The Journal of CPRI,
Vol. 9, No. 2, June 2013 pp. 271–278

Design and Implementation of Micro-Stepping Drive of Stepper Motor for Solar Array Drive Assembly using FPGA

Sachin Angadi*, Satyanarayana Kumari B** and Saikumar H V*

This paper describes the design of a control system for a micro step driving method of stepper motor using VHDL. The control system so designed is employed in the application of Solar Array Drive Assembly (SADA) in terrestrial solar plant and in satellites. This solution uses a digital controller with Sine-Cosine variation to control the currents of the two phases of the motor, making them change according to the orthogonal sinusoidal scalar form curve, fulfilling the micro step driving of stepper motor and improving its position control precision by hundreds of times. At the same time, this driving method solves problems such as large noise, resonance and non-even step angle and so on. The work presented in this paper implements the control system in Xilinx ISE.

Keywords: *Micro-step, Sine-cosine variation, Satellites, Terrestrial solar plant, VHDL and Xilinx ISE.*

1.0 INTRODUCTION

Spacecraft goes through intermediate orbits called the transfer orbit and drift orbit before it is placed into the geo stationary orbit. The operational mission lifetime of any satellite is 12 years. Power requirement for such a complex system for long time is a serious issue and compulsorily renewable energy is desired, the only feasible energy that is available in space in abundance is solar energy. Photovoltaic solar array systems are the most common method for providing spacecraft power generation. Solar array technologies and their system configurations changed dramatically over the years as more aggressive and demanding requirements were imposed. Also the weight is constraint on the spacecraft; hence there is need for the design of low weight solar array drive to drive the solar panels to track the power from the sun [1].

Permanent magnet stepper motors (PMSMs) are used in positioning applications due to their durability, high efficiency, and power density, as well as their high torque to inertia ratio and absence of rotor winding [2]. Stepper motors are a good choice for open-loop applications because the position error is not cumulative with rotation as long as synchronism is maintained [3]. Authors in [4] brief about the phenomenon that is observed when a closed loop commutation delay is present in stepper motor control. In the presence of this delay, the feedback linearization is not exact and when algorithms are employed for correction the stability of the system reduces considerably. So the fact that the stepper motor gives satisfactory results in open loop mode is exploited by authors in [5, 6]. In [5] the authors propose novel controller for remarkable performance that uses pulse width modulation technique on a compact hardware, the method demands high switching devices and high operating frequencies. In [6] authors implement the closed loop control of drive using PI controller but the stepper motor is operated in open loop for better stability. Owing to the requirements for

*The National Institute of Engineering, Mysore - 570008, Karnataka, India.
**Indian Space Research Organization, Bangalore - 560 017, Karnataka, India. E-mail: sachin.36.bvb@gmail.com.

SADA [7] best defines the solution. In our paper we aim at designing the micro-stepping stepper motor drive with micro-stepping ratio of upto 128. The methodology proposed here uses Read Only Memory (ROM) as Look-up Table (LUT) to store the values. Further LUT produces the values of sine and cosine for the windings of the stepper motor in digital form which is then converted to analog signal using digital to analog converter (DAC). This proposed methodology reduces the switching duty up to 100 times as compared to PWM technique used in [4]. Also a single ROM is utilised to generate Sine wave, the Cosine wave being derived from the Sine is an added advantage. This paper aims to explain, the design of micro stepping stepper motor drive using VHDL for its application in SADA in IRS satellite carried out in ISRO satellite centre (ISAC). Also brings out the comparison between the proposed control methodology and the conventional full step control methodology of stepper motor drive.

The paper is organised as follows: To start with the introduction and literature review in section I. The concept of micro-stepping with its advantages and disadvantages are discussed in section II. The Problem statement is defined in section III which includes the requirement specifications of the project. Section IV is reserved to the study of the design approach which includes the discussion of the proposed methodology in terms of the functional block and sub-blocks. Section V deals with the discussion of results followed by conclusion in Section VI.

2.0 CONCEPTUAL BACKGROUND

In high precision stepper motor applications, it is necessary to use motors with small steps whose size is imposed by the required resolution. Another alternative is the technique of micro stepping; Micro-stepping is a way of moving the stator flux of stepper motor more smoothly than in full or half-step drive modes. This results in less vibration and makes noiseless stepping possible down to less frequency. It also makes smaller step angles and better positioning possible, where the motor step size is further reduced by means of control.

Micro-stepping can be used to solve noise and resonance problems, and to increase step accuracy and resolution. If the system damping is low there is an obvious risk of losing steps or generating noise when the motor is operated at or around the resonance frequency. Depending on motor type, total inertia, and damping; this problems can also appear at or close to integer multiples and fractions of F0, that is: F0⁄4, F0⁄3, F0⁄2, 2F0, 3F0, 4F0. Normally the frequencies closest to F0 give the most problems. When a non-micro stepping driver is used, the main cause of these resonances is that the stator flux is moved in a discontinuous way, 90 or 45 (full-step and half-step mode) electrical degrees at a time. This causes a pulsing energy flow to the rotor. The pulsations excite the resonance. The energy transferred to the rotor, when a single step is taken, is in the worst case (no load friction) is given by Eqn. 1.

$$\text{Energy} = \left(\frac{4 * T_h}{n}\right)\left(1 - \cos f_e\right) \quad \text{....(1)}$$

wheren is number of full steps per revolution, T_h is the holding torque of the motor and f_e is electrical step angle, 90º for full-step, 45º for half-step. This shows that using half-steps instead of full-steps reduce the excitation energy to approximately 29% of the full-step energy. If we move to micro stepping 1/32 of full-step mode only 0.1% of the full-step energy remains. It appears that, by using micro stepping techniques, this excitation energy can be lowered to such a low level that all resonances are fully eliminated. Unfortunately this is only true for an ideal stepper motor. In reality there are also other sources that excite the system resonances. Never the less, using micro stepping will improve the movement in almost all applications and in many cases micro stepping will alone give a sufficient reduction of the noise and vibrations to satisfy the application. Micro-stepping can also be used to increase stepper motor resolution. If this principle is employed the motor can execute up to 25,600 steps/revolution (i.e., 128 micro steps, 200 full steps) for a motor with typical step angle of 1.8º.

In conventional methods stepper motor windings are energized in sequence in which case the stator

flux rotates through 90º for full step or 45º for half step, but in micro step mode both the windings of the stepper motor are energized simultaneously to align the stator flux in position with the resultant magnetic field at an intermediate angles depending upon the micro stepping ratio required. The look up table for winding currents of motor for micro stepping ratio of 8 is listed below in Table 1.

TABLE 1

Micro step No.	% Phase-A Amp	% Phase-B Amp
0	0	100
1	19.51	98.08
2	38.27	92.39
3	55.56	83.15
4	70.71	70.71
5	83.15	55.56
6	92.39	38.27
7	98.08	19.51
8	100	0

Likewise the table can be stored in ROM for various micro-stepping ratios for the operation of stepper motor. Also the sequence in which the currents are applied to motor windings is very important. The sequence in which sine(S)/ cosine(C) generators are to be applied to the motor windings for micro stepping direction control in all four quadrants is as shown in Table 2 for Counter clockwise (CCW) and Clockwise (CW) direction.

TABLE 2

Direction of rotation	Quadrant for this direction	Quadrant as per standard convention	Motor winding			
			A+	B-	C+	D-
CCW	I	I	S			C
CCW	II	II	S		C	
CCW	III	III		S	C	
CCW	IV	IV		S		C
CW	I	II	S		C	
CW	II	I	S			C
CW	III	IV		S		C
CW	IV	III		S	C	

Considering the maximum current in both the windings to be I_r the currents in micro-stepping mode will be $I_r \cos\theta$ and $I_r \sin\theta$ in individual stator windings, their resultant is I_r itself as shown in Eqn. 2.

$$I_r = \sqrt{\left(\left(I_r \cos\theta\right)^2 + \left(I_r \sin\theta\right)^2\right)} \quad \text{....(2)}$$

Consequently the stepper motor develops the same torque as developed under full step methodology. There is no reduction in motor torque on account of micro stepping [8].

In theory, micro stepping is quite simple, and theoretically, the technique solves all resonance, vibration and noise problems in a stepper motor system. In reality, a lot of different phenomena arise which set limits for the system performance. Some are related to the driver and others to the digital controller that is used. The operating frequency is one such problem. However with advancement in the technology high speed switching drivers are available in the form of integrated chip. Also the Field Programmable Gate Array (FPGA) controllers designed using VHDL increase the speed of controller considerably up to few MHz and further more it is compatible with ASICs and mass production methods [6].

3.0 PROBLEM FORMULATION

The proposed control methodology deals with dividing normal step angle of the Stepper motor with the following features

- Facilitate various micro steps viscerally 2 or 4 or 8 or 16 or 32 or 64 or 128 as per the requirement.
- The control system should allow different micro stepping rate for the drive which decides the speed of the motor in (steps/sec).
- Bi-directional control in clockwise and counter clockwise rotation.
- Orthogonal sine-cosine current Variation in windings of the stepper motor in order to maintain the torque constant.

- Least possible delay for the control system.
- Maximum possible frequency.
- Reduced switching losses.
- Least hardware possible.

4.0 DESIGN APPROACH

To control the stepper motor, the driver can apply an open loop control algorithm or closed loop algorithm. In an open loop stepper motor driver, in order to perform position control, the current in the phase of the motor must be changed gradually. Therefore, the performance of the current controller in an open loop motor driver is important. The approach adapted here is to have Read Only Memory (ROM) of Sine and Cosine wave for windings A and B respectively to facilitate various micro stepping ratios. Here in this approach FPGA is used as digital controller, which provides better control and high speed operation. Controller designed in this paper will receive the inputs from MA31750 processor. This processor supports military standard specifications for communication between the earth stations and the satellite in space using telecommand (to send data from earth stations to satellite) and Telemetry (to receive data from satellite to earth stations). The transfer of data using telemetry and telecommand is governed by the read and write cycle of the MA31750 processor. Figure 1 represents Black box approach of the entire control system with all the inputs and the outputs.

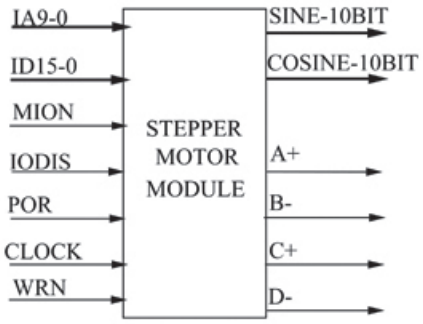

FIG. 1 INPUTS OUTPUTS OF MICRO-STEPPING STEPPER MOTOR DRIVE.

The inputs being IA-10 bit Address bus (Gives address of the Data that has to be written to the Output port), ID-16 bit Data bus (Data that is written to the output port), MION-Memory I/O active low (is low for writing data from memory to Output port), IODIS-input output disables, POR-Power on reset, CLK-Clock (Decides the operating frequency) and WRN(write enable pin). The output being Sine and Cosine-10 bit-binary data, input to DAC.A+, B-, C+ and D- are the control signals for the driver circuits to facilitate proper channelling of the signals to the windings of the stepper motor.

4.1 Digital Controller

The Figure 2 is representation of the block level implementation of the digital controller to be simulated in VHDL. As discussed earlier the digital controller requires the interface between itself and MA31750 processor.

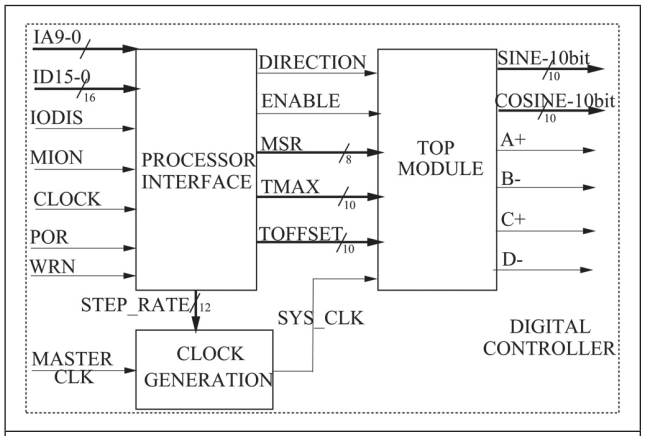

FIG. 2 OVERALL BLOCK DIAGRAM OF DIGITAL CONTROLLER.

The outputs of the processor interface will act as the inputs to the Digital controller which in this case are the inputs like direction bit-DIR (if this bit is 0 then motor will rotate in counter clockwise direction and rotate in clockwise direction otherwise), Enable bit (For controller to perform its function this bit should be high, Controller is disabled if this bit is low), MSR bit (this bit defines Micro stepping Ratio and can take value 2 or 4 or 8 or 16 or 32 or 64 or 128 as per requirement), Tmax most of the times is set to maximum value and Toffset value is changed suitably to overcome detent torque of the stepper motor. These available inputs are then fed to the Top Module of the digital controller as shown in

the Figure 2. This block considers all the inputs to give the required outputs which are the Sine and Cosine wave in binary form and also the control signals required to enable the stepper motor drivers. Another sub block is the Clock Generation which generates the system clock responsible for synchronizing the operations of the controller and also determines the speed of operation of the controller. Each of the modules listed will be discussed in detail in what follows.

4.2 Processor Interface

Figure 3 represents the processor interface. It consists of the 3–8 Decoder and eight 16 bit data latches. The output of the decoder act as chip select for the 16-bit data latch. The system is provided with synchronous clock, asynchronous power on reset and asynchronous chip select. Depending on the status of MION, IODIS and IA9-IA0 the decoder outputs one chip select pin (CS0-CS7). The chip select pin further act as the enable pin for the 16bit data latch. The latch designed is the D-Flip Flop with synchronous clock, asynchronous power on reset and asynchronous enable (CS0-CS7) pin. Monitoring the clock, power on reset and chip select pin suitable data ID15-ID0 is output on the appropriate output port. PORTA outputs Step_rate (12-bit), direction bit and enable bit, PORTB gives the Micro stepping ratio (MSR), PORTC gives the maximum torque value and PORTD gives offset value. PORTE-PORTH are Miscellaneous. The decoder enable pin is suitably designed to select the addresses from 100-107H only. Any eight addresses can be chosen as per requirement.

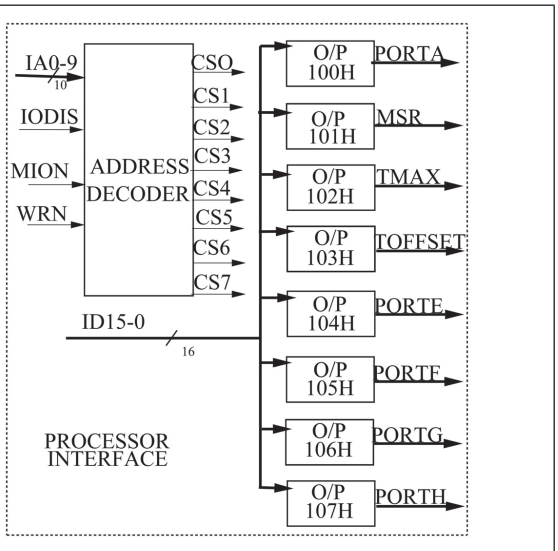

FIG. 3 FUNCTIONAL BLOCK DIAGRAM OF PROCESSOR INTERFACE.

4.3 Top Module

The Figure 4 is detailed representation of Top Module. The inputs to this block are MSR, Tmax, Toffset, Sys_clk, POR, Enable and Direction. The expected outputs of the block are drive enable signals that are A+, B-, C+ and D-. The sine and cosine wave in digital form are Sine-10 bit and Cosine-10 bit. The block contains various sub-blocks for its functional requirements namely Steps Counter, Angle to Sine/Cosine Converter, Multiplier and 10-bit sub-tractor.

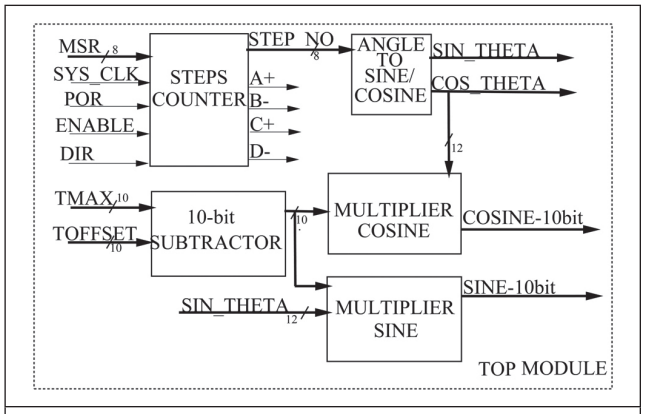

FIG. 4 FUNCTIONAL BLOCKS OF TOP MODULE.

Steps counter block takes inputs as shown in Figure 4 and outputs the step number and all 4-drive enable signals. The step number will act as an angle input to the Angle to sine/cosine block.

Angle to Sine/Cosine block is the ROM table storing up to 129 values of Sine/Cosine varying from Sine/cosine 0º to Sine/Cosine 90º to facilitate micro stepping ratio of up to 128. So given θ as input it outputs sinθ and cosθ simultaneously. The output of angle to sine/cosine block is fed to the two multipliers as shown in Figure 4. 10-bit subtractor in Figure 4 gives the difference between the maximum torque and offset torque value that is differential amplitude which is fed to two multiplier blocks along with sinθ and cosθ.

The multiplier multiplies the two inputs which are of 10 bit and 12 bit and outputs 10 bit data in digital form which are the final Sine and Cosine outputs to be fed to the external Digital to Analog converter (DAC). It is specially designed using shift and add method to synthesize less hardware [9, 10].

4.4 Clock Generation

This block is concerned with generation of clock and the functional diagram is as shown in Figure 5. The clock so generated is synchronous in nature that is to say the clock is applied to all the blocks simultaneously. The clock named sys_clock in Figure 2 decides the speed of operation (speed of the stepper motor in steps/sec) based on the micro stepping rate input (12 bit binary) given to the clock generation block by processor interface block.

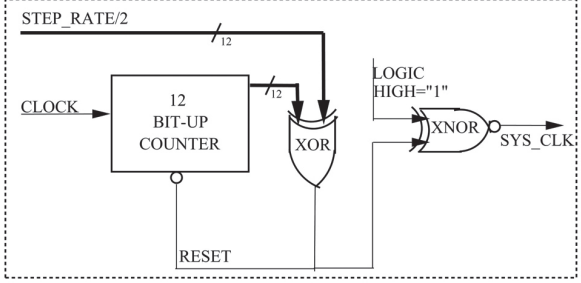

FIG. 5 CLOCK GENERATION BLOCK.

All the above blocks discussed are programmed using Very high speed integrated circuit (VHSIC) Hardware Descriptive Language (VHDL) [11]. The tool used for simulation is MODELSIM SE from Mentor Graphics and that for implementation is XILINX ISE.

5.0 DISCUSSION OF RESULTS

This section shows the implementation of the control algorithm discussed in previous section. The signals are labeled appropriately as shown in the Figure 6. From the Figure 6 we can infer that outputs are active only when POR=0 and Enable=1, not otherwise. In above test case DIR=0 hence cosine wave is reversed in nature to provide direction of rotation in CCW direction. Drive enable signals are suitably generated to sequence the sine and cosine wave to the windings of the stepper motor. Sys_clk is derived from clock depending on step_rate input. In this case step_rate is 2 hence period of sys_clk is twice that of period of clock and it decides the speed of the motor, so in other words to control the speed of the motor the step_rate is varied.

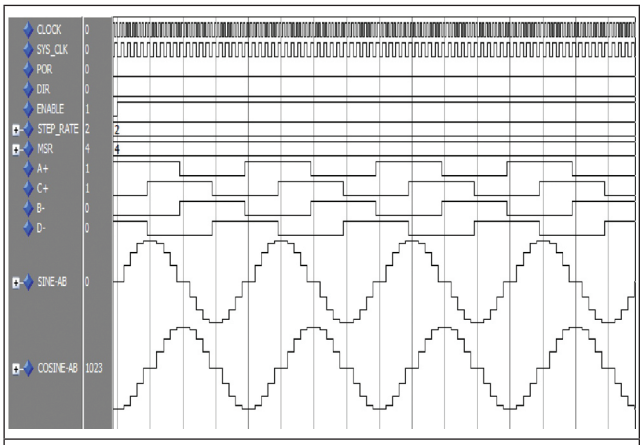

FIG. 6 RESULT FOR MICRO-STEPPING RATIO OF 4 AND CCW DIRECTION.

Figures 7 and 8 illustrate the direction control for micro-stepping ratio of 8. The results satisfy the look up table presented in Table 1 and also obey the quadrant operation as listed in Table 2. Likewise the waveforms can be obtained for micro-stepping ratio up to 128 as shown in Figures 9 and 10. The speed of motor is decided by the period of the clock pulses controlled by clock generation block which can be varied by user defined input over a wide range.

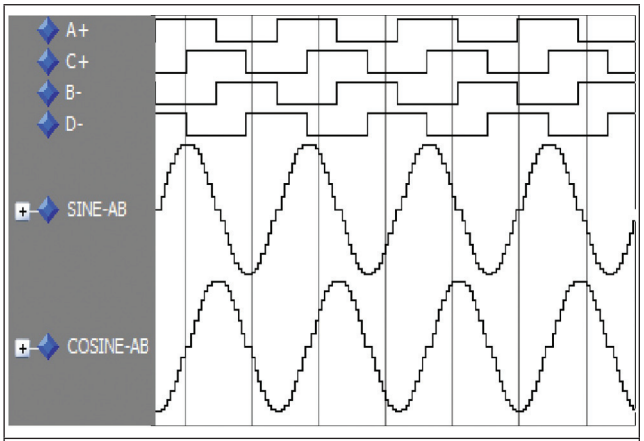

FIG. 7 RESULTS FOR MICRO-STEPPING RATIO OF 8 AND CCW DIRECTION.

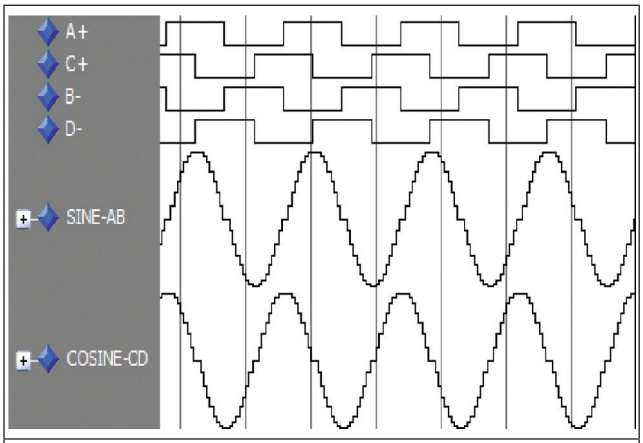

FIG. 8 RESULTS FOR MICRO-STEPPING RATIO OF 8 AND CW DIRECTION.

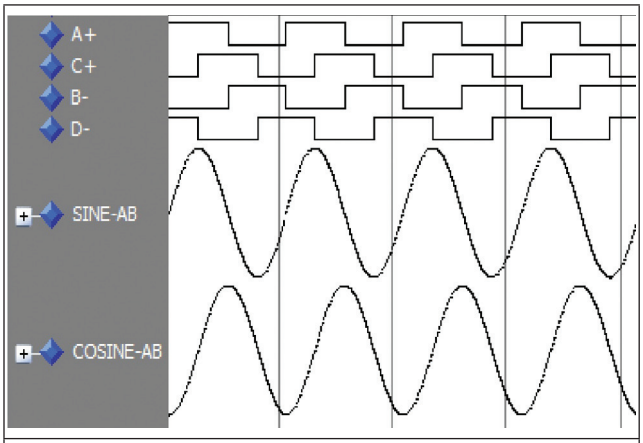

FIG. 9 RESULTS FOR MICRO-STEPPING RATIO OF 128 AND CCW DIRECTION.

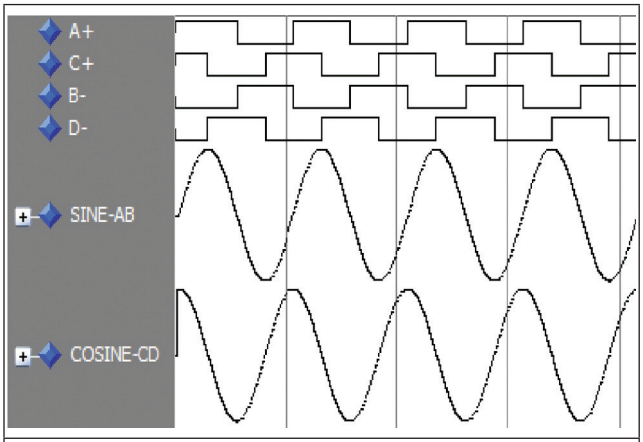

FIG. 10 RESULTS FOR MICRO-STEPPING RATIO OF 128 AND CCW DIRECTION.

Table 3 shows the synthesis report of the micro-stepping drive and that of conventional full-step method is summarized in Table 4. The Comparison reveals that the micro-step method requires more number of logic elements as compared to conventional methodology. The excess cells required is justified by sophisticated control algorithm that is being implemented to facilitate various merits in the system and to overcome the demerits of the conventional methods. Also in Table 5 timing Summary for proposed micro-stepping Methodology is discussed and it reveals that the digital controller designed in this paper can provide satisfactory performance with maximum operating frequency of 19.028 MHz.

TABLE 3

Resources	Quantity
Selected device	rh1280-s
Number of slices	757 out of 1232 62%
Number of flip flop	241 out of 608 41%
Number of IO's	52
Number of IOB's	52
Number of GCLKs	2

TABLE 4

Resources	Quantity
Selected device	3s50pq208-5
Number of slices	492 out of 768 64%
Number of flip flops	47 out of 1536 3%
Number of IO's	68
Number of IOB's	68 out of 124 54%
Number of GCLKs	2 out of 8 25%

TABLE 5

Resources	Quantity
Minimum period	52.554 ns (Maximum Frequency 19.028 MHz)
Minimum input arrival time before clock	29.483 ns
Minimum input arrival time after clock	7.842 ns
Maximum combinational path delay	7.588 ns

6.0 CONCLUSION

The work has demonstrated the design of a VHDL based current control mechanism employing micro-stepping scheme using a Xilinx and ModelSim. It can be used for solar tracking in satellites in order to obtain a high resolution

in rotational movement. The micro-stepping ratio up to 128 can be chosen with bidirectional control and speed control over wide range. The micro-stepping drive compared to conventional full-step methodology utilizes more gates at the cost of merits it adds to the system viz., motor acts like DC motor, reduced resonances, reduced noise and very high resolution. To put everything in nutshell, the work attempted in this paper has accounted all the requirements listed in section 3.0.

ACKNOWLEDGEMENT

Authors wish to thank the management of ISRO Satellite Centre (ISAC) for allowing to conduct this research work and permitting to publish this paper.

REFERENCES

[1] Satyanarayan Kumari B and Latha R. "Comprehensive design review antenna deployment electronics", *ISRO*, Bangalore, Karnataka, August 2011.

[2] Acarnley. "Stepping motors: a guide to modern theory and practice", *4th Edition Newgen Imaging Systems (P) Ltd*. India, 1982.

[3] John R Rogers and Kevin Craig. "On-hardware optimization of stepper-motor system dynamics", *Elsevier Journal in Mechatronics*, Vol. 15, pp. 291–316, 2005.

[4] Krishnamurthy and Prashanth. "Analysis of the effects of the closed-loop commutation delay on stepper motor control and application to parameter estimation", *IEEE Transactions on Control Systems Techology*, Vol. 16, No. 1, pp. 70–77, January 2008.

[5] Daniel Carrica, Macros A Funes and Sergio A Gonzalez. "Novel stepper motor controller based on FPGA hardware implementation", *IEEE Trans. Mechatronics*, Vol. 08, No. 1, pp. 120–124, March 2003.

[6] Ngoc Quy Le and Jae Wook Joen, "An open loop stepper motor driver based on FPGA," *International Conference on Control, Automation and Systems*, 2007.

[7] Anish N K, Deepak Krishnan, Moorthi S and Selvan M P. "FPGA based micro-stepping scheme for stepper motor in space-based solar power systems", *IEEE International Conference on Industrial and Information Systems.* August 2012.

[8] Athani V V. "Stepper motors fundamentals, applications and design", *1st Edition, New Age International(P) Limited*, 1964.

[9] Richard F Tinder. "Engineering digital design", *2nd Edition, Academic Press an Imprint of Elsevier Science*, USA, 2000.

[10] Morris Mano M and Michael D Ciletti. "Digital design", *4th Edition, Pearson Prentice Hall(P)*.

[11] Douglas L Perry. "VHDL Programming by Example", *4th Edition, Tata McGraw-Hill (P),* 2002.