.8. Summary of motion step motion simulation

Having obtained the results of simulation testing can be made summary of values and direction as the Table 1. Based on Table 1 is summary of the degree values and the direction of movement of the stepper motor is obtained begin from 0 degrees until to 360 degrees moving in accordance with a predetermined degree and moving in accordance with the direction specified. Overall the movement of the stepper motor works in real time.

Table 1. Summary of stepper motor movement simulation value

Case	EN	Motion		Direction		Input Value				Output Value				Level
		FS	HS	CW	CCW	In1	In2	In3	In4	O1	O2	O3	O4	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0°
Full1	1	0	-	-	1	0	1	1	1	0	1	1	1	90°
Half1	1	0	-	0	-	0	0	1	1	1	0	1	1	180°
Half1	1	0	-	-	1	0	0	1	1	0	1	1	1	90°
Full2	1	-	1	0	-	1	0	1	1	1	0	0	1	135°
Full2	1	0	-	-	1	1	0	1	1	1	1	0	1	270°
Half2	1	0	-	0	-	1	0	0	1	1	1	0	1	270°
Half2	1	0	-	-	1	1	0	0	1	0	1	1	1	90°
Full3	1	-	1	0	-	1	1	0	1	1	1	0	0	225°
Full3	1	0	-	-	1	1	1	0	1	1	1	1	0	360°
Half3	1	0	-	0	-	1	1	0	0	1	1	1	0	360°
Half3	1	0	-	-	1	1	1	0	0	1	1	0	1	270°
Full4	1	-	0	0	-	1	1	1	0	0	1	1	0	315°
Full4	1	0	-	-	1	1	1	1	0	0	1	1	1	90°
Half4	1	0	-	0	-	0	1	1	0	0	1	1	1	90°
Half4	1	0	0	-	1	0	1	1	0	1	1	1	0	360°

4. CONCLUSION

Provide a statement that what is expected, as stated in the "Introduction" chapter can ultimately Based on the results of program design and simulation testing obtained at the time of research, it can be concluded as follows:

a. From the results of this study obtained the arrangement of digital logic circuit that are slips flip-flop 21 gates, number 4 input LUTs as much 17, Occupied Slice as much as 17, total 4 input LUTs as much 33, number BOBs IOBs as much as 9, and number of BUFGUMXs 1.

b. Simulation results of motor movement testing is that if the value at the input reset = 0 then the stepper motors can not move in accordance with the predetermined degree. If the input value is given 0111, fs_hs = 1, and cw_ccw = 0 the motor moves half step clockwise. The values at the output are o1 = 0, o2 = 0, o3 = 1, and o4 = 1. If input variable fs_hs gave the value = 0, the motor move full stepp. If the input value is 0011, fs_hs = 0, and cw_ccw = 0 the motor moves full step clockwise. The value at the output is o1 = 1, o2 = 0, o3 = 1, and o4 = 1. If the input value is 1011, fs_hs = 1, and cw_ccw = 0 the motor moves half step clockwise. The value at the output is o1 = 1, o2 = 0, o3 = 0, and o4 = 1. If the input value is 1001, fs_hs = 0, and cw_ccw = 0 the motor moves full step clockwise. The value at the output is o1 = 1, o2 = 1, o3 = 0, and o4 = 1. If the input value is 1101, fs_hs = 1, and cw_ccw = 0 the motor moves half step clockwise. The value at the output is o1 = 1, o2 = 1, o3 = 0, and o4 = 0. If the input value is 1100, fs_hs = 0, and cw_ccw = 0 the motor moves full step clockwise. The value at the output is o1 = 1, o2 = 1, o3 = 1, and o4 = 0. If the input value is 1110, fs_hs = 1, and cw_ccw = 0 the motor moves half step clockwise. The value at the output is o1 = 0, o2 = 1, o3 = 1, and o4 = 0. If the input value is 0110, fs_hs = 0, and cw_ccw = 0 the motor moves full step clockwise. The value at the output is o1 = 0, o2 = 1, o3 = 1, and o4 = 1.

REFERENCES

[1] L. Przenioslo and M. Holub, "Development of microprocessor, time optimized stepper motor driving algorithm," *International Conference on Methods and Models in Automation and Robotics (MMAR)*, pp. 174-179, 2017.

[2] I.Virgala, M.Kelemen, A. Gmiterko, and T.Liptak, "Control of Stepper Motor by Microcontroller," *Journal of Automation and Control*, vol. 3, no. 3, pp 131-134, 2015.

[3] B. Liu, P. Fransson, A. Ronquist, and B. Winroth, "Compensation of Load-Dependent Position Error for A Hybrid Stepper Motor," *IEEE International Conference on Mechatronics and Automation ICMA*, pp. 846-851, 2017.

[4] Mohan, Arun G and Vijina K., "Design and Implementation Of Based Multipurpose Real-Time Controller For Hybrid Stepper Motor," *Internasional Journal Of Modern Trends In Engineering and Research*, SJIF:3518, 2016.

[5] C-Keng Lai, J-Shan Ciou, and Chia-Che T., "FPGA-based Stepper Motor Vector Control System Design," *International Automatic Control Conference (CACS)*, pp. 1-5, 2017.

[6] J. Wadgaonkar, K. Bhole, and P. Singh., "Floating Point FPGA Architecture of PID Controller," *International Conference on Industrial Instrumentation and Control (ICIC)*, pp. 1259-1263, 2015.

[7] R. Rustam and M. Ibrahim Al Khoory, "FPGA-Based Stepper Motor Controller: An FPGA Based Design Project to Improve Level of the Complexity in Project–Based Learning," *Advances in Science and Engineering Technology International Conferences(ASET)*, pp. 1-5, 2018.

[8] Prof.T. Zhao, W-J Shen, N-yi Ji, and H-hong Liu, "Study and implementation of SPWM Microstepping Controller For Stepper Motor," *IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pp. 2298-2302, 2018.

[9] A. Ramesh, *et al.*, "A Novel Three Phase Multilevel Inverter with Single DC Link for Induction Motor Drive Applications," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 2, pp. 763-770, 2018.

[10] Shamala N., and C. Lakshminarayana, "Performance Enhancement in Active Power Filter (APF) by FPGA Implementation," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no.2, pp. 689~698, Apr 2018.

[11] A. Rincon, F. Angulo, and F. Hoyos, "Controlling a DC Motor through Lypaunov-like Functions and SAB Technique," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 4, pp. 2180 – 2198, Aug 2018.

[12] G. Renuka,V. Usha Shree, P. Chandra, S. Reddy, "Comparison of AES and DES Algorithms Implemented On Virtex-6 FPGA and Microblaze Soft Core Processor," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 5, pp. 3544-3549, Oct 2018.

[13] H. Lin Oo, S Anatolii, Ye Naung, K. Zaw Ye, and Z. Min Khaing, "Modelling and Control of an Open-Loop Stepper Motor in Matlab/Simulink," *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 869-872, 2017.

[14] Ioana Ionica, Mircea Modreanu, A. Morega, and Cristian Boboc, "Design and Modeling of a Hybrid Stepper Motor," *International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pp. 192–195, 2017.

[15] L. Zhang, Li Liu, J. Shen, J. Lai, Kesen Wu, Z. Zhang, and Jian Liu, "Research on stepper motor motion control based on MCU," *Chinese Automation Congress (CAC)*, pp. 3122-3125, 2015.

[16] S. Thakra, and A. Mahesh Joshi, "PWM Waveform Generation Using Rotary Encoder on Spartan-3E Starter Kit," *International Conference on Computational Intelligence & Communication Technology (CICT)*, pp. 1-4, 2017.

[17] W. Hong-bin, ZZhe, C. Xu-hui, and W. Yuan-bin, "Stepper Motor SPWM Subdivision Control Circuit Design Based on FPGA," *IEEE/ACIS 16th International Conference on Computer and Information Science (ICIS)*, pp. 889 – 893, 2017.

[18] Suma, and V. Sridhar, "ROI-based Compression on Radiological Image by Urdhva-Tiryagbhyam and DWT Over FPGA," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 2, pp. 211-218, 2017.

[19] T. Nguyen, "The control structure for DC motor based on the flatness control," *International Journal of Power Electronics and Drive System (IJPEDS)*, vol. 8, no. 4, pp. 1814-1821, 2017.

[20] D. Castro, L. Gaia, H Seiti Ogawa, and B. de Carvalho Albertini, "Automated Generation of HDL Implementations of Dadda and Wallace Tree Multipliers," In *Computing Systems Engineering (SBESC) 2017 VII Brazilian Symposium on IEEE*, pp. 17-22, 2017.

[21] A. Y. Yousef, and M. H. Mostafa, "Simulation and Implementation of Multiple Unipolar Stepper Motor Position Control in the Three Stepping 1 Modes using Microcontroller," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 4, no. 1, pp. 29-40, Oct 2016.

[22] Gadakh, S. N., and A. S. Khade, "FPGA implementation of high speed vedic multiplier," *International Conference & Workshop on Electronics & Telecommunication Engineering (ICWET 2016)*, pp. 4-4, 2016.

[23] U. Farooq, R. Chotin-Avot, M. Azeem, M. Ravoson, M. Turki, and H. Mehrez, "Inter-FPGA routing environment for performance exploration of multi-FPGA systems," *Proc. 27th Int. Symp. Rapid Syst. Prototyping (RSP)*, pp. 107-113, 2016.

[24] W Kim, D Shin, Y Lee, CC Chung, "Simplified torque modulated microstepping for position control of permanent magnet stepper motors[J]," *Mechatronics*, vol. 35, pp. 162-172, 2016.

[25] W. Zheng, R. Liu, M. Zhang, G. Zhuang, and T. Yuan, "Design of FPGA Based High-Speed Data Acquisition and Real-Time Data Processing System on J-TEXT Tokamak," *Fusion Eng. Des.*, vol. 89, no. 5, pp. 698–701, 2014.

BIOGRAPHIES OF AUTHORS

Freddy Artadima Silaban received his Master of Engineering (MT) degree in Electrical Engineering, Information Technology Specialization, Gunadarma University, Depok, 2016. He currently teaches at Universitas Mercu Buana, Jakarta (Indonesia).

Setiyo Budiyanto received his Ph.D in Electrical Engineering, University of Indonesia (2016). Currently he is Research Reviewer - with Certification Uses Standardized Methods of Ministry of Research, Technology and Higher Education (Indonesia) Regulation No. 69 - 2016; He is Associate Professor in Department of Electrical Engineering - Universitas Mercu Buana, Jakarta (Indonesia).

Wahyu Kusuma Raharja Head of Postgraduate Department of Electrical Engineering Gunadarma University's until now. He is currently teaches and research in Gunadarma University, Depok (Indonesia).