Application Note, V1.0, Aug. 2008

AP08086

XC88x/XC878 Series

CORDIC and MDU for Constant V/F Control of Induction Motor

Microcontrollers

Never stop thinking

Edition 2008-08-27

Published by
Infineon Technologies AG
81726 München, Germany

LEGAL DISCLAIMER

THE INFORMATION GIVEN IN THIS APPLICATION NOTE IS GIVEN AS A HINT FOR THE IMPLEMENTATION OF THE INFINEON TECHNOLOGIES COMPONENT ONLY AND SHALL NOT BE REGARDED AS ANY DESCRIPTION OR WARRANTY OF A CERTAIN FUNCTIONALITY, CONDITION OR QUALITY OF THE INFINEON TECHNOLOGIES COMPONENT. THE RECIPIENT OF THIS APPLICATION NOTE MUST VERIFY ANY FUNCTION DESCRIBED HEREIN IN THE REAL APPLICATION. INFINEON TECHNOLOGIES HEREBY DISCLAIMS ANY AND ALL WARRANTIES AND LIABILITIES OF ANY KIND (INCLUDING WITHOUT LIMITATION WARRANTIES OF NON-INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF ANY THIRD PARTY) WITH RESPECT TO ANY AND ALL INFORMATION GIVEN IN THIS APPLICATION NOTE.

Information

For further information on technology, delivery terms and conditions and prices please contact your nearest Infineon Technologies Office (**www.infineon.com**).

Warnings

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies Components may be used in life-support devices or systems only with the express written approval of Infineon Technologies, if a failure of such components can reasonably be expected to cause the failure of that life-support device or system, or to affect the safety or effectiveness of that device or system. Life support devices or systems are intended to be implanted in the human body, or to support and/or maintain and sustain and/or protect human life. If they fail, it is reasonable to assume that the health of the user or other persons may be endangered.

AP08086

Revision History:	2008-08	V1.0
Previous Version:	none	

Page	Subjects (major changes since last revision)

We Listen to Your Comments

Any information within this document that you feel is wrong, unclear or missing at all?
Your feedback will help us to continuously improve the quality of this document.
Please send your proposal (including a reference to this document) to:

mcdocu.comments@infineon.com

Table of Contents **Page**

1 Introduction

1.1 Overview

This application note describes the implementation of a constant V/F control algorithm for control of an Induction motor using space vector modulation (SVPWM). Constant V/F control is a popular algorithm for open loop speed control of AC induction motors. This method is most suitable for applications without position control requirements or the need for high precision speed control. Examples of these applications include air conditioning, fans and blowers etc.

In this application note, the principles of constant V/F control, SVPWM and the software implementation for the XC88x/XC878 microcontrollers are discussed. Also the advantages of the microcontroller peripherals: CAPCOM6E (Capture and Compare Unit for modulation and PWM generation) and the fast 10-bit ADC (Analog-to-Digital Converter), which are specifically designed for the motor control applications are discussed.

This motor control software makes use of the advantages of the XC88x/XC878 peripherals, like the implementation of SVPWM using the CORDIC and MDU coprocessor units. The software for Induction motor control is written both in C and assembly, specifically the main algorithms (e.g. V/F control, SVPWM and current calculation subroutines) are written in assembly to reduce the execution time.

1.2 Motor Theory

AC Induction motors (ACIM) are widely used in industrial and residential motor applications due to their simple construction and durability. These motors have no brushes to wear out or magnets that add to the cost.

An induction motor has basically two parts, the Stator and the Rotor. The stator is made up of a number of stampings with slots to carry three phase windings. It is wound for a definite number of poles. The windings are geometrically spaced 120^{o} apart. Two types of rotors are used in induction motors – Squirrel cage rotor and Wound rotor. The most common type of rotor is the squirrel cage rotor. The rotor consists of a stack of steel laminations with evenly spaced conductor bars around the circumference. The conductor bars are mechanically and electrically connected with end rings.

1.2.1 Principle of Operation

When a three phase AC voltage is applied to stator windings of an induction motor, a rotating magnetic field is produced. The rotating magnetic field travels at an angular speed equal to its stator frequency. It is assumed that the rotor is at standstill. The rotating magnetic field in the stator induces electromagnetic forces in the rotor windings. As the rotor windings are short circuited, current circulates in them, producing a reaction. As known from Lenz's law, the reaction is to counter the source of the rotor current, i.e., the induced emf in the rotor creates a rotating magnetic field in the rotor. The induced emf will be countered if the difference in the speed of the rotating magnetic fields from the stator and the rotor becomes zero. When the differential speed between the rotor and magnetic field in the stator becomes zero, there is zero emf and hence zero rotor current resulting in zero torque production in the motor. This is called the synchronous speed of the machine. Now, depending on the shaft load, the rotor will settle down to a speed, always less than the speed of rotating magnetic field and torque will be produced,.The speed differential is known as the slip speed.

Synchronous speed is given as

$$\boldsymbol{\omega_s = 2\pi f_s \text{ [rad/sec]}} \quad \text{where } f_s \text{ - supply frequency} \qquad (1.1)$$

Synchronous speed or speed of the stator magnetic field in rpm, given as

$$\mathbf{N_s = \frac{120 f_s}{P_p} \text{ [RPM]}} \quad \text{where } P_p \text{ – Number of poles} \qquad (1.2)$$

2 Speed control of Induction motor

The induction motor always runs at less than its synchronous speed and by controlling the synchronous speed, the actual rotor speed can be controlled. The relationship between the synchronous speed, stator poles and the supply frequency is given in equation (1.2).

2.1 Speed Control

For inverter driven induction motors, the speed can be controlled by changing the supply frequency. In order to maintain the constant air gap flux (and not allow it to saturate), the magnitude of the applied voltage needs to be varied in accordance with the frequency variation.

The rms value of air gap induced emf in an induction motor is given by

$$E_{emf} = \frac{1}{\sqrt{2}} k_w \Phi_m \omega_s T \ [v] \tag{2.1}$$

$$E_{emf} = 4.44 k_w \Phi_m f_s T \ [v] \tag{2.2}$$

Where k_w - Stator winding factor

Φ_m - Peak air gap flux

f_s - Supply frequency [Hz] ($\omega_s = 2\pi f_s$)

T - Number of turn per phase in the stator

Neglecting the stator impedance R_s+jX_{ls}, the induced emf approximately equals the supply phase voltage. Hence,

$$V_{ph} \approx E_{emf} \tag{2.3}$$

The flux is then written as

$$\Phi_m = \frac{Vph}{Kb \ \ fs} \tag{2.4}$$

Where $k_b = 4.44 k_w T$

K_b is constant, so flux is approximately proportional to the ratio between the supply voltage and frequency. This is represented as

$$\Phi_m \ \alpha \ \frac{Vph}{fs} \ \alpha \ K_{vf} \tag{2.5}$$

Where K_{vf} is the ratio between V_{ph} and f_s

From equation (2.4), it is seen that, to maintain the flux constant, K_{vf} has to be maintained constant. Therefore, whenever stator frequency is changed, the stator input voltage has to be changed accordingly so as to keep K_{vf} constant.

A number of control strategies have been formulated, depending on how the voltage to frequency ratio is maintained:

a. Constant V/F control
b. Constant slip-speed control
c. Constant air gap flux control
d. Vector control

2.2 Principle of Constant V/F Control

If the effect of stator impendence, R_s+jX_{ls}, is not neglected, than relation between applied voltage and induced emf is given below

$$V_{ph} = E_{emf} + I_s(R_s+jX_{ls}) \qquad (2.6)$$

Where I_s - Fundamental stator phase current, A

R_s - Stator resistance per phase, Ω

X_{ls} - Stator leakage reactance per phase,Ω

Induced emf is give as

$$E_{emf} = j(L_m I_m)\omega_s = j\,\lambda_m\omega_s \qquad (2.7)$$

Where L_m - Magnetizing inductance per phase, H

I_m - Magnetizing current per phase, A

ω_s - Supply angular velocity, rad/sec

λ_m - Mutual air gap flux linkages, V-s

Substituting equation (2.6) in equation (2.5), Phase voltage is derived as follow

$$V_{ph} = j\,\lambda_m\omega_s + I_s(R_s+jX_{ls}) = I_sR_s + j\,(\lambda_m\omega_s + I_sX_{ls})$$

$$V_{ph} = I_sR_s + j\omega_s\,(\lambda_m + I_s\,L_{ls}) \qquad (2.8)$$

Where $X_{ls} = L_{ls}\omega_s$

L_{ls} - Stator leakage inductance per phase, H

Hence the magnitude of phase voltage is given as

$$|V_{ph}| = \sqrt{\left((I_sR_s)^2 + \omega_s^2\,(\lambda_m + I_s\,L_{ls})^2\right)} \qquad (2.9)$$

From equation (2.8), it is clear that V/F ratio needs to be adjusted based on, the supply frequency, the air gap flux magnitude, the stator impedance and the magnitude of the stator current. Such a complex implementation is not desirable for low performance applications, such as fans and pumps; therefore it is usual to have a preprogrammed voltage to frequency relationship as shown in Figure 1. Also an offset voltage should be added at low stator frequency to overcome the stator resistance drop.

The relationship between the applied voltage and frequency is written as

$$V_{ph} = V_{offset} + K_{vf}f_s \qquad (2.10)$$

Where $V_{offset} = I_sR_s$

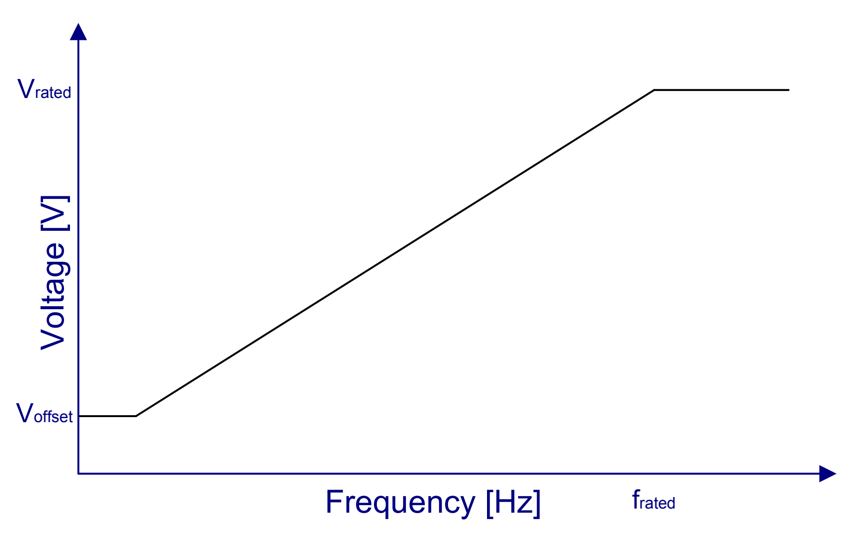

Figure 1 Voltage and Frequency profile for Constant V/F control

2.3 Principle of SVPWM Technique

Pulse with modulation can be used to create a sinusoidal voltage by adjusting the duty cycle. The inductance of the motor will filter the PWM into a smooth signal. There are different ways to generate sinusoidal voltages with a three phase inverter. Space vector pulse with modulation is one of the popular methods to produce three phase sinusoidal voltages because it generates higher voltages with lower total harmonic distortion than sinusoidal PWM techniques.

2.3.1 Three Phase Inverter

An inverter is an electronic circuit for converting direct current to alternating current. The structure of a typical three phase voltage source power inverter is shown in Figure 2. V_a, V_b and V_c are the phase voltages applied to the windings of the motor. Q_1 through Q_6 are the six MOSFETs which are controlled by the input PWM signals (A^+, A^-, B^+, B^-, C^+ and C^-), that shape the input voltages supplied to the motor terminals.

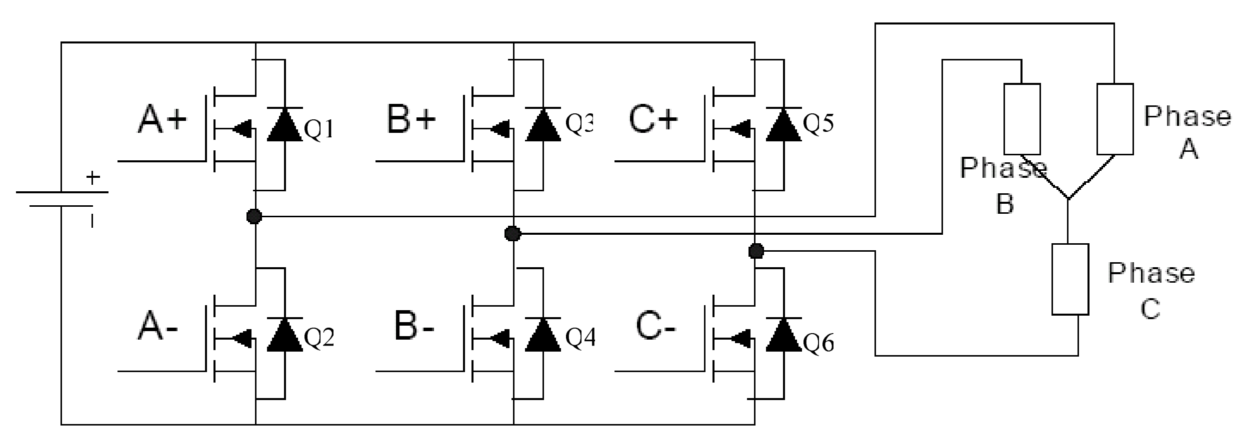

Figure 2 Three Phase Voltage Source Inverter

Note that whenever the MOSFET A^+ is switched on, MOSFET A^- must be switched off and visa versa, to prevent damaging shoot-through current.

This makes it easy to adopt a simple notation for describing the state of the inverter. For example, the state when transistor A^+, B^- and C^- are on can be represented with the notation (100). The state where transistors A^-, B^+ and C^+ are on is denoted bye (011)

2.3.2 Six Step Mode

In a three phase inverter, the three outputs can supply voltage at the motor terminals in any one of six active states "100", "110", "010", "011", "001" and "101". In addition, two inactive states that produce no voltage at the motor terminals are "000" and "111".

Six resultant nonzero voltage vectors generated from six switch states respectively is illustrated in Figure 3. They are of the same magnitude and each shifted by an angle of 60° in space. Two zero vectors (inactive states) are at the origin and supply zero voltage to a motor. These eight vectors are called the basic space vectors and are denoted by U_0, U_1, U_2, U_3, U_4, U_5, O_{000} and O_{111}.

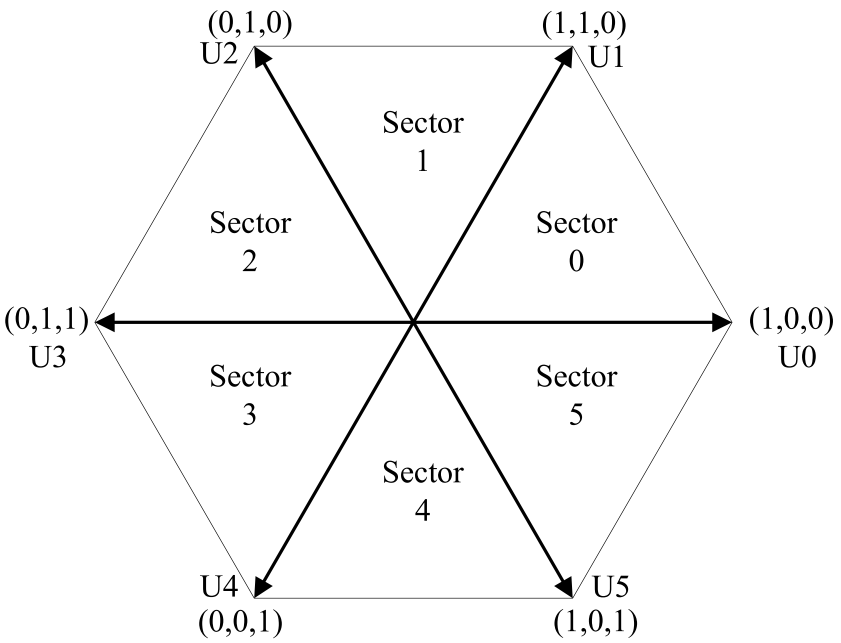

Figure 3 Space Vector Pulse Width Modulation – Sector

Consider the following sequence of states:

(100, 110, 010, 011, 001, 101)

Running the inverter through this switching sequence will produce the line-to-neutral voltages shown in Figure 4. This mode of operation is called "six-step mode".

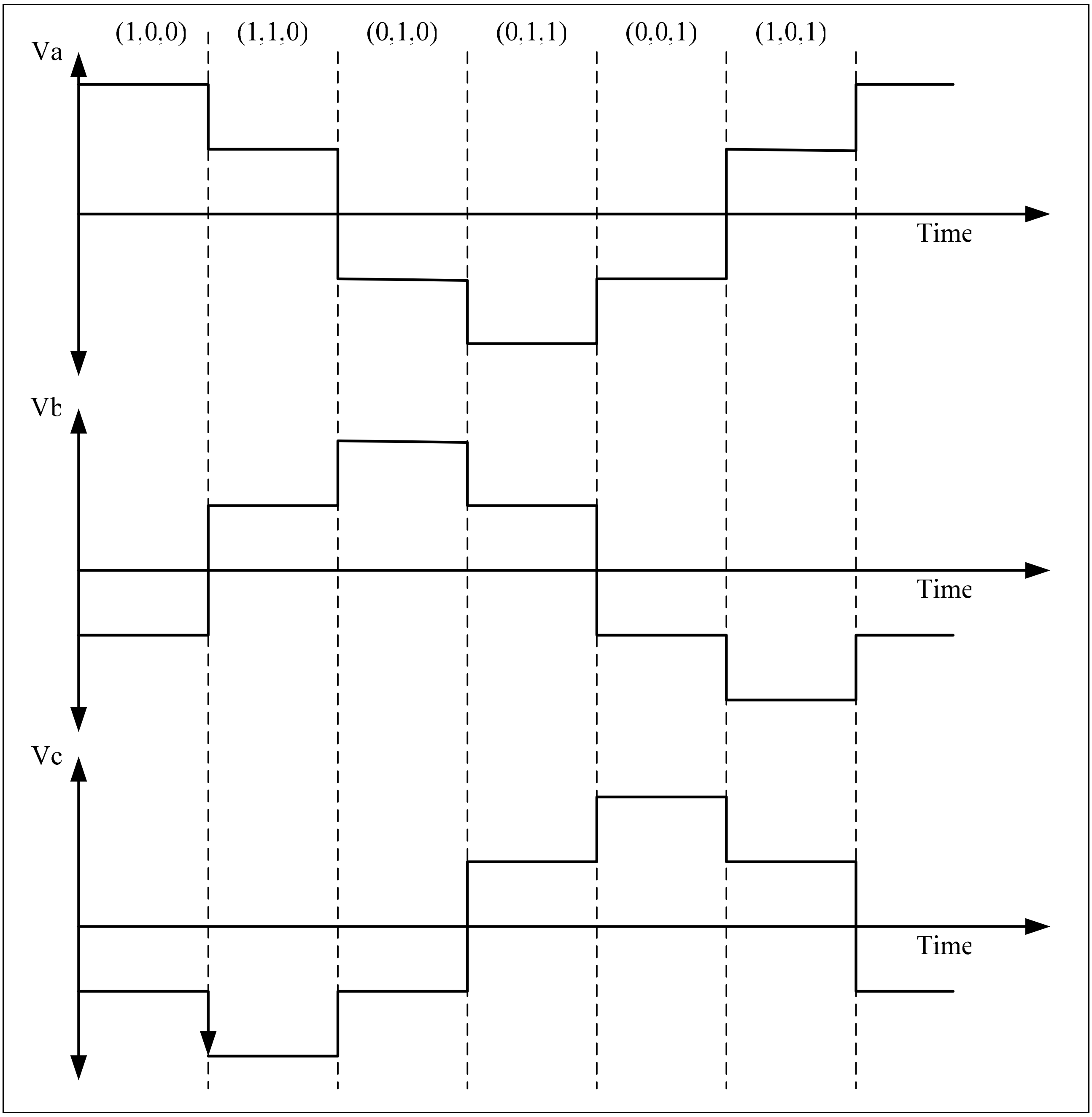

Figure 4 Phase Voltage in Six Step Mode

2.3.3 Space Vector Pulse Width Modulation

Space vector pulse width modulation is based on six step mode, but smoothes out the steps through averaging techniques. For example, if a voltage is required that is between two step voltages, the corresponding inverter states can be activated in such a way that the average of the step voltages produces the desired output voltage.

A space vector is denoted by a magnitude and an angle. As time increases, the angle of the space vector increases, causing the vector to rotate. This produces three line-to-line sinusoidal voltages that have 120^0 phase shifts.

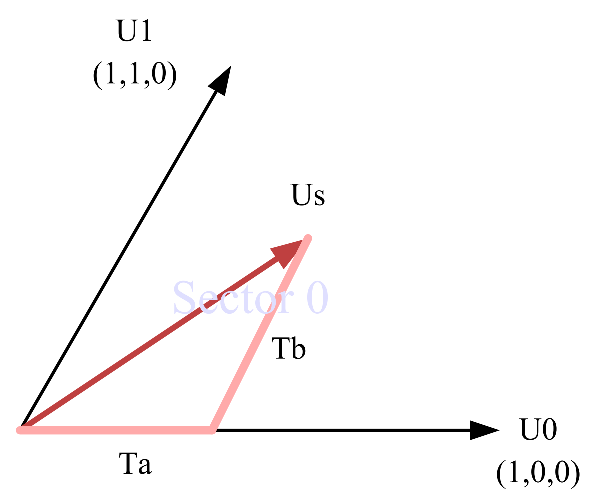

Figure 5 **Approximate of Space Vector U_s by T_a and T_b**

The goal of space vector pulse width modulation is to generate the appropriate PWM signals so that any vector can be produced. Consider a space vector (U_s) located in the sector 0 defined by U_0 and U_1, as shown in the Figure 5. The desired space vector (U_s) can be obtained by applying U_0 for a percentage of time (T_a) and U_1 for a percentage of time (T_b). In other words, U_0 (100) state is active for time T_a, U_1 (110) is active for time T_b and one of the null vectors $\mathbf{U_z}$ is active for T_0 , where $T_0=T_p- T_a - T_b$. The pulse period is called T_p.

Space vector is represented as

$$U_S T_P = U_0 T_a + U_1 T_b + U_z T_0 \quad (2.11)$$

When the modulation index (the magnitude of Us) is less than 0.866, the sum of T_a and T_b will be less than T_p. The maximum modulation index for space vector pulse with modulation is 0.866. Figure 6 shows a symmetric or center aligned Space vector modulation implementation.

For a given space vector (U_s), switching times can be calculated using the following formulas:

$$Ta = U[\cos(\alpha) - \frac{\sin(\alpha)}{\sqrt{3}}]$$

$$Tb = \frac{2}{\sqrt{3}} U \sin(\alpha) \quad (2.12)$$

Where U = $|U_s|$ and α = angle (U_s)

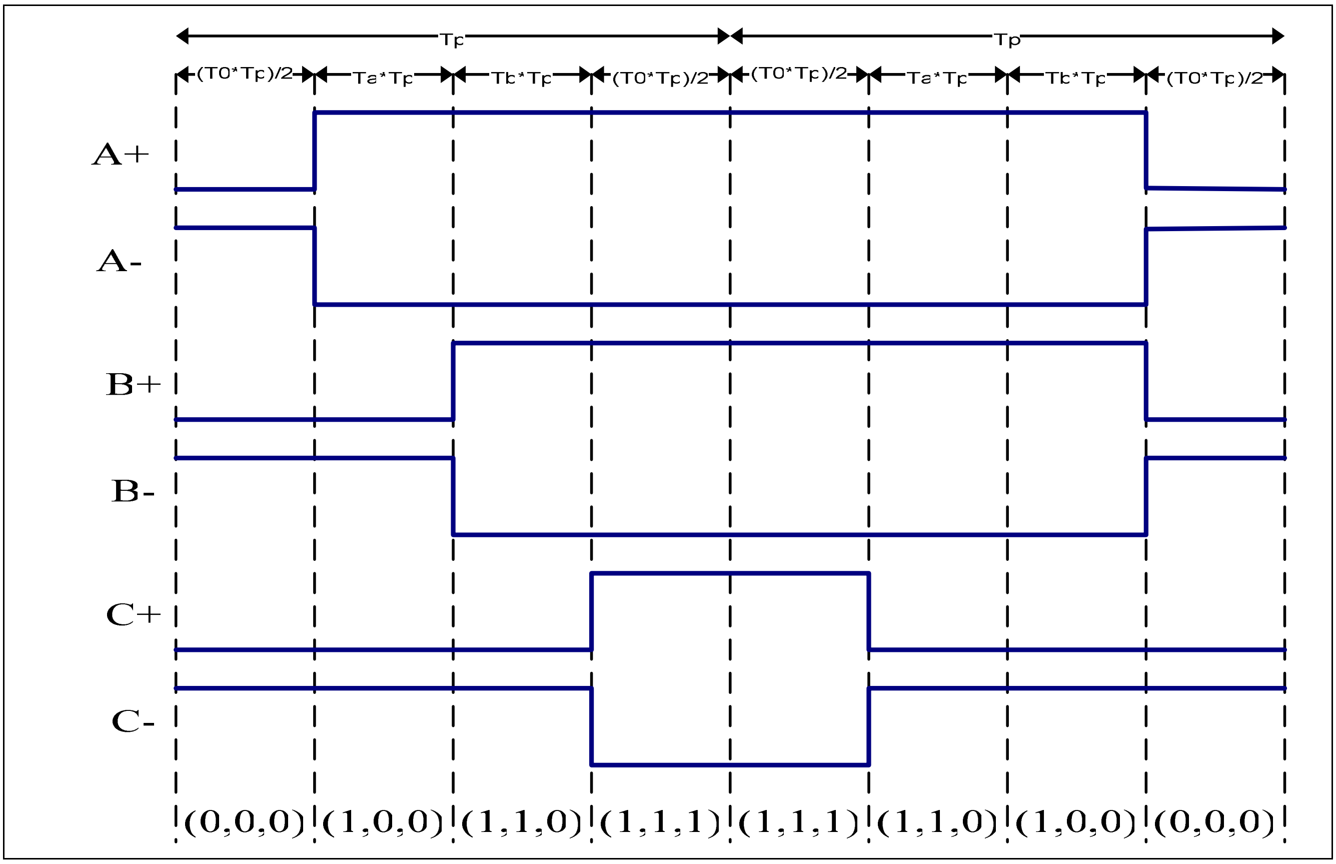

Figure 6 SVPWM using symmetric PWM for Sector 0

3 Software Implementation

In this chapter, the implementation of a constant V/F control and SVPWM generation in the XC88x/XC878 microcontrollers are discussed in detail.

3.1 Control System Overview

An implementation of a constant V/F control algorithm for inverter fed induction motors in open loop is shown in Figure 7. To implement this application in the XC88x/XC878 microcontroller, three on-chip peripheral modules are needed and they are CCU6E (CAPCOM6E), ADC (Analog-to-Digital Converter) and Timer T2.

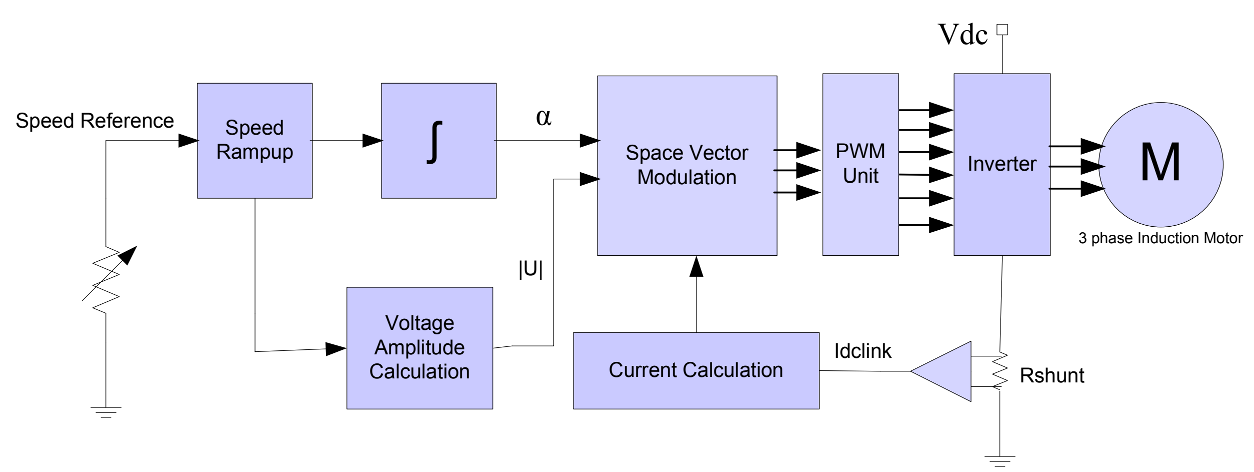

Figure 7 Implementation of Constant V/F control for Induction motor.

To generate the PWM control signals for the inverter, the CCU6E module is used. For this purpose, timer T12 and the CC60SR, CC61SR, CC62SR compare registers are used. Timer T12 operation is configured for center aligned Mode. Dead-time control is enabled for the six PWM signals to avoid shoot-through current. The control algorithm and over current protection algorithm are executed in timer the T12 period match ISR.

The timer T13 is configured in single shot mode. It starts synchronously with a T12 zero match. The timer T13 period match and CC63SR compare events are used to trigger ADC module for current measurement. The period register of timer T13 (CCU6_T13PR) and CCU6 channel 3 compare register (CCU6_CC63SR) are updated every second timer T12 period. And this value is always greater than 1.5 times of T12 period value.

The timer T12 period match interrupt is blocked in the timer T12 period match ISR and released in timer T13 period match ISR. As a result of this setup, the timer T12 period match ISR is triggered in every second T12 period match.

The ADC module is used for measurement of the motor current and speed reference value. Channel 3 and Channel 4 are used for the measurement of current at different time instants from a single current shunt. The measurement results of channel 3 and channel 4 are stored in result register 0 and 1 respectively. Also channel 0 is configured to measure the speed reference value and the result is stored in result register 2.

In the Timer T2 overflow interrupt service routine, the voltage amplitude is calculated from the input speed reference value which is stored in result register 2. Also the Speed ramp up rate is controlled using Timer T2.

3.2 CCU6 Timer T12 Period Match Interrupt

During this interrupt routine, all calculations necessary for constant V/F control is executed like the angle calculation, SVPWM generation, update of the CCU6E compare registers and over current check.

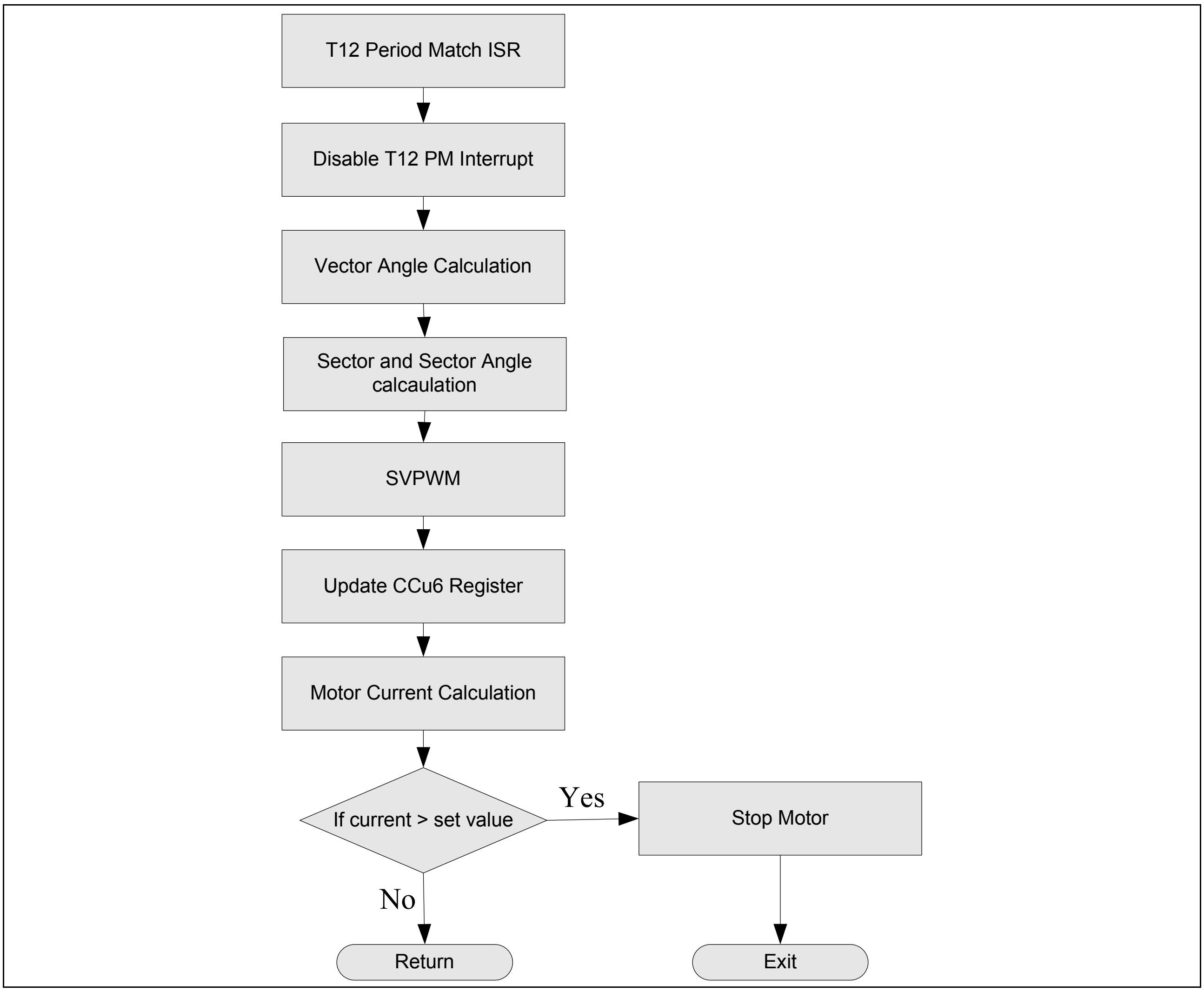

Figure 8 **Flow chart T12 Period Match ISR**

3.2.1 T12 Period Match ISR Disabled

As the execution of all calculations takes longer than one T12 period cycle, T12 period match ISR has to be disabled during calculation. It is enabled after measuring the current value by timer T13 period match ISR.

3.2.2 Angle Calculation

The vector angle is derived by integrating the motor speed over a time period. As the control strategy used here is open loop, the actual motor speed is neither measured nor calculated by any feedback mechanism. Thus the reference speed is used for the vector angle calculation.

In the actual implementation, integral calculation of vector angle is achieved by simple summation of the reference speed over a time period.

Now the scaling of all parameters involved in this angle calculation should be taken care of both at the input and output side as the final result of the angle calculation will be used by the SVPWM algorithm for which the CORDIC coprocessor is used. The value range of input to the CORDIC processor is [2^{15}-1, 2^{15}]. Hence the output vector angle range [$-\pi,\pi$] is represented as [2^{15}-1, 2^{15}]

At the input side, the scaling for speed should be taken care of in such a way that the summation of this value gives the vector angle value.

The speed scaling value is calculated as shown below:

$$\mathbf{N_{speed}} = \frac{2^{15} * 60}{\Delta T * 2^{16} * P_p} \qquad (3.1)$$

Where P_p - Number of Poles

ΔT - Function call time ($2*T_{12PM}$)

The relation between actual value, normalization value and target value is given below

$$Target\_Value = \frac{Actual\_Value * 2^{15}}{Normalization\_value} \qquad (3.2)$$

3.2.3 Space Vector Modulation

3.2.3.1 Calculation of PWM duty cycle for SVM

To generate six PWM signals (two signals for each phase), a reference voltage vector (U_s) is required to determine the switching states and the corresponding duty cycle values. The voltage vector is represented in terms of magnitude (U) and sector angle (α). The voltage magnitude (U) can be obtained by multiplying the reference speed value with V/F constant. This calculation is done in every Timer T2 overflow ISR. Sector and sector angle (α) are calculated from vector angle, and this calculation is done using the MDU coprocessor.

The Code for the sector angle calculation is shown in Figure 9

```
//Sector = (int)(((Angle)*6)>>16) & 0x0007);

          MOV  MDU_MD0,Angle+1                //Vector Angle Low Byte Value
          MOV  MDU_MD1,Angle                  //Vector Angle High Byte Value
          MOV  MDU_MD4,#006h
          MOV  MDU_MD5,#000h
          MOV  MDU_MDUCON,#10H;

loop1:
          MOV    A,MDU_MDUSTAT
          JB     ACC.2,loop1                  // Check whether MDU is ready

          MOV  Sector+1,MDU_MR2               // Sector value

//AngleTab= (unsigned int)Angle-10922*Sector;

          MOV  MDU_MD0,MDU_MR0                //Vector Angle*6  Low Byte Value
           MOV  MDU_MD1,MDU_MR1               //Vector Angle*6 Low Byte Value
          MOV MDU_MD4,#006h
          MOV MDU_MD5,#000h
          MOV  MDU_MDUCON,#11H;

loop2:
          MOV    A,MDU_MDUSTAT                // Check whether MDU is ready
          JB     ACC.2,loop2

          MOV SectorAnlge,MDU_MR1             // Sector Angle value
          MOV SectorAnlge+1,MDU_MR0
```

Figure 9 Sector angle calculation using MDU

Given the voltage amplitude (U) and sector angle (α), switching times (T_a and T_b) can be calculated using the following formulae.

$$\mathbf{T_a = U[\cos(\alpha) - \frac{\sin(\alpha)}{\sqrt{3}}] \quad = \quad \frac{2}{\sqrt{3}} U \sin(60 - \alpha)}$$

$$\mathbf{T_b = \frac{2}{\sqrt{3}} U \sin(\alpha)} \tag{3.3}$$

Where U = $|U_s|$ and α = angle (U_s)

In general, a look-up table could be used for calculating sine and cosine of the sector angle. In a look-up table approach, the table needs to be updated if the user changes PWM period (timer T12 period value) or modulation index. In this current implementation, it is possible to avoid the look-up table by using the CORDIC coprocessor for the sine and cosine calculations. For the calculation of the switching times, CORDIC coprocessor is used.

CORDIC Configuration		**Circular Rotation Mode**
X_{inital} = U	Y_{inital} =0	Z_{inital} = α
X_{final} = K[X cos(Z) –Y sin(Z)]	Y_{final} = k[Ycos(Z) + X sin(Z)]	Z_{final} =0
MPS=0	T_b = Y_{final}	Interrupt Disabled

Table 1 CORDIC Configuration for switching time calculation

Compare values are calculated from T_a and T_b values using following formulae:

$$V_{Ta} = \frac{1}{2}(T_{12P} - T_a - T_b)$$

$$V_{Tb} = \frac{1}{2}(T_{12P} + T_a - T_b)$$

$$V_{Tc} = \frac{1}{2}(T_{12P} + T_a + T_b)$$

$$V_{Td} = \frac{1}{2}(T_{12P} - T_a + T_b) \tag{3.4}$$

Where T_{12P} - PWM period (Timer T12 period value)

3.2.4 Update of CAPCOM6 Registers

The detailed compare values of CAPCOM6 compare registers at each sector is given in Table 2.

Based on the current sector, the corresponding compare values are loaded into the respective shadow registers. Shadow transfer will happen during timer T13 period match ISR.

Sector	**PWM Generation**			**ADC Trigger**	
	CC60SR	**CC61SR**	**CC62SR**	**CC63SR**	**T13PR**
0	V_{Ta}	V_{Tb}	V_{Tc}	$(3T_{12P} + T_b)/2 + D$	$(5T_{12P} + Ta)/2 + D$
1	V_{Td}	V_{Ta}	V_{Tc}	$(3T_{12P} + T_b)/2 + D$	$(5T_{12P} + T_a)/2 + D$
2	V_{Tc}	V_{Ta}	V_{Tb}	$(3T_{12P} + T_b)/2 + D$	$(5T_{12P} + T_a)/2 + D$
3	V_{Tc}	V_{Td}	V_{Ta}	$(3T_{12P} + T_b)/2 + D$	$(5T_{12P} + T_a)/2 + D$
4	V_{Tb}	V_{Tc}	V_{Ta}	$(3T_{12P} + T_b)/2 + D$	$(5T_{12P} + T_a)/2 + D$
5	V_{Ta}	V_{Tc}	V_{Td}	$(3T_{12P} + T_b)/2 + D$	$(5T_{12P} + T_a)/2 + D$

Table 2 Compare Values for CAPCOM6 compare registers at each sector

3.2.5 Current Measurement

In order to trigger the current measurement very accurately, the amplified voltage of the current shunt is fed to two ADC channels, which are measured at different time intervals and stored in separate result registers. The ADC conversions are consecutively started every second period (2*T12) of the modulation. The first current measurement is triggered by the compare match of CC63 and sampled by the ADC channel 3. The second measurement by ADC channel 4 is triggered by the period match of timer T13. ADC channel 3 is used to measure the positive current and channel 4 measures the negative current. Figure 10 shows the usage of timers T12 & T13, the interaction of the timer interrupts, and the event triggers for the start of ADC conversions.

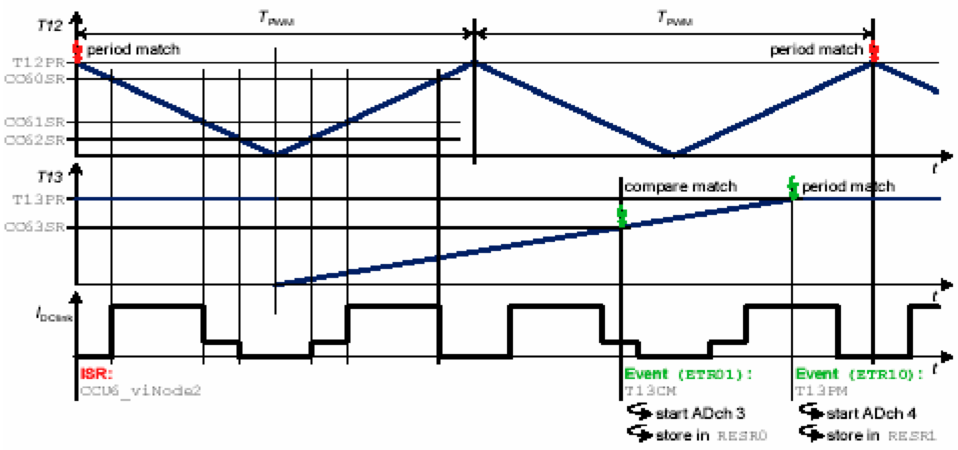

Figure 10 Timing Diagram of Timer T12 and T13

The following table describes, the motor phase current measured at each sector during both trigger events.

Sector		0	1	2	3	4	5
$I_{dc\ link}$	CC63 CM ISR	I_a	I_b	I_b	I_c	I_c	I_a
	T13 PM ISR	$-I_c$	$-I_c$	$-I_a$	$-I_a$	$-I_b$	$-I_b$

Table 3 Phase Current measured at each Sector

Two phase currents are calculated for each sector from the ADC registers. Motor current is computed from these two phase current values using the MDU coprocessor unit. Over current protection is also implemented in this software. If the motor current value exceeds the set limit value, motor will be stopped.

The maximum current range is defined as follows:

$$Imax = \frac{Vadcref}{Rshunt * Gop} \tag{3.5}$$

Where V_{adcref} - ADC reference Voltage

R_{Shunt} - Current shunt resistor value

G_{OP} - Amplifier gain

In this implementation 10 bit ADC value is multiplied by 8. The current scaling is given below.

$$N_I = \frac{Imax * 2^{15}}{8 * 2^{10}} \tag{3.6}$$

3.3 Timer T2 Overflow Interrupt

In this motor control application, Timer T2 is used to generate interrupts for every 300 µS. During this interrupt routine, the speed reference value is read from ADC result register 2 and voltage amplitude value is calculated.

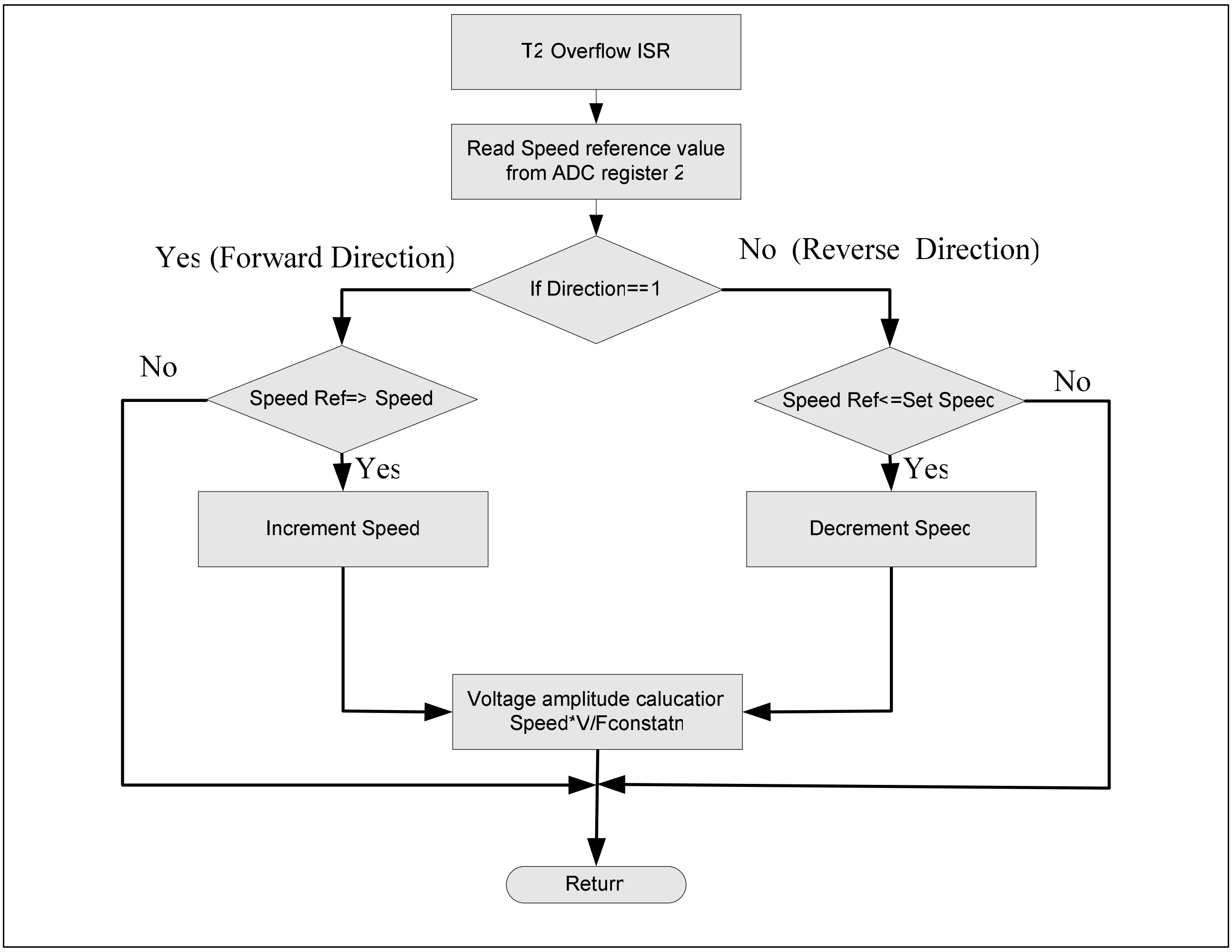

Figure 11 Flow chart Timer T2 Overflow ISR

Given speed reference value, the voltage amplitude value can be calculated using the following formulae

$$\mathbf{U = V_0 + K_{VF} * f_s} \tag{3.7}$$

Where V_0 - Voltage offset value (I_s*R_s)

K_{VF} - V/F Constant

f_s - Supply frequency

$$f_s = \frac{N_s * P_p}{120} \tag{3.8}$$

Where Ns - Speed reference value

The voltage amplitude value depends on the maximum duty cycle of the PWM which is defined by the period value of timer T12. So V/F constant (K_{VF}) value is scaled depending upon PWM period value (Timer T12). Normalization value of K_{VF} is given below.

$$N_{VF} = \frac{2^{15} * 120 * N_{V}}{N_{speed} \quad * P_{p}} \qquad (3.9)$$

Where N_V - Voltage Normalization value

N_{speed} - Speed Normalization value

P_p - Number of poles

Voltage normalization value calculation is given below

$$N_{V} = \frac{2^{15} * \sqrt{3} * V_{dclink}}{3 * 2} * \frac{2 * f_{pwm}}{f_{cpu}} \qquad (3.10)$$

Where V_{dclink} - DC link Voltage [v]

f_{PWM} - PWM frequency

f_{CPU} - CPU frequency

Speed slew rate is controller by Timer T2. Slew rate calculation is given below

$$Slew\_Rate = \frac{N_{speed}}{(T2 * TR * 2^{15})} \text{ [RPM/S]} \qquad (3.11)$$

Where N_{speed} - Speed Normalization Value

T2 - Timer Overflow (µS)

TR - Rampup Counter

Appendix A Code for Switching Time Calculation

```
       ORL    SYSCON0,#01               //switch to mapped SFRs
        CLR    a
        MOV    CD_STATC,#00
        MOV    CD_CON,#10
        MOV    CD_CORDYL,#0
        MOV    CD_CORDYH,#0
        MOV    CD_CORDZH,AngleTab        //sector angle Value
        MOV    CD_CORDZL,AngleTab+1
        MOV    CD_CORDXH,Amplitude       //Voltage Amplitude Value
        MOV    CD_CORDXL,Amplitude+1     //CORDIC starts autmatically

loop3:
        MOV    A,CD_STATC                //wait for CORDIC with circular rotation
        JNB    ACC.2,loop3

        MOV    a,Sector+1                //switching time calculation
        JB     ACC.0,sector_odd1
        MOV    Tb+1,CD_CORDYL            //if sector 0,2,4; Tb=M*sin(sector_angle)
        MOV    Tb,CD_CORDYH
        SJMP   sector_exit1

sector_odd1:
        MOV    Ta+1,CD_CORDYL           //if sector 1,3,5; Ta=M*sin(sector_angle)
        MOV    Ta,CD_CORDYH
sector_exit1:

        SETB   C
        MOV    a,AngleTab+01             //Calculation of  60-sector angle Value
        SUBB   A,#0AAH
        MOV    A,AngleTab
        SUBB   A,#02AH
        JNC    value_high

        CLR    C
        MOV    A,#0AAH
        SUBB   A,AngleTab+01H
        MOV    AngleTab+01H,A
        MOV    A,#02AH
        SUBB   A,AngleTab
        MOV    AngleTab,A
       SJMP   value_exit
value_high:
        CLR    A
        MOV    AngleTab,A
        MOV    AngleTab+01H,A
 value_exit:
```

```
        ORL    SYSCON0,#01                  //switch to mapped SFRs
        CLR    a
        MOV    CD_STATC,#00
        MOV    CD_CON,#10
        MOV    CD_CORDYL,#0
        MOV    CD_CORDYH,#0
        MOV    CD_CORDZH,AngleTab           // 60-sector angle Value
        MOV    CD_CORDZL,AngleTab+1
        MOV    CD_CORDXH,Amplitude          //Voltage Amplitude Value
        MOV    CD_CORDXL,Amplitude+1        //CORDIC starts autmatically

loop4:
        MOV    A,CD_STATC                   //wait for CORDIC with circular rotation
        JNB    ACC.2,loop4


        MOV    a,Sector+1                   //switching time calculation
        JB     ACC.0,sector_odd2
        MOV    Ta+1,CD_CORDYL               //if sector 0,2,4; Ta=M*sin(60-sector_angle)
        MOV    Ta,CD_CORDYH
        SJMP   sector_exit2
sector_odd2:
        MOV    Tb+1,CD_CORDYL               //if sector 1,3,5; Tb=M*sin(60-sector_angle)
        MOV    Tb,CD_CORDYH
sector_exit2:
        ANL    SYSCON0,#0FEH                //switch to standard  SFRs
```

http://www.infineon.com

Published by Infineon Technologies AG