International Journal of Review in Electronics & Communication Engineering (IJRECE)
Volume 2 - Issue 1 February 2014

CNC MACHINE CONTROLLER USING FPGA

Dipali B. Bhatt [1], Hemant R. More[2]
[1]Asst. Prof. Dept. of Electronics and communication Vadodara Institute Of Engineering And Research, Vadodara, Gujarat,
[2]M.Tech (Digital System), College of Engineering, Pune,
Emails: [1]dipali.bhat18@gmail.com, [2]hemantmore9786@gmail.com

***Abstract-* In many industries use of CNC (Computer Numerated Control) machine goes increasing day by day. So the motion of these machines should be controlled accurately. Motion control of CNC machine is becoming challenging stuff. To carry out the motion control of CNC machine, this work introduces an approach to implement 3-axis motion controller based on Field Programmable Gate Array (FPGA). Since the straight-line motion is a fundamental motion of CNC machine, this project introduces a linear interpolation method to do approximate straight-line motion within any 3-axis space. As Interpolation calculation of hardware interpolation is implemented by hardware logic circuit such as FPGA in the controller, therefore this method can avoid a large amount of complex mathematical calculation, which hints that this controller has high real-time performance**

Keywords: CNC, FPGA, Motion Control, IC design, VLSI.

I. INRODUCTION

Motion control system represents a major subsystem that is responsible for actuation of all sorts of devices in industrial automation such as robotics, wire bonding machines, CNC machining, and high speed assembly machines. In these tasks, higher speed with the same error performance means higher productivity. Over the past years, there have been proposed many different control algorithms. Most of these algorithms can be considered as special cases of the class of computed torque controllers [1]. Computed torque control is a special application of feedback linearization of nonlinear systems, which has gained popularity in modern systems. One way to classify robot control schemes is to classify them as computed torque like or non-computed torque like. As specified in [1], computed torque-like controls appear in robust control [2], adaptive control [3], learning control [4], hybrid control [5], and impedance control [6] etc. The non-computed torque like control appears in PID feedback [7] [8], neural network and fuzzy logic controls [9] etc.

Success of a motion control systems depends not only on the control algorithm but also on the control hardware structure. The control hardware structures used by existing controllers include i) purely relying on a motion control DSP [10]; ii) utilizing both a DSP and motion control ASIC (Application Specific Integrated Circuit) [11]; iii) using general-purpose DSP and external control resources such as a FPGA [12]; iv) using a modular structure [13]; and v) using multiple DSPs [14]. A combination of a DSP and an FPGA is the most popular structure at the present time, where FPGA implements I/O functions according to specific requirements. Increase in productivity i.e. throughput, usually not higher accuracy, is the central driving force behind advances in the design of motion control hardware. What designers strive for is higher speed with the same error performance, providing higher productivity – all at low cost. It has been theoretically demonstrated that many advanced control algorithms (i.e. model-based controls) exhibit far superior performance over PID especially for high-speed systems with nonlinear dynamic characteristics. The relatively low update rate leads to degraded performance especially at high robot speeds. Further, due to the limited computational power of existing low cost control architectures, many advanced control algorithms are impractical since the complexity of these algorithms result in a low sampling frequency, leading poor control performance. A multi-DSP system may be a solution to realize a complex model based control algorithm, but its structure incurs limitation with regard to the data exchange between different DSPs [15]. New controller architecture is therefore required, which can implement advanced controls resulting in substantially improved performance especially during high speed motions.

FPGA have historically been used to create custom hardware logic circuit (such as I/O functions) that could not be performed by microcontroller or DSP. In recent years, FPGAs have been greatly improved due to enhancement of submicron process technology. Some researchers incorporated control algorithms into FPGA chip to enhance the performance of servo control systems. The system developed by Takahashi and Goetz [16] could run a current control algorithm with a Xilinx FPGA to increase the bandwidth of the current loop control. Tzou and Kuo [17] performed the vector and velocity controls of a PMAC servo motor by using FPGA technology successfully. Other works on FPGA based motion controls include Paramasivam [18], Bielewicz [19], and Dubey [20] PID control was used as the control algorithm in these works.

Main aim of this work is to implement linear interpolation technique to control the 3-axes of CNC machine.

1. To receive X, Y and Z coordinates from User Interface using UART (Universal Asynchronous Receiver Transmitter) in FPGA.
2. To store the received coordinate in FIFO.
3. To read the coordinate one by one and process it.
4. According to proportion in the coordinate generate the step and direction signal for X, Y and Z axis stepper motors.

II. IMPLEMENTAION

What is a CNC?

A CNC computer numerated is controller that drives a cutting machine. The most popular cutting machine is Milling machine or Laser cutting machine.

Milling machine

A simple milling machine has 3 axes, while more complex machines have 4 axes or more. Here's a simple 3-axis milling machine. Each axis is controlled manually with a handle.

On a CNC machine, each axis is moved by a motor and a controller to precisely control the motors rotations. For example, let's say each turn of a motor makes an axis move by 1mm.To move by 3mm, you need 3 turns.

Motors and loop control

Two types of motor are commonly used: stepper motors and Continuous Current (CC) motors. Stepper motors typically use open loop motion control, while CC motors use closed loop motion control. Stepper motors can be seen as motors controlled by electrical pulses. Common stepper motors require 200 pulses per rotation, so by sending 600 pulses, the motor should make exactly 3 turns. This is called "open loop motion control" because there is no mechanism that checks that the motor turns as expected. Instead the user relies on the known characteristics of the machine/motors combination. The main drawback of stepper motors is the possibility of "missing steps" if driven too fast, so stepper motors are always used at conservative speeds (i.e. with enough margin to avoid missing steps). CC motors don't use pulses, but continuous current, which is simpler than pulses, but getting exact motion, is more complex. To get 3 turns, the controller needs to send some current to the motor, and needs a way to monitor the rotation, so that it stops sending current once the 3 turns are achieved. This is called "closed loop control" because of the mechanism that monitors the motor rotation. This allows for faster operation and efficiency, but is more complex and slightly less precise than stepper motors.

A. Stepper Motor Control

a. Parallel interface

Here's a typical CNC setup.

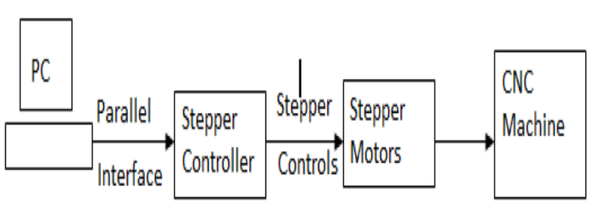

Figure 1: Typical CNC setup

A PC uses its parallel interface to connect to the stepper controller, which in turn drives the stepper motors (sometimes simply called "steppers") of the CNC mill.

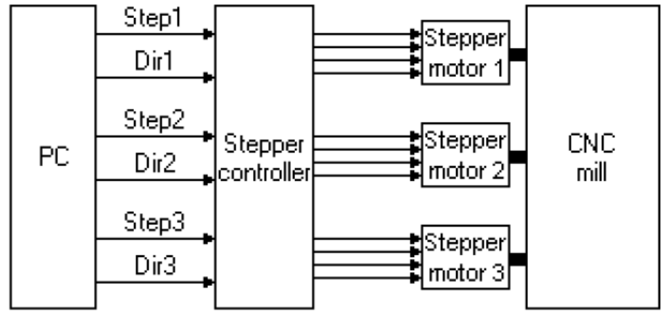

Figure 2: Parallel interface to connect to the stepper controller

Assuming our mill has three stepper motors, let's zoom in to see more details. The PC sends two signals per stepper motor (Step & Direction).Then the stepper controller generates the stepper outputs (4 to 8 wires per stepper, depending on their types).

The PC must be dedicated to the CNC task because the timing of the Step/Direction signals must be as precise as possible. The PC must either run in DOS mode, or in Windows/Linux but with all other tasks shut down.

B. Motion controller

A motion controller has the following advantages:

1. Each axis has a dedicated motion unit. That means that unlimited number of axes can be driven simultaneously. If your mill had 10 axes, all running together, that'd be no problem.
2. The timing resolution of the stepper pulses is much higher than with a software solution (10 to 100 times better).
3. The PC doesn't need to be dedicated, as the hardware motion units all work in parallel with the PC.

A simple motion controller is built using a FPGA board, which has a RS232 serial interface so is easy to interface to PC. The FPGA is placed between the PC and the Stepper controller.

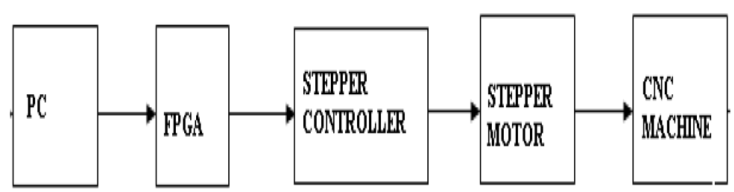

Figure 3: General block diagram of CNC machine motion controller

C. FPGA design

Here's the FPGA motion controller block diagram:

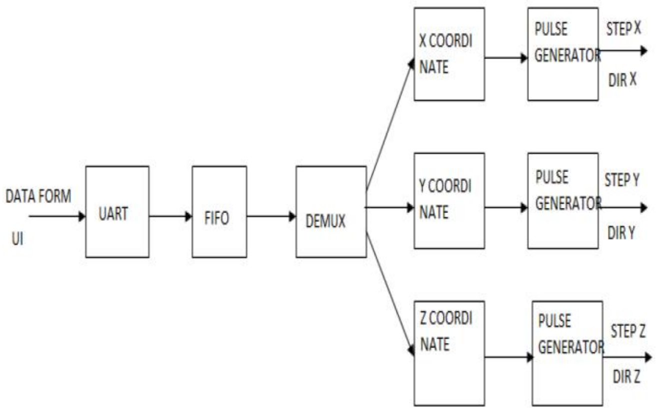

Figure 4: Block diagram of motion controller

The coordinates from user interface are received through RS232 cable using UART (Universal Asynchronous Receiver Transmitter).Then received coordinates are stored in FIFO. Then these coordinates are demultiplexed into X, Y&Z coordinates and feed to the motion controller unit. Motion controller generates necessary step and direction signal for respective motors according to the coordinates.

D. UART (Universal Asynchronous Receiver Transmitter) Operation

The data frame used contains total ten bits, one start and stop bit and remaining eight bits are data bits. The data is transmitted serially LSB first at a given bit rate (baud rate) known by the transmitter and receiver. Since the transmitter can start sending this data at any time, the receiver needs a method of identifying when the first (LSB) is being sent. This is achieved by the transmitter sending an active Low start signal for the duration of one bit.

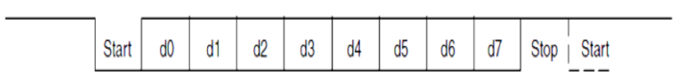

Figure.5: UART data frame

The receiver uses the falling edge of the start bit to begin an internal timing circuit. This timing is then used to sample the value of the serial input at a point approximately at the mid position of each data bit. This is where the data should be most stable. After the last data bit (MSB) has been sampled, the receiver checks to see if the transmitted stop bit (High) is the value expected to help confirm correct operation.

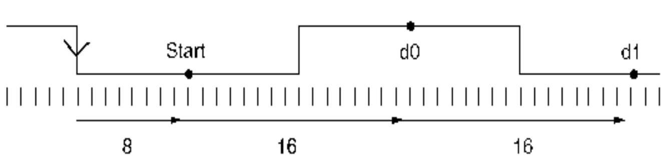

Figure.6: UART sampling data at most stable value

Since the receiver resynchronizes (starts the internal timing circuit) to the falling edge of each start bit, the transmitter and receiver timing must be the same to an accuracy of only one half of a bit period every 10-bit periods. This 5-percent tolerance is usually easy to achieve in digital systems.

State machine for UART receiver:

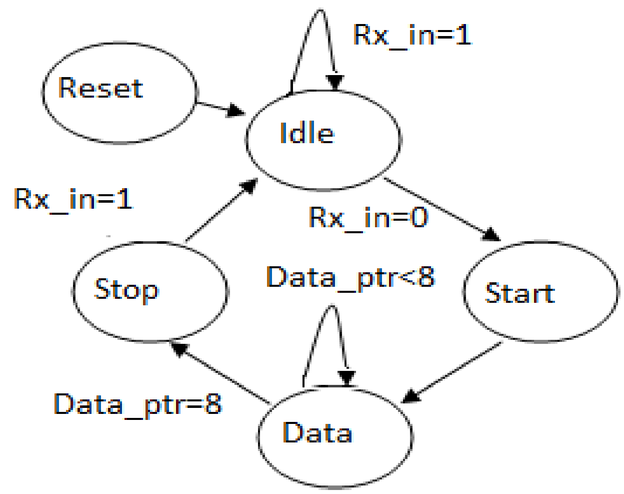

Figure.7: State machine of UART

Motion Controller State Machine:

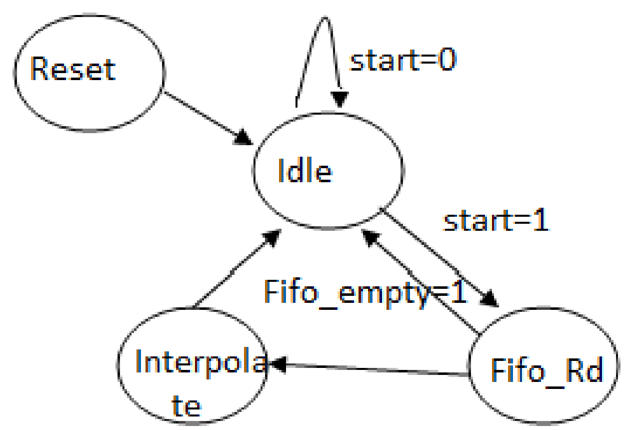

Figure.8: State machine of motion controller

III. RESULT

To verify the 3-D linear interpolation, a test bench is written to make verification. In simulation, the start point is (0, 0, 0), the end point is (10, 2, 0). As X is main axis so maximum frequency is allotted to X-axis and according to proportion in X and Y coordinate frequency for Y-axis is generated. As X coordinate is 10 and Y coordinate 2 then no. of pulses sent to X and Y motor should be 10 and 2 respectively and it is verified with simulation result. Figure 9 shows the simulation wave of the 3-D linear interpolation. Also the system is also implemented on Altera Cyclone III FPGA and results are verified on DSO (Digital Spectrum Oscilloscope)

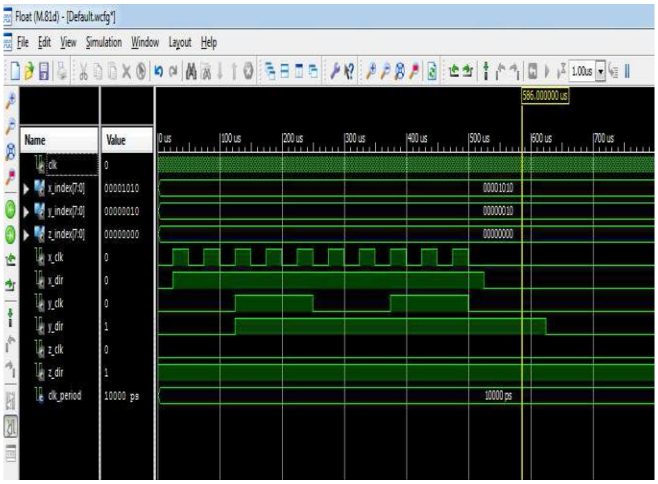

Figure 9: Simulation wave of the 3-D linear interpolation

IV. SUMMARY AND CONCLUSIONS

This work introduces the implementation of linear interpolation module in 3-axis FPGA based motion controller. Through introducing the algorithm principle/hardware structure; the implementation of 2-axis/3-axis linear interpolation module is discussed. Implementation algorithm and concurrent processing hardware make the module have good real-time performance. The approximating precision is excellent, however the module cost few hardware resource, the significant characteristics indicate that the FPGA-based linear interpolation module has excellent performance and can be used for practical motion control. The simulation and implementation on FPGA of straight-line motion within 3D space verifies the effectiveness of the proposed approach.

This system can work efficiently in the following areas

1. Computer Numerated Control such as
    a. Drilling Machine
    b. Milling Machine
    c. Laser Cutting Machine
    d. Embroidery machine
2. In Robotic Automation to perform the task repetitively with accuracy
3. In medical instruments

REFERENCES

[1] F. L. Lewis, M. D. Darren, and T. A. Chaouki, Robot Manipulator Control: Theory and Practice, Second edition, Marcel Dekker, Inc., New York, Basel, 2004.

[2] M. W. Spong, J. S. Thorp, and J. M. Kleinwaks, "Robust microprocessor control of robot manipulators", Automatica, Vol. 23, pp. 373-379, 1987.

[3] J. J. E. Slotine, and W. Li, "On the adaptive control of robot manipulators", Int. J. Robotics Research, Volume. 6, no. 3, pp. 49-59, 1987.

[4] S. Arimoto, S. Kawamura, and F. Miyazaki, "Bettering operation of robotics by learning", J. of Robotics System, No. 1, pp. 123-140, 1984. [5] M. H. Raibert, and J. J. Craig, "Hybrid position/force control of manipulators", JDyn. Syst., Meas., and Contr., vol. 102, June 1981.

[5] N. Hogan, "Impedance control: An approach to manipulation", J. Dyn. Syst., Meas., and Control, vol. 107, pp. 1-7, March 1985.

[6] D. M. Dawson, "Uncertainties in the control of robot manipulators", School of Electrical Engineering, Georgia Institute of Technology, Mar. 1990.

[7] S. Arimoto, Control Theory of Non-Linear Mechanical Systems: A Passivity-Based and Circuit-Theoretic Approach, Oxford: Clarendon Press, 1996.

[8] W. T. Miller, R. S. Sutton, and P. J. Werbos, Neural Networks for Control, Cambridge: MIT Press, 1991.

[9] M. D. Murphy, and F. G. Turnbull, "Power Electronic Control of AC Motors", Oxford, England, Pergamon Press plc, 1988.

[10] L. Debowski, and M. T. Hartman, "A new flexible architecture of digital control systems based on DSP and complex CPLD technology for power conversion application", Proc. PCIM 2000 Europe Intelligent Motion, Nurnberg, Germany, 2000.

[11] D. D. Bester, J. A. Du Toit, and J. H. R. Enslin, "High Performance DSP/FPGA Controller for Implementations of Computational Intensive Applications", Proc. IEEE-ISIE, 1998.

[12] P. Malapelle, and G. Torri, "A New Modular Programmable High Speed Digital Control for Large Drives", Proc. IEEE-IECON, 1994.

[13] R. W. Doncker, and V. R. Stefanovic, "Microprocessor Control of Motor Drives and Power Converters", IEEE Industry Applications Society, Toronto, Canada, 1993.

[14] C. U. Karipidis, "A versatile DSP/FPGA structure optimized for rapid prototyping and digital real-time simulation of power electrnic and electrical drive systems", PhD thesis, Institute of power electronics and electrical drives of the Aachen University of Technology, Germany, 2001.

[15] T. Takahashi, and J. Goetz, "Implementation of complete AC servo control in a low cost FPGA and subsequent ASSP conversion", Applied Power Electronics Conference, Volume 1, pp: 565-570, 2004.

[16] Y. Y. Tzou, and T. S. Kuo, "Design and implementation of an FPGAbased motor control IC for permanent magnet AC servo motors", 23rd International Conference on Industrial Electronics, Control and Instrumentation, Volume 2, pp:943 – 947, Nov. 1997.

[17] S. Paramasivam, R. Arumugam, and S. Vijayan, etc, "Ingenious digital controller for switched reluctance motor using Verilog (HDL)", Conference on Convergent Technologies for Asia-Pacific Region, Volume 3, pp:993 – 997, Oct. 2003.

[18] Z. Bielewicz, L. Debowski, and E. Lowiec, "A DSP and FPGA based integrated controller development solutions for high performance electric drives", Proceedings of the IEEE International Symposium on Industrial Electronics, Volume 2, pp: 679 – 684, June, 1996.

[19] R. Dubey, P. Agarwal, and M K. Vasantha, "FPGA based PMAC motor control for system-on-chip applications", Proceedings of International Conference on Power Electronics Systems and Applications, pp:194 – 200,Nov. 2004

[20] Weihai Chen, Zhaojin Wen, Zhiyue Xu& Jingmeng Liu "Implementation of 3-axis Linear Interpolation in a FPGA-based 4-axis Motion Controller"

[21] Ming-Tzong, Lin, Hong-TzongYau, Hao-Wei Nien and Meng Shiun Tsai "FPGA-Based Motion Controller with Real-time Look-ahead Function" Proceedings of the 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics July 2 - 5, 2008, Xi'an, China.

[22] Hong-Tzong Yau and Ming-Tzong Lin and Yao-Ter Chan and Kuo-Chin Yuan "Design and Implementation of Real-time NURBS Interpolator using a FPGA-Based Motion Controller" Proceedings of the 2005 IEEE International Conference on Mechatronics July 10-12, 2005, Taipei, Taiwan.