Digital System Control for Three-Degrees of Freedom Mechanical Arm with FPGA

José Rafael Guzmán-Sepúlveda[1], René de Jesús Romero-Troncoso[2]
Universidad de Guanajuato
Facultad de Ingeniería Mecánica, Eléctrica y Electrónica
Salamanca, Guanajuato, México
jrafaelgs@ieee.org[1], troncoso@salamanca.ugto.mx[2]

Abstract

This work has as objective to show the design process and implementation of three-degrees of freedom mechanical arm, controlled by a Digital System, described under VHDL, and implemented into a FPGA (Field Programmable Gate Array). The Digital System for control is designed through the interconnection of circuits and functional blocks. The FPGA terminals are used to control the sequences of three stepper motors and this way to achieve the movement of the three-degrees of freedom arm, the interface stage among the control and the motors, is developed with a power electronic circuit based on transistors. This design proposal is a didactic model that seeks to illustrate the stages of the design and, mainly, the Digital System for control the movements of the mechanical arm. For this case, the mechanical design of the arm does not depend on a specific trajectory, it is a free trajectory and user can decide, trough the FPGA interface, the mechanical arm movement. The mechanical arm proposal does not have the capacity to exercise big forces, not to lift big weight, but rather it seeks to be an easy construction model and accessible for everyone, since it can be built until with recycled material, just as it was made in this case, for educational purposes.

1. Introduction

For modern men, it is practically impossible to conceive their life without the presence of the Digital Electronics because the quantity of electronic applications that appears with the daily life is huge and Digital Electronics applications are so many and so common [1]. Sensors, micro-chips and actuators are commonly found in modern products. Even the automobile industry which is traditionally a Mechanical Engineering domain is putting micro-chips in modern vehicles and mechanical structures [2]. Mechatronics, with its multidisciplinary approach to design and develop of engineering solutions, provides an important direction to following teaching and research. Hence, the teaching of such a multidisciplinary degree has become necessary for successful engineers [3]. Universities around the world have responded with the introduction of new degrees in Mechatronics and Electromechanical Engineering arises because in the same way that it happens in Digital Electronics and in Electronics in general, there are so many applications where we can find electronic devices involved with mechanical devices, one of those applications is the mechanical structures control [4]. In control and construction terms, a mechanical arm of pedagogic and illustrative purposes have the same structure of the big industrial manipulators, for this reason, it is of great importance to understand the design techniques employed for their control.

Every Digital System design begins with a group of specifications and characteristic that should complete the pattern to develop and it culminates with a logical diagram that can be represented by Booleans functions, which are programmed in the FPGA [5]. Actually, the logical designer task is simplified with the use of descriptive languages of circuits. Although diverse descriptive languages exist, the VHDL (Very high-speed integrated Hardware Description Language) it is the most popular and their use most of the time. The descriptive language VHDL has for objective to carry out the realization of the circuit in programmable logic, this technique is known like Design for the Synthesis [1].

The proposed Digital System to control the mechanical arm was done with top-down Hierarchical Design technique. With this technique, the designer has the easiness to separate the design stages and to determine several abstraction levels, being able to treat the systems

978-0-7695-3320-9/08 $25.00 © 2008 IEEE
DOI 10.1109/CERMA.2008.74

like black boxes to develop them in a parallel way. This project was divided fundamentally in three main parts: the mechanical part of the arm, the power electronic interface circuit and the FPGA stage. It is very important to say that the proposed Digital System for the stepper-motors control was not made thinking about a specific trajectory. In this project, the Digital System, as in the same way that the mechanical design, does not depend on any movement trajectory. It is proposed an FPGA interface so the movement trajectory is determined from user. Figure 1 shows the main parts this project was divided:

Figure 1. Design Process Stages

2. Development

2.1. Mechanical Stage

Mechanical stage focus to the construction of the mechanical arm structure, for this, a great variety of elements were used, such as aluminum links, gears, pulleys, bands, stepper motors, axes, guides and some elements of utility that were obtained from electronic devices in disuse. This stage of the project was very interesting because the mechanical unions among gears, bands, pulleys, they were own production, and all mechanical devices were positioned in a precise way to obtain the wanted movement. Figure 2 shows how the mechanical arm was built:

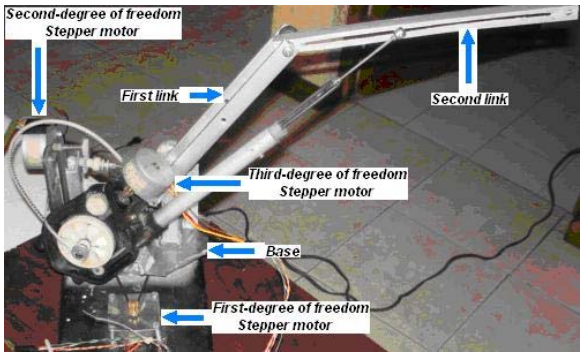

Figure 2. Three-Degrees of Freedom Mechanical Arm Structure

Figure 2 shows how the arm was built to achieve the three-degrees of freedom movement: in the inferior part of the image the first motor is appreciated that allows the movement of the whole base where it is the arm; in the left superior part of the image the motor that helps to carry out the movement of the second-degree of freedom that corresponds to the movement of the first link, with its respective mechanical joining that allows the wanted movement leave; finally, Figure 2 also shows the stepper motor that was positioned to carry out the movement of the third-and-last-degree of freedom, with the appropriate joining this movement it is reflected in the second link of the mechanism. The mechanical arm structure has been built in a free way, for this project it is not necessary to take a synthesis procedure to determine the structure dimensions, like the links length, based on a specific precision points trajectory. The structure dimensions were selected arbitrary and the mechanical arm structure was not built thinking about a specific movement trajectory. Figure 2 does not show the mechanical arm actuator because for the present case the actuator can be a pincer, an imam, a supporting, an electromagnet, a pencil, it can be practically anything, the mechanical arm actuator is the user decision.

2.2. Electronical Stage

Electronical stage is mainly the used power electronic interface to connect the FPGA and the stepper motors mounted in the mechanical arm structure. The FPGA contains a very extensive logical package for the present application but its terminals can not be directly connected to the stepper motors phases because the demanded current for the motors is much bigger that the one that would support the FPGA.

To solve this, it is necessary to implement a denominated “interface circuit”, which allows to interpret the FPGA outputs and to take the corresponding actions in the motors giving the necessary and enough power, at the same time the FPGA is protected. For this, optic insulators were used to eliminate the physical contact between the FPGA outputs and stepper motors inputs. Figure 3 shows schematic diagram of the implemented circuit:

The power electronic interface circuit shown at Figure 3 is just for one stepper motor phase, it is necessary to implement twelve times this same power electronic interface circuit because every stepper motor has four phases and this project uses three stepper motors for its movement. Some times FPGA outputs have a very low voltage level, that is why another option for interface electronic circuit is to put a buffer between FPGA and Optic-Insulator, but for the present project it is not nec-

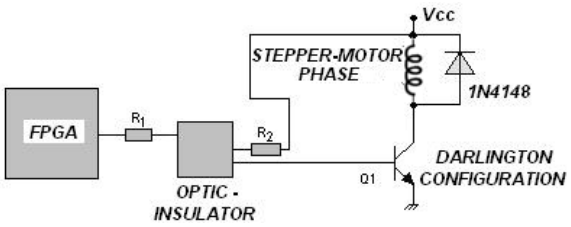

Figure 3. Power Electronic Interface Circuit

essary. Optic-Insulator output has the same state that the corresponding stepper motor phase (activated or disabled), but it is isolated from the FPGA and has a voltage level a little higher than its input. The optic insulators output cannot provide the demanded current for the stepper motors and that is why a power stage is advisable. Darlington configuration transistors, with high output gain, were used in this power stage. Finally, interface circuit also has available the connectors for the corresponding stepper motor phase terminals: stepper motor common terminal is connected to feeding voltage (Vcc) and the corresponding stepper motor phase is connected to Darlington configuration transistor collector. Figure 4 shows the complete implemented interface electronic circuit and its connection to FPGA:

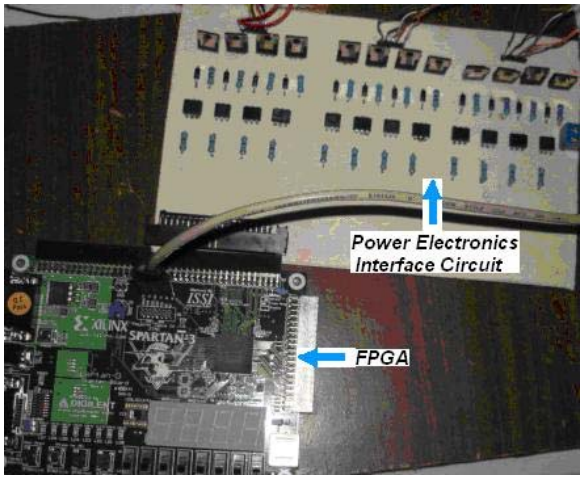

Figure 4. Power electronic interface circuit connected to FPGA

There are many different ways to implement the power electronic interface circuit because there are a great variety of power electronic devices that can be used, this is just a standard power electronic interface circuit and user can decide which ones electronic devices is going to use to implement it.

2.3. FPGA Stage

The FPGA stage consists on a Digital System described under VHDL that is able to control the movements of the mechanical arm. This design offers the option of activating the sense and the movement qualification: the user selects the motor that wants to activate, the sense in which wants that it moves and finally it activates the beginning of the movement.

The programmable logic can be used as a way of personalizing the logical designs, that is to say, to design thinking about the own hardware. The first PLDs was programmed by mask and they were developed by makers of computers, at the beginning of the 60´s the programmable logic arrived for fusible and from then on this technique was available so much for small as for big users [6].

To describe in a simpler way the Digital System for control the mechanical arm movement diverse functional blocks they were used that allowed to be carried out the necessary functions. With foundation in Hierarchical Design, the Digital System was built with the following functional blocks: the Time-Base, the Stepper Motor Finite States Machine and the Control Finite States Machine.

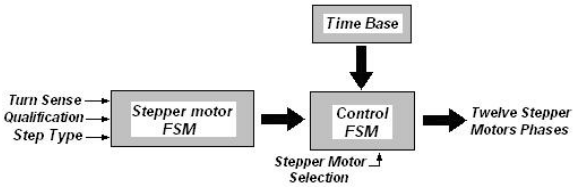

Figure 5. Digital System functional blocks

Figure 5 shows the Digital System functional blocks and gives an idea about the interconnection between them, including the user options for selecting some mechanical arm movement.

Time-Base functional block is like an oscillator, it does not receive any value or command from user but it sends a predetermined and constant period pulse to the Control Finite States Machine, for this case, the period is 1ms; the relatively low frequency pulse generated in Time-Base functional block is used to process the corresponding stepper motors phases through the power electronic interface circuit and, at the same time, it allows the stepper motors to interpret the present phases states and to react when there is a change in the movement sequence; specifically, the Time-Base functional block can be designed like a constant module counter, and the count module is determined by the user, this way, the Time-Base pulse period can take so

many values, but once given it does not change; it is necessary the Time-Base pulse allows a good operation to the power electronic interface circuit devices.

Stepper motor Finite State Machine receives from user the stepper motor turn sense (CW or CCW) and the step type (Half-Step or Full-Step) values, also user begins the selected stepper motor movement with the qualification signal. As it can be seen, user activates the corresponding stepper-motor turn sense signal, the step type signal and selects the stepper-motor he wants to move, so the mechanical arm structure can take many and so different movement trajectories. This is the reason why the mechanical arm movement does not depend on a specific movement trajectory, because user can decide, and in fact determines, the mechanical arm movement trajectory. Stepper Motor FSM sends to Control FSM the corresponding stepper motor phases states with the movement characteristics selected from user; Stepper Motor FSM contains the appropriate stepper motor phases sequences for generate both senses movements, also the half-step and full-step movements.

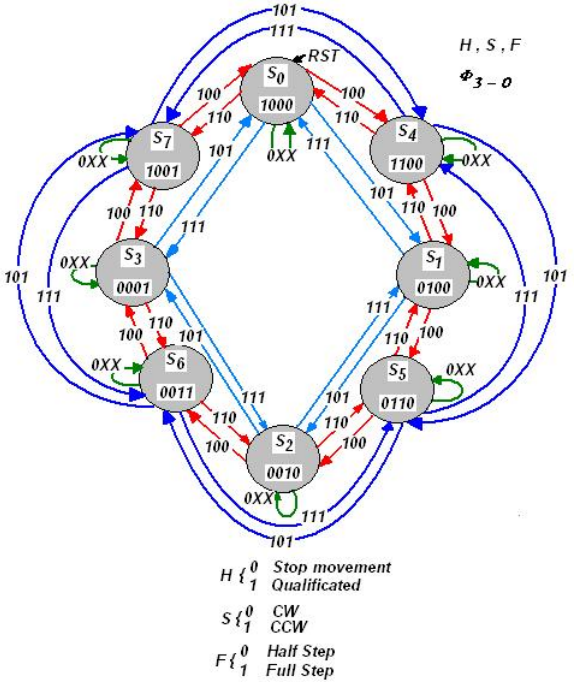

Figure 6. Stepper Motor Finite States Machine

Figure 6 shows Stepper Motor FSM states and its characteristics. Qualification signal, Turn-Sense signal and Step-Type signal are input signals and their values are defined directly from user. The stepper motor phases can polarize in two different ways: the first one with just one polarized phase, and the second one with two polarized phases. In Summary, the Stepper Motor FSM considers four states with just one polarized phase and another four states with two polarized phases, this way, Half-Step and Full-Step movements can be generated. There is not any state with three or four polarized phases because it is not a stepper motor movement for interest. Figure 6 shows the possible movement sequences too: CW and Full-Step, CW and Half-Step, CCW and Full-Step, and finally, CCW and Half-Step. The Stepper Motor FSM in Figure 6 presents a reference State that the FSM takes when the Reset signal (RST) is activated. Reset and FPGA-clock signals are the highest hierarchy signals because RST signal restarts all digital system and CLK signal makes this design a synchronous digital system determining the digital system operation speed, but another high hierarchy signal is the qualification signal, with qualification signal user allows or impedes the selected stepper motor movement, this way, when qualification signal is disabled the stepper motor does not move. Turn-sense and step-type signals do not have a high hierarchy but they are important because they determine the movement characteristics. For the present project it is necessary to consider the Half-Step movement because it is more precise movement than Full-Step movement, this way, the proposed Control Digital System offers two precision levels for stepper motor movement, and consequently, the possibility of a better mechanical arm positioning.

Control Finite State Machine receives the Time-Base pulse, the corresponding stepper motor phases states depending of the movement characteristics selected from user and the stepper motor selection signal; Time-Base pulse and stepper motor phases states come from internal functional blocks but the stepper motor selection signal comes from user; Control FSM interprets the stepper motor phases states depending of the stepper motor movement characteristics selected in Stepper Motor FSM, user selects the stepper motor that wishes to move and the Control FSM assigns the phases states to the corresponding stepper motor. This process is carried out in the following way: Control FSM receives the stepper motor phases states at FPGA-clock speed, it is to say, every FPGA-clock pulse the Control FSM updates the information corresponding to the stepper motor phases states and the movement qualification, this way, any movement sequence change produces the pertinent actions almost immediately and the FPGA outputs states reflect the effect of such a change almost immediately too; the Time-Base functional block works as a constant module counter and sends to Control FSM a constant period pulse determined by the designer;

Control FSM sends to FPGA outputs the corresponding stepper motor phases states to the selected stepper motor, but not at FPGA-clock speed, it send that values at Time-Base pulse speed. All Digital System proposed functional blocks are totally synchronized with the FPGA-clock but it is necessary to reduce its frequency and to carry out the corresponding stepper motor phases states with an appropriate speed.

3. Results

The results of each one of the stages in that the design was divided are the following ones: the structure of the mechanical arm was built mounting the motors in appropriate positions to generate a free movement in each one of the three-degrees of freedom; an interface circuit was designed, based on power transistors, to connect the FPGA with the stepper motors; finally, a Digital System was designed that generates the necessary and appropriate signals to control the mechanical arm movement.

As mentioned, both the mechanical arm structure and the Digital System for control the stepper-motors movement were not made thinking about a specific movement trajectory, but it is proposed an FPGA interface to take the corresponding values about turn sense, step type and the selected stepper-motor from user.

In the next four pictures, it can be seen the results of the Control Digital System simulation procedure. It is important to say that the pictures show all the possible kind of movements, CW and Full-Step, CCW and Full-Step, CW and Half-Step, CCW and Half-Step, and finally, an additional "hold on" state, but only for one stepper-motor. The simulation results that figures 7, 8, 9 and 10 show will be only the movement options of the selected stepper-motor.

To obtain a correct simulation they were defined some control signals. These signals are a kind of flags that can give some information about the stepper-motor movement conditions: Reset signal (RST) has the capacity to restart the complete Digital System, it is activated in the beginning of the simulation, for a short time, and then it keeps disabled the rest of the simulation; Clock signal (CLK) is exactly the FPGA clock signal, it has a standard 50MHz frequency; Turn sense signal (C) selects the CW or CCW turn sense, if it is disabled the default turn sense is the CW turn sense, if the turn sense signal is activated then the stepper-motor movement is in CCW turn sense; Step type signal (F) selects the stepper-motor step type, it will select a Full-Step sequence when it is activated, and when it is disabled the default step type is the Half-Step sequence; Start Type Movement signal (STM) is just a control signal that indicates the exactly instant when there is a change in the stepper-motor movement characteristics; Step signal (S) is just a control signal that indicates the number of steps necessary to complete the step type selected sequence; the Information Transfer signal (S) is a flag signal that indicates the information transfer is being in the correct way, it is disabled when the information transfer is correct; Ready signal (RDY) is a flag signal that is activated to indicate the selected movement has been completed, and it is also activated when the selected stepper-motor is in the "hold on" state; finally, the Phases signal (PHI) is a signal bus that represents the selected stepper-motor phases states.

An important explanation about the simulation procedure is in the results it can be seen the stepper-motor sequence changes are done in a lower frequency than FPGA standard frequency.

Figure 7 shows the first part of the simulation, it can be seen the complete Half-Step and CW sequence, and some of the Half-Step CW sequence. Figure 8 shows the second part of the simulation, it can be seen the end of the Half-Step and CCW sequence, and the begining of the Full-Step CW sequence. Figure 9 shows the third part of the simulation, it can be seen the end of the Full-Step and CW sequence, and the begining of the Full-Step CCW sequence. Figure 10 shows the last part of the simulation, it can be seen the end of the Full-Step and CCW sequence, and the hold state.

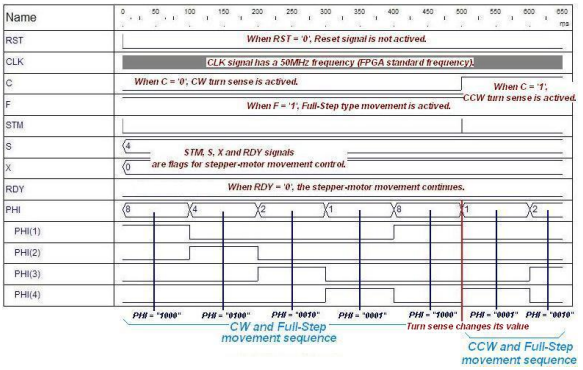

Figure 7. Stepper Motor Simulation Part 1 (Half-Step, CW and CCW)

4. Conclusion

Mechatronics uses the knowledge of other branches of the Engineering, but it has its technological problems, applications and particular designs as independent Engineering. This project requires

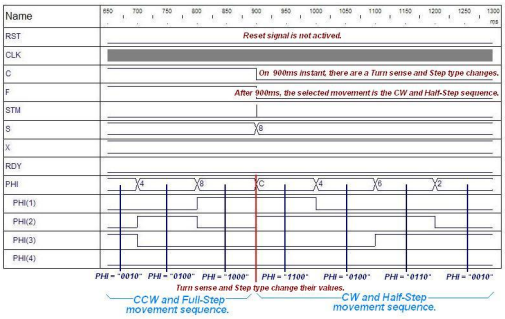

Figure 8. Stepper Motor Simulation Part 2 (Half-Step CCW, Full-Step CW)

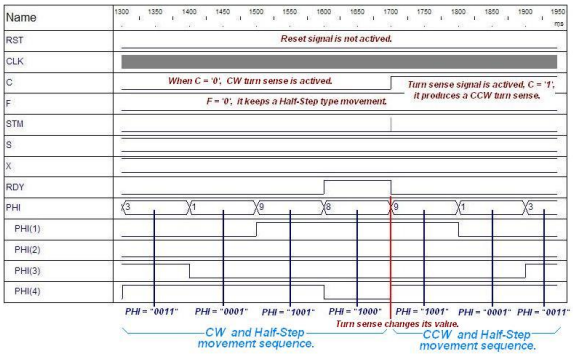

Figure 9. Stepper Motor Simulation Part 3 (Full-Step, CW and CCW)

applying knowledge and skills previously acquired, such as: Digital logic design (Finite State Machine); Hardware description languages (VHDL); Hierarchical design technique in the implementation using FPGA; Electronics circuit design (interface and drive circuit); and finally, Motors and actuators (stepper motor).

The main reasons for choosing an FPGA design is to provide tools to achieve rapid prototyping of digital systems that is essential in the current electronic industry reflecting current industry practice; and to use VHDL hardware description language to describe digital systems, synthesis and actual hardware implementation as an alternative design platform [3].

The user interface for this project can be a PC keyboard, a computer or an electronic circuit, but for this case the selected user interface was the own FPGA controls, push-buttons and switches. This user interface is not the best one but it is very practical and functional.

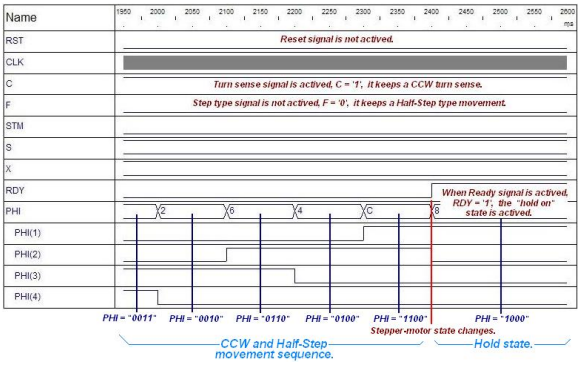

Figure 10. Stepper Motor Simulation Part 4 (Full-Step CCW, Hold)

The biggest complications arose in the mechanical stage because the mechanical transmissions and unions structures are own production, it is necessary to verify that the position of the elements is the appropriate one to generate a free movement. This mechanical arm proposal was built with recycled material.

This project takes a very common industrial application: to control the mechanical structures movement, but also, at the same time, it allows to do the easiness with which an FPGA can control complex mechanisms. The three-degrees of freedom mechanical arm proposed cannot exercise big forces and cannot lift heavy objects either, it can be built even with recycled material, like in this case, so the main objective of this project is to illustrate about the digital design process and to be focused to FPGA stage.

References

[1] R. de Jesús Romero Troncoso, *Electrónica Digital y Lógica Programable*. Lascuráin de Retana No. 5 C.P. 36000: Universidad de Guanajuato, 1st ed., 2007.

[2] V. Giurgiutiu, J. Lyons, D. Rocheleau, and W. Liu, "Mechatronics/microcontroller education for mechanical engineering students at the university of south carolina," *Mechatronics*, vol. 15, pp. 1025–1036, 2005.

[3] K. C. Aw, S. Q. Xie, and E. Haemmerle, "A fpga-based rapid prototyping approach for teaching of mechatronics engineering," *Mechatronics*, vol. 17, pp. 457–461, October 2007.

[4] R. Isermann, "Mechatronics design approach," in *The mechatronics handbook* (R. H. Bishop, ed.), CRC Press, 2002.

[5] National Semiconductor, *Programmable Logic Design Guide*, May 1986.

[6] National Semiconductor, *Programmable Logic Devices Databook and Design Guide*, 1990.