

บทที่ 2

ภาษาแลคเกอร์สำหรับพีแอลซี และวงจรตรรกะเชิงลำดับ

บทนำ

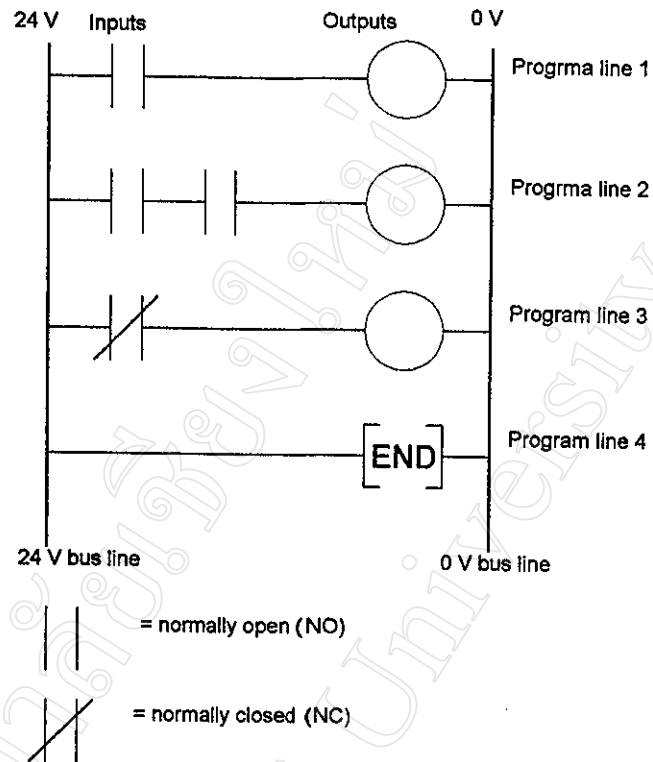
ในการออกแบบระบบใด ๆ ก็ตามเราต้องมีความเข้าใจองค์ประกอบพื้นฐานของระบบนั้นเสียก่อน เพื่อที่จะได้เลือกใช้ทฤษฎีให้เหมาะกับระบบที่ต้องการจะออกแบบขึ้นมามีดังเช่นในงานวิจัยนี้เป็นการออกแบบสร้างระบบซอฟต์แวร์ เพื่อใช้สำหรับการออกแบบโปรแกรมสำหรับพีแอลซีเพื่อใช้ในการควบคุมการทำงานของเครื่องจักร และระบบควบคุมอื่น ๆ ตามต้องการ ดังนั้นในบทนี้จึงต้องกล่าวถึงภาษาแลคเกอร์หรือแลคเกอร์โคอะแกรม การทำงานของพีแอลซีโดยวิธีการโปรแกรมด้วยภาษาแลคเกอร์ การเปรียบเทียบแลคเกอร์โคอะแกรมและวงจรตรรกะ และกล่าวถึงวงจรตรรกะ คุณสมบัติของวงจร รวมถึงการออกแบบวงจร ซึ่งจะได้เป็นพื้นฐานในการออกแบบสร้างระบบซอฟต์แวร์

2.1 แลคเกอร์โคอะแกรม [7]

การโปรแกรมพีแอลซีส่วนมาก จะดำเนินการในลักษณะที่เหมือนกับการวาดวงจรสวิตชิงในรูปแบบของขั้นบันได (Ladder Diagram Format) รูปแบบดังกล่าวมีข้อกำหนดคือ

- 1) วงจรจัดตามลำดับในแนวนอนประกอบด้วย อินพุต (อ้างถึงหน้าสัมผัส) และเอาต์พุต (อ้างถึงคอยล์) ดังแสดงในรูปที่ 2.1
- 2) อินพุตจะต้องอยู่หน้าเอาต์พุตเท่านั้น และต้องอยู่ในรูปแบบของหน้าสัมผัสแบบปกติเปิด (Normally Open หรือ NO) หรือหน้าสัมผัสแบบปกติปิดเท่านั้น (Normally Close หรือ NC)
- 3) วงจรในแต่ละขั้นต้องมีเอาต์พุตอย่างน้อย 1 เอาต์พุตและต้องเป็นเอาต์พุตในพีแอลซีเร็นรีเลย์ของพีแอลซี สัญลักษณ์ใช้เป็น () หรือวงกลม
- 4) ไม่ใช้การเขียนวงจรในแนวตั้ง
- 5) อินพุตและเอาต์พุตมีการกำหนดตัวเลขประกอบ
- 6) ส่วนอื่น ๆ เช่น ตัวตั้งเวลา ตัวนับ และชิฟต์รีจิสเตอร์ (Shift Register) สามารถนำมาใช้ในแลคเกอร์โคอะแกรมได้ (จะกล่าวถึงในหัวข้อต่อไป)

เนื่องจากการวาดวงจรอยู่ในรูปแบบของขั้นบันไดจึงเรียกว่า แลคเกอร์ เส้นในแนวตั้งสองเส้นเรียกว่า บัสไลน์ (Bus Line) ซึ่งเชื่อมต่อเข้ากับแหล่งจ่ายจากตัวอย่างรูปที่ 2.1 เป็นแหล่งจ่ายขนาด 24 โวลท์ และ 0 โวลท์ ในแนวนอนแสดงโปรแกรมไลน์ เอาต์พุตที่อยู่บนโปรแกรมไลน์ทำงานเมื่อหน้าสัมผัสอินพุตเชื่อมต่อกันครบวงจร (เมื่ออินพุตและเอาต์พุตครบวงจรกับแหล่งจ่าย)



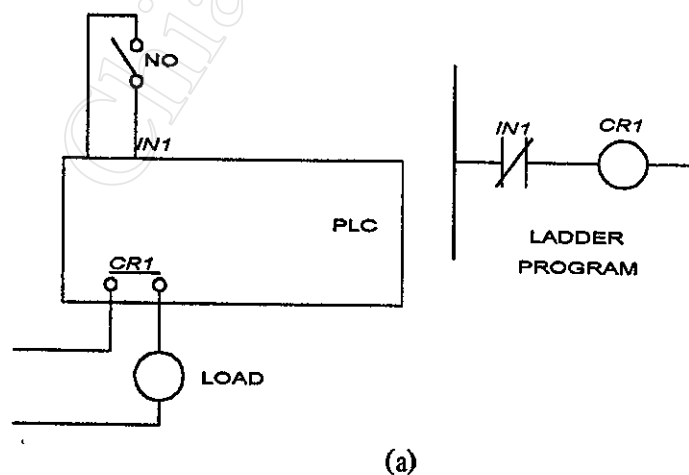
รูปที่ 2.1 แสดงรูปแบบของแลคเคอร์ไคอะแกรม

แลคเคอร์ไคอะแกรมสามารถเปลี่ยนเป็นโปรแกรมที่ประกอบด้วย คำสั่ง (Instruction) และ ข้อมูล (Data) ดังแสดงในตารางที่ 2.1 อธิบายคำสั่งบูลีน (Boolean Instruction) ที่ใช้ในการโปรแกรมพีแอลซี โปรแกรมแลคเคอร์จะถูกเก็บที่หน่วยความจำ (Memory) ของพีแอลซีด้วยรูปแบบ ตำแหน่ง-คำสั่ง-ข้อมูล (Address-Instruction-Data) ตำแหน่งคือตำแหน่งในหน่วยความจำของพีแอลซี คำสั่งและข้อมูลจะถูกเก็บเรียงกันตามลำดับ เริ่มที่ตำแหน่ง 0000

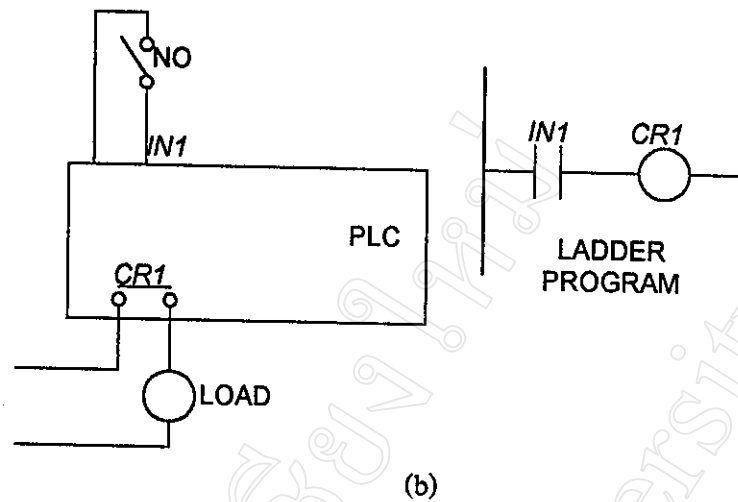
ตารางที่ 2.1 แสดงคำสั่งแลคเคอร์

คำสั่ง	คำอธิบาย
LOAD	โหลดสถานะทางตรรกะของอินพุต
LOAD NOT	โหลดสถานะทางตรรกะของอินพุตและทำการกลับค่า (0 เป็น 1 และ 1 เป็น 0)
AND	คำสั่ง AND ทางตรรก
AND NOT	คำสั่ง AND NOT ทางตรรก
OR	คำสั่ง OR ทางตรรก
OR NOT	คำสั่ง OR NOT ทางตรรก
OUT	เอาต์พุต

วงจรแลคเคอร์เป็นตัวกำหนดว่าเอาต์พุตของพีแอลซีจะทำงานอย่างไร พิจารณารูปที่ 2.2 (a) และ (b) แสดงการเชื่อมต่อพีแอลซีเข้ากับอินพุตและเอาต์พุต สังเกตว่าทั้งสองแบบต่อเข้ากับสวิตช์ที่เป็นแบบปกติเปิดเข้าที่อินพุต *IN1* ของพีแอลซีในรูปที่ 2.2 (a) ในวงจรแลคเคอร์กำหนดให้ *IN1* เป็นแบบปกติปิดเอาต์พุต *CR1* จะหยุดทำงานถ้า *IN1* ทำงาน ในรูปที่ 2.3 (b) ในวงจรแลคเคอร์กำหนดให้ *IN1* เป็นแบบปกติเปิด เอาต์พุต *CR1* ทำงานเมื่อ *IN1* ทำงาน เนื่องจากการทำงานควบคุมเอาต์พุตของพีแอลซีใช้วิธีการอ่านค่าของอินพุตเข้ามาแทนค่าทางตรรกเพื่อหาว่าเอาต์พุตควรเป็นอย่างไร ในรูปที่ 2.2 (a) มีการใช้ *NOT* ฟังก์ชันกับอินพุต *IN1* เมื่อพีแอลซีอ่านค่าจาก *IN1* เข้ามาก็จะถูกทำให้เป็นค่าที่ตรงกันข้ามกับความเป็นจริงแล้วนำไปควบคุมการทำงานของเอาต์พุต



รูปที่ 2.2 แสดงแลคเคอร์ควบคุมการทำงานของเอาต์พุต



รูปที่ 2.2 (ต่อ) แสดงแลคเคอร์ควบคุมการทำงานของเอาต์พุต

การนำพีแอลซีไปใช้ในระบบควบคุม วงจรแลคเคอร์ภายในพีแอลซีเท่านั้นที่เป็นตัวกำหนดให้เอาต์พุตของพีแอลซีทำงาน โดยที่สัญญาณอินพุตเป็นเพียงเหตุการณ์ที่เข้ามาทำให้วงจรแลคเคอร์ที่บันทึกอยู่ภายในหน่วยความจำของพีแอลซีต้องตัดสินใจเท่านั้น และเช่นเดียวกันเอาต์พุตของพีแอลซีก็คือหน่วยที่จะต้องทำงานตามคำสั่งที่ออกมาจากแลคเคอร์เท่านั้น การโปรแกรมพีแอลซีด้วยแลคเคอร์ไคอะแกรมจะอธิบายในหัวข้อ 2.2 ต่อไปจากนี้

2.2 การโปรแกรมพีแอลซีด้วยภาษาแลคเคอร์ [7]

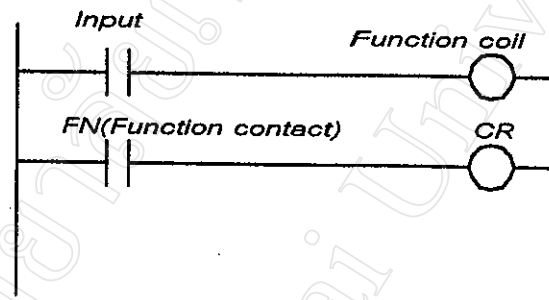
หัวข้อ 2.1 ได้กล่าวถึงแลคเคอร์ไคอะแกรมพื้นฐาน เพื่อเป็นแนวทางในการนำพีแอลซีไปใช้งานและเป็นพื้นฐานในการโปรแกรมพีแอลซีด้วยแลคเคอร์โปรแกรม ในหัวข้อนี้จะอธิบายการโปรแกรมพีแอลซีด้วยแลคเคอร์ไคอะแกรม

2.2.1 โครงสร้างของแลคเคอร์ (Ladder Structure)

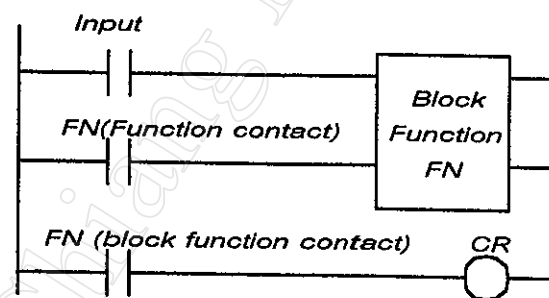
วงจรแลคเคอร์ประกอบขึ้นด้วยหน้าสัมผัส และเอาต์พุตเท่านั้น การทำงานของแลคเคอร์ก็จะถูกจำกัดอยู่ที่การสวิทช์เท่านั้นเช่นกัน แต่สำหรับบางโรงงานที่ผลิตพีแอลซีขึ้นมาเพื่อรองรับการทำงานที่ซับซ้อนมาก ๆ ก็จะผลิตให้พีแอลซีมีฟังก์ชันพิเศษ ซึ่งมีเฉพาะรุ่นและแต่ละผู้ผลิต ดังนั้นฟังก์ชันพิเศษเหล่านี้จะมีไม่เหมือนกันในพีแอลซีแต่ละรุ่นและยี่ห้อ แต่ก็จะมีฟังก์ชันที่เป็นพื้นฐานในทุกๆ พีแอลซีเช่น ตัวตั้งเวลา (Timer) ตัวนับ (Counter) และ ฟังก์ชันในการจัดการกับข้อมูล (Data-Handling) ในแลคเคอร์ไคอะแกรมฟังก์ชันต่าง ๆ เหล่านี้อาจหาให้อยู่ในรูปแบบที่พิเศษไปจากเอาต์พุตธรรมดา ดังแสดงในรูปที่ 2.3 ดังนั้นแลคเคอร์ไคอะแกรมจึงประกอบขึ้นด้วย หน้าสัมผัส เอาต์พุต และ ฟังก์ชันพิเศษ สัญลักษณ์ต่าง ๆ ต่อไปนี้ใช้ในการอธิบายเกี่ยวกับการโปรแกรมภาษาแลคเคอร์ในบทนี้

ตาราง 2.2 สัญลักษณ์ที่ใช้แทนอุปกรณ์ของพีแอลซี

IN	=	หน้าสัมผัสอินพุต
CR	=	รีเลย์ควบคุม
AR	=	รีเลย์ช่วย
T	=	ตัวตั้งเวลา
C	=	ตัวนับ



(a)

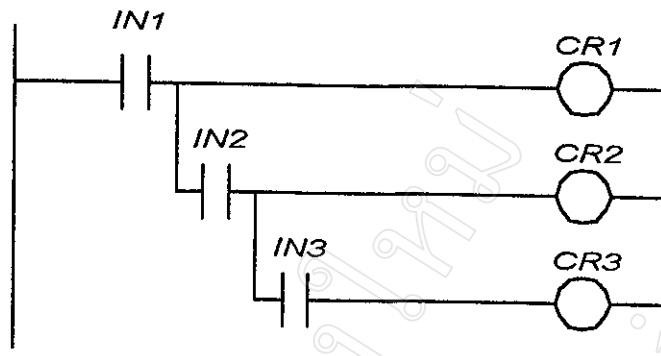


(b)

รูปที่ 2.3 แสดงฟังก์ชันของแลคเคอร์ (a) เอาต์พุตฟังก์ชัน (b) บล็อกฟังก์ชัน

2.2.2 การมีหลายเอาต์พุต (Multiple Outputs)

วงจรแลคเคอร์แต่ละขั้นอาจมีเอาต์พุตได้มากกว่าหนึ่งเอาต์พุต ดังตัวอย่างแลคเคอร์รูปที่ 2.4 เป็นการประยุกต์ใช้ในการควบคุมตามลำดับนั้นคือ เอาต์พุต CR2 จะต้องทำงานหลังจาก CR1 ทำงานไปแล้วและ เอาต์พุต CR3 จะสามารถทำงานได้หลังจาก CR2 ทำงานแล้วเช่นกัน

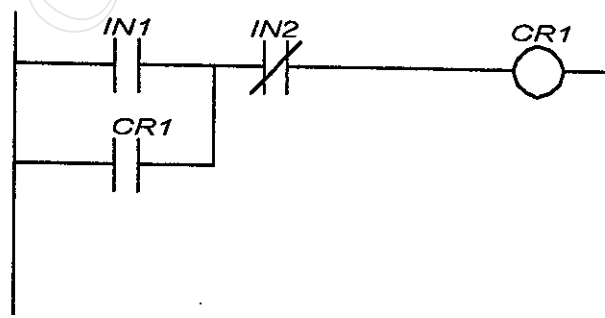


รูปที่ 2.4 แลคเคอร์ 1 ชั้นที่มีเอาต์พุตมากกว่า 1 เอาต์พุต

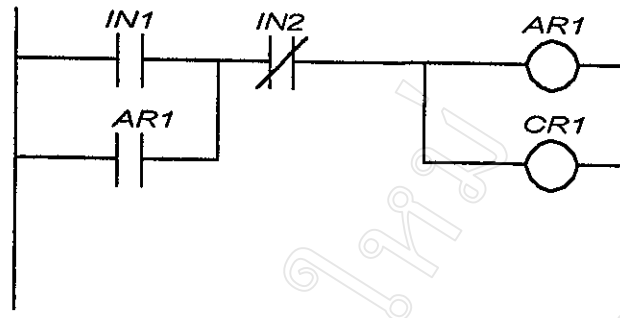
2.2.3 วงจรแลคซ์ (Latch Circuit)

วงจรงแลคซ์ ใช้สำหรับทำให้เอาต์พุตยังคงทำงานค้างอยู่เมื่ออินพุตที่สั่งให้เอาต์พุตทำงานเปลี่ยนแปลงสถานะหลังจากนั้น เช่น การกด Pushbutton SW ตัวอย่างวงจรงแลคซ์ ดังรูปที่ 2.5 เมื่อหน้าสัมผัส *IN1* ถูกสั่งให้ทำงาน ทำให้ *CR1* ทำงาน และสถานะของ *CR1* ถูกนำกลับมาเป็นอินพุตเพื่อรับการ ทำงานของ *CR1* หลังจาก *IN1* หยุดทำงานวงจรงแลคซ์ จะถูกรีเซตเมื่อ *IN2* ถูกสั่งให้ทำงานเปิดการเชื่อมต่อวงจรของ *CR1* การทำเช่นนี้ได้เนื่องจาก แลคเคอร์ยอมให้นำสถานะของเอาต์พุตกลับมาเป็นอินพุตให้กับวงจรงได้ (หรือเรียกว่าการป้อนกลับ)

ในบางครั้งเพื่อป้องกันไม่ให่วงจรภายในเสียสถานะ เนื่องจากปัญหาของกระแสไฟฟ้าภายนอกตัวพีแอลซี สามารถใช้รีเลย์ช่วยของพีแอลซีมาเป็นตัวช่วยในการสร้างสัญญาณป้อนกลับ เนื่องจากภายในพีแอลซีมีเบตเตอร์สำรองอยู่ดังรูปที่ 2.6 เอาต์พุต *CR1* และ *CR1* ถูกสั่งให้ทำงานโดยหน้าสัมผัสอินพุต *IN1* *CR1* สร้างการป้อนสัญญาณกลับมาเป็นอินพุตเพื่อสร้างกรแลคซ์ ให้กับตัวเองและเอาต์พุต *CR1* ในพีแอลซีบางรุ่นจะมีคำสั่งสำหรับการแลคซ์ ได้โดยตรง



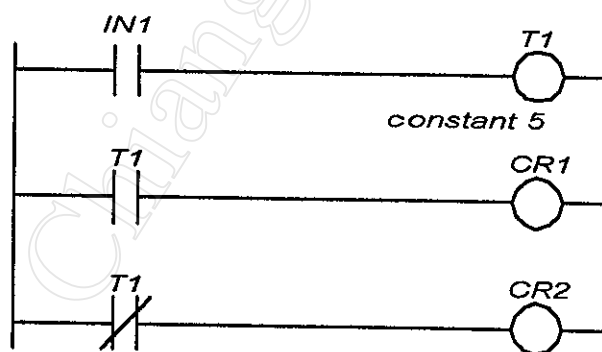
รูปที่ 2.5 การสร้างวงจรงแลคซ์ด้วยเอาต์พุตของพีแอลซี



รูปที่ 2.6 การสร้างวงจรแลตซ์ด้วยรีเลย์ช่วยภายใน

2.2.4 ตัวตั้งเวลา (Timer)

การใช้ตัวตั้งเวลาในโปรแกรมแลตซ์เกอร์ มีส่วนที่ต่างจากการควบคุมเอาต์พุตของแลตซ์เกอร์ ในส่วนของค่าเวลาที่กำหนดให้กับตัวตั้งเวลาเท่านั้น ดังรูปที่ 2.7 แสดงตัวอย่างวงจรแลตซ์เกอร์ที่ประกอบด้วยตัวตั้งเวลา $T1$ การใช้เหมือนกันกับเอาต์พุตแต่ต้องกำหนดค่าคงที่ให้กับตัวตั้งเวลา อินพุต $T1$ และ $T2$ ในรูปเป็นการนำหน้าสัมผัสของตัวตั้งเวลามาใช้ในการควบคุมเอาต์พุต $CR1$ และ $CR2$ ตามลำดับ เมื่อเริ่มต้นการทำงานของวงจร $CR2$ ทำงานทันทีเมื่อไรที่ $IN1$ ถูกสั่งให้ต่อวงจรตัวตั้งเวลา นับหลังจากนั้นไป 5 วินาที $CR1$ จะถูกสั่งให้ทำงานโดยหน้าสัมผัสแบบปกติเปิดของ $T1$ และในขณะเดียวกัน $CR2$ ก็จะถูกสั่งให้หยุดทำงานด้วยหน้าสัมผัสแบบปกติปิดของ $T1$

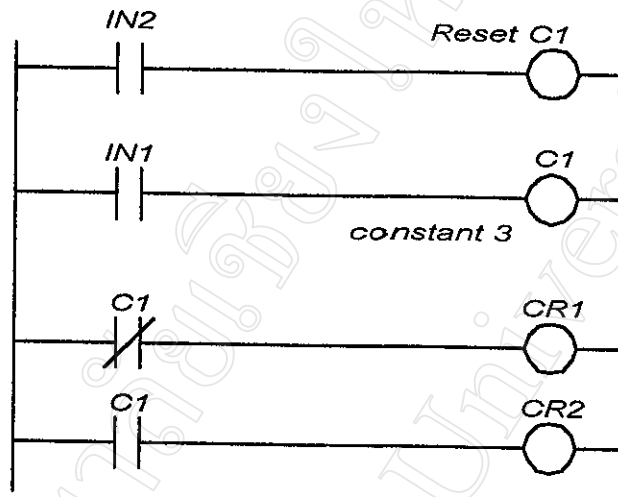


รูปที่ 2.7 แสดงวงจรแลตซ์เกอร์ของตัวตั้งเวลา

2.2.5 ตัวนับ (Counter)

ตัวนับใช้ในการนับการทำงานของหน้าสัมผัสที่ต่ออยู่ เมื่อนับครบตามจำนวนที่ตั้งไว้ก็จะสั่งให้หน้าสัมผัสของตัวนับทำงาน การใช้ตัวนับในวงจรแลตซ์เกอร์ใช้ 2 เอาต์พุต เอาต์พุตที่เป็นตัวสั่งให้นับและเอาต์พุตที่เป็นตัวรีเซ็ตการนับค่าดังแสดงในรูปที่ 2.8 การทำงานของตัวนับเหมือนกับตัวตั้งเวลาคือ

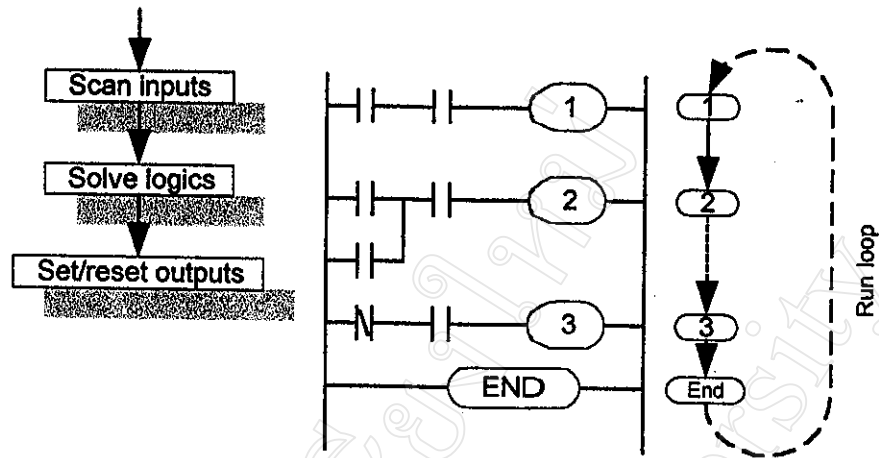
เมื่อนับเท่ากับค่าที่ตั้งไว้ก็จะทำการสั่งให้หน้าสัมผัสทำงาน และเมื่อใดที่เอาต์พุตรีเซตของตัวนับถูกสั่งให้ทำงานค่าที่นับก็จะถูกล้างทั้งหมดและเริ่มต้นนับค่าใหม่ตั้งแต่ต้น และสถานะของหน้าสัมผัสก็จะกลับเหมือนค่าตั้งต้น



รูปที่ 2.8 การใช้งานตัวนับในวงจรแลคเคอร์

2.3 การทำงานของพีแอลซี [7]

การทำงานของพีแอลซีขึ้นอยู่กับโปรแกรมระบบปฏิบัติการของพีแอลซี เมื่อพีแอลซีถูกสั่งให้เริ่มต้นทำงาน โปรแกรมระบบปฏิบัติการเริ่มต้นที่การตรวจสอบสถานะอินพุตและสถานะทางเอาต์พุต ในขณะนั้นก่อน สถานะอินพุตและเอาต์พุตของพีแอลซีถูกเก็บอยู่ในส่วนของหน่วยความจำที่เรียกว่า อิมเมจเมโมรี่ (Image Memory) หลังจากนั้นแลคเคอร์แต่ละขั้น (Rung) จะถูกนำขึ้นมาแก้สมการ โดยการแทนค่าที่ได้จากอิมเมจเมโมรี่เพื่อหาสถานะเอาต์พุตต่อไปของอุปกรณ์เอาต์พุตของพีแอลซี ขั้นตอนหลังจากการเซตและ/หรือรีเซตสถานะของเอาต์พุตภายในแล้วขั้นตอนต่อมาคือนำแลคเคอร์ขั้นต่อ ๆ ไปขึ้นมาแก้สมการตรรกและสั่งการให้เอาต์พุตทำงานตามคำตอบที่ได้ทำงานกระทั่งพบแลคเคอร์ขั้นที่เป็นคำสั่ง 'END' จึงวนกลับไปขั้นต้นแรกคือการตรวจสอบสถานะของอุปกรณ์อินพุตและเอาต์พุตในอิมเมจเมโมรี่ ขั้นตอนที่กำลังมานี้เรียกว่า วงรอบเวลาของพีแอลซี (PLC Cycle Time) ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 แสดงรอบการทำงานของพีแอลซี

2.4 วงจรตรรกะเชิงลำดับ [11]

วงจรตรรกะเชิงลำดับ เอาต์พุตของวงจรไม่ขึ้นอยู่กับสถานะทางค่านอินพุตเพียงอย่างเดียวเท่านั้น แต่ยังขึ้นอยู่กับสถานะของวงจรอีกด้วย โมเดลของวงจรตรรกะเชิงลำดับดังแสดงในรูปที่ 2.10

- 1) S เป็นเซตของสถานะภายในวงจรตรรกะเชิงลำดับ ซึ่งเป็นเซตที่มีขอบเขต ดังนั้น

$$S = \{s_0, s_1, \dots, s_n\}$$

- 2) X เป็นเซตของสัญญาณอินพุต ซึ่งเป็นเซตที่มีขอบเขต ดังนั้น

$$X = \{x_0, x_1, \dots, x_n\}$$

- 3) Z เป็นเซตของสัญญาณเอาต์พุต ซึ่งเป็นเซตที่มีขอบเขต ดังนั้น

$$Z = \{z_0, z_1, \dots, z_n\}$$

- 4) F เป็นฟังก์ชันสำหรับกำหนดสถานะต่อไป s^{+1} ซึ่งเป็นฟังก์ชันของสถานะปัจจุบัน s' และ อินพุตปัจจุบัน X' ดังนั้น

$$s^{+1} = F(s', X')$$

- 5) G เป็นฟังก์ชันในการกำหนดค่าเอาต์พุตของสถานะภายในของวงจร สำหรับ มีลลี (Mealy)

$$Z' = G(s', X')$$

เมื่อเป็นแบบ มัวร์ (Moore) ซึ่งฟังก์ชันเองเอาต์พุตเป็นฟังก์ชันของสถานะภายในเท่านั้นคือ

$$Z' = G(s')$$

นิยาม [11] วงจรตรรกะเชิงลำดับ $M = (X, Z, S, f, \delta)$ เมื่อ X คือเซตของสัญญาณอินพุตที่ไม่เป็นเซตว่าง Z คือเซตของสัญญาณเอาต์พุต S เป็นเซตของสถานะภายใน f และ δ เป็นฟังก์ชันของสถานะต่อไป

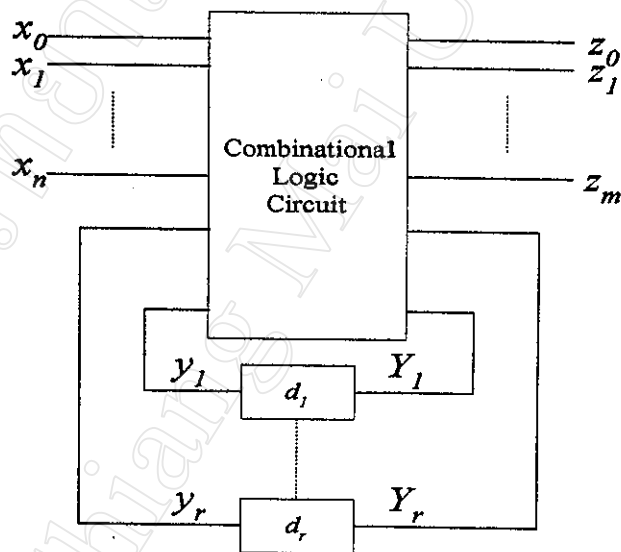
$$s^{t+1} = f(s^t, X^t) \text{ และ ฟังก์ชันของเอาต์พุตคือ}$$

$$Z^t = \delta(s^t, X^t) \text{ สำหรับวงจรแบบมีถลึงและ}$$

$$Z^t = \delta(s^t) \text{ สำหรับวงจรแบบมัลล์}$$

2.4.1 โมเดลวงจรตรรกะเชิงลำดับ

ในรูปที่ 2.1 แสดงโมเดลของวงจรตรรกะเชิงลำดับ ประกอบด้วยส่วนที่เป็นวงจรประกอบ (Combination Circuit) และสายสัญญาณป้อนกลับ (Feedback Line) จากเอาต์พุตกลับมายังอินพุตผ่านส่วนที่ทำหน้าที่เป็นหน่วยความจำ



รูปที่ 2.10 โมเดลของวงจรตรรกะเชิงลำดับของ Huffman

สายสัญญาณป้อนกลับ $y_1, y_2, \dots, y_r$ ผ่านส่วนที่หน่วงเวลา (Delay Element) คือ $d_1, d_2, \dots, d_r$ มาเป็นสัญญาณอินพุตให้กับวงจร หน้าที่ของสายสัญญาณป้อนกลับคือ เป็นสัญญาณกระตุ้นเพื่อให้เกิดสถานะ ปัจจุบัน (Present State) ของวงจรเนื่องจากสถานะที่อาจจะเกิดขึ้นได้ภายในวงจรหรือสถานะที่เป็นไปได้นั้นสามารถเป็นไปได้มากที่สุดเท่ากับ 2^r ดังนั้นจำนวนของสายสัญญาณป้อนกลับได้จากความสัมพันธ์ $2^r \geq s \geq 2^{r-1}$ เมื่อ s เป็นสถานะภายในของวงจร สำหรับ $x_1, x_2, \dots, x_n$ และ $z_1, z_2, \dots, z_m$ คืออินพุตและเอาต์พุตของวงจร

ไปนี้

2.4.2 กระบวนการออกแบบวงจรตรรกเชิงลำดับ

ลำดับขั้นตอนการออกแบบวงจรตรรกเชิงลำดับ [8] โดยทั่วไปประกอบด้วยขั้นตอนดังต่อไปนี้

- 1) การอธิบายการทำงานของวงจรที่ต้องการจะออกแบบ โดยใช้การอธิบายด้วยตารางสถานะ หรือใช้ไคอะแกรมสถานะ ฯ
- 2) ลดจำนวนสถานะ เนื่องจากบางครั้งในขั้นตอนการอธิบายทำให้เกิดสถานะ
- 3) มากจนเกินความจำเป็นซึ่งทำให้สับสนเปลือง การลดจำนวนสถานะใช้อิมพลีเคชันเมตริกซ์ (Implication Matrix) ซึ่งเป็นการค้นหาสถานะที่เหมือนกันและสามารถรวมเข้าด้วยกันได้
- 4) หาสมการสำหรับเอาต์พุตและสมการสำหรับกระตุ้นสถานะภายใน
- 5) ลดรูปสมการให้มีขนาดที่เหมาะสม

2.4.3 ฮาซาร์ดและคริติคัลเรซ (Hazard and Critical Race)

เวลาในการนำสัญญาณทางอินพุตหลาย ๆ อินพุตเข้าสู่ตัวดำเนินการเพื่อให้ได้เอาต์พุตตามที่ต้องการ ในทางปฏิบัติไม่อาจที่จะสามารถหลีกเลี่ยงการเหลื่อมของเวลาได้ ดังนั้นจึงทำให้เอาต์พุตชั่วขณะของวงจรไม่ตรงกับฟังก์ชันที่ได้ออกแบบไว้ เรียกความผิดพลาดนี้ว่า ฮาซาร์ด และในวงจรตรรกเชิงลำดับฮาซาร์ดจะทำให้เกิดสถานะที่ผิดพลาดได้ถ้าสายสัญญาณป้อนกลับถูกรบกวนกลับโดยสายสัญญาณที่มีฮาซาร์ดเกิดขึ้นในระหว่างการทำงานเรียกว่า คริติคัลเรซ