Reconfigurable Hardware Based Programmable Digital Circuit Design for a Rotational Stepper Motor

Yahya Taher Qassim yahyaaldabbagh@yahoo.com

Rabee Mouffag Hajim rabeemoufag@yahoo.com

Department of Computer Engineering

College of Engineering/ University of Mosul

Abstract

In this research, a hardware digital circuit was designed for a programmable rotational stepper motor using VHDL as a design tool and the FPGA as a target technology. The design is implemented on a Spartan 3 starter kit (supported with XC3S200 field programmable gate array). The 50 MHz provided by the starter kit is divided to obtain the necessary delay time between the motor phases that ranges between 2-10 m seconds. Through output selections, the direction of rotation of the stepper motor besides the magnitude of the angle of movement and the rotation speed can be controlled. The advantage of using reconfigurable hardware (FPGA) instead of a discrete digital component is that the designer can make modifications to the design easily and quickly, and the total design represents an embedded system (which works without computer). The total programmable hardware design that make control on the stepper motor movement, occupy an area that did not exceed 12% of the chip resources.

Keywords: Stepper Motor, Motion control, FPGA.

تصميم دائرة رقمية مبرمجة لدوران محرك الخطوة اعتمادا على الكيان المادي القابل للتشكيل

يحيى طاهر قاسم الدباغ **ربيع موفق حاجم الحيالي**

كلية الهندسة/ قسم هندسة الحاسبات/ جامعة الموصل

الخلاصة

في هذا البحث، تم تصميم دائرة رقمية عملية مبرمجة تسيطر على محرك الخطوة من ناحية اتجاه حركة الدوران ومقدار الزاوية المطلوبة للدوران والسرعة باستخدام لغة وصف الدوائر المتكاملة ذات السرعة الفائقة كطريقة للتصميم. تم تنفيذ الدائرة المسيطرة على اللوح الالكتروني الرقمي Spartan 3 (مزود بمصفوفة بوابات قابلة للبرمجة حقلياً تتسع لتصاميم تصل إلى 200000 بوابة منطقية كحد أقصى). اللوح المذكور مزود بمولد نبضات ترددية مقداره 50 MHz، في حين أن الأزمنة المطلوبة والفاصلة بين شفرات التتابع لعمل محرك الخطوة تتراوح قيمها بحدود ٢-١٠ ملي ثانية، لذا فقد تم تقسيم تردد اللوح أعلاه لغرض النزول به إلى قيم تتلاءم وعمل محرك الخطوة. تم التحكم باتجاه ومقدار زوايا بالإضافة إلى سرعة دوران محرك الخطوة باستخدام المفاتيح المتوفرة على اللوح الحاوي لمصفوفة البوابات القابلة للتشكيل. إن فائدة استخدام الكيان المادي القابل للتشكيل بدلاً من الدوائر المتكاملة التقليدية والمنفصلة، هو أن المصمم يستطيع عمل تحويرات على التصميم بسهولة وسرعة كما أن مجمل التصميم للسيطرة على محرك الخطوة يمثل نظام مطمور (بالإمكان تشغيله بمعزل عن الحاسبة). التصميم المنفذ على اللوح الحاوي لـ FPGA شغل ما لا يزيد عن ١٢% من موارد الرقاقة المذكورة.

Received ٢٨ Aug. 2006 **Accepted 19 April. 2007**

1. Introduction:

Advancements in digital motion control systems are mainly due to advances in power electronics, microprocessor technology and digital control systems. A modern motion control system such as a robotic manipulator arm or a positioning system consists in general of several electrical drives such as stepper motors. There are many previous methods for controlling the stepper motor motion. These methods like operating under the command of a computer system [1], or using microprocessor system (μp) [2]. In [3] the stepper motor motion is controlled by using microcontroller and CPLD. This type of control can be also done by using a modern programmable logic control (PLC), with a suitable selection for the necessary timers in its software.

Another way for designing the required control circuit is by using discrete digital components [4]. These components must be connected together to obtain the desired logic circuit, but the last method is difficult in making modifications to the design.

In this research, another hardware solution is used: Field programmable gate array (FPGA) is suitable for fast implementation and quick hardware verification. FPGA based systems are flexible and can be reconfigured unlimited number of times. These reconfigurable devices

are with no pre-determined function. Each individual is represented as a Bit string that is downloaded to the chip as configuration data. This data includes a definition of each cell's functionality as well as the topology of the system [5].

To create a digital design into the FPGA, one of two methods must be used as a design entry. The first one is by using schematic capture to implement the design on the FPGA; the second method is by using hardware description languages such as VHDL [7]. One of the key features of using VHDL is that it can be used to achieve all the goals for documentation, simulation, verification and synthesis of digital design, thus saving a lot of effort and time. In this paper, the implementation of a programmable rotational stepper motor circuit on an FPGA using VHDL is presented.

This paper is organized as follows: in section 2, basics on the work of the stepper motor is presented. Section 3 is interested with VHDL & FPGA and design implementation for the control circuit of the stepper motor, including synthesis of logic circuit. Section 4 included the conclusions.

2. Stepper motor basics:

Stepper motor is well known in position control field applications and is particularly used in robotics, computer peripherals such as printers and servo quality derivers and so on [6]. It can be considered as an electromechanical transducer whose input is a binary voltage waveform and whose output is quantized angular movement [2].

Contrary to conventional motors, stepper motors do not rotate continuously. Instead they rotate a predetermined amount each time; they are excited by a digital signal.

The motor step depends on the number of poles, where many applications require that we be able to take much smaller steps. One way to achieve smaller steps is to add a gear head to the stepper motor, but this adds the expense of a gear head, takes up more space and introduces

the error of backlash in the gear head. A more practical way to achieve smaller steps is to modify the design of the stepper motor. Adding more poles in between the existing poles lets the motor step increments that are proportional to the number of poles:

4 poles = 90^0 / step

8 poles = 45^0 / step

16 poles = 22.5^0 / step

And so on.

The stepper motor used in this paper has 48 poles and therefore steps in 7.5^o increments; its type is CI- 23003, where:

Motor shaft step = One revolution / No. of poles = $360^0/ 48 = 7.5^0$

So, At the low- cost end, the motor shaft rotates through 7.5 degrees per step or 48 steps per revolution.

Its important to notice that regardless of the number of poles; the excitation still requires a 4- phase sequence because of the principles of stepping [6].

The control sequence of a four- phase motor is achieved by activating one phase at a time.

The programmable rotational stepper motor using FPGA would require the following components: the stepper motor for converting digital information into mechanical motion, a phase sequence generator circuit to generate appropriately timed inputs to the stepper motor (this part will be designed here in the Spartan3 FPGA), and driver circuit to provide appropriate voltage and current levels to the stepper motor as shown in figure 1.

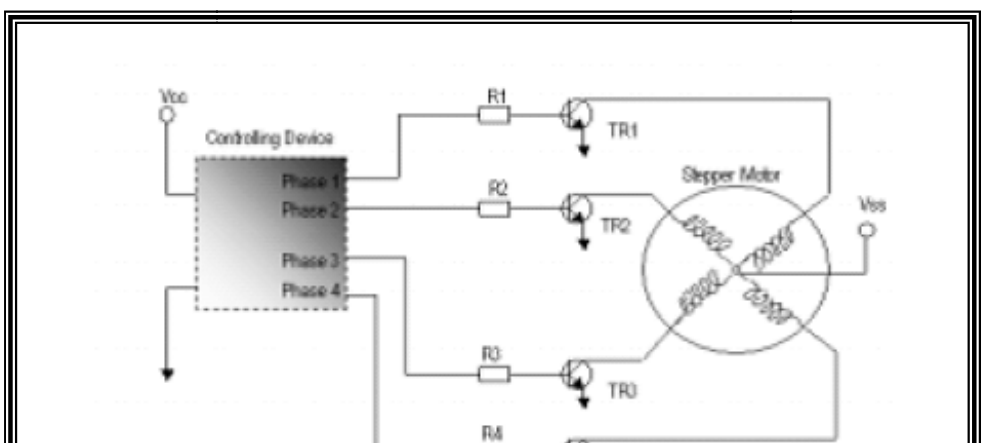


Figure 1: Stepper Motor Controller

R1 to R4 are selected to limit the driving current. Tr1 to Tr4 are power NPN transistors. The four windings have a common connection to the motor supply voltage (Vcc), which equal to 5 volt in the motor under test.

The stepper motor that will be controlled by the FPGA has four sets of coils. When logic level patterns are applied to each set of coils, the motor steps through its angles.

A very common stepping code is given by the following hexadecimal numbers:

A 9 5 6

Each hex digit is equal to four binary bits:

1010 1001 0101 0110

These binary bits represent voltage levels applied to each of the coil driver circuits. The steps are:

1010	5V	0V	5V	0V
1001	5V	0V	0V	5V
0101	0V	5V	0V	5V
0110	0V	5V	5V	0V

If this pattern is send repeatedly, then the motor shaft rotates in clockwise direction.

By reversing the previous pattern, which means:

6 5 9 A

Then, a counterclockwise direction rotation must be achieved.

The speed of the stepper motor shaft depends on how fast the logic level patterns are applied to the four sets of coils. The stepper motor that used in this paper needs 2- 10 m seconds between the binary bits that represent voltage levels which applied to the coil driver circuits. By varying this standard time delay between the binary bits that applied to the coil driver circuits, a various rotating speed can be achieved. The 2 m sec. delay gives high rotating speed; while the 10 m sec. delay make the stepper motor rotate slowly.

3. FPGA Implementation:

As mentioned in the introduction, the Spartan3 FPGA is used to contain the logic design of the stepper motor controller as a pure hardware. Implementing any digital design on the FPGA required completing several design processes, these processes must be executed one by one. The first one is the design entry using hardware description language such as VHDL.

3.1 VHDL and design flow:

Very High Speed IC **H**ardware **D**escription **L**anguage now is the most moderate tool for designing complex digital designs that occupy thousands or million logic gates. Its device independent, and by using it, we can make simulation to design before implementation, or make design simple to change. To develop the design with HDL, we can change the program which presents the hardware design then make re-programmability on the FPGA chips by the bit streams generated from the VHDL program.

VHDL deals with logic gates as the simplest design level: gate, register, chip and system is the most complex level. Designing with VHDL can finally be implemented on two types of integrated circuits: FPGA and ASIC. Figure 2 explains the design flow on the FPGA chips [3][7].

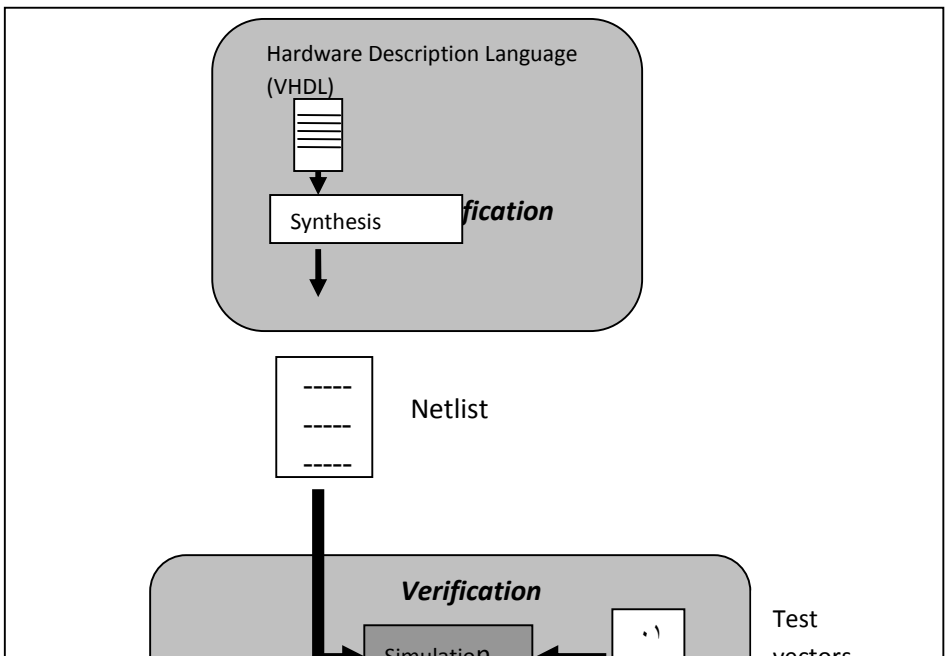


Figure (2) The FPGA design flow

3.2 Spartan 3 FPGA: the target technology

Spartan 3 FPGA is the target technology for the control circuit of the stepper motor. The Digilent Spartan 3 development board provides a powerful, self contained development platform for designs targeting the new Spartan3 FPGA from Xilinx.

The board contain many components that allow the user to develop and evaluate a system centered on the spartan3 FPGA such as: the 8 slide switches (connected to the FPGA inputs), 50 MHz oscillator clock source, JTAG port, 200,000 gate in a thin ball grid array package, and 3*40- pin expansion connectors (four of them are connected to the inputs of the driver circuit of the stepper motor as shown in figure 3), etc…. [8].

The combination of speed and price of the Spartan 3 FPGA means that complete system functions can be implemented in a very cost and space efficient manner.

Figure 3 is shown the designed circuit that represent the FPGA control on the driver circuit of the stepper motor and the stepper motor.

Figure 3: The stepper motor controlled by Digilent Spartan 3 development board as

The 50 MHz provided by the development board, is divided through the VHDL program to obtain the required time delay between the hexadecimal digits of the pattern. This pattern must be send to the driver circuit of the motor, and as mentioned in section 2, this delay time is between 2- 10 m seconds.

The design is configured in way that we can make useful from five of the 8 slide switches that provided by the board to make the necessary control process to the motion of the stepper motor, table 1 is shown the functions of the switches 1 to 3,

while in table 2, the functions of the switches 4 to 5 besides 2 is presented:

Table 1: the functions of the switches 1 to 3 in the development board

Switch number	In the off position '0'	In the on position '1'
1	Stops the stepper motor	The stepper motor can rotate
2	The rotation is in clockwise direction	The rotation is in counter- clockwise direction
3	The stepper motor rotate in high speed	The rotation is in low speed

Table 2: the functions of the switches 4 to 5 in the development board

The logic of the switches number 5 & 4	The logic of switch number 2	The type of rotation
00	0	Continuous clockwise direction rotation
01	0	45^0 in clockwise direction of rotation
10	0	90^0 in clockwise direction of rotation
11	0	135^0 in clockwise direction of rotation
00	1	Continuous counterclockwise direction rotation
01	1	45^0 in counterclockwise direction of rotation

10	1	90^0 in counterclockwise direction of rotation
11	1	135^0 in counterclockwise direction of rotation

3.3 Synthesis of logic circuit

The ISE (Integrated Software Environment) represent the compatible software package to deal with the Spartan3 starter kit. This package is supported with a synthesis tool called XST (Xilinx Synthesis Technology) which make synthesis to the design before implementation. The implementation process represents the following steps: translate, map, place & route, and lastly generating the programming file.

From the synthesis report, some useful information on the chip resources is obtained. This information is shown in table 3:

Table 3: the device utilization summary

Device utilization summary

Selected device: 3S200ft256-5

Number of slices: 60 out of 1920 3%

Number of slice flip flops:	38 out of	3840	0%
Number of 4 input LUTs:	100 out of	3840	2%
Number of bonded IOBs:	8 out of	173	4%
Number of GCLKs:	1 out of	8	12%

The maximum critical path for the design is equal to 7.576 ns (5.731 ns logic and 1.845 ns route).

Figure 4 show the control circuit of the stepper motor synthesized with XST as a logic design circuit configured on the Spartan 3 FPGA.

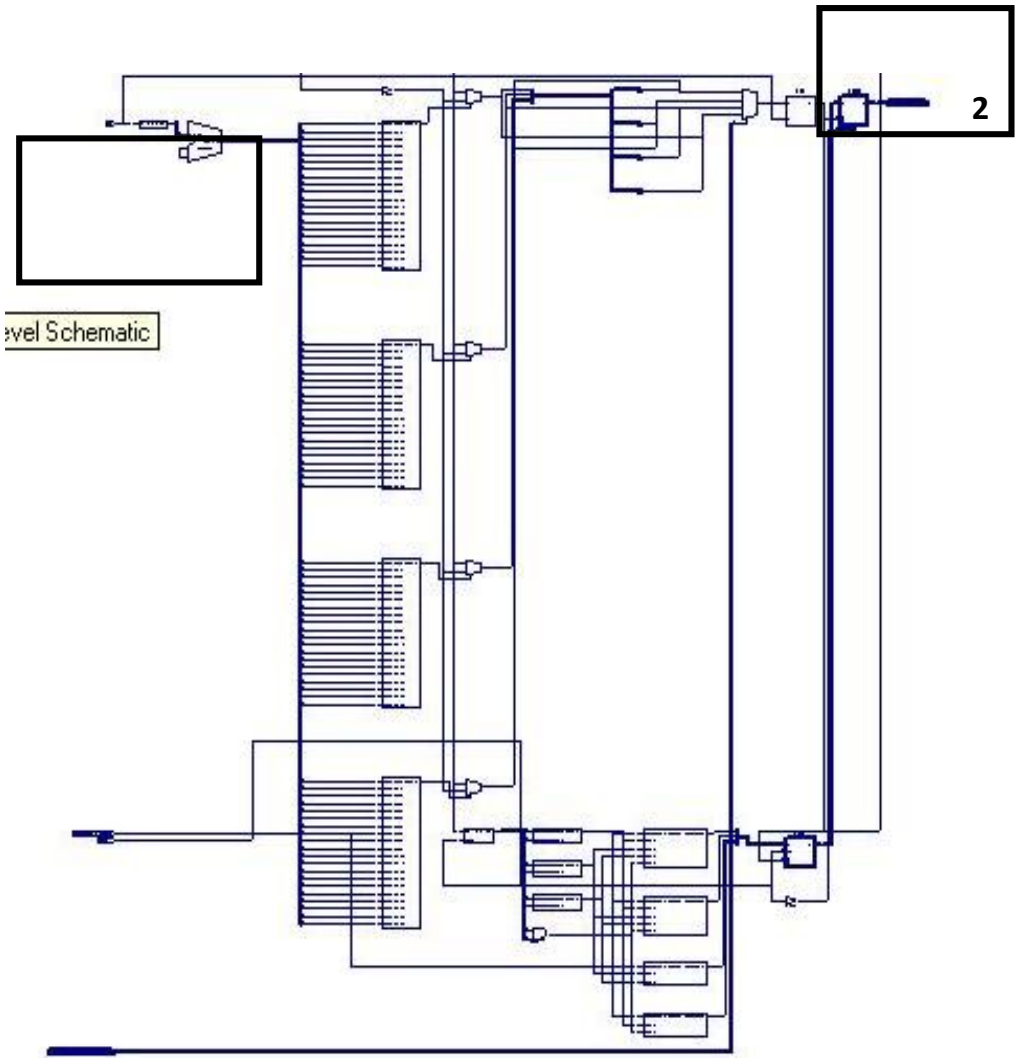
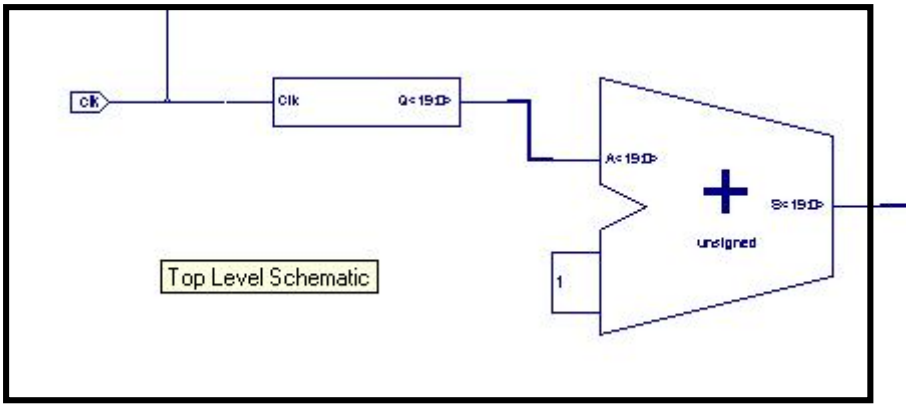
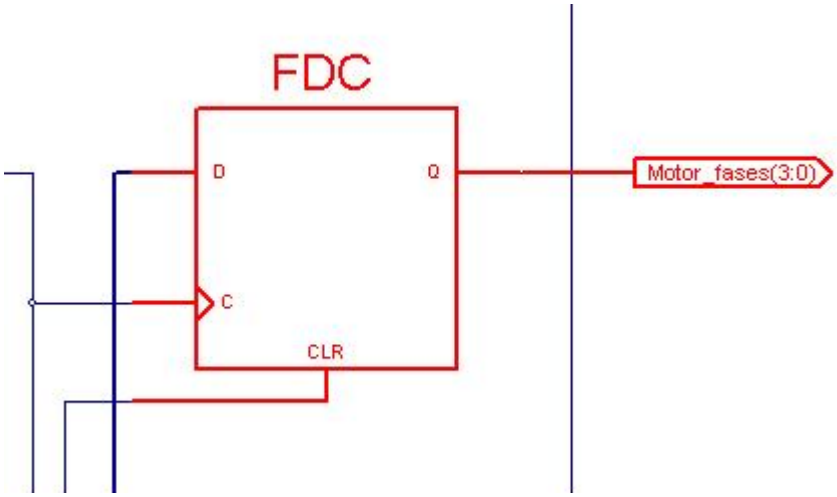


Figure 4-A: Top level schematic for the control circuit of the stepper motor synthesized with XST.

3




FDC
D
Q
C
CLR
Motor_fases(3:0)


direction_of_rotation
on_off

4. Conclusions

In this research, Xilinx Spartan 3 FPGA is used to configure a digital design on it to make a control digital circuit of stepper motor motion. This type of design comparing with the methods mentioned in the introduction, has an important issue represented by disconnecting the development board from the computer after configuring the FPGA that lie in it, this is called an embedded system. Also the advantage of using FPGA is that the designer can make modifications to the design easily and quickly. The resources used from the FPGA in this design was not exceeded the 12%.

5. References

[1]: Mike J. Johnson and Guru Subramanyam ***"A Parallel Port Interface Circuit for Computer Control Applications Involving Multiple Stepper Motors"*** University of Northern Iowa, IEEE, Circuits and Systems, Vol 2, pp 889- 892, 1997.

[2]: C. S. Chen ***"Microcomputer Speed Control of Stepper Motor "*** University of Akron, Ohio, IEEE, Control Systems Magazine, Vol 2, No.1, pp17- 20, 1982.

[3]: Karen Parnell and Nick Mehta ***" Programmable Logic Design Quick Start Handbook "*** Xilinx, Inc. 2004, ***www.xilinx.com.***

[4]: ***Digital Logic Lab KL-300 Experiment Manual***, King Instrument co., Ltd., pp. 387.

[5]: Renato A. Krohling, Yuchao Zhou, and Andy M. Tyrrell ***"Evolving FPGA – based robot controllers using an evolutionary algorithm"*** University of York, UK, 2002, available at: rk8@ohm.york.ac.uk

[6]: ***Stepper Motor CI- 23003 Experiments Manual***, King Instrument Inc.

[7]: Ahmed Al- Sulaifanie, and Yahya Al-Dabbagh, ***"Single Chip DWT- IDWT Processor Design with VHDL"***. Al- Rafidain Engineering Journal, ISSN 1813- 0526, Vol. 14, pp. 59, 2006.

[8]: Xilinx Spartan 3 FPGA Introduction Webpage, Xilinx

http://www.xilinx.com/xlnx/xil_prodcat_landingpage.jsp?title=spartan-3

The work was carried out at the college of Engg. University of Mosul