บทที่ 6
การประยุกต์ใช้เพทริเน็ตในส่วนควบคุมเฟสลอจิก

ความนำ

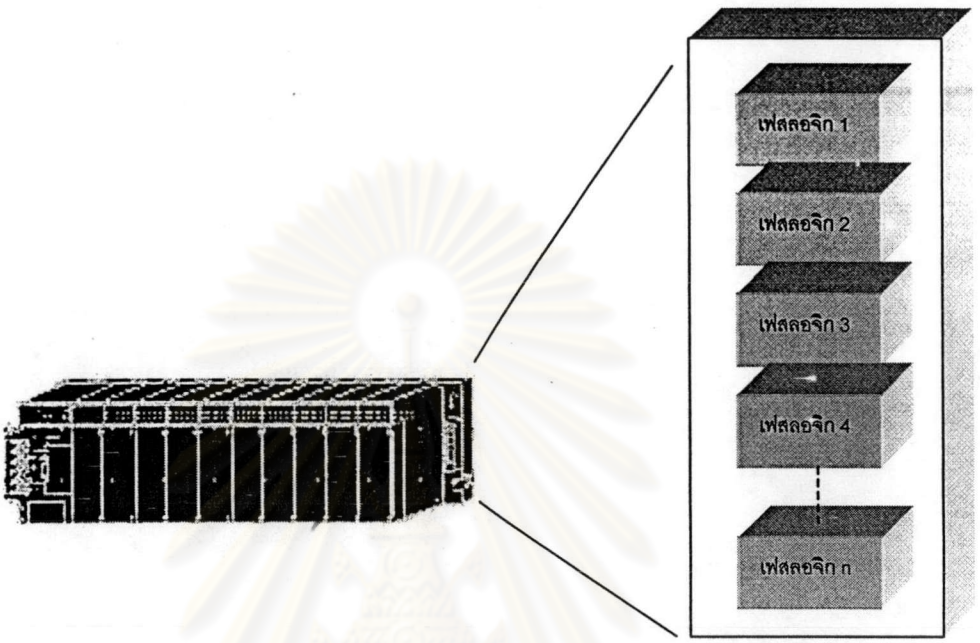
ในการควบคุมการผลิตในโรงงานอุตสาหกรรมนั้น การควบคุมส่วนใหญ่เป็นการควบคุมลำดับการทำงานของอุปกรณ์ต่างๆ ในกระบวนการ ตัวควบคุมลำดับที่นิยมใช้กันแพร่หลายในอุตสาหกรรมคือเครื่องควบคุมแบบโปรแกรมได้ ลักษณะการทำงานของเครื่องควบคุมแบบโปรแกรมได้นั้นจะทำงานตามเงื่อนไขต่างๆ ตามที่กำหนดไว้โดยเก็บอยู่ในหน่วยความจำภายใน การเปลี่ยนแปลงขั้นตอนการทำงานต่าง ๆ ของระบบสามารถทำได้โดยการเปลี่ยนโปรแกรมการเขียนโปรแกรมที่อยู่ภายใน ภาษาที่นิยมใช้กันคือภาษาแลดเดอร์ (Ladder Logic Diagram) ซึ่งเป็นภาษาที่มีรูปแบบคล้ายการเขียนวงจรไฟฟ้า ในปัจจุบันขั้นตอนการผลิตมีขนาดใหญ่และซับซ้อนมากขึ้น ทำให้การออกแบบโปรแกรมรวมถึงการเปลี่ยนแปลง แก้ไข ภาษาแลดเดอร์ทำได้ยากขึ้น ที่ผ่านมาเพทริเน็ตเป็นเครื่องมือที่ได้รับความนิยมในการวิเคราห์ระบบไม่ต่อเนื่อง เนื่องจากเพทริเน็ตสามารถใช้ในการอธิบายเหตุการณ์ต่างๆ ได้อย่างดี ทำให้มีความพยายามออกแบบโปรแกรมภาษาแลดเดอร์โดยใช้เพทริเน็ต ทำให้สามารถเขียนโปรแกรมสำหรับควบคุมลำดับได้ง่ายขึ้น

จากมาตรฐานของ ISA – S 88.01-1995 ได้มีการจัดแบ่งลำดับขั้นการทำงานของระบบเป็นลำดับขั้นต่างๆ เพื่อให้เกิดความสะดวกในการแก้ไขสูตรการผลิต ซึ่งทำให้โปรแกรมภายในเครื่องควบคุมแบบโปรแกรมได้มีลักษณะเป็นโมดูลทำให้เกิดความยืดหยุ่นในการออกแบบโปรแกรม ในหัวข้อนี้ได้กล่าวถึงการประยุกต์ใช้เพทริเน็ตในการออกแบบการออกแบบส่วนควบคุมระดับเฟสสำหรับเครื่องควบคุมแบบโปรแกรมได้ซึ่งเรียกว่าเฟสลอจิก [10] ทำให้สามารถวิเคราะห์ระบบที่ต้องการออกแบบได้ และการออกแบบโปรแกรมแลดเดอร์มีลักษณะเป็นขั้นตอนวิธีที่แน่นอน ง่ายต่อการแก้ไข ปรับปรุง ในภายหลัง

การจัดแบ่งการทำงานแบบเฟส

จากมาตรฐานของ ISA – S 88.01-1995 ได้มีการจัดแบ่งลำดับขั้นการทำงานของระบบเป็นลำดับขั้น โดยส่วนที่เล็กที่สุดคือเฟส ซึ่งในแต่ละเฟสคือลำดับการทำงานชุดหนึ่งของอุปกรณ์ ตัวอย่างเช่น ยูนิตผสมวัตถุดิบอาจประกอบด้วยเฟสใส่วัตถุดิบ A เฟสใส่ส่วนผสม เฟสก

วน เฟสให้ความร้อน เฟสปล่อยวัตถุดิบออก ดังนั้นเมื่อเราใช้เครื่องควบคุมแบบโปรแกรมได้เป็นเครื่องควบคุมในระดับนี้แล้ว หมายความว่าภายในเครื่องควบคุมแบบโปรแกรมได้ จะต้องมีโมดูลย่อยของการทำงานแต่ละเฟสเพื่อรอการเรียกใช้งาน แสดงดังรูป

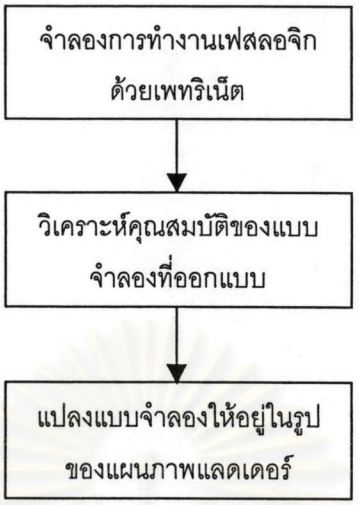


รูปที่ 6.1 รูปแสดงเฟสลอจิกในหน่วยความจำของเครื่องควบคุมแบบโปรแกรมได้

ขั้นตอนการออกแบบเฟสลอจิกด้วยเพทริเน็ต

ขั้นตอนการออกแบบเฟสลอจิกด้วยเพทริเน็ต เริ่มจากการจำลองการทำงานเฟสลอจิกด้วยเพทริเน็ต ในขั้นตอนนี้ต้องทำการกำหนดสถานะและเหตุการณ์ที่เป็นไปได้ทั้งหมดของเฟสลอจิกแล้วเขียนอยู่ในรูปของแบบจำลองเพทริเน็ต เมื่อเขียนแบบจำลองได้แล้วจะนำแบบจำลองมาวิเคราะห์คุณสมบัติต่างๆ ประกอบด้วย Safeness, Boundedness, Liveness เพื่อแสดงว่าเฟสลอจิกที่ได้ออกแบบนั้นสามารถนำมาสร้างเป็นฮาร์ดแวร์ได้ และขั้นตอนสุดท้ายเป็นขั้นตอนการแปลงเพทริเน็ตเป็นแผนภาพแลดเดอร์เพื่อใช้ในเครื่องควบคุมแบบโปรแกรมได้

ขั้นตอนการออกแบบเฟสลอจิกด้วยเพทริเน็ตแสดงดังรูป



รูปที่ 6.2 ขั้นตอนการทำงานของส่วนจัดแบ่งการทำงาน

หลักการทำงานของเฟสลอจิก

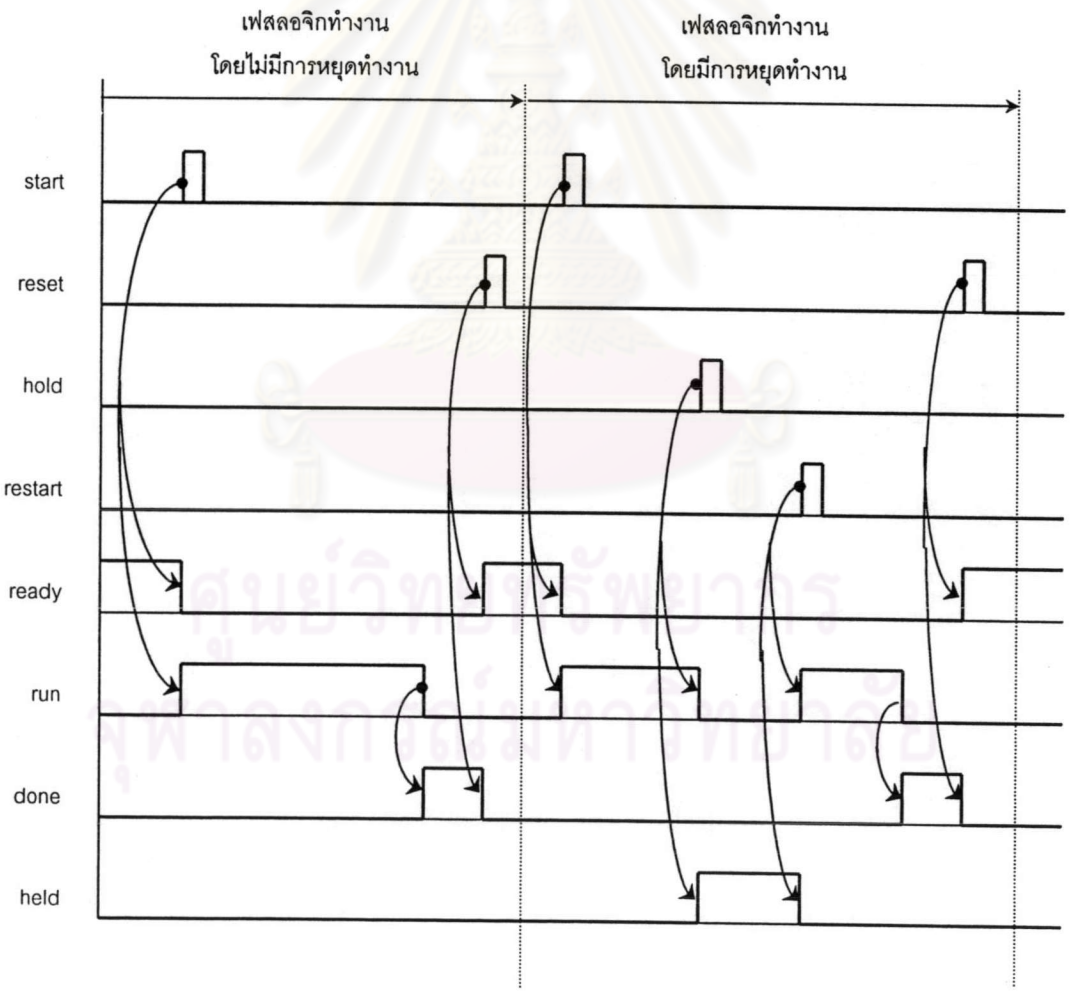
เฟสลอจิกสามารถเขียนจำลองได้ดังรูป

รูปที่ 6.3 แบบจำลองเฟสลอจิก

การทำงานของเฟสลอจิกอธิบายได้ดังนี้ เมื่อเริ่มต้นสัญญาณที่ตำแหน่งขา ready จะอยู่ที่ตำแหน่ง "high" เพื่อแสดงความพร้อมในการทำงาน เมื่อต้องการให้เฟสลอจิกทำงานทำได้โดยการส่งสัญญาณ "high" ไปที่ขา start ของเฟสลอจิก เมื่อเฟสลอจิกได้รับสัญญาณเริ่มทำงานดังกล่าวแล้ว สัญญาณขาออก ready จะอยู่ที่ตำแหน่ง "low" และ สัญญาณขาออก run จะอยู่ที่ตำแหน่ง "high" จนกระทั่งเมื่อเฟสลอจิกทำงานเสร็จสิ้นแล้ว สัญญาณที่ตำแหน่ง run จะกลับสู่สถานะ "low" และส่งสัญญาณ "high" ออกไปที่ตำแหน่ง done เพื่อแสดงสถานะเสร็จสิ้นการ

ทำงาน เมื่อต้องการสั่งให้เฟสลอจิกเริ่มทำงานอีกครั้งหนึ่งสามารถทำได้โดยการส่งสัญญาณ "high" ไปที่ขา reset ทำให้เฟสลอจิกอยู่ในสถานะพร้อมทำงานอีกครั้งหนึ่ง

อย่างไรก็ตาม ในขณะที่เฟสลอจิกกำลังทำงาน (สัญญาณขาออก run จะอยู่ที่ตำแหน่ง "high") อาจมีเหตุการณ์ที่จะต้องให้เฟสลอจิกหยุดทำงานชั่วคราว ทำได้โดยการส่งสัญญาณ "high" เข้าที่ขา hold เมื่อเฟสลอจิกได้รับสัญญาณดังกล่าว เฟสลอจิกจะหยุดทำงานชั่วคราวโดยการให้สัญญาณขาออก run จะอยู่ที่ตำแหน่ง "low" และแสดงสถานะการหยุดทำงานชั่วคราวที่ด้วยการให้สัญญาณขาออก held จะอยู่ที่ตำแหน่ง "high" ในขณะนี้เมื่อต้องการให้เฟสลอจิกได้รับสัญญาณ "high" ที่ขา restart เฟสลอจิกจะกลับมาทำงานอีกครั้งหนึ่ง (สัญญาณ - ขาออก run จะอยู่ที่ตำแหน่ง "high") จนกระทั่งเฟสลอจิกทำงานเสร็จสิ้น แผนภาพเวลาแสดงการทำงานแสดงได้ดังรูป



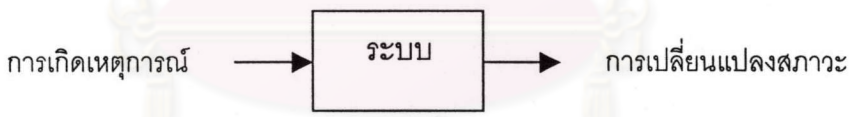
รูปที่ 6.4 แผนภาพเวลาแสดงการทำงานของเฟสลอจิก

การจำลองเฟสลอจิกโดยใช้เพทริเน็ต

จากลักษณะการทำงานที่แสดงในแผนภาพเวลาการทำงานเห็นว่า เมื่อมองเฟสลอจิกในรูปแบบของวงจรไฟฟ้าแล้วสัญญาณขาเข้าที่ใช้ควบคุมระบบคือขอบขาขึ้นของสัญญาณทางไฟฟ้า และระดับสัญญาณขาออกของระบบคือสัญญาณทางไฟฟ้าเช่นกัน แสดงดังรูป

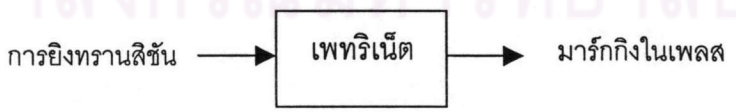
รูปที่ 6.5 เฟสลอจิกอยู่ในรูปแบบของฮาร์ดแวร์

เมื่อเราต้องการแปลงเฟสลอจิกดังกล่าวให้สามารถอธิบายการทำงานโดยใช้เพทริเน็ตจึงจำเป็นต้องจัดรูปแบบความสัมพันธ์เพื่อเชื่อมโยงถึงกัน เมื่อเราเปรียบเทียบการทำงานของเฟสลอจิกในรูปแบบฮาร์ดแวร์กับการทำงานของระบบทั่วไปแล้วเห็นว่า การทำงานของระบบเกิดจากเหตุการณ์ต่างๆ ที่อาจเกิดขึ้นได้ในระบบ และการเกิดเหตุการณ์ดังกล่าวจะทำให้ระบบเกิดการเปลี่ยนแปลงของสภาวะ ดังนั้นเห็นได้ว่าสัญญาณขาออกอาจเทียบได้กับเหตุการณ์และสัญญาณขาออกอาจเทียบได้กับสภาวะของระบบ แสดงดังรูป



รูปที่ 6.6 ลักษณะความสัมพันธ์ของระบบ

และเมื่อแสดงการทำงานของระบบด้วยเพทริเน็ต แสดงดังรูป



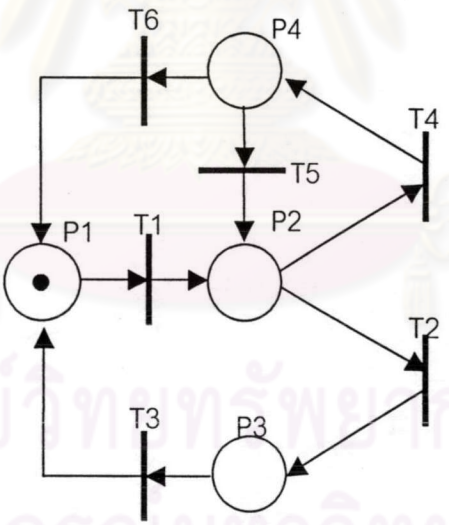
รูปที่ 6.7 ลักษณะความสัมพันธ์ของเพทริเน็ต

จากการทำงานของเฟสลอจิกสามารถสรุป สถานะและเหตุการณ์ที่กำหนด แสดงดังตารางที่ 1

สถานะ		เหตุการณ์	
1.	เฟสลอจิกพร้อมทำงาน	1.	สั่งให้เฟสลอจิกเริ่มทำงาน
2.	เฟสลอจิกกำลังทำงาน	2.	เฟสลอจิกทำงานเสร็จสิ้น
3.	เฟสลอจิกทำงานเสร็จสิ้น	3.	สั่งให้เฟสลอจิกเตรียมการทำงานต่อไป
4.	เฟสลอจิกหยุดรอ	4.	สั่งให้เฟสลอจิกหยุดรอ
		5.	สั่งให้เฟสลอจิกทำงานต่อ
		6.	ยกเลิกการทำงานของเฟสลอจิก

ตารางที่ 6.1 ตารางสถานะและเหตุการณ์ของเฟสลอจิก

เหตุการณ์จากตารางที่ 1 ทำให้เกิดการเคลื่อนที่จากสถานะก่อนและสถานะหลัง ย้ายมาร์กกิงสามารถเขียนรูปแบบการแสดงเพทริเน็ตได้ดังรูป



รูปที่ 6.8 แบบจำลองเพทริเน็ตของเฟสลอจิก

มาร์กกิงเริ่มต้นคือ $M_0 = [1000]'$

ตารางสภาวะก่อนและหลังการเกิดเหตุการณ์ต่างๆ แสดงดังตาราง

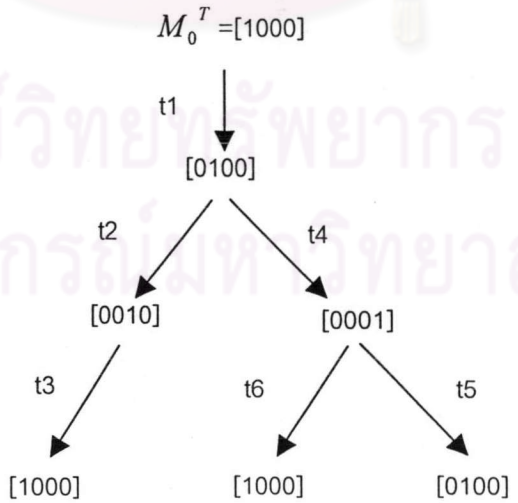
เหตุการณ์		สถานะก่อนการเปลี่ยนแปลง	สถานะหลังการเปลี่ยนแปลง
1.	สั่งให้เฟสลอจิกเริ่มทำงาน	1	2
2.	เฟสลอจิกทำงานเสร็จสิ้น	2	3
3.	สั่งให้เฟสลอจิกเตรียมการทำงานต่อไป	3	1
4.	สั่งให้เฟสลอจิกหยุดรอ	2	4
5.	สั่งให้เฟสลอจิกทำงานต่อ	4	2
6.	ยกเลิกการทำงานของเฟสลอจิก	4	1

ตารางที่ 6.2 ตารางสภาวะก่อนและหลังการเกิดเหตุการณ์

การวิเคราะห์เฟสลอจิกจากเพทริเน็ต

จากที่เราได้จำลองการทำงานของเฟสลอจิกแล้ว ขั้นตอนต่อไปคือการวิเคราะห์คุณสมบัติของเฟสลอจิกว่าสามารถออกแบบเป็นฮาร์ดแวร์ได้หรือไม่ โดยการมีคุณสมบัติของเพทริเน็ตต่างๆ ดังนี้ Safeness, Boundedness, Liveness โดยการพิจารณาจากรีชเอบิลิตีทรี

จากแบบจำลองเพทริเน็ตของเฟสลอจิกสามารถแสดงรีชเอบิลิตี้ทรีได้ดังรูป



รูปที่ 6.9 รีชเอบิลิตีทรีของเฟสลอจิก

จากรีชเอบิลิตีทรีสรุปได้ดังนี้

1. เพทริเน็ตมีคุณสมบัติ Safeness เนื่องจากในทุกๆ เพลสในทุกๆ โนดของรีชเอบิลิตี้ทรีมีค่าไม่มากกว่าหนึ่ง
2. เพทริเน็ตมีคุณสมบัติ Boundedness เนื่องจากในทุกๆ เพลสในทุกๆ โนดของรีชเอบิลิตี้ทรีมีค่าไม่มากกว่าค่าคงที่ค่าหนึ่ง นั่นคือหนึ่งนั่นเอง
3. เพทริเน็ตมีคุณสมบัติ Liveness เนื่องจากในทุกๆ โนดของรีชเอบิลิตี้ไม่เป็นเทอร์มินัลโนด

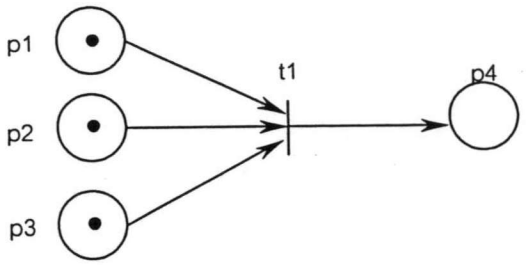
จากข้อสรุปข้อที่ 1 และ 2 ได้ว่าแบบจำลองเฟสลอจิกที่ออกแบบสามารถสร้างเป็นฮาร์ดแวร์ได้ โดยใช้อุปกรณ์แบบเปิด/ปิดเท่านั้น และไม่จำเป็นต้องใช้ตัวนับ เป็นส่วนประกอบของฮาร์ดแวร์ จากข้อสรุปที่ 3 ได้ว่าแบบจำลองเฟสลอจิกที่ออกแบบไม่มีโอกาสที่จะเกิด Deadlock เลย

ตัวอย่างการแปลงเพทริเน็ตเป็นแผนภาพแลดเดอร์

จากการวิเคราะห์แบบจำลองเฟสลอจิกแล้วสรุปว่าเราสามารถนำมาใช้ได้ ในขั้นตอนต่อจากนี้เป็นขั้นตอนการแปลงแบบจำลองเฟสลอจิกให้เป็นแผนภาพแลดเดอร์เพื่อให้สามารถนำไปโปรแกรมเครื่องควบคุมแบบโปรแกรมได้

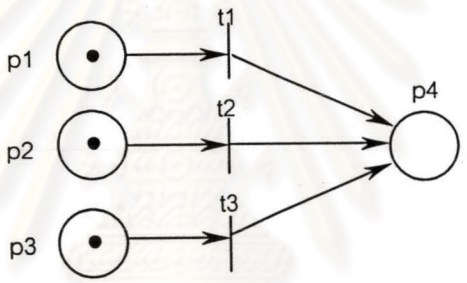
การแปลงรูปจากเพทริเน็ตเป็นแผนภาพแลดเดอร์นั้น เริ่มจากการจัดแบ่งส่วนของอินพุตและเอาต์พุตของระบบออกจากกัน โดยตลอดการทำงานของเพทริเน็ตคือการเปลี่ยนแปลงสัญญาณขาออก ซึ่งการเปลี่ยนแปลงสัญญาณขาออกเป็นผลของสัญญาณขาเข้า เมื่อแสดงการทำงานดังกล่างได้แล้วจึงเริ่มการเขียนแผนภาพแลดเดอร์เพื่อจัดความสัมพันธ์ของอินพุตและเอาต์พุตในสถานะของระบบต่างๆ ดังกล่าว

การเขียนแผนภาพแลดเดอร์นั้น เงื่อนไขสำคัญที่มีความสำคัญในการแสดงส่วนประกอบต่างๆ ในรูปแบบเพทริเน็ตนั้นคือ ลอจิก AND และลอจิก OR สามารถอธิบายความสัมพันธ์ของแผนภาพแลดเดอร์และเพทริเน็ต [16,17] ได้ดังรูป



รูปที่ 6.10 เพทริเน็ตของลอจิก AND

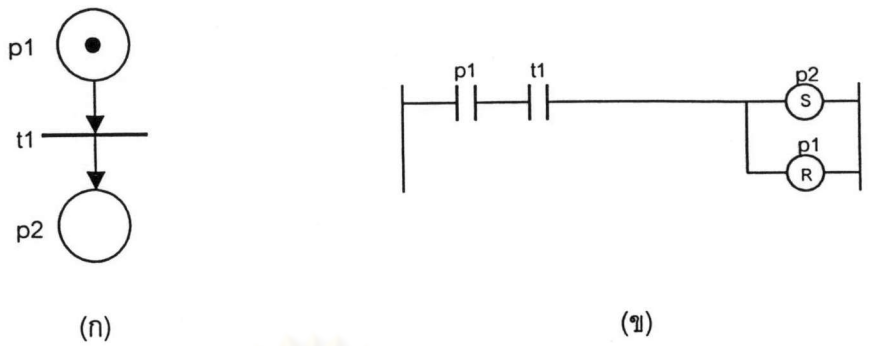
พิจารณาในรูปที่ 6.10 แสดงเพริเน็ตของลอจิก AND เพลส p_4 จะมีมาร์กกิงได้ก็ต่อเมื่อเพลส p_1 และ p_2 และ p_3 ทั้งหมดถูกมาร์กกิงและทรานสิชัน t_1 ถูกยิงทรานสิชัน



รูปที่ 6.11 แผนภาพแลดเดอร์และเพทริเน็ตของลอจิก OR

พิจารณาในรูปที่ 6.11 แสดงเพริเน็ตของลอจิก OR เพลส p_4 จะมีมาร์กกิงได้ก็ต่อเมื่อ ชุดของ p_1,t_1 หรือ p_2,t_2 หรือ p_3,t_3 ชุดใดชุดหนึ่งหรือหลายชุดถูกยิงทรานสิชัน

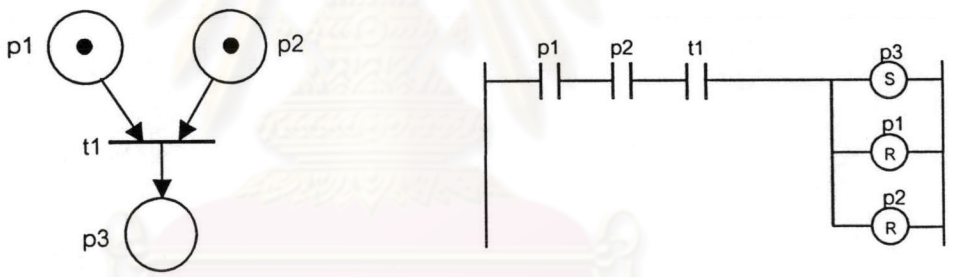
จากเงื่อนไขของ AND และ OR พิจารณาตัวอย่างการแปลงเพทริเน็ตให้เป็นแผนภาพแลดเดอร์แสดงดังนี้



(ก) (ข)

รูปที่ 6.12 ตัวอย่างการแปลงเพทริเน็ตเป็นแผนภาพแลดเดอร์ (1)

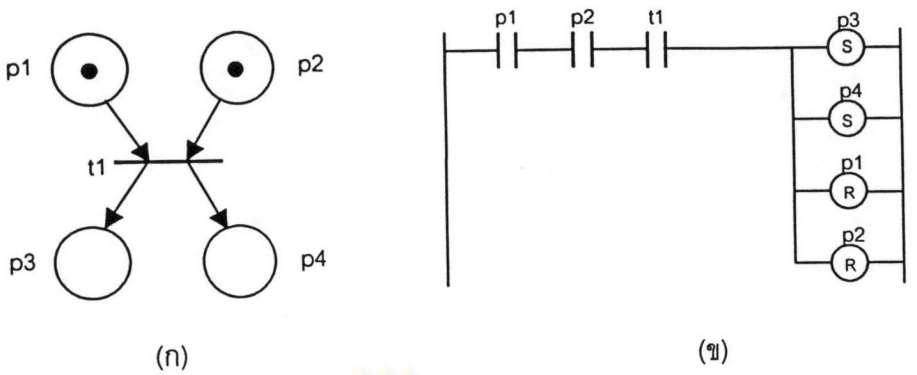
จากรูปที่ 6.12 (ก) เพลส p_2 จะถูกมาร์กได้ก็ต่อเมื่อทรานสิชัน t_1 ถูกอีนาเบิลและเกิดการยิงทรานสิชัน t_1 เมื่อพิจารณาทรานสิชัน t_1 เห็นว่าจะอีนาเบิลได้ก็ต่อเมื่อที่เพลส p_1 มีโทเค็นอยู่ภายใน 1 โทเค็น



(ก) (ข)

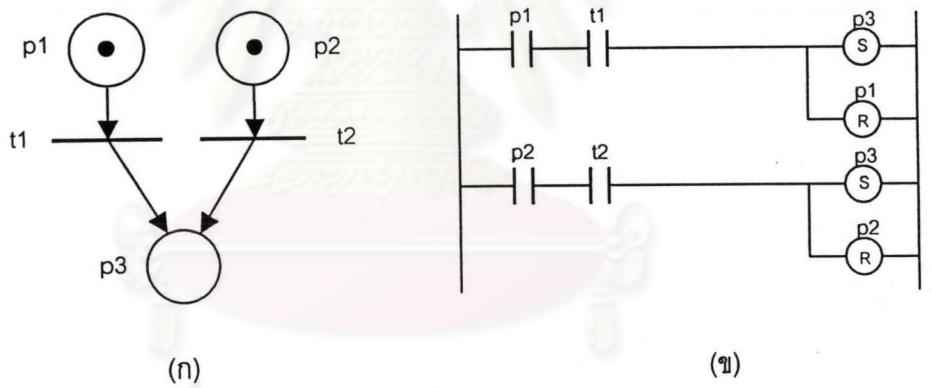
รูปที่ 6.13 ตัวอย่างการแปลงเพทริเน็ตเป็นแผนภาพแลดเดอร์ (2)

จากรูปที่ 6.13 (ก) เพทริเน็ตมีลักษณะการทำงานคล้ายรูปที่ 6.12 แต่มีเงื่อนไขของเพลส p_2 กล่าวคือเพลส p_3 จะถูกมาร์กได้ก็ต่อเมื่อทรานสิชัน t_1 ถูกอีนาเบิลและเกิดการยิงทรานสิชัน t_1 เมื่อพิจารณาทรานสิชัน t_1 เห็นว่าจะอีนาเบิลได้ก็ต่อเมื่อที่เพลส p_1 และเพลส p_2 มีโทเค็นอยู่ภายในเพลสละ 1 โทเค็น



รูปที่ 6.14 ตัวอย่างการแปลงเพทริเน็ตเป็นแผนภาพแลดเดอร์ (3)

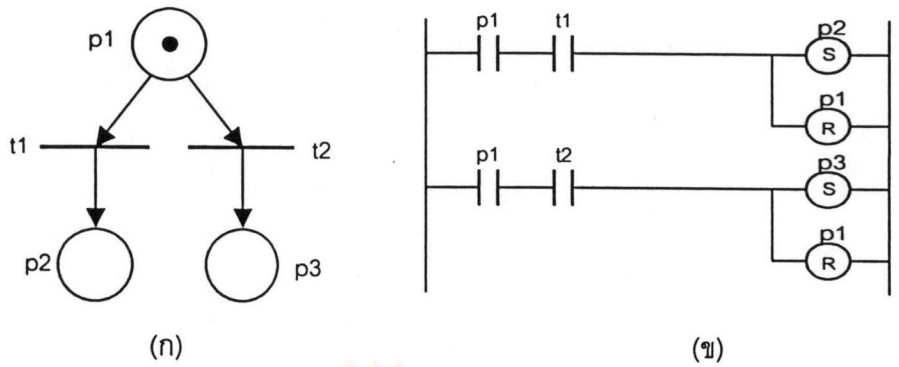
จากรูปที่ 6.14 (ก) เมื่อทรานสิชัน t_1 ถูกอีนาเบิลและเกิดการยิงทรานสิชัน t_1 จะทำให้โทเค็นปรากฏที่เพลส p_3 และเพลส p_4 และทรานสิชัน t_1 เห็นว่าจะอีนาเบิลได้ก็ต่อเมื่อที่เพลส p_1 และเพลส p_2 มีโทเค็นอยู่ภายในเพลสละ 1 โทเค็น



รูปที่ 6.15 ตัวอย่างการแปลงเพทริเน็ตเป็นแผนภาพแลดเดอร์ (4)

จากรูปที่ 6.15 (ก) เพลส p_3 จะถูกมาร์กได้ก็ต่อเมื่อทรานสิชัน t_1 ถูกอีนาเบิลและเกิดการยิงทรานสิชัน t_1 หรือทรานสิชัน t_2 ถูกอีนาเบิลและเกิดการยิงทรานสิชัน t_2

เมื่อพิจารณาทรานสิชัน t_1 เห็นว่าจะอีนาเบิลได้ก็ต่อเมื่อที่เพลส p_1 มีโทเค็นอยู่ภายใน 1 โทเค็น และทรานสิชัน t_2 เห็นว่าจะอีนาเบิลได้ก็ต่อเมื่อที่เพลส p_2 มีโทเค็นอยู่ภายใน 1 โทเค็น

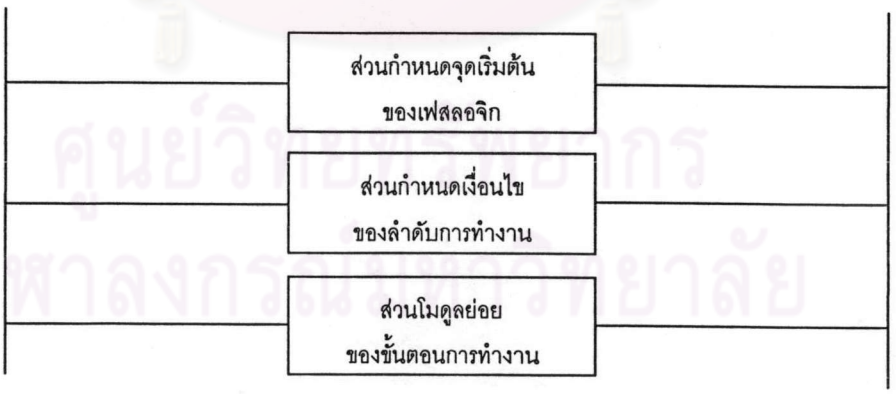


รูปที่ 6.16 ตัวอย่างการแปลงเพทริเน็ตเป็นแผนภาพแลดเดอร์ (5)

จากรูปที่ 6.16 (ก) เพลส p_2 จะถูกมาร์กได้ก็ต่อเมื่อทรานสิชัน t_1 ถูกอีนาเบิลและเกิดการยิงทรานสิชัน t_1 เพลส p_3 จะถูกมาร์กได้ก็ต่อเมื่อทรานสิชัน t_2 ถูกอีนาเบิลและเกิดการยิงทรานสิชัน t_2 และเมื่อพิจารณาทรานสิชัน t_1 และ t_2 เห็นว่าจะอีนาเบิลได้ก็ต่อเมื่อที่เพลส p_1 มีโทเค็นอยู่ภายใน 1 โทเค็น ซึ่งอาจมองเป็นการจัดแบ่งทรัพยากรก็ได้คือขณะใดขณะหนึ่ง ระบบสามารถอยู่ในสถานะ p_2 หรือ p_3 ได้เท่านั้น

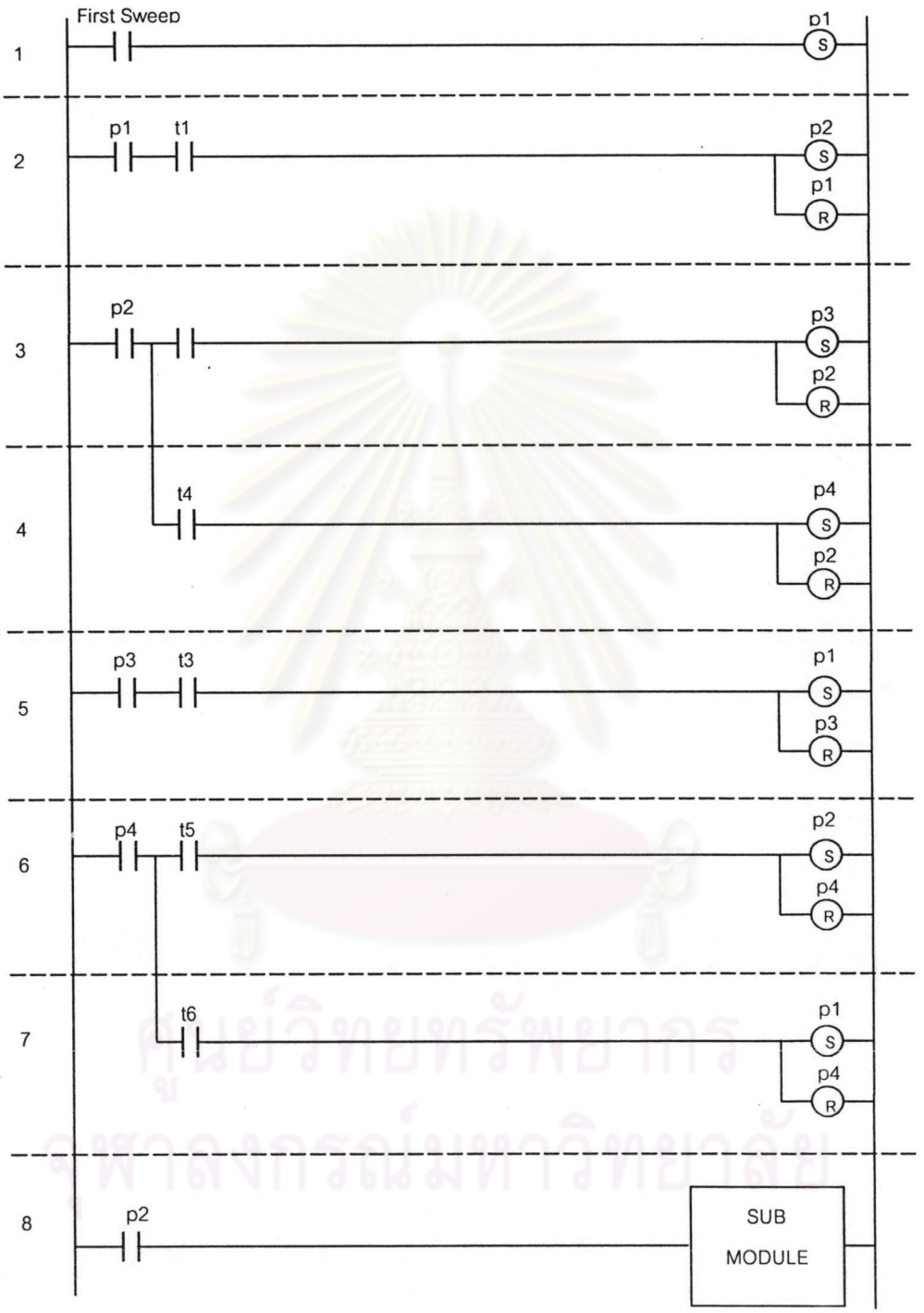
แผนภาพแลดเดอร์ของเฟสลอจิก

การแปลงเพทริเน็ตที่อธิบายการทำงานของเฟสลอจิกให้อยู่ในรูปของแผนภาพแลดเดอร์นั้น ส่วนต่างๆ ของแผนภาพแลดเดอร์ที่แปลงมาสามารถแบ่งออกเป็น 3 ส่วนดังรูป



รูปที่ 6.17 ส่วนต่างๆ ของแผนภาพแลดเดอร์ของเฟสลอจิก

จากแบบจำลองเพทริเน็ตของเฟสลอจิก สามารถออกแบบแผนภาพแลดเดอร์ได้ดังรูป



รูปที่ 6.18 แผนภาพแลดเดอร์ของเฟสลอจิก

ส่วนที่ 1 ส่วนกำหนดจุดเริ่มต้นของเฟสลอจิก ประกอบด้วยแถวที่ 1

<u>แถวที่ 1</u> มาร์กกิงเริ่มต้นของระบบคือ $M_0 = [1000]^t$ คือมีโทเค็นอยู่ที่เพลส p_1 ดังนั้นเมื่อเขียนในรูปของแผนภาพแลดเดอร์จึงออกแบบให้การทำงานรอบแรกเป็นการเซตเอาต์พุตรีเลย์ p_1

ส่วนที่ 2 ส่วนกำหนดเงื่อนไขของลำดับการทำงาน ประกอบด้วยแถวที่ 2 ถึงแถวที่ 7

<u>แถวที่ 2</u> โทเค็นจะปรากฏโทเค็นในเพลส p_2 ได้ก็ต่อเมื่อมีโทเค็นอยู่ที่เพลส p_1 และทรานสิชัน t_1 ถูกยิง และหลังจากทรานสิชัน t_1 ถูกยิงทำให้โทเค็นเคลื่อนที่ออกจากเพลส p_1 ดังนั้นเมื่อเขียนในรูปของแผนภาพแลดเดอร์จึงออกแบบให้การเซตเอาต์พุตรีเลย์ p_2 และรีเซตเอาต์พุตรีเลย์ p_1 ขึ้นกับสวิทช์ p_1 และ t_1

<u>แถวที่ 3</u> โทเค็นจะปรากฏโทเค็นในเพลส p_3 ได้ก็ต่อเมื่อมีโทเค็นอยู่ที่เพลส p_2 และทรานสิชัน t_2 ถูกยิง และหลังจากทรานสิชัน t_2 ถูกยิงทำให้โทเค็นเคลื่อนที่ออกจากเพลส p_2 ดังนั้นเมื่อเขียนในรูปของแผนภาพแลดเดอร์จึงออกแบบให้การเซตเอาต์พุตรีเลย์ p_3 และรีเซตเอาต์พุตรีเลย์ p_2 ขึ้นกับสวิทช์ p_2 และ t_2

<u>แถวที่ 4</u> โทเค็นจะปรากฏในเพลส p_4 ได้ก็ต่อเมื่อมีโทเค็นอยู่ที่เพลส p_2 และทรานสิชัน t_4 ถูกยิง และหลังจากทรานสิชัน t_4 ถูกยิงทำให้โทเค็นเคลื่อนที่ออกจากเพลส p_2 ดังนั้นเมื่อเขียนในรูปของแผนภาพแลดเดอร์จึงออกแบบให้การเซตเอาต์พุตรีเลย์ p_4 และรีเซตเอาต์พุตรีเลย์ p_2 ขึ้นกับสวิทช์ p_2 และ t_4

<u>แถวที่ 5</u> เพทริเน็ตจะกลับไปสู่สถานะเริ่มต้นคือ โทเค็นอยู่ในเพลส p_1 ได้ก็ต่อเมื่อมีโทเค็นอยู่ที่เพลส p_3 และทรานสิชัน t_3 ถูกยิง และหลังจากทรานสิชัน t_3 ถูกยิงทำให้โทเค็นเคลื่อนที่ออกจากเพลส p_3 ดังนั้นเมื่อเขียนในรูปของแผนภาพแลดเดอร์จึงออกแบบให้การเซตเอาต์พุตรีเลย์ p_1 และรีเซตเอาต์พุตรีเลย์ p_3 ขึ้นกับสวิทช์ p_3 และ t_3

<u>แถวที่ 6</u> หลังจากโทเค็นปรากฏที่เพลส p_4 แล้ว เมื่อทรานสิชัน t_5 ถูกยิงจะทำให้โทเค็นปรากฏโทเค็นในเพลส p_2 อีกครั้งหนึ่ง ดังนั้นเมื่อเขียนในรูปของแผนภาพแลดเดอร์จึงออกแบบให้การเซตเอาต์พุตรีเลย์ p_2 และรีเซตเอาต์พุตรีเลย์ p_4 ขึ้นกับสวิทช์ p_4 และ t_5

<u>แถวที่ 7</u> เพทริเน็ตจะกลับไปสู่สถานะเริ่มต้นคือ โทเค็นอยู่ในเพลส p_1 ได้ก็ต่อเมื่อมีโทเค็นอยู่ที่เพลส p_4 และทรานสิชัน t_6 ถูกยิง และหลังจากทรานสิชัน t_6 ถูกยิงทำให้โทเค็นเคลื่อนที่ออกจาก

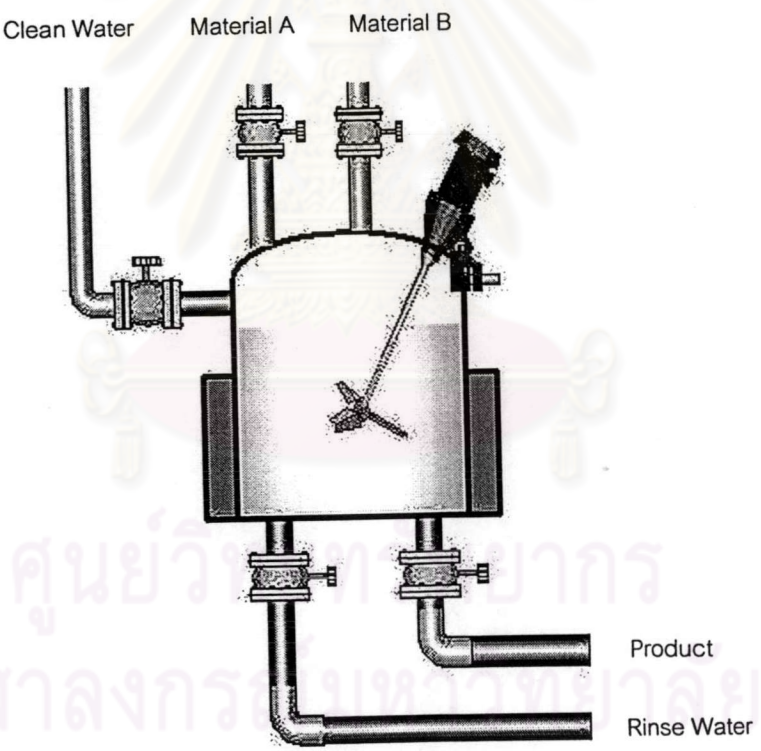
เพลส p_4 ดังนั้นเมื่อเขียนในรูปของแผนภาพแลดเดอร์จึงออกแบบให้การเซตเอาต์พุตรีเลย์ p_1 และรีเซตเอาต์พุตรีเลย์ p_4 ขึ้นกับสวิทช์ p_4 และ t_6

ส่วนที่ 3 ส่วนสับโมดูลย่อยของขั้นตอนการทำงาน ประกอบด้วย แถวที่ 8

<u>แถวที่ 8</u> เมื่อพิจารณาสถานะของระบบในเพลส p_2 เห็นว่าเป็นส่วนการทำงานจริงของระบบ ฉะนั้นจึงมีส่วนของสับโปรแกรมย่อยเพื่อให้ออกแบบการทำงานของเฟสลอจิกได้

ตัวอย่างเฟสลอจิก

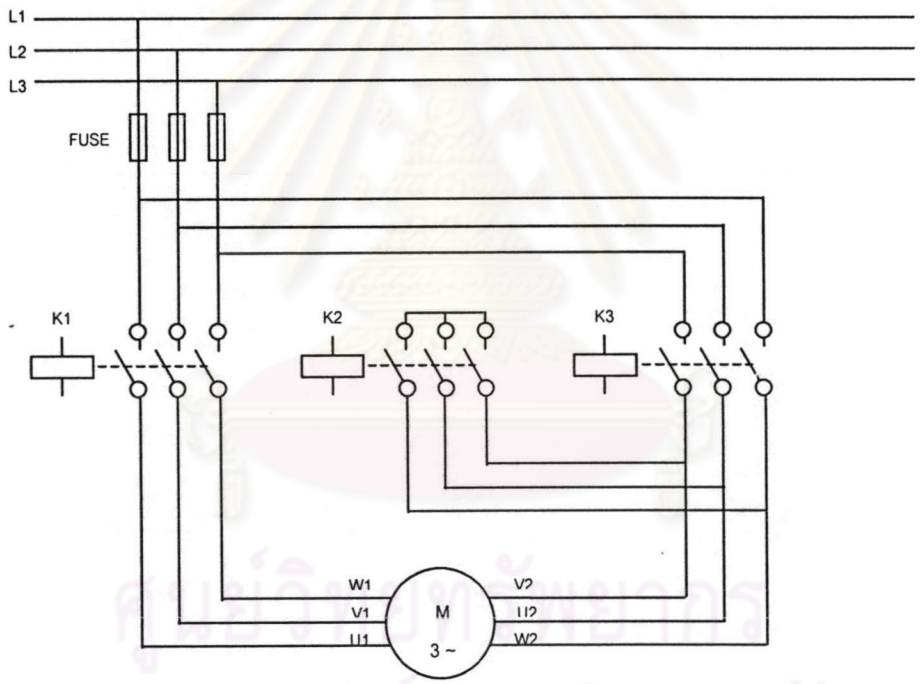
ในหัวข้อนี้เป็นการแสดงตัวอย่างโมดูลย่อยภายในเฟสลอจิก พิจารณาตัวอย่างยูนิตถังกวนดังรูป



รูปที่ 6.19 ยูนิตถังกวน

จากมาตรฐาน ISA-S88.01 ได้มีการแบ่งการทำงานเป็นส่วนย่อยๆ โดยส่วนที่เล็กที่สุดคือเฟส เมื่อพิจารณารูปยูนิตถังกวน อาจแบ่งเป็นเฟสย่อยๆ คือ เฟสกวนส่วนผสม, เฟสใส่วัตถุดิบ A, เฟสใส่วัตถุดิบ B, เฟสปล่อยผลิตภัณฑ์ออก, เฟสล้างยูนิต เป็นต้น จากหลักการทำงานของเฟสลอจิกทำให้การควบคุมการทำงานในแต่ละเฟสย่อยๆ เปรียบเสมือนการทำงานภายในเฟสลอจิก

เมื่อพิจารณาการทำงานของเฟสกวนส่วนผสม โดยสมมุติให้การทำงานของเฟสกวนดังนี้ เมื่อพิจารณาจากกระบวนการจริงแล้วเห็นว่าการทำงานของเฟสกวนส่วนผสมคือการควบคุมทำงานของมอเตอร์สามเฟส ดังนั้นเมื่อมีการสั่งให้เฟสกวนส่วนผสมทำงานคือการสั่งให้มอเตอร์สามเฟสหมุน โดยเริ่มเดินแบบสตาร์-เดลต้า ซึ่งมีวงจรเดินสายกำลัง (Power Wiring) ดังรูปที่ 6.20



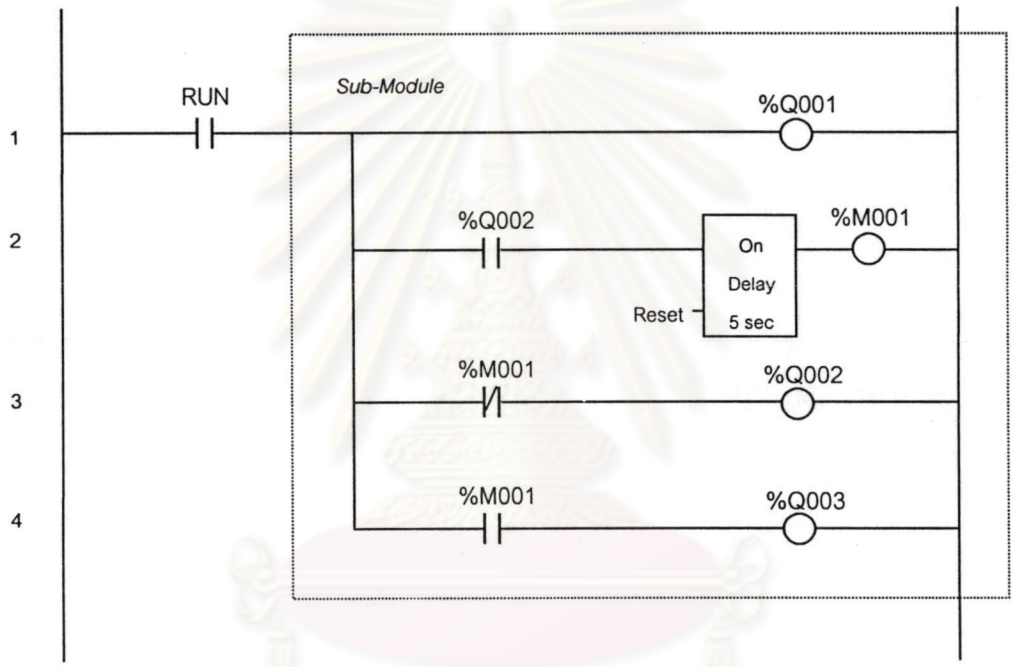
รูปที่ 6.20 การเริ่มเดินมอเตอร์แบบสตาร์-เดลต้า

การทำงานของการเริ่มเดินแบบสตาร์-เดลต้าอธิบายได้ดังนี้ จากรูปเป็นการต่อมอเตอร์สามเฟสเข้ากับคอนแทคเตอร์ K1, K2, K3 เห็นว่าสวิทช์เอาต์พุตของคอนแทคเตอร์ทั้งสามอยู่ในสถานะเปิดวงจร เมื่อต้องการเริ่มเดินมอเตอร์ การต่อขดลวดต้องต่อแบบสตาร์เพื่อลดกระแส

ในขณะสตาร์ต คืออยู่ในสถานะสวิทช์เอาต์พุตของคอนแทคเตอร์ K1 และ K2 อยู่ในสถานะปิดวงจรและสวิทช์เอาต์พุตของคอนแทคเตอร์ K3 อยู่ในสถานะเปิดวงจร เมื่อมอเตอร์หมุนไปแล้วช่วงเวลาหนึ่ง การต่อขดลวดจะเป็นแบบเดลต้า คือสวิทช์เอาต์พุตของคอนแทคเตอร์ K1 และ K3 อยู่ในสถานะปิดวงจรและสวิทช์เอาต์พุตของคอนแทคเตอร์ K2 อยู่ในสถานะเปิดวงจร

กำหนดการต่อคอนแทคเตอร์ K1, K2, K3 เข้ากับเอาต์พุตของ PLC คือ %Q001, %Q002, %Q003 ตามลำดับ

เมื่อเขียนแผนภาพแลดเดอร์ในส่วนโมดูลย่อยของขั้นตอนการทำงาน แสดงได้ดังรูปที่ 6.21



รูปที่ 6.21 แผนภาพแลดเดอร์ในส่วนการเดินมอเตอร์สามเฟส

จากรูป อธิบายการทำงานของแผนภาพแลดเดอร์ได้ดังนี้

แถวที่ 1 เมื่อเริ่มทำงานเอาต์พุต %Q001 จะทำงานทำให้สวิทช์เอาต์พุตของคอนแทคเตอร์ K1 ปิดวงจร เป็นการต่อสายไฟฟ้าทั้งสามเฟสเข้าที่มอเตอร์ที่ยังไม่มีการต่อขดลวดทำให้มอเตอร์ยังไม่เริ่มหมุน

แถวที่ 2 เมื่อเอาต์พุต %Q002 ทำงานแล้ว On Delay Timer จะเริ่มจับเวลา เมื่อเวลาผ่านไปเป็นเวลา 5 วินาที จะทำให้รีเลย์ %M001 ต่อวงจร

แถวที่ 3 ในขณะเริ่มต้นเอาต์พุต %Q002 ต่อวงจรทำให้สวิทช์เอาต์พุตของคอนแทคเตอร์ K2 ปิดวงจร ซึ่งทำให้มอเตอร์อยู่ในสถานะการต่อวงจรแบบสตาร์

แถวที่ 4 รีเลย์ %M001 จะทำให้เอาต์พุต %Q003 ต่อวงจรทำให้สวิทช์เอาต์พุตของคอนแทคเตอร์ K3 ปิดวงจร ซึ่งทำให้มอเตอร์อยู่ในสถานะการต่อวงจรแบบเดลต้า