SMART DOOR SYSTEM IMPLEMENTATION BY USING FPGAs

Tee Han Shen, Hairul Hadee and Wan Rahiman[1]

School of Electrical and Electronic Engineering
Universiti Sains Malaysia, Engineering Campus
Pulau Pinang, Malaysia.

Corresponding Author's Email: [1]wanrahiman@usm.my

ABSTRACT: The project is to develop a system focusing on the robotic philosophy to be implemented by using Quartus II software and Altera DE2-115 FPGA board. The project is designing in a way that an automated door system will open when people is approaching the door from either side and close when there are nobody around automatically. The door users are detected by using IR sensors, then the inputs are interpreted by digital circuit in field-programmable gate array (FPGA) board which has been designed using Quartus II software and the door is operated by using stepper motor. D flip flops and some logic gates are included to generate clock pulses which will activate the stepper motor to move 90° whenever the sensors are triggered or not triggered. A bidirectional shift register is used to control the direction of rotation of motor which is clockwise 90° or anti-clockwise 90°.

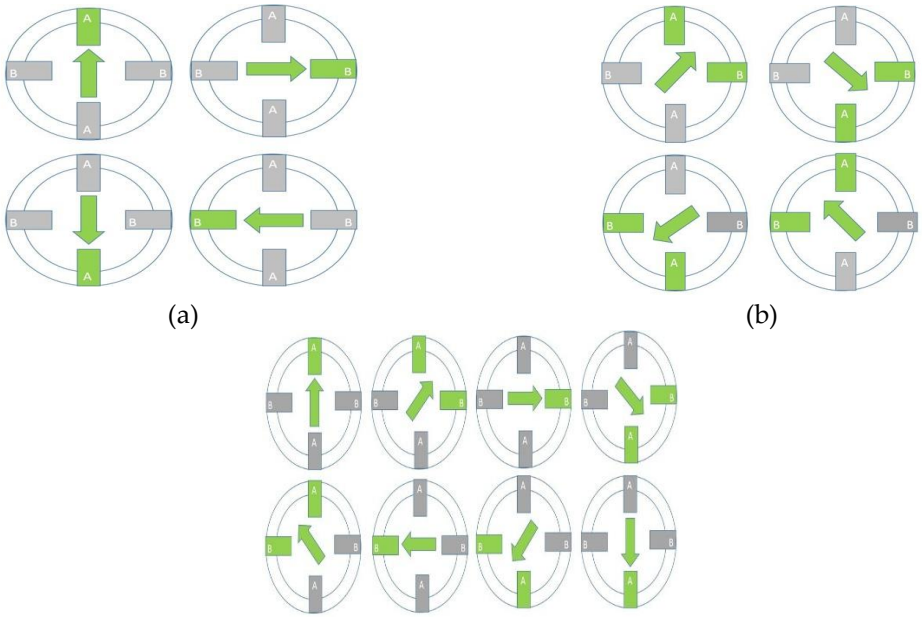
KEYWORDS: *Smart Door, Stepper motor, Bidirectional shift register*

1.0 INTRODUCTION

A smart door is an automated door which will open and close automatically by using sensor to detect the presence of door user. Rajiwade et al. (2016), Hung et al. (2015) and Chowdhury (2013) proposed few techniques and methods for the door system is controlled by digital circuit with inserted inputs from the sensor. The door will move by using stepper motor or servo. In our project, we are using stepper motor to move the door. This paper contains design and implementation of using FPGA platform with analysis on the suitable data interfaces for real-time information.

Condit (2004) explains that stepper motor internal structure consists of a permanent magnetic rotating shaft, called the rotor, and electromagnets on the stationary portion that surrounds the motor, called the stator.

Based on Figure 1, the works have decided to make the stepper motor moving in Full Step with low torque. In order to move the rotor clockwise, in our case, the A electromagnet is magnetised to become south pole, the north pole of the rotor will be attracted to it as was mentioned by Roy et al. (2014). Then, B will be magnitised and the A will be demagnetised. This causes the north pole of the rotor to point to the position in of B. The action moves the stepper motor a step which is 7.5°. Next, B will be demagnetised, $\bar{A}$ will be magnetised. It moves the rotor's north pole and stop at $\bar{A}$. After that, $\bar{B}$ will be magnetised and the rotor will be at $\bar{B}$. The sequences of magnetisation of electromagnet are repeated to run the stepper motor clockwise. The sequences magnetising the electromagnet will need to be reversed in order to turn the rotor in anti-clockwise direction.



(a) (b)

(c)

Figure 1: Illustrate the effect of sequence on rotational motion: (a) Full Step (Low Torque) with one phase HIGH (b) Full Step (High Torque) with two phases HIGH (c) Half Step (Best Precision) with one two phases HIGH

Since the step angle of the stepper motor is 7.5°, in order to move 90° per time, the motor needs to move 12 steps as our door system only involves 0° and 90°. Let the input pins controlling the 4 phases of stepper motor be A, B, C and D. The inputs for each phase of the stepper motor from step 1 to step 12 are shown below.

Table 1: Steps and inputs of stepper motor to rotate 90°

Step	**Input**			
	A	**B**	**C**	**D**
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	0	0	0
5	0	1	0	0
6	0	0	1	0
7	0	0	0	1
8	1	0	0	0
9	0	1	0	0
10	0	0	1	0
11	0	0	0	1
12	1	0	0	0

Clock pulses generating system is designed to generate 12 clock pulses whenever the sensors are triggered or not triggered. The 12 clock pulses will be connected to the clock of the bidirectional shift register which gives output to the ULN2003 motor driver to change the state of stepper motor for 12 times. The usage of ULN2003 is to amplify the output from GPIO pin to power up the stepper motor.

Frequency divider is included in the digital circuit to reduce the extremely high 50MHz build-in frequency of FPGA board to desired frequencies which are 5.9Hz and 2.98Hz.

2.0 SYSTEM DESCRIPTION

The circuit consists of 2 clock pulses generators to create 12 pulses whenever sensors are triggered or not triggered, 1 bi-directional shift register to control the direction of the stepper motor, 2 frequency dividers, one is for the clock pulses generators and bidirectional shift register, another one is for the delay system, 2 IR sensors as the inputs (O is outside sensor, I is inside sensor), 1 delay system to delay the inputs and 1 ULN2003 motor driver and 1 stepper motor as the outputs which is built externally.

A. Delay System

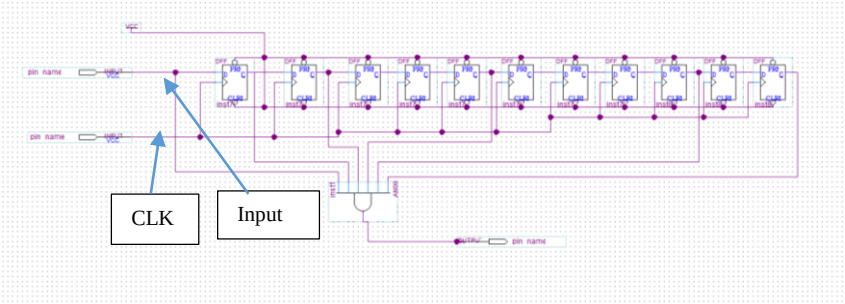


Figure 2: Delay system

In order to avoid immediate closure of the door after the person leave the sensor detection range, a delay circuit is set. In this circuit, the input is from frequency divider 2 then passing through a series of D flip-flops which act as a serial in serial out (SISO) shift register. This will cause the current input to come out as final output after a certain period of delay. The delay time is 0.335*11 = 3.69s, which means that the input will come out as final output after 3.69s. The input of this delay system is from the output of NOR gate connecting both sensors. When both sensors are LOW, then the input will be HIGH and when one or both of the sensors are HIGH, the input will be LOW.

B. Clock Pulses Generating System

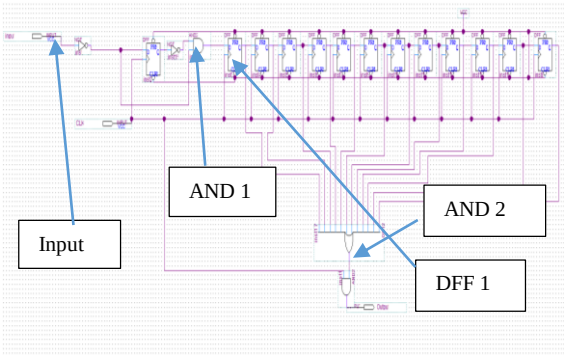


Figure 3: Clock Pulses Generating System

The function of the clock pulses generating system is to generate 12 clock pulses for the input of the Bidirectional Shift Register to shift 12 times as the door is only required to rotate 90° in clockwise or anticlockwise. The stepper motor will move 7.5° per shifting, thus 12 times of shifting are required in order to make the stepper motor to rotate 90°.

The input of the clock pulses generating system 1 is connected to the AND gate from the delay system passing through a NOT gate, whereas clock pulses generating system 2 is directly connected to the AND gate of delay system. All the flip-flops in the system are connected with the

same clock, the output of frequency divider 1.

C. Bidirectional Shift Register

The main purpose of the bidirectional shift register is to enable the motor to rotate clockwise and anticlockwise when the respective inputs are inserted. All the AND gates at left side is to shift serial data to the right when there is a HIGH input whereas the AND gates at right side is to shift serial data to the left when there is a HIGH input. The input for R is from the output of OR gate with inputs from both sensors. Whereas the input for L is from the output of NOR gate with inputs from both sensors. When R is HIGH and L is LOW, the AND gates at right side will be activated for serial data entry to shift to the right one step at every subsequent clock edge.

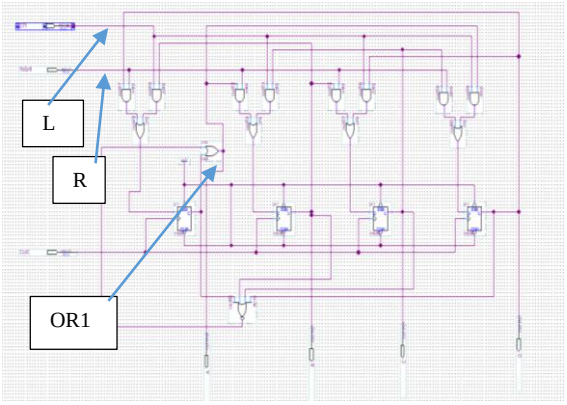


Figure 4: Bidirectional Shift Register

When L is HIGH and R is LOW, the AND gates at left side will be activated to shift the data to the left one step at every subsequent clock edge. The NOR gate and the OR1 are functioned together as mechanism for inserting and isolating only 1 HIGH in the shift register to be shifted right or left. Later, the outputs of the shift register will be assigned to the General Purpose Input Output (GPIO) pins and will be connected to the motor driver to drive the stepper motor. As an effect, shifting of data to the right will cause the stepper motor to rotate 90° in clockwise direction, whereas shifting of data to the left wil cause the stepper motor to rotate 90° in anticlockwise direction.

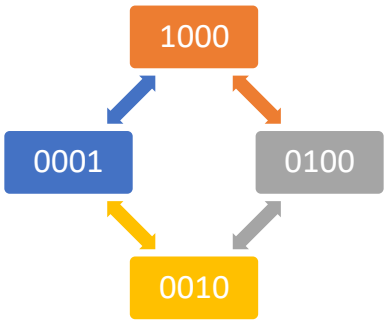


Figure 5: State Diagram of Bidirectional Shift Register

The above state diagram shows the way of shifting used in our project. Whereas the table below is the next state table of the bidirectional shift register.

Table 2: Next State Table of Bidirectional Shift Register

CLK	Input		Present State				Next State			
	L	R	Q_3	Q_2	Q_1	Q_0	Q_{3+}	Q_{2+}	Q_{1+}	Q_{0+}
1	0	1	1	0	0	0	0	1	0	0
2	0	1	0	1	0	0	0	0	1	0
3	0	1	0	0	1	0	0	0	0	1
4	0	1	0	0	0	1	1	0	0	0
5	0	1	1	0	0	0	0	1	0	0
6	1	0	0	0	0	1	0	0	1	0
7	1	0	0	0	1	0	0	1	0	0
8	1	0	0	1	0	0	1	0	0	0
9	1	0	1	0	0	0	0	0	0	1
10	1	0	0	0	0	1	0	0	1	0

3.0 RESULTS AND DISCUSSION

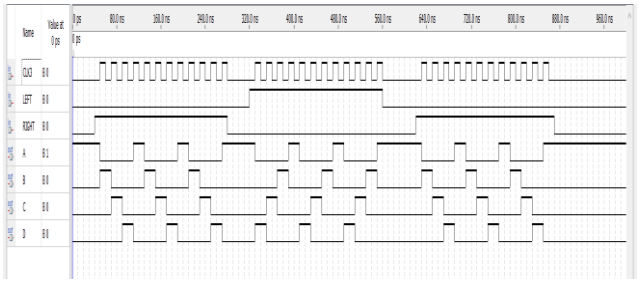


Figure 6: Simulation of Bidirectional Shift Register

The data shifted to the right for 12 times when right input is activated, whereas the data shifted to the left for also 12 times when left input is activated.

Besides, as this smart door is an electronic door, so it requires power source to activate it. The stepper motor is very firm and cannot be rotated by external force easily when the stepper motor is active. However, it will become loose when there is no power source. So a manual lock is set in order to lock the door when the owner is at outside or during the night time for extra safety purpose by double lock the door and for the case when there is power failure occurs where the door has to be locked manually. In order to warn the owner to close the door manually during power failure, a buzzer and also some sort of warning system which will send warning message to the owner's electronic gadget is set up with the inputs connected to an NAND gate. The inputs of this NAND gate will be the voltage source, Vcc of this door system and the backup power source. During normal condition, the voltage source is HIGH and the backup power source will also be HIGH, thus the output will be LOW and the buzzer will not be triggered. When power failure occurs, the voltage source will be LOW and thus the buzzer will be triggered as the output of NAND gate is HIGH now.

4.0 CONCLUSION

Arithmetic Logic Unit is the part of a computer that performs all arithmetic computations, such as addition and subtraction, increment, decrement, shifting and all sorts of basic logical operations. Here, using FPGA which can perform the various arithmetic operations of Addition, Subtraction, Increment, Decrement, Transfer, logical operations such as NAND, OR, XOR, NOT and also the

shift operation. All the above mentioned operations are then verified to see whether they match theoretically or not. The above given waveforms show that they match completely thereby verifying our results

5.0 ACKNOWLEDGEMENT
This project was supported by the Universiti Sains Malaysia under the Bridging Grant 304.PELECT.6316121. We would like to thanks the university for allowing us to use the FPGA Laboratory at the School of Electrical and Electronic Engineering, Universiti Sains Malaysia to complete the project.

6.0 REFERENCES

Chowdhury N, Nooman S. & Sarker S. (2013). Access Control of Door and Home Security by Raspberry Pi Through Internet, *International Journal of Scientific & Engineering Research, Vol. 4, Iss. 1*

Condit R. & Jones D. W. (2004). *Stepping Motors Fundamentals*. AN907, Microchip

Hung C. H., Bai Y. W. & Ren J. H. (2015). Design and Implementation of a Door Lock Control Based on a Near Field Communication of a Smartphone. *International Conference on Consumer Electronics-Taiwan (ICCE-TW), pp.45-46*

Rajiwade B., Thakar S., Pokharkar P. & Malbhare S. (2016). Design and Implementation of Smart Door Lock Control System using Bluetooth Controller of Smart Phone. *International Research Journal of Engineering and Technology, 3(11), 482-484*

Roy T. S., Kabir H. &, Chowdhury M. A. M. (2014). Simple Discussion on Stepper Motors for the Development of Electronic Device. *International Journal of Scientific & Engineering Research, 5(1), 1089-1096*