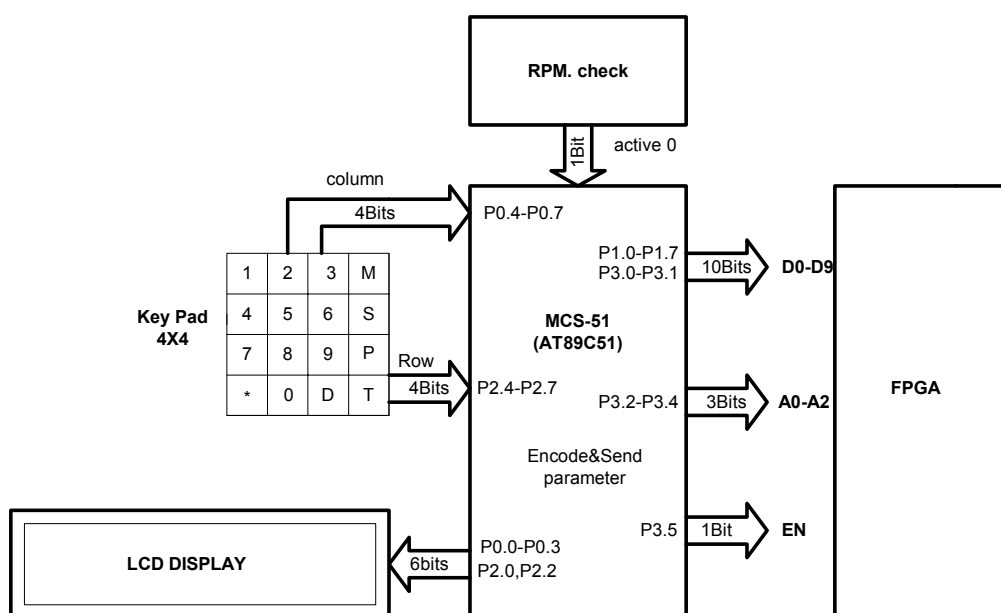


บทที่ 4

การออกแบบโปรแกรม

ในส่วนการออกแบบโปรแกรมเพื่อสร้างสัญญาณ SVM PWM นั้นจะประกอบไปด้วยสองส่วนได้แก่ ส่วนของไมโครคอนโทรลเลอร์สำหรับรับค่าอินพุตตัวแปร และส่วนของ FPGA ที่ถูกใช้งานด้านการคำนวณ และสร้างสัญญาณ SVM PWM เพื่อขับมอเตอร์

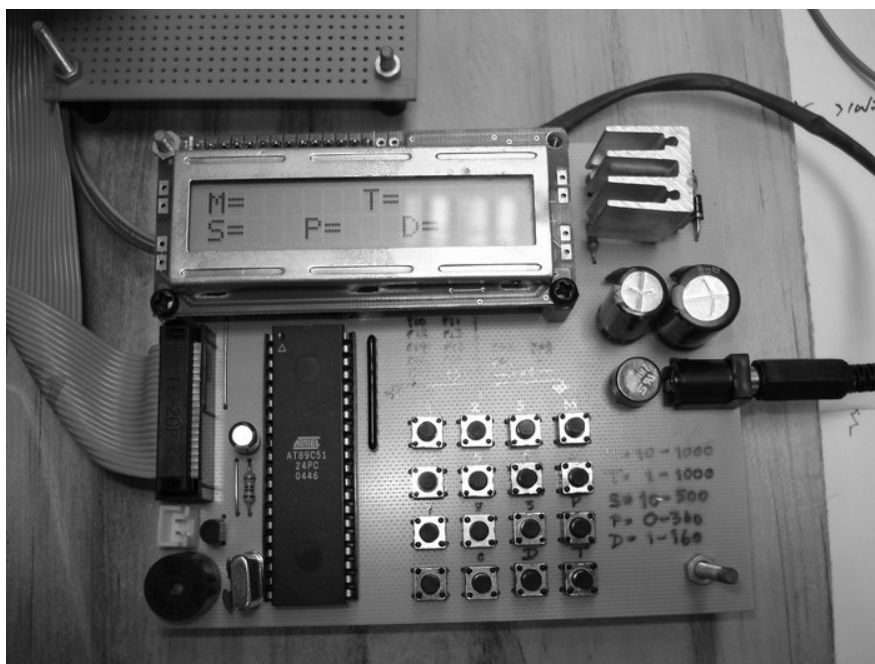
4.1 การออกแบบส่วนไมโครคอนโทรลเลอร์ส่งค่าตัวแปรให้ FPGA



รูปที่ 4.1 บล็อกไดอะแกรมส่วนรับ-ส่งค่าตัวแปรเพื่อส่งให้ FPGA

ไมโครคอนโทรลเลอร์ MCS-51(AT89C51) ใช้ในการติดต่อกับผู้ใช้งานโดยการตั้งค่าตัวแปรผ่านทางคีย์บอร์ด เพื่อตั้งค่าตัวแปรต่างๆที่ใช้สำหรับการควบคุมการทำงานของมอเตอร์ อาทิ เช่น ธรรมชาติการมอดูเลต , ความถี่มอเตอร์ , ความถี่การสวิตช์ และ Dead time โดยแสดงผลของการตั้งค่าบนจอแอลซีดี โดยค่าตัวแปรที่ทำการปรับตั้งจะถูกส่งไปให้กับ FPGA เพื่อใช้ในการคำนวณเพื่อสร้างสัญญาณ SVM PWM โดยจะแบ่งข้อมูลที่ส่งให้ FPGA เป็น 3 ส่วนคือ

- ส่วนของค่าช่วงตัวแปร(DATA) โดยแสดงค่าช่วงตัวแปรดังตารางที่ 4.1 จำนวน 10 Bits
- ส่วนของการเลือกตัวแปร(ADDRESS) โดยแสดงดังตารางที่ 4.2 จำนวน 3 Bits



รูปที่ 4.2 บอร์ดไมโครคอนโทรลเลอร์ MCS-51 สำหรับรับ-ส่งค่าตัวแปร

ส่วนของสัญญาณ Enable เพื่อให้ FPGA รับข้อมูลที่ส่งจากไมโครคอนโทรลเลอร์ MCS-51 จำนวน 1 Bit

ตารางที่ 4.1 กำหนดตัวแปรที่แสดงบนจอแอลซีดี และช่วงค่าตัวแปร

ตัวแปรใน MCS-51	ตัวแปรใน FPGA	ค่าช่วงตัวแปร ที่ MCS-51 ส่ง ให้ FPGA	ความหมายตัวแปร
M	m_index	1-1000	ครรชนิการมอดดูเลต
S	step	1-500	ความถี่มอเตอร์
T	Ts	1-1000	ความถี่การสวิตช์
P	pshift	1-360	มุมแรงดันระหว่างเฟส A และ B
D	dtime	1-100	Dead time

การกำหนดพอร์ตข้อมูล

จากตารางที่ 4.1

1) ค่าช่วงตัวแปร ที่ส่งให้กับ FPGA อยู่ในช่วง 0-1000 โดยใช้ 10 Bits ดังนั้นจำเป็นต้องใช้ พอร์ตของ MCS-51 ทั้งหมด 10พอร์ตโดยที่ใช้งานพอร์ต P1.0 ถึง P1.7 และ P3.0 P3.1 ใน MCS-51 ดังรูปที่ 4.1 โดย P1.0 เป็น Bit ค่าน้อยสุด หรือ LSB และ P3.1 เป็น Bit ค่ามากที่สุด หรือ MSB

2) ค่าตัวแปรที่ปรับค่าได้มี 5 ตัวแปรดังนั้น กำหนดการเลือกปรับค่าตัวแปรแต่ละตัวตามค่าเลขฐานสองโดยใช้ทั้งหมด 3 Bits จำนวน 3 พอร์ตโดยที่ใช้งานพอร์ต P3.2 ถึง P3.4 ใน MCS-51 ดังรูปที่ 4.1 โดย P3.2 เป็น Bit ค่าน้อยสุด หรือ LSB และ P3.4 เป็น Bit ค่ามากที่สุด หรือ MSB

ตารางที่ 4.2 ตัวแปรเทียบกับค่าเลขฐานสอง

ตัวแปร	ค่าเลขฐานสอง
m_index	000
step	001
pshift	010
T	011
dtime	100

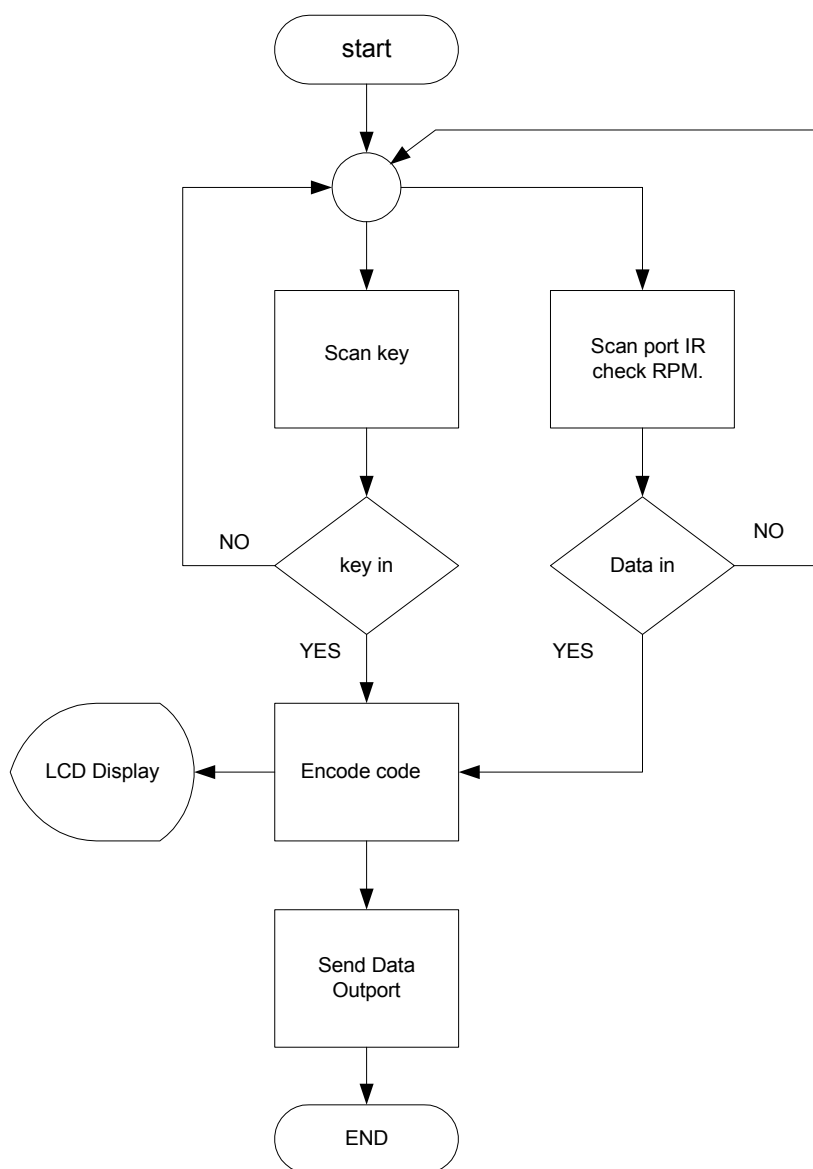
3) บิต Enable สำหรับการตรวจสอบค่าที่ส่งมาถูกต้องหรือไม่ โดยปกติจะมีค่าสถานะลอจิกเป็น 1 และเมื่อมีการเปลี่ยนแปลงค่าตัวแปร และค่าช่วงตัวแปร จึงจะมีค่าสถานะลอจิกเป็น 0 เพียงช่วงเวลา 1ms เท่านั้นแล้วจะกลับมาเป็นสถานะลอจิก 1 เหมือนเดิมโดย กำหนดให้ FPGA ทำการอ่านค่าที่พอร์ตนี้ โดยกำหนด เป็น active 0 เท่านั้น FPGA จึงจะรับค่าตัวแปรเข้าไปคำนวณโดยการออกแบบ จะให้ FPGA จะทำการวนอ่านค่าที่พอร์ตนี้ ถ้าสถานะลอจิกเป็น “0” ก็จะดำเนินการรับค่าจาก พอร์ต data 10 Bits และ select 3 bits เพื่อเก็บค่าเข้าไปคำนวณเพื่อสร้างสัญญาณ SVM PWM โดยที่ใช้งานพอร์ต P3.5 ของ MCS-51 เป็นบิต Enable ดังรูปที่ 4.1

4) คีย์บอร์ดใช้สำหรับป้อนค่าช่วงตัวแปรและตัวแปร โดยตัวคีย์บอร์ดมีจำนวน 16 คีย์และมีโครงสร้างในรูปแบบ 4 แถว และ 4 หลัก ซึ่ง MCS-51 จะต้องใช้พอร์ตจำนวน 8 พอร์ตในการเชื่อมต่อกับคีย์บอร์ดโดยที่ใช้งานพอร์ต P0.4 ถึง P0.7 เป็นตำแหน่งหลัก และพอร์ต P2.4 ถึง P2.7 เป็นตำแหน่งแถว ของ MCS-51 ดังรูปที่ 4.1

5) แอลซีดี ใช้แสดงผลของการตั้งค่าตัวแปร โดยเชื่อมต่อกับพอร์ตของ MCS-51 จำนวน 6 พอร์ต โดยที่ใช้งานพอร์ต P0.0 ถึง P0.3 เป็น LCD data(ข้อมูลที่ส่งให้ LCD) และ P2.0 เป็น LCDRS(เป็นสัญญาณการเขียนและอ่าน LCD) P2.1 เป็น LCDE (เป็นสัญญาณ Enable จอ LCD) ดังรูปที่ 4.1

4.2 โปรแกรมส่วนของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ถูกนำมาใช้ในการติดต่อกับผู้ใช้งาน โดยแสดงดังรูปที่ 4.3 เริ่มต้นการทำงานของโปรแกรมจะทำการรอรับค่าจากคีย์บอร์ด เมื่อมีการป้อนค่าโดยให้ป้อนค่าตัวแปร แล้วตามด้วยค่าช่วงของตัวแปร โดยป้อนเป็นเลขฐาน 10 เมื่อรับค่าจะคีย์บอร์ดแล้ว ไมโครคอนโทรลเลอร์จะทำการเปลี่ยนข้อมูลที่ป้อนเป็นเลขฐานสอง เพื่อส่งให้กับ FPGA และจะแสดงค่าตัวแปร และค่าช่วงตัวแปรที่เป็นเลขฐาน 10 ที่ทำการตั้งค่าแสดงบนจอ LCD ด้วย

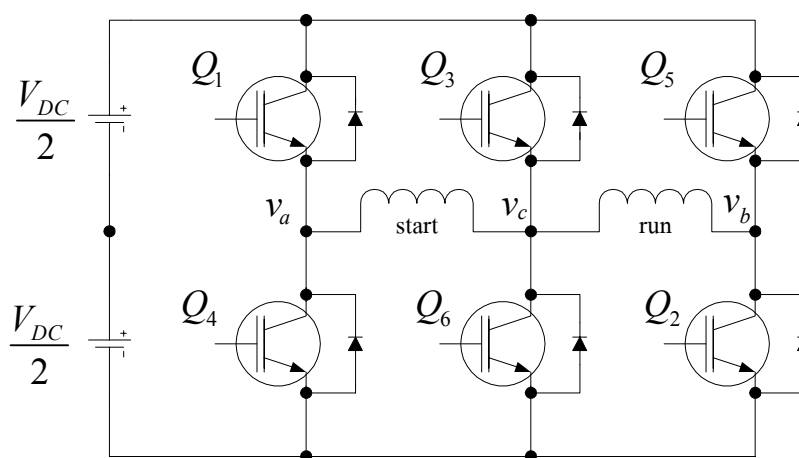


รูปที่ 4.3 แผนภาพแสดงการทำงานของโปรแกรมในส่วน MCS-51

4.3 การออกแบบสัญญาณ SVM PWM ของ FPGA

4.3.1 การขับมอเตอร์เหนี่ยวนำกระแสสลับชนิด 1 เฟสโดยอินเวอร์เตอร์ 3 เฟส

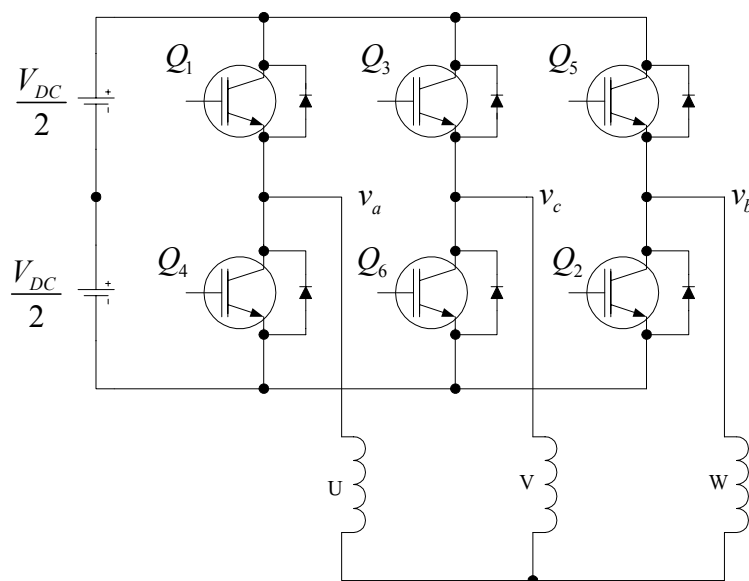
โครงสร้างวงจรอินเวอร์เตอร์ 3 เฟส ที่ใช้ขับมอเตอร์เหนี่ยวนำกระแสสลับชนิด 1 เฟส ดังแสดงในรูปที่ 4.4 โดย แรงดัน v_a คือแรงดันเฟสมีมุม 0° , v_b คือแรงดันเฟสมีมุม 180° , v_c คือแรงดันเฟสมีมุม θ° และ IGBT แต่ละตัวถูกควบคุมด้วยสัญญาณ SVM PWM จาก FPGA โดยชุดบน Q_1, Q_3, Q_5 คือ a, b, c และชุดล่าง Q_4, Q_6, Q_2 คือ $\bar{a}, \bar{b}, \bar{c}$



รูปที่ 4.4 วงจรขับอินเวอร์เตอร์ สำหรับมอเตอร์เหนี่ยวนำกระแสสลับ 1 เฟส

4.3.2 การขับมอเตอร์เหนี่ยวนำกระแสสลับชนิด 3 เฟสโดยอินเวอร์เตอร์ 3 เฟส

โครงสร้างวงจรอินเวอร์เตอร์ 3 เฟส ที่ใช้ขับมอเตอร์เหนี่ยวนำกระแสสลับชนิด 3 เฟส ดังแสดงในรูปที่ 4.5 โดย แรงดัน v_a คือแรงดันเฟสมีมุม 0° , v_b คือแรงดันเฟสมีมุม 120° , v_c คือแรงดันเฟสมีมุม 240° และ IGBT แต่ละตัวถูกควบคุมด้วยสัญญาณ SVM PWM จาก FPGA โดยชุดบน Q_1, Q_3, Q_5 คือ a, b, c และชุดล่าง Q_4, Q_6, Q_2 คือ $\bar{a}, \bar{b}, \bar{c}$

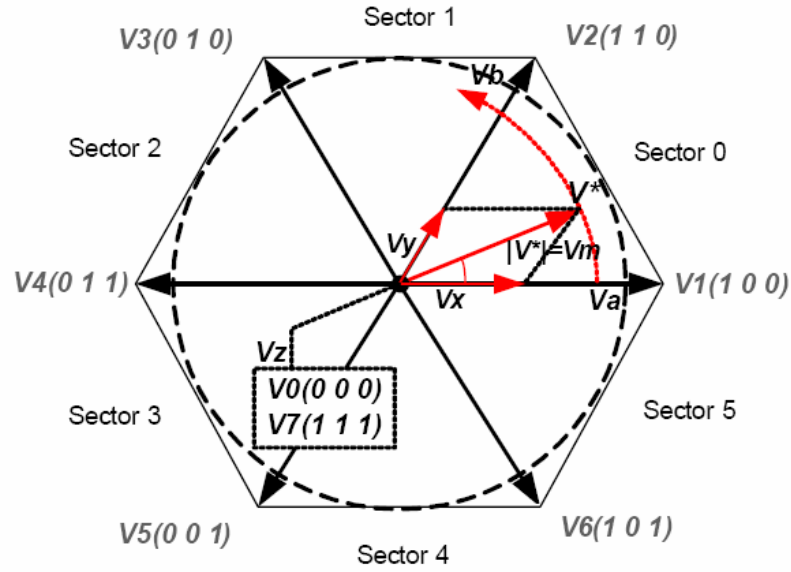


รูปที่ 4.5 วงจรขับเคลื่อนอินเวอร์เตอร์ สำหรับมอเตอร์เหนี่ยวนำกระแสสลับ 3 เฟส

4.3.3 หลักการควบคุมแบบ Vector Modulation

หลักการทำงานของ Space vector PWM อธิบายได้โดย เวกเตอร์แรงดัน ($V_0, \dots, V_7$) แสดงในรูปที่ 4.6 ซึ่งประกอบไปด้วยเวกเตอร์ที่มีค่าจำนวนหกเวกเตอร์ ($V_1, \dots, V_6$) และสองเวกเตอร์ศูนย์ (V_0, V_7) ซึ่งยังคงใช้ในส่วนของการสวิตช์ (T_s) พื้นฐานของเทคนิคการควบคุมแบบ Vector Modulation เป็นการการบวกของเวกเตอร์ศูนย์กับเวกเตอร์อีก 2 ตัวใกล้เคียง โดยจะได้ผลเป็นเวกเตอร์ตัวใหม่ V^* ดังแสดงในรูปที่ 4.6 โดยค่าของ V^* จะมีค่าคงที่ตลอดช่วงเวลาการสุ่ม T

ตัวอย่างเช่นเวกเตอร์ V^* ใน Sector 0 จะประกอบไปด้วยเวกเตอร์ศูนย์ V_z และเวกเตอร์ที่มีค่า V_x, V_y ซึ่งเป็นองค์ประกอบของ V_a และ V_b ตามลำดับ ดังนั้นใน Sector 0 จะมีค่าดังนี้ $V_1(100), V_2(110)$ โดยตัวเลข 1 และ 0 แสดงสถานการณ์ เปิด ปิดตามลำดับ ของชุด IGBT ชุดบน Q_1, Q_3, Q_5 ของวงจรอินเวอร์เตอร์ 3 เฟส



รูปที่ 4.6 เวกเตอร์แรงดันไฟฟ้าที่สร้างโดยอินเวอร์เตอร์

เวกเตอร์ $\mathbf{V}^*$ อธิบายได้ดังสมการ

$$\mathbf{V}^* = \mathbf{V}_a \frac{ta}{T} + \mathbf{V}_b \frac{tb}{T} + \mathbf{V}_z \frac{tz}{T} \quad (4.1)$$

โดยที่ ta, tb, tz เป็นเวลาในการสร้างสัญญาณเวลาการสวิตช์เวกเตอร์แรงดัน $\mathbf{V}_a, \mathbf{V}_b, \mathbf{V}_z$ ผลรวมของเวลาจะเป็นคาบเวลาการสวิตช์

$$ta + tb + tz = T = \frac{T_s}{2} \quad (4.2)$$

เมื่อ $\mathbf{V}_z = 0$ และ $\mathbf{V}^* = \mathbf{V}_x + \mathbf{V}_y$ สมการของ $\mathbf{V}^*$ เขียนใหม่ได้เป็น

$$\mathbf{V}^* = \mathbf{V}_x + \mathbf{V}_y = (\mathbf{V}_a \frac{ta}{T} + \mathbf{V}_b \frac{tb}{T}) \quad (4.3)$$

ดังนั้น

$$ta = \frac{|\mathbf{V}_x|}{|\mathbf{V}_a|} T, \quad tb = \frac{|\mathbf{V}_y|}{|\mathbf{V}_b|} T \quad (4.4)$$

จากตรีโกณมิติ ความสัมพันธ์ของ $\mathbf{V}_x, \mathbf{V}_y$ สามารถแสดงได้เป็น

$$\mathbf{V}_x = V_m (\cos \alpha - \frac{1}{\sqrt{3}} \sin \alpha) \quad (4.5)$$

$$\mathbf{V}_y = \frac{2}{\sqrt{3}} V_m \sin(\alpha)$$

ถ้า $|\mathbf{V}_a| = |\mathbf{V}_b| = \frac{2}{3} V_{dc}$ [1], เมื่อทำการแทนค่า $\mathbf{V}_x, \mathbf{V}_y$ ลงในสมการ (4.4) ค่าของ ta, tb, tz จะมีค่าดังนี้

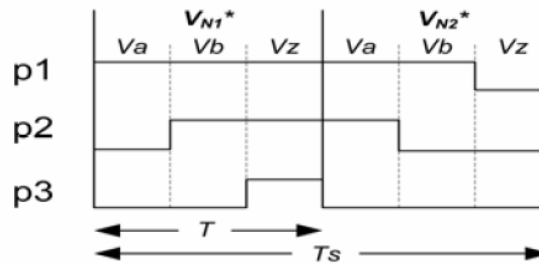
$$\begin{aligned} ta &= \frac{3}{4} M (\cos \alpha - \frac{1}{\sqrt{3}} \sin \alpha) T \\ tb &= \frac{\sqrt{3}}{2} M (\sin \alpha) T \\ tz &= T - ta - tb \end{aligned} \quad (4.6)$$

เมื่อ $M = \text{modulation index} = V_m / (V_{dc} / 2)$

$T = ta + tb + tz = T_s / 2$ ($T_s = \text{switching time}$)

$\alpha = \text{vector angle in each sector}, 0^\circ < \alpha < 60^\circ$

วิธีการลดทอนผลของการสูญเสียในสวิตช์ เป็นวิธีที่ได้รับความนิยมแพร่หลายในการออกแบบวงจรอินเวอร์เตอร์ โดยอาศัยคุณสมบัติของ เวกเตอร์ศูนย์ $\mathbf{V}_0(000)$ และ $\mathbf{V}_7(111)$ จากตัวอย่างในรูปที่ 4.7 เวกเตอร์ศูนย์ $\mathbf{V}_7$ มีค่ามากกว่า $\mathbf{V}_0$ ซึ่งเป็นการใช้สร้าง $\mathbf{V}_{N1}^*$ ใน Sector 0 โดยสังเกตได้ว่าจะมีเฉพาะ P3 ที่สวิตช์จากสภาวะลอจิกต่ำเป็นลอจิกสูง

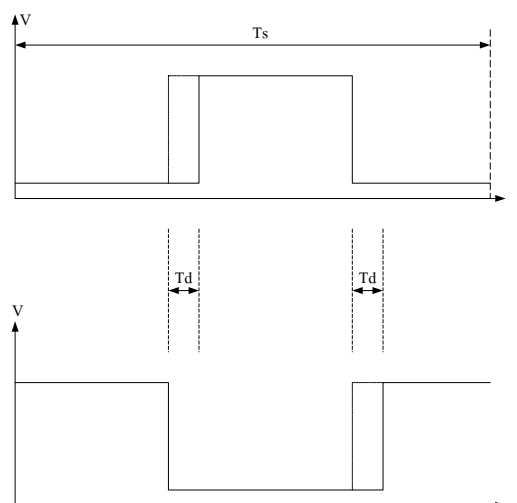


รูปที่ 4.7 กราฟแสดงการสวิตช์

4.4 ออกแบบและพัฒนาส่วนซอฟต์แวร์ควบคุมวงจรมอเตอร์

4.4.1 Dead Time

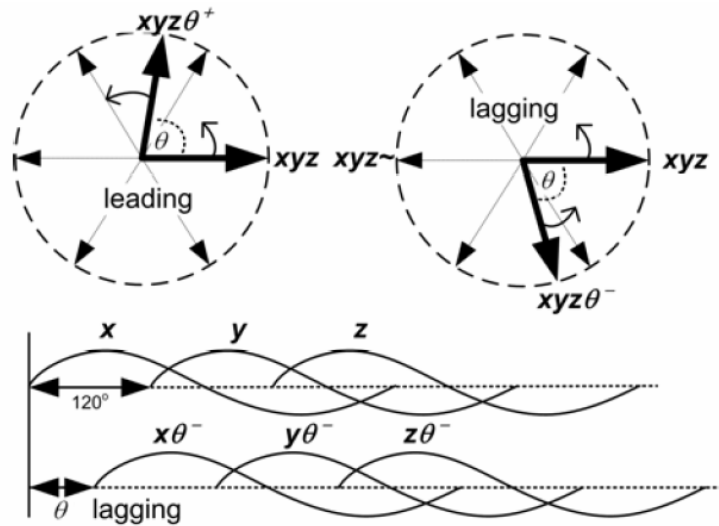
เงื่อนไขการสวิตช์ในทางอุดมคตินั้น สัญญาณควบคุม IGBT ที่กึ่งเดียวกันจะมีค่าตรงกันข้ามกัน (Complementary) เช่นกึ่งสวิตช์ Q_1, Q_4 ในรูปที่ 4.5 แต่ในทางปฏิบัตินั้นเป็นไปได้เนื่องจาก IGBT หรือสวิตช์กำลังทางไฟฟ้านั้นจะมีช่วงเวลาในการทำงาน และหยุดทำงาน ช่วงหนึ่งโดยช่วงเวลาเริ่มการทำงานของสวิตช์จะมีค่าน้อยกว่าช่วงเวลาที่ให้หยุดทำงาน ถ้าหากให้สัญญาณควบคุม สวิตช์ Q_1 หยุดทำงาน และ Q_4 เริ่มทำงานในเวลาเดียวกัน ในทางปฏิบัติจะทำให้มีช่วงเวลาที่สวิตช์ทั้งสองยังคงทำงานอยู่ จะทำให้เกิดการลัดวงจรของแหล่งจ่ายกำลังงานได้จะทำให้เกิดความเสียหายกับสวิตช์ หรือแหล่งจ่ายได้ ดังนั้นเราจึงต้องป้องกันการลัดวงจรดังกล่าวด้วยการหน่วงเวลาของสัญญาณเพื่อให้ Q_1 หยุดทำงาน และเริ่มให้ Q_4 ทำงานหลังจากหน่วงเวลาดังรูปที่ 4.8



รูปที่ 4.8 สัญญาณ SVM PWM ที่มี delay time

4.4.2 ออกแบบและพัฒนาส่วนซอฟต์แวร์ควบคุมวงจรมอเตอร์สำหรับมอเตอร์ 1 เฟส

การสร้างสัญญาณ เลื่อนเฟส SVM PWM นั้นใช้วิธี Vector Modulation จากรูปที่ 4.9 เมื่อ Sector เริ่มต้นของเวกเตอร์ที่เลื่อนเฟส ($xyz\theta$) คำนวณโดยพิจารณาจากเงื่อนไขการนำหน้า และตามหลังเฟส เทียบกับเวกเตอร์เริ่มต้น โดย $xyz, xyz \sim, xyz\theta$ มีค่าเฟสของเวกเตอร์ $0^\circ, 180^\circ, \theta^\circ$ ตามลำดับ



รูปที่ 4.9 Vector modulation และการเลื่อนเฟส

ดังแสดงในรูปที่ 4.10 เป็นฟังก์ชันบล็อกในระบบ ซึ่งทั้งหมดเขียนโดยใช้ภาษามาตรฐานของ Verilog โดยค่าตัวแปรอินพุตที่ต้องการประกอบไปด้วย Dead time (DTIME), ครรชนีการมอดดูเลต (MOD) ระยะเวลาการสุม (Ts) มุมของเฟสเลื่อน (θ°) Memory incremental step (STEP) โดยการออกแบบนี้ ค่าครรชนีการมอดดูเลต และ มุมของเฟสเลื่อน นั้นสามารถหาความสัมพันธ์กับระบบที่ออกแบบ สำหรับค่าตัวแปรอินพุต ซึ่งได้แก่คือ Ts , STEP, dead time สามารถคำนวณได้จาก

$$T_s = \frac{F_{clk}}{2 \times F_s} \quad (4.7)$$

$$STEP = \frac{F_{fund} \times D_{mem} \times 6}{2 \times F_s}$$

$$td = \frac{2 \cdot dtime}{f_{clk}}$$

เมื่อ T_s = คาบเวลาของความถี่การสวิตช์

F_{clk} = สัญญาณนาฬิกาของ FPGA

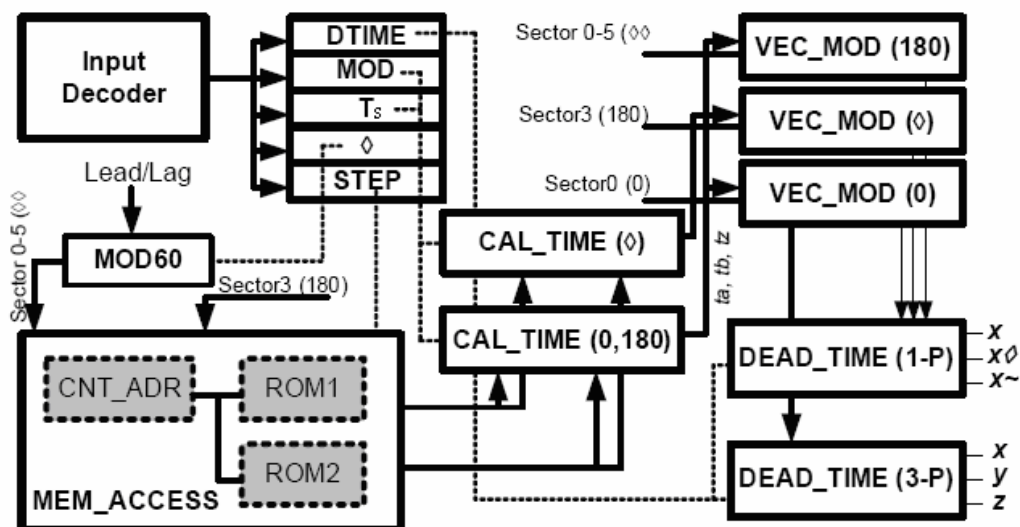
F_s = ความถี่การสวิตช์

F_{fund} = ความถี่หลักมูลของไฟฟ้าสลับที่จ่ายให้กับมอเตอร์

D_{mem} = ความละเอียดของ ROM = 512 ระดับ (9 Bit)

td = ช่วงเวลาของ dead time

$dtime$ = ตัวแปรสำหรับปรับค่า dead time โดยมีค่าช่วง 0-100



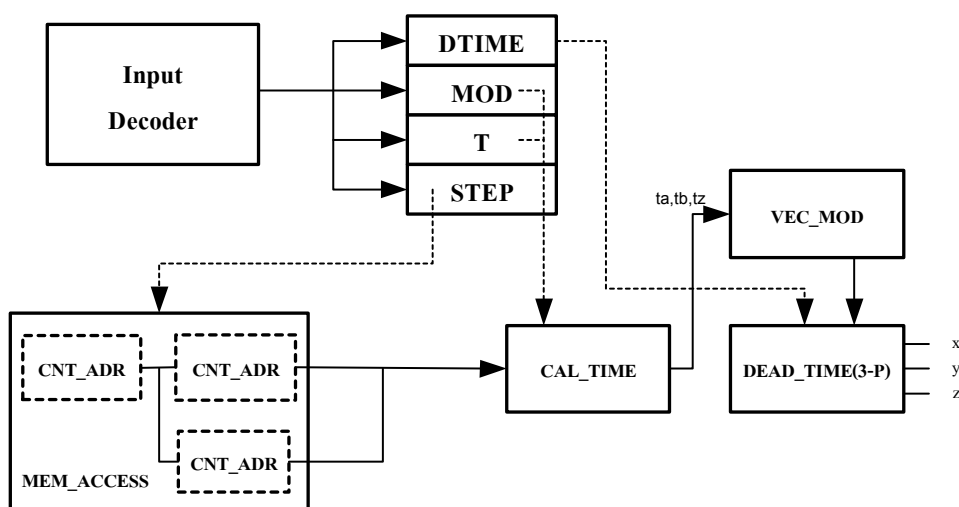
รูปที่ 4.10 ฟังก์ชันบล็อกในการออกแบบระบบ

MOD60 ใช้ในการหา Start sector ของเวกเตอร์เฟสเลื่อน ซึ่งขึ้นอยู่กับค่า Lag/Lead ที่กำหนดโดยตัวแปร θ เมื่อทราบ Start sector แล้วก็จะทำการหาค่า ตำแหน่ง Address ที่มีความสัมพันธ์กับค่า STEP ซึ่ง ROM1 และ ROM2 จะเก็บค่า $\frac{3}{4}(\cos \alpha - \frac{1}{\sqrt{3}} \sin \alpha)$ และ $\frac{\sqrt{3}}{2} \sin(\alpha)$ ตามลำดับ ($0 < \alpha < 60$) โดยค่าคงที่ ที่ได้จะนำไปใช้ในการคำนวณ ta, tb, tz ตามสมการที่ (4.6) หน่วยความจำจะส่งค่าของเวกเตอร์ $xyz, xyz \sim, xyz\theta$ โดย Multiplex มาเก็บ Address จากนั้น CAL_TIME จะทำการคำนวณค่า ta, tb, tz แยกกันสำหรับแต่ละเวกเตอร์ $xyz, xyz \sim, xyz\theta$ ต่อจากนั้น VEC_MOD ทั้งสามส่วน จะทำการสร้างรูปแบบสัญญาณ SVM PWM สำหรับแต่ละเฟส สุดท้ายก่อนจะส่งสัญญาณ SVM PWM ไปควบคุม transistor ในวงจรอินเวอร์เตอร์ 3 เฟส จะทำการแทรก Dead time เข้าไปเพื่อป้องกันการลัดวงจร ในรูปที่ 4.10 สัญญาณ SVM PWM (x, y, z) ใช้ควบคุมมอเตอร์ 3 เฟส และสัญญาณ SVM PWM ($x, x\theta, x \sim$) ใช้ควบคุมมอเตอร์ 1 เฟส

4.4.3 ออกแบบและพัฒนาส่วนซอฟต์แวร์ควบคุมวงจรอินเวอร์เตอร์สำหรับมอเตอร์ 3 เฟส

จากหลักการของวิธี SVM PWM จะเห็นได้ว่าการสร้างเวกเตอร์รับที่เกิดจากเวกเตอร์แรงดันสองตัวจะทำให้สามารถควบคุมแรงดัน 3 เฟสได้โดยการแปลงเวกเตอร์ให้เป็นคาบเวลาในการสร้างสัญญาณ SVM PWM โดยจะได้สัญญาณควบคุมแรงดัน 3 เฟส ได้พร้อมกัน โดยการทำงานในส่วนของการสร้างสัญญาณ SVM PWM สำหรับมอเตอร์ 3 เฟสแสดงดังรูปที่ 4.11 การทำงานคล้ายคลึงกับโปรแกรมในส่วนของการควบคุมมอเตอร์ 1 เฟส แตกต่างกันเพียงไม่ต้อง

สร้างสัญญาณ SVM PWM 3ชุด แบบใน 1เฟส โดยอธิบายการทำงานของโปรแกรมในระบบ3เฟส ได้ดังนี้ ในส่วนของ Input Decoder ใช้สำหรับรับค่าตัวแปร และค่าช่วงตัวแปรจาก ไมโครคอนโทรลเลอร์ และทำการเข้ารหัสเพื่อปรับตัวแปรที่ต้องการโดย ตัวแปร DTIME ใช้ สำหรับการปรับค่า Deadtime ในส่วนของบล็อก DEAD_TIME(3-P) เพื่อแทรก delay time เข้าไป กับสัญญาณ SVM PWM ในส่วนของตัวแปร MOD ใช้สำหรับปรับค่า modulation index ในส่วน ของตัวแปร Ts ใช้ในการปรับค่าความถี่ในการสวิตช์โดยเป็นการปรับคาบเวลาของสัญญาณที่ได้มา จากบล็อก MEM_ACCESS ส่วนตัวแปร STEP ใช้ในการปรับค่าความถี่ของมอเตอร์และความ ละเอียดของมุมที่เวกเตอร์เลื่อนโดย บล็อก MEM_ACCESS มีหน้าที่ในการคำนวณค่าคงที่ และการ หาค่า ตำแหน่งAddress ที่มีความสัมพันธ์กับค่า STEP ซึ่งROM1 และ ROM2 จะเก็บ ค่า $\frac{3}{4}(\cos \alpha - \frac{1}{\sqrt{3}} \sin \alpha)$ และ $\frac{\sqrt{3}}{2} \sin(\alpha)$ ตามลำดับ ($0 < \alpha < 60$) โดยค่าคงที่ ที่ได้จะนำไปใช้ในการ คำนวณ t_a, t_b, t_z ตามสมการที่ (4.6) ซึ่งจะคำนวณโดยบล็อก CAL_TIME จากนั้นสัญญาณ สัญญาณ t_a, t_b, t_z ที่ได้จะส่งเข้าไปยังบล็อก VEC_MOD เพื่อทำการกำหนด Sector เริ่มต้นและ สร้างรูปแบบการสวิตช์ตามแต่ละ Sector ที่กำหนดไว้



รูปที่ 4.11 ฟังก์ชันบล็อกในการออกแบบระบบ 3เฟส