Hardware Implementation of a High Speed Inverse Park Transformation using CORDIC and PLL for FOC Brushless Servo Drive

A. S. N. Mokhtar[1], M. B. I. Raez[1], Mohd. Marufuzzaman[1], M. A. M. Ali[1]
[1]*Department of Electrical, Electronic and Systems Engineering, Universiti Kebangsaan Malaysia,*
Bangi, 43600 UKM, Selangor, Malaysia
mamun.reaz@gmail.com

***Abstract*—Field Oriented Control in Permanent Magnet Synchronous Motor needs a quick computation in extremely short time to get an optimal efficiency and stability of the motor. The aim of this research is to present a fully integrated solution of inverse Park transformation by incorporating Coordinate Rotation Digital Computer, ARITHMETIC and Phase-locked Loop (PLL) module. But, the main problem in presenting such solution lies in completing the transformation within a limited clock cycles to pledge the steadiness of the motor where presently is the main issues of this industry. This research presents a transformation that works within 4 clock cycles with an execution time of 160ns of 24MHz frequency with an accuracy of 99.9 % which is the lowest computational cycle for the era.**

***Index Terms*—Phase locked loops, brushless machines, CORDIC, inverse park transformation, servomotors, field oriented control.**

I. INTRODUCTION

In this recent era, the number of drive systems offered to designer has increased tremendously. Among them brushless servomotor becomes one of the most useful drive options for a wide range of applications due to their high power density, high efficiency, robust operation, lower maintenance and high mechanical reliability [1]. A brushless servomotor has permanent magnet, which rotate and fixed armature. Most brushless motors need an alternating current (AC) source. This type of motor is also known as Permanent Magnet Synchronous Motor (PMSM).The vector control of the motor, which is known as Field Oriented Control (FOC) of the PMSM drive, is employed to achieve high performance. FOC has been proven a very powerful control method for motor control.

The important transformations in FOC are Park and Clarke transformation and their inverses. Park transformation is used to convert two-phase stationary frame (*α-β*) into two-phase rotating frames (*d-q*). The two-phase *d-q* is fed to a vector rotation block where it is rotated by an angle θ to follow the frame *α-β*. The rotation over an angle θ is given by equation (1) and (2):

$$V_a = V_d \cos\theta - V_q \sin\theta, \quad (1)$$

$$V_\beta = V_d \sin\theta + V_q \cos\theta. \quad (2)$$

Coordinate rotation digital computer (CORDIC) computing technique is used to realize these transformations. It was proposed by Volder [2]. CORDIC is an algorithm to compute trigonometric functions in hardware such as field-programmable gate array (FPGA). The transformation of FOC needs very quick mathematical calculations, thus CORDIC algorithm was selected for solving FOC tasks.

Ghariani et. al used a modified CORDIC to implement the Park transformation with a frequency of 8MHz in Xilinx FPGA [3]. The execution time of their design was 125ns even solely for the CORDIC algorithm execution. Rachid et. al. used two different frequencies, which were 100MHz and 8MHz where both of them need *6* clock cycles to produce the output [4]. Furthermore, their research used a rotor flux estimator for calculation of *cos θ* and *sin θ*. The calculated angle is then used in Park transformation and inverse Park transformation. So the transformation needs *6* clock cycles excluding the calculation of *cos θ* and *sin θ*. Apart from that, a research from Ying-Shieh Kung et. al. shows that inverse Park sub-module could be accomplished in *5* steps [5]. Finite state machine (FSM) method was proposed which interprets each step equals to *1* clock cycle. The operation of each step could be completed within 40 ns at 25MHz frequency, thus, execution time for whole inverse Park transformation is 0.2 µs. However, the values of *cos θ* and *sin θ* were pre-calculated in the look-up table which used a total of 24,576 bits of memory. Ricardo de Castro et. al. reported that the combination of Park transformation and Clarke transformation produces 30 clock cycles with a maximum frequency of 78MHz [6]. This research used a CORDIC algorithm that available from Xilinx for calculating the module and angle of the vector. This CORDIC implementation takes 20 clock cycles to complete.

Infineon Technologies applied FOC in their motor product [7]. It is reported that inverse Park transformation module needed *31* clock cycles (775ns) using frequency of 40MHz but it is also stated that this information do not cover the complete control algorithms. The Freescale semiconductor on the other hand requires *45* clock cycles

Manuscript received February 25, 2012; accepted November 12, 2012.

[8]. No other description given on this transformation regarding the calculation of *cos θ* and *sin θ*. Texas instrument also provides a data on the product of TMS320C2XX, which gives the performance information [9]. In this product, Linear Interpolation and single function with look-up table is used for calculating *sin θ* and *cos θ*. It shows that, with 20MHz, the inverse Park module with *sin θ* and *cos θ* calculation takes *101* cycles with 5.05µs execution time in the assembly calling functions while the fully C compatible functions requires *126* cycles with 6.03µs execution time.

Table I shows the summary of other research. This summary table compare in term of frequency used, the execution time and the clock cycle. The aim of the research is to present a method of getting a very quick computation in extremely short time in order to get an optimal efficiency and stability of the motor through the FOC in PMSM Motor.

TABLE I. SUMMARY OF OTHER RESEARCH.

Researcher	**Frequency**	**Execution time**	**Clock cycle**
Ghariani et. al. [3]	8 MHz		-
Rachid. et. al. [4]	100MHz & 8MHz	-	6
Ying-Shieh Kung et. al. [5]	25Mhz	0.2µs	5
Ricardo de Castro et.al. [6]	78MHz		30
Infineon Tech [7]	40MHz	775ns	31
Freescale Semicond. [8]	-	-	45
Texas Instrument [9]	20Mhz	5.05µs 6.03µs	101 126

II. Architecture of Inverse Park Transformation Module

The whole architecture of inverse Park transformation of this research is shown in Fig. 1. The architecture consists of three modules; Phase-Locked Loop (PLL), CORDIC and ARITHMETIC. PLL is used for frequency controls where it generates an output signal based on the reference input signal. The clock output of PLL is used as clock input for all other modules. The CORDIC module that used in this research is an existing CORDIC which originated by Volder [2]. The rotation mode is used where the input vector is rotated by a specific angle to calculate the sine and cosine function. This module is used to perform the calculation of *cos θ* and *sin θ*. This is the most important part in order to get an accurate result as it will affects the rest of the system. The ARITHMETIC module that comes after the CORDIC module is used to complete the transformation. This ARITHMETIC module consists of adder, subtractor, multiplexer and shifter. A well blend of all modules produced a complete inverse Park transformation with precise output. Fig. 1 shows the overall block diagram of inverse Park transformation.

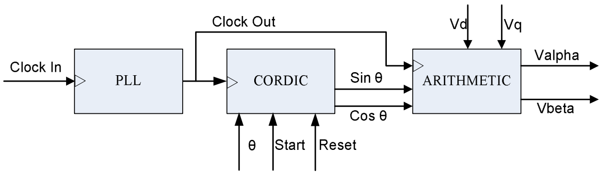

Fig. 1. Overall block diagram of inverse Park transformation

A. Phase-Locked Loop

PLL is a closed-loop frequency-control system based on the phase difference between the input and feedback clock signal of a controlled oscillator which is used on top of all modules. Fig. 2 shows the basic PLL block diagram. Phase frequency detector (PFD) is used in PLL circuit to align the rising edge of the reference input clock to a feedback clock [10]. The other input of PFD is from the output of a divide by N counter. The output of the phase detector is a voltage proportional to the phase difference between the two inputs.

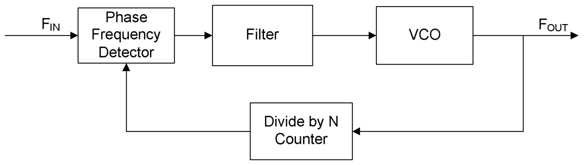

Fig. 2. Basic PLL block diagram.

This signal is applied to the loop filter. The filtered signal controls the voltage controlled oscillator (VCO). Based on the control voltage of the filter, the VCO oscillates at upper or lower frequency, which affects the phase and frequency of the feedback clock. If the PFD produces a higher signal then the VCO increases its frequency and vice versa. A divide by N counter is inserted in the feedback loop to increase the VCO frequency above the input reference frequency. VCO frequency (F_{out}) is equal to N times the input reference clock (F_{IN}). The PFD input reference clock (F_{IN}) is equal to the input clock (F_{IN}) divided by the pre-scale counter (N).

B. ARITHMETIC

ARITHMETIC module is used to complete the transformation module after getting result from CORDIC module. Both floating point and fixed point can be used to implement CORDIC algorithm [11]. Floating point is slow and consumed numerous of logic elements while fixed point is easy to deal with [12]. Fixed point arithmetic is the same as integer arithmetic, so it is very fast and mostly used in FPGAs. Therefore, in this research, fixed point is used in all modules. Fixed point format *Q1.14* is used for all inputs. The input angle *θ* is represented in radian and it ranges between *–90*^{o} and *90*^{o}. The mirror properties are used for other angle values.

Inverse Park transformation plays an important role in implementing FOC in PMSM motor. The proposed architecture of inverse Park transformation is designed in such a way that it gives accurate result with very short execution time. This will influence the overall FOC performance for faster and accurate operations.

III. Results and Discussion

The architecture has been developed using Verilog hardware description language in Quartus II Altera environment. The simulation outcome were obtained using ModelSim simulator. All modules that were described in Fig. 1 are developed. The first left module is the PLL module which employed from Megawizard function in Altera [10]. PLL module uses input frequency of 24MHz, 50 % duty clock cycle and it gives out output frequency of 144MHz. The output of this PLL is used as clock input for the others modules. The calculation of *cos θ* and *sin θ* is done before the transformation commence where the CORDIC is applied. This CORDIC module is synchronized with *start* and *reset* signal where the values in the registers are set to *0* when the *reset* signal is triggered. There is 16bits input *θ* that goes through this module and it gives 20bits output of calculated *sin θ* and *cos θ*. This 20bits output of CORDIC is directly sent to the multiplier to multiply with the 16bits input of V_d and V_q. To follow the requirement and standard, these two values will be shifted right 3bits to the right by using arithmetic shift operator. Thus, output of the ARITHMETIC module is 32 bits of V_{alpha} and V_{beta}. To maintain the signed values, the empty position in the most significant bit (MSB) is filled with the original MSB.

To verify the module, two different sets of 16bits random data are tested and compared with calculated values (numerical calculation) shown in Table II. In this module, all inputs are multiplied by *16384* or 2^{14} according to fixed point format. All of these values are represented in signed decimal number. Throughout the process, the output values need to be divided by 2^{29} in order to obtain the original real data output. To calculate the accuracy of this module, equation (3) is implemented

$$\frac{\textit{Simulation values}}{\textit{Calculation values}} \times 100\%. \tag{3}$$

From Table II, by using this formula it shows that the accuracy of all data is 99.99 %. After synthesizing and simulating the Verilog code, the result is validated by viewing the waveform generated. The number of clock cycle required to compute the output for each input is measured. Fig. 3 and Fig. 4 shows the gate level simulation (GLS) output using 2 different sets of input. The computation of this module is measured from the first rising edge of the clock after the *start* signal is activated. From this figures, it is clearly shown that the output is produced within 4 clock cycles. The execution time is about 160ns for 24MHz frequency, which means this module can work in an extremely fast environment. This fast and precise result will certainly help in the robustness of the PMSM.

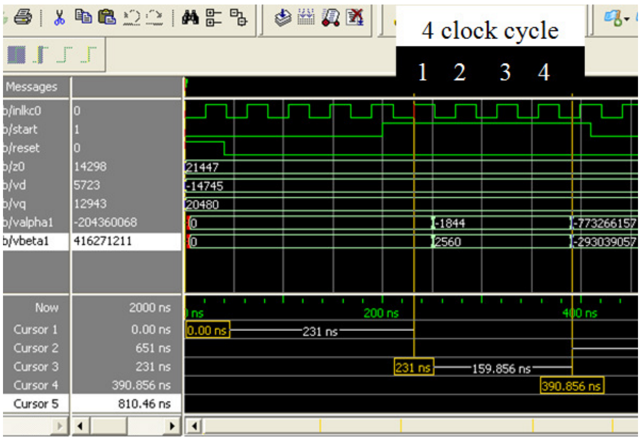

Fig. 3. Simulation results of first input data.

The FPGA resource usage summary is shown in Table III. The overall logic utilization, individual utilization for combinational adaptive look-up table, memory and registers are shown in this table. The design is used only 2 % of the logic elements as well as used no memory blocks of the target FPGA. Analyzing Table III, it is been understandable that the design is used less logic elements which will maximize area efficiency. The hardware implementation of CORDIC algorithm is used fixed point format numbers so that it requires less FPGA resource.

TABLE II. COMPARISON BETWEEN SIMULATION AND CALCULATED VALUE.

Input Data	$V_d = -0.9$ $V_q = +1.25$ $Z_0 = 75^o /$ $1.308 radian$	$V_d = 0.35$ $V_q = 0.79$ $Z_0 = 50^o /$ $0.873 radian$
Simulation Value	$V_\alpha = -773266157$ $V_\beta = -293039057$ $V_\alpha = \frac{-773266157}{2^{29}}$ $= -1.440320$, $V_\beta = \frac{-293039057}{2^{29}}$ $= -0.545827$	$V_a = V_d \cos\theta - V_q \sin\theta$ $V_a = -0.9 * \cos(75) -$ $1.25 * \sin(75) = -1.440344$ $V_\beta = V_d \sin\theta + V_q \cos\theta$ $V_\beta = -0.9 * \sin(75) +$ $1.25 * \cos(75) = -0.545809$
Calculation value	$V_\alpha = 204360068$ $V_\beta = 416271211$ $V_\alpha = \frac{204360068}{2^{29}}$ $= -0.380650$, $V_\beta = \frac{-416271211}{2^{29}}$ $= -0.775366$	$V_a = V_d \cos\theta - V_q \sin\theta$ $V_a = -0.35 * \cos(50) -$ $0.79 * \sin(50) = -0.380199$ $V_\beta = V_d \sin\theta + V_q \cos\theta$ $V_\beta = -0.35 * \sin(50) +$ $0.79 * \cos(50) = -0.775918$

TABLE III. DESIGN SUMMARY OF INVERSE PARK TRANSFORMATION.

Total logic elements	563/33,216 (2 %)
Total combination functions	499/33126 (2 %)
Dedicated logic registers	212/33216 (<1 %)
Total register	212
Total pins	115/475 (24 %)
Total virtual pins	0
Total memory bits	0/483,840 (0 %)
Embedded Multiplier 9-bits elements	16/70 (23 %)
Total PLLs	1/4 (25 %)

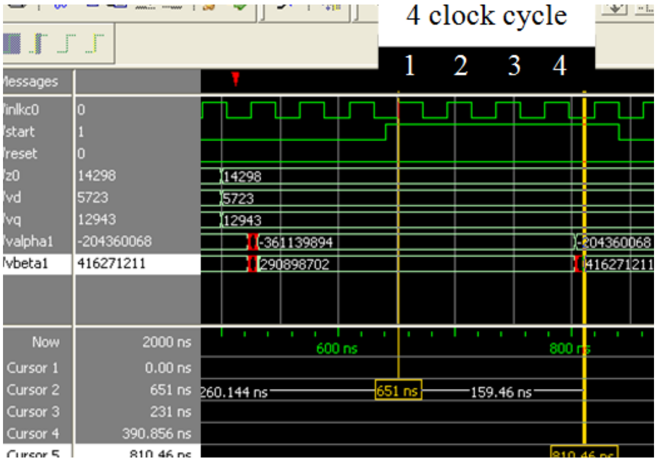

Fig. 4. Simulation results of second input data.

This inverse Park transformation module has successfully completed and in comparison with Table I, it is being shown that this work can accomplish the transformation within 4 clock cycles which is the lowest among others. Besides, this inverse Park transformation implements CORDIC for calculating *cos θ* and *sin θ* as well as the transformation itself whereas others implemented only the transformation. This research also shows the lowest execution time of 160ns compared to other works.

IV. CONCLUSIONS

This paper presents a complete inverse Park transformation including the calculation of *sin θ* and *cos θ*. After conducting a thorough evaluation and performance analysis, it is found that the hardware implementation is a prominent contribution in this area of research on the motion movement where it is able to complete the fast computational calculation within 4 clock cycles (160ns). Thus, this fast and precise result manages to prove that the overall motor system and performance of FOC PMSM have been improved in terms of its efficiency and stability.

REFERENCES

[1] Wenxiang Zhao, K. T. Chau, Cheng Ming, Ji Jinghua, Zhu Xiaoyong, “Remedial Brushless AC Operation of Fault-Tolerant Doubly Salient Permanent-Magnet Motor Drives”, *IEEE Trans. Industrial Electronics*, vol. 57, pp. 2134–2141, 2010. [Online]. Available: http://dx.doi.org/10.1109/TIE.2009.2033824

[2] J. E. Volder, “The Cordic Trigonometric Computing Technique”, *IRE Trans. Electronics Computer*. vol. EC-8, pp. 330–334, 1959. [Online]. Available: http://dx.doi.org/10.1109/TEC.1959.5222693

[3] M. Ghariai., N. Masmoudi, M. W. Kharrat, L. Kamoun, “Design and Chip Implementation of Modified CORDIC Algorithm for Sine and Cosine Functions Application: Park Transformation”, in *Proc. of the Int. Conf. Microelectronics*, 1998, pp. 241–244.

[4] B. Rachid, M. Jean-Gabriel, S. Stephane, T. Arnaud, “Towards the System On-Chip Realization of the Sensorless Vector Controller with the Microsecond-Order Computation Time”, in *Proc. of the IEEE Canadian Conf. Electrical and Computer Engineering,* 2006, pp. 1073–1077.

[5] K. Ying-Shieh, T. Ming-Hung, “FPGA-Based Speed Control IC for PMSM Drive with Adaptive Fuzzy Control”, *IEEE Trans. Power Electronics*, vol. 22, no. 6, pp. 2476–2486, 2007. [Online]. Available: http://dx.doi.org/10.1109/TPEL.2007.909185

[6] R. de Castro, R.E. Araujo, and H. Oliveira, “Control in Multi-Motor Electric Vehicle with a FPGA Platform”, *IEEE Int. Symp. Industrial Embedded Systems,* pp. 219–227, 2009. [Online]. Available: http://dx.doi.org/10.1109/SIES.2009.5196218

[7] *Application Notes of 3-phase Motor Drive Implementation of Space Vector Modulation Based on XC164CS/XC167CI,* Infineon Technologies, 2003.

[8] *Datasheet of Motor Control Library, In Inverse Park Transformation,* Freescale Semiconductor, 2009.

[9] Application Reports of Clarke & Park transforms on the TMS320C2XX, Texas Instrument, 1997, p. 46. [Online]. Available: http://www.nalanda.nitc.ac.in/industry/appnotes/Texas/motcon/bpra048.pdf

[10] *Phase-Locked Loop Megafunction User Guide,* Altera Corporation. [Online]. Available: http://www.altera.com/support/devices/pll_clock/basics/pll-basics.html

[11] K. Kota, J. R. Cavallaro, “Numerical accuracy and hardware tradeoffs for CORDIC arithmetic for special-purpose processors”, *IEEE Trans. Computers.*, no. 42, pp. 769–779, 1993. [Online]. Available: http://dx.doi.org/10.1109/12.237718

[12] Mohd. Marufuzzaman, M. B. I. Reaz, M. A. M. Ali, L. F. Rahman, “Hardware Approach of Two Way Conversion of Floating Point to Fixed Point for Current *dq* PI Controller of FOC PMSM Drive”, *Elektronika ir Elektrotechnika (Electronics and Electrical Engineering),* no. 7, pp. 79–82, 2012.