Abstract

This paper is about the implementation of Forward Clarke Transform (FCT) module on Field Programmable Logic Controller (FPGA) using Verilog hardware description language (HDL). FCT that is mathematically represented in matrix form is used to convert a three-phase Permanent Magnet Synchronous Motor (PMSM) stator current into two-phase quadrature current. The input and output (I/O) interphase signals i.e. $i_a$ and $i_b$, $i_\alpha$ and $i_\beta$ are all in 2’s complement 16 bits. To address the complicated hardware based floating arithmetic, floating point calculations involved are handled as fixed-point. Apart from that, the module is also required to complete the FCT calculation within four clock cycles.

Keywords: Clarke Transform, Space Vector, Reluctance Motor Control, HDL

1. Introduction

1.1. Green Technology Connection

One of the considerations in ensuring an environmentally friendly technology is the efficiency of energy used in driving electric motors. Examples of areas where efficiency improvement can be made in electric motors driving are electrical braking, oscillation and overshooting reduction, field weakening control, motion profiling, etc. In many parts of these processes, multiphase system calculations are used. A common example for this is the calculation involving transformations between two phase system and three phase system.

This article is to publish the findings obtained in a development process of green technology compliant motor controller. The research was funded under Technofund¹ scheme by The Ministry of Science, Technology and Innovation of Malaysia (MOSTI). The findings in this paper are part of the development that involved Forward Clarke Transform (FCT).

1.2. Mathematical Background

Clarke Transform is a mathematical transformation of three-phase coordinates into two-phase orthogonal time-varying coordinates system represented in matrix form. It was introduced in 1950 by Edith Clarke, a professor of electrical engineering from the University of Texas at Austin [10] [6]. Clarke’s derivation involved space vector concept. It is also sometimes known as $a\beta$, $a\beta\gamma$ or $dq\theta$ transformations. In contrast to the complex representation in Clarke’s derivation, real vectors representation is more convenient to be handled, where distinction between space and time phasor [11] is more evident. Equation (1.2) and Figure 1 express the simple derivation of three vectors in a space by two axis system.

¹ Project no. TF1008C130 – Sub-project No. 7 – Green Motion Controller.
Equalizing coefficient to modify the actual locus represented by the inner circle to form the outer locus as indicated in Figure 1 i.e. $\sqrt{2/3}$ is applied to the equation to conserve power between the coordinate systems [12] [5]. The $1/\sqrt{3}$ is obtained from Hermitian symmetrical components transformation while $\sqrt{2}$ coefficient is to produce the same instantaneous active power in the original coordinates and space vector frame in a symmetrical system. In the theory, factor $\sqrt{2/3}$ is used to achieve power-invariant transformation [8], however, as far as the module is concerned, the coefficient $2/3$ can be used instead of $\sqrt{2/3}$. Equation (1.2) is the FCT equation used in this module.

From Figure 1,

$$\begin{bmatrix} x \\ y \end{bmatrix} = a + b + c$$

$$= a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} -1/2 \\ \sqrt{3}/2 \end{bmatrix} + c \begin{bmatrix} -1/2 \\ -\sqrt{3}/2 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & -1/2 & -1/2 \\ 0 & \sqrt{3}/2 & -\sqrt{3}/2 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$$\begin{bmatrix} I_a \\ I_y \end{bmatrix} = \begin{bmatrix} 2/3 & -1/3 & -1/3 \\ 0 & 1/\sqrt{3} & -1/\sqrt{3} \end{bmatrix} \begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix}$$

(1.1)

(1.2)

In a balanced three phase power system, we know that,

$$i_a + i_b + i_c = i_a \angle 0^\circ + i_a \angle 120^\circ + i_a \angle -120^\circ$$

$$= i_{peak} (1 - 0.5 + j0.866 - 0.5 - j0.866)$$

Thus,

$$i_a + i_b + i_c = 0$$

(1.3)

From Equation (1.2),
In basic Field Oriented Control (FOC) motor, Clarke’s transform is utilized as reference to construct Park’s transform frame. The pairing of Clarke’s and Park’s transforms constitutes collaboration to convert stationary reference plane to rotating reference plane [7]. It is also used as reference signal for the space vector modulation (SVM) of the three-phase inverter. The implementation of both Clarke’s and Park’s transform in FOC motor makes AC motor controllable as DC motor under steady state condition [7].

Mathematically describing motor requires a study of both rotor and stator electrical and magnetic elements. The interaction between rotor and stator magnetic flux (\(\Phi_r\) and \(\Phi_s\)) triggers mechanical movement. Stator induces rotating magnetic field (\(B_S\)) and it will agitate rotor’s magnetic field (\(B_r\)). This will induce a mechanical force on the rotor to rotate to get back the right alignment against \(\Phi_s\). In a reluctant machine such as PMSM, rotor’s flux is fixed relative to its physical structure. Hence, the implementation of controller for reluctant machines shall include the manipulation of stator current (I) value.

2. Arithmetic Operation in HDL

The main approach used to complete this project was through a design methodology called data flow design [13] while the platform used was Altera DE II Board with Quartus II software IDE. With data flow design, it is possible to achieve the result in only 1 clock cycle while RTL (register transfer level) design may require more. However the design clearly shows that the requirement for FPGA resources is relatively high for a simple type of system which is 17 logic elements. The reason for this is our utilization of Altera’s built-in fast multiplier that uses a lot of logic elements.

Arithmetic operation is easy in general computer programming but not with RTL or data flow design. To assist the design, several languages have been developed to simplify the process of interconnecting electronics components. Such languages are Verilog [1], VHDL, Handel-C, etc. For this module, Verilog IEEE Std 1364-1995 and IEEE Std 1364-2001 is used.

Signed number in binary can be either in sign-magnitude or 2’s complement [15] form. The one most commonly used in digital system is the 2’s complement. In this representation, mathematical operations such as subtraction, multiplication, and division is done using adder. However, there are several cases where more improvement can be achieved by manipulating some parameters. For
example multiplying 10 in decimal number can simply be done by shifting the floating point of the
number to the right as shown in Figure 2 below. The same approach is applied in this module for
multiplication and division operations.

\[
123 \times 10^2 = 1.23 \times 10^2 \\
\text{Shift two places to right hand side}
\]

**Figure 2.** Decimal Multiplication with powers of 10

By default, Verilog handles signed number in 2’s complement representation [9]. Declaration of
input and output parameters in Verilog includes the sign. The size of input (no. bit) is extended
according to the size of output to avoid overflow error. Figure 3 shows an example of an adder being
used to perform addition operation. Notice that the number of bits of the input is internally extended to
match the size of output.

As explained by Figure 2, one can tell that arithmetic shift is also applicable in binary multiplication
and division. As shown in Figure 3, arithmetic shift in Verilog for signed number must be done using
symbol `>>>` and `<<<` instead of `>` and `<`. Example below shows \( i_a \times 2^2 \) operation.

**Figure 3.** Input Extended to match the output size

IEEE prepared Std 754 Single-Precision (32-bits), Double-Precision (64-bits), Single-Extended
Precision and Double-Extended Precision to represent floating point numbers. These standards are
directly usable using `real` keyword in Verilog. However that keyword is not available in the platform
chosen for the development (Quartus II). (Error (10172): Verilog HDL unsupported feature error xxx:
real variable data type values are not supported)

**Figure 4.** Binary multiplication with powers of 2
There are many algorithms have been created to perform fast multiplier including Booth algorithm invented by Andrew Donald Booth in 1951 [14]. In Altera Quartus II library, there is an embedded multiplier supporting signed number multiplication [4] [3] available. This is the one that has been used in this module. Figure 5 below indicates this embedded multiplier.

```
module multiplier_test
(input signed [17:0] signa, signb,
output signed [35:0] result);
assign result = signa * signb;
endmodule
```

**Figure 5.** Embedded Multiplier

3. Algorithm Framework

Input of this module is actually received from the stator as power is valued in per unit system. The Verilog code is based on the assumption that the stator current is 16 bits per sample. Following the Micro Architecture Specification (MAS) [2] provided for the project, the output of this module must be both in 16 bits also. Figure 6 illustrates the input, output and control signal included.

Assuming \( i_a \) and \( i_b \) are digitized signals of stator current. From Equation (1.5), \( i_a = i_b \) hence, \( i_{\alpha} \) is simply direct wire from \( i_a \). While from Equation (1.6), \( i_\beta = \frac{2i_b - i_a}{\sqrt{3}} \). Figure 7 shows the data flow design approach used to calculate \( i_{\beta} \).

4. Result and Discussion

Both functional and timing analysis is done using waveform simulation by assuming the range of \( i_a \) and \( i_b \) is 0000 0000 0111 1111_2 (+12710) to 1111 1111 1000 0001_2 (−12710).
Data in Table 1 is used to plot graph in Figure 8. \( i_a \), \( i_b \) and \( i_c \) are lagging 120° from each other while \( i_{\beta} \) leading \( i_{\alpha} \) by 90°.
Referring to Figure 9, at frequency of 50MHz, the calculation is completed approximately within 4 clock cycles with output error ±1. Besides that there are also visible glitches occurs at every signal transition at duration of less than 10ns.

**Figure 8.** The desired plot of Clarke transformation

**Figure 9.** Timing Simulation Waveform
5. Conclusion

The mathematical three phase conversion concept used in Clarke’s Transformation employs space vector concepts instead of the conventional line to line power conversion. Simulation proves our implementation is able to perform the calculation in 4 clock cycles.

Verilog handles signed number in 2's complement form. Besides, the output values are both required to be in 16 bits. Hence, the designer of the ADC used to convert stator current should consider transforming the analog values into 2's complement digital number. It should also use a suitable sampling rate and quantization to avoid overflow error.

Fixed point arithmetic is much easier than floating point. However, the trade-off is the precision of the result. Precision, on the other hand, can theoretically be improved by adding more clock cycles.

6. References