บทที่ 3
การกำหนดพฤติกรรมของระบบควบคุม และการสังเคราะห์วงจรตรรกเชิงลำดับ

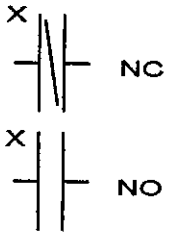
บทนำ

ในบทนี้กล่าวถึงการทำงานของระบบปฏิบัติการของพีแอลซีในการปฏิบัติตามคำสั่งตามแลดเดอร์ไดอะแกรม ซึ่งจะทำให้สามารถเข้าใจขั้นตอนการสังเคราะห์วงจรตรรกเชิงลำดับที่จะนำมาแปลเป็นแลดเดอร์ไดอะแกรมและรหัสคำสั่งของพีแอลซีต่อไป ในหัวข้อต่อมาอธิบายวิธีการกำหนดพฤติกรรมของระบบควบคุมโดยแสดงการกำหนดพฤติกรรมแบบกราฟอธิบายการเปลี่ยนแปลงสภาวะ และแสดงตัวอย่างการสังเคราะห์เพื่อหาสมการทางตรรกจากกราฟอธิบายการเปลี่ยนแปลงสภาวะ วิธีการสังเคราะห์ที่อธิบายเป็นตัวอย่างในบทนี้จะนำไปใช้ในการสังเคราะห์โดยวิธีอัตโนมัติซึ่งจะกล่าวถึงในหัวข้อต่อไปในบทนี้

3.1 อุปกรณ์เสมือนภายในพีแอลซี (PLC Virtual Devices)

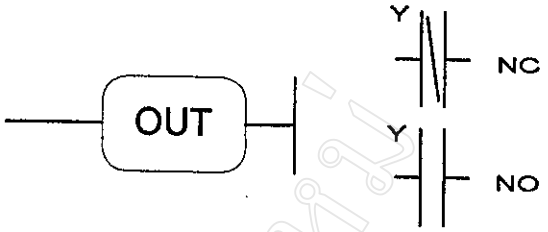
ได้มีการกล่าวถึงไว้แล้วในตอนต้นวิทยานิพนธ์เล่มนี้ว่า พีแอลซีจำลองเอาอุปกรณ์ต่าง ๆ ที่มีใช้อยู่ในระบบควบคุมที่เป็นระบบรีเลย์ โดยพีแอลซีใช้กระบวนการทางซอฟต์แวร์เป็นตัวเลียนแบบการทำงานจึงเรียกอุปกรณ์จำลองเหล่านี้ว่า อุปกรณ์เสมือน (Virtual Device) พีแอลซีใช้รหัสนีโมนิกส์เพื่อสั่งให้เกิดการเชื่อมต่อของอุปกรณ์เสมือนภายในพีแอลซีตามแลดเดอร์ไดอะแกรมที่ผู้ใช้ได้ออกแบบ อุปกรณ์เสมือนพื้นฐานที่มีในพีแอลซีมีดังต่อไปนี้

อินพุต เป็นอุปกรณ์เสมือนภายในพีแอลซีที่รอรับการติดต่อจากภายนอกพีแอลซีโดยผ่านอุปกรณ์ที่เป็นอุปกรณ์อินพุตของพีแอลซี



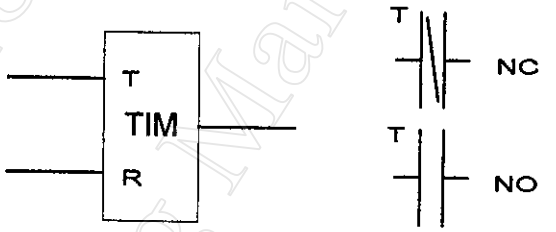
รูปที่ 3.1 สัญลักษณ์ทางอินพุตในแลดเดอร์ไดอะแกรม

เอาต์พุต เป็นอุปกรณ์เสมือนภายในพีแอลซีชนิดเดียวที่สามารถสั่งการควบคุมออกไปยังภายนอกตัวพีแอลซีโดยผ่านรีเลย์หรืออุปกรณ์โซลิดเสตที่เป็นเอาต์พุตภายนอกของพีแอลซี การโปรแกรมสามารถนำหน้าสัมผัสของเอาต์พุตมาเป็นอินพุตให้กับวงจรได้



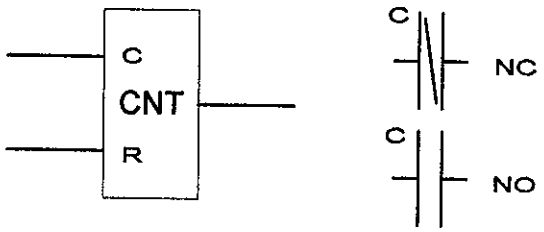
รูปที่ 3.2 สัญลักษณ์ทางเอาต์พุตในแลดเดอร์ไดอะแกรม

ตัวตั้งเวลา(Timer) อุปกรณ์เสมือนภายในพีแอลซีสำหรับตั้งเวลา มีสัญญาณเอาต์พุตที่ทำงานตามเงื่อนไขของการตั้งเวลา เอาต์พุตของอุปกรณ์ตั้งเวลาไม่สามารถเชื่อมต่อกับอุปกรณ์เอาต์พุตภายนอกพีแอลซีโดยตรง เนื่องจากการทำงานต่างๆ ของอุปรณ์ตั้งเวลาเป็นการกระทำทางซอฟต์แวร์ทั้งหมด แต่เอาต์พุตของอุปกรณ์ตั้งเวลาจะใช้เป็นเงื่อนไขอินพุตของวงจรภายในได้



รูปที่ 3.3 สัญลักษณ์ตัวตั้งเวลาในแลดเดอร์ไดอะแกรม

ตัวนับ(Counter) เป็นอุปกรณ์เสมือนภายในพีแอลซีมีหน้าที่เป็นตัวนับการทำงานและการใช้งานเหมือนกับตัวตั้งเวลา แตกต่างกันที่อินพุต C ของตัวนับที่ต้องเข้ามาในรูปของสัญญาณกระตุ้น

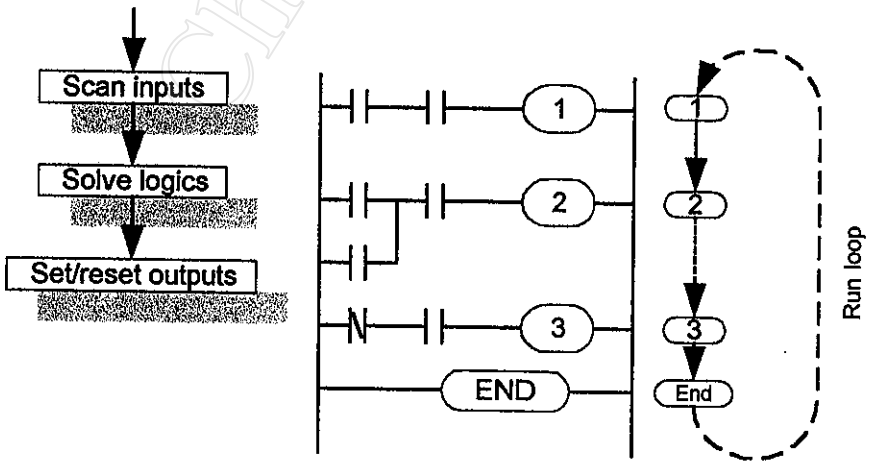


รูปที่ 3.4 สัญลักษณ์ตัวนับในแลดเดอร์ไดอะแกรม

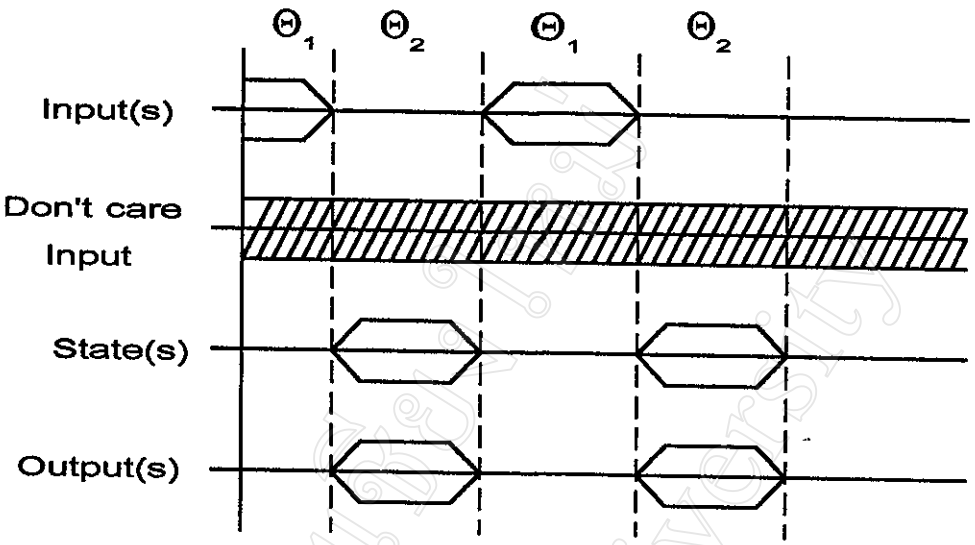
3.2 วงรอบเวลาของพีแอลซี (PLC Cycle Time)

การสังเคราะห์วงจรจำเป็นจะต้องรู้พฤติกรรมการทำงานของวงจรก่อน เนื่องจากวิทยานิพนธ์ฉบับนี้เป็นการสังเคราะห์วงจรตรรกเพื่อนำไปสร้างโปรแกรมควบคุมพีแอลซีโดยใช้ภาษาแลดเดอร์(แลดเดอร์ไดอะแกรม) การทำงานของพีแอลซีขึ้นอยู่กับโปรแกรมระบบปฏิบัติการของพีแอลซี เมื่อพีแอลซีถูกสั่งให้เริ่มต้นทำงานโปรแกรมระบบปฏิบัติการเริ่มต้นที่การตรวจสอบสภาวะอินพุตและสภาวะทางเอาต์พุตในขณะนั้นก่อน โดยสภาวะของอินพุตและเอาต์พุตของพีแอลซีจะถูกเก็บอยู่ในส่วนของหน่วยความจำที่เรียกว่าอิมเมจเมโมรี่ (Image Memory) หลังจากนั้นแลดเดอร์แต่ละขั้น (Rung) จะถูกนำขึ้นมาแก้สมการโดยการแทนค่าที่ได้จากอิมเมจเมโมรี่เพื่อหาสภาวะเอาต์พุตต่อไปของอุปกรณ์เอาต์พุตของพีแอลซี ขั้นตอนหลังจากการเซตและ/หรือรีเซตสภาวะของเอาต์พุตภายในแล้วขั้นตอนต่อมาคือนำแลดเดอร์ขั้นต่อ ๆ ไปขึ้นมาแก้สมการตรรกและสั่งการให้เอาต์พทำงานตามคำตอบที่ได้ทำจนกระทั่งพบแลดเดอร์ขั้นที่เป็นคำสั่ง *'END'* จึงวนกลับไปที่ขั้นตอนแรกคือการตรวจสอบสภาวะของอุปกรณ์อินพุตและเอาต์พุตในอิมเมจเมมโมรี่ ขั้นตอนที่กล่าวมานี้เรียกว่า วงรอบการเวลาของพีแอลซี (PLC Cycle Time) ดังแสดงในรูปที่ 3.5 แสดงวงรอบการทำงานของพีแอลซี

การทำงานของพีแอลซีในแต่ละขั้นเมื่อเปรียบเทียบกับการทำงานของวงจรตรรกเชิงลำดับเราแยกเอาต์พุตและรีเลย์ช่วยของพีแอลซีเป็นเอาต์พุตและตัวแปรสภาวะ (State Variable) เมื่อเทียบกับวงจรตรรกเชิงลำดับ พีแอลซีสามารถนำสัญญาณเอาต์พุตและตัวแปรสภาวะกลับมาเป็นอินพุตของวงจรเพื่อสร้างสภาวะต่อไปให้กับระบบควบคุม เมื่อวิเคราะห์การทำงานจากวงรอบการทำงานของพีแอลซีสามารถสรุปได้ว่าในการเกิดเอาต์พุตและสภาวะใหม่แต่ละครั้งจะต้องมีการเปลี่ยนแปลงของอินพุตเกิดขึ้นก่อนเสมอ ดังนั้นจึงแยกการทำงานออกเป็น 2 เฟสคือ เมื่ออินพุตเปลี่ยนแปลง เอาต์พุตและสภาวะเปลี่ยนแปลงพร้อมกันดังรูปที่ 3.6



รูปที่ 3.5 แสดงรอบการทำงานของพีแอลซี



รูปที่ 3.6 แสดงการทำงานในแต่ละรอบการทำงานของพีแอลซี

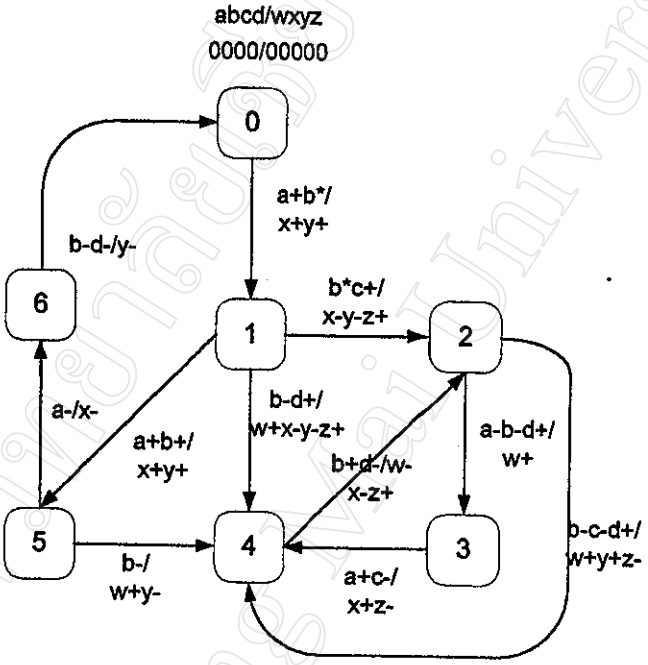
3.3 การกำหนดพฤติกรรมของระบบควบคุม

การกำหนดพฤติกรรมของระบบควบคุมใช้วิธีการกำหนดที่เรียกว่า *"ภาษาอธิบายการเปลี่ยนแปลงสภาวะ"* (State Transition Description Language) ซึ่งมีรูปแบบของภาษาอธิบายใน ภาคผนวก ก.ที่มาของภาษาอธิบายการเปลี่ยนแปลงสภาวะมาจากการใช้กราฟอธิบายการเปลี่ยนแปลงสภาวะ (State Transition Description Graph) เพื่ออธิบายพฤติกรรมของระบบควบคุมแต่อธิบายกราฟด้วยการใช้สัญลักษณ์และตัวอักษรแทน ในการอ้างถึงตัวแปรอินพุตและตัวแปรเอาต์พุตในเงื่อนไขการเปลี่ยนแปลงสภาวะของระบบควบคุม ตัวแปรอิน พุตจะหมายถึงอินพุตภายนอกของพีแอลซีและเอาต์พุตของฟังก์ชันพิเศษของพีแอลซี ตัวแปรเอาต์พุตจะหมายถึงเอาต์พุตภายนอกของพีแอลซีและอินพุตของฟังก์ชันพิเศษของพีแอลซี ซึ่งอุปกรณ์ต่าง ๆ ของพีแอลซีที่สามารถนำมาเป็นตัวแปรให้กับการกำหนดพฤติกรรมของระบบควบคุมได้กล่าวถึงไว้ในหัวข้อก่อนหน้านี้แล้ว

รูปที่ 3.7 แสดงตัวอย่างกราฟอธิบายการเปลี่ยนแปลงสภาวะที่ใช้ในการกำหนดพฤติกรรมการทำงานของระบบควบคุมที่ใช้พีแอลซีเป็นตัวควบคุมหลัก การเปลี่ยนแปลงสภาวะของระบบควบคุมจากสภาวะหนึ่งไปยังอีกสภาวะหนึ่งเกิดขึ้นเมื่อเงื่อนไขของการเปลี่ยนแปลงสภาวะถูกต้องตามที่กำหนดไว้ กราฟอธิบายการเปลี่ยนแปลงสภาวะเป็นกราฟแบบมีทิศทาง (Directed Graph หรือ Digraph) ทิศทางแสดงการเปลี่ยนสภาวะ แสดงได้จากเส้นตรงหรือเส้นโค้งที่มีหัวลูกศรชี้ไปด้านใดด้านหนึ่ง สภาวะปัจจุบันของระบบควบคุมอยู่ทางด้านหางและสภาวะต่อไปของระบบควบคุมอยู่ทางด้านหัวลูกศรชี้ดังแสดงในรูปที่ 3.7 เงื่อนไขด้านอินพุตและค่าของเอาต์พุตจะถูกกำหนดอยู่ที่เส้นตรงหรือเส้นโค้งแสดงการเปลี่ยนแปลงสภาวะนี้ สังเกตชื่อของสัญญาณที่กำหนดเป็นเงื่อนไขของการเปลี่ยนแปลง

สภาวะ เครื่องหมายบวก (+) และเครื่องหมายลบ (-) ที่กำหนดอยู่ที่ชื่อของสัญญาณแต่ละสัญญาณ เครื่องหมายเหล่านี้เป็นตัวบอกให้รู้ว่าสัญญาณมีค่าเป็น '1' หรือเป็น '0' ตามลำดับ ตัวอย่างเช่น a+ หมายความว่าสัญญาณ a เป็น '1' และเมื่อนำมาเป็นเงื่อนไขในการเปลี่ยนแปลงสภาวะดังตัวอย่างต่อไปนี้

a+b+c-/z- หมายความว่า "ถ้าสัญญาณ a เป็น 1 และสัญญาณ b เป็น 1 และสัญญาณ c เป็น 0 ดังนั้นเอาต์พุต z เป็น 0 "



รูปที่ 3.7 กราฟอธิบายการเปลี่ยนสภาวะ

ชื่อของสัญญาณที่ไม่ได้ถูกกำหนดด้วยเครื่องหมาย '+' หรือ เครื่องหมาย '-' แต่ถูกกำหนดด้วยเครื่องหมายดอกจันทน์ (*) เป็นการแสดงบอกให้รู้ว่าสัญญาณนั้นเป็นสัญญาณที่ไม่มีการกำหนดว่าจะเป็น '1' หรือ '0' หรือเรียกว่า directed don't care สัญญาณเหล่านี้เมื่อถูกกำหนดขึ้นในช่วงของการเปลี่ยนแปลงสภาวะใด ดังเช่นสัญญาณ *b** ในระหว่างการเปลี่ยนแปลงสภาวะจากสภาวะ S_0 ไปยังสภาวะ S_1 หลังจากสภาวะ S_1 ไปแล้วสัญญาณ b จะต้องถูกกำหนดลงในเงื่อนไขของการเปลี่ยนแปลงสภาวะต่อไปจนกว่าสัญญาณ b ถูกกำหนดให้เป็น '+' หรือ '-' ในการเปลี่ยนแปลงสภาวะครั้ง ต่อ ๆ ไป สัญญาณอื่น ๆ ที่มีอยู่แต่ไม่มีการกล่าวถึงหรือไม่มีการกำหนดลงในเงื่อนไขของการเปลี่ยนแปลงสภาวะ สัญญาณเหล่านั้นจะมีค่าคงที่หลังจากสภาวะครั้งสุดท้ายที่เปลี่ยนแปลงซึ่งทำให้ไม่ต้องกำหนดทุก ๆ สัญญาณที่ทุก ๆ การเปลี่ยนแปลงสภาวะ

นิยามและข้อกำหนดต่อไปนี้อ้างถึงการกำหนดพฤติกรรมของวงจรตรรกเชิงลำดับที่เรียกว่า Burst-Mode [8] และได้มีการเปลี่ยนแปลงบางส่วนเพื่อให้สามารถนำมาใช้ได้กับการกำหนดพฤติกรรมการทำงานของระบบควบคุมแบบที่ใช้พีแอลซี $G = (V,E,v_0,I,O,in,out)$ และ V เป็นเซตของสภาวะภายในของระบบควบคุม $E \subseteq VXV$ เป็นเซตของการเปลี่ยนสภาวะภายในโดยที่ $e=(v_i,v_j)$ I และ O เป็นเซตของสัญญาณอินพุตและสัญญาณเอาต์พุต v_0 เป็นสภาวะเริ่มต้น (Initial State) ของระบบควบคุม in และ out เป็นฟังก์ชันที่ใช้ในการกำหนดดิวบ์ของอินพุตและเอาต์พุตของแต่ละสภาวะ และมีฟังก์ชัน inpc และ outv สำหรับการกำหนดการเปลี่ยนแปลงของอินพุตและเอาต์พุต

กำหนดให้ $(u,v) \in E$ และ $x_i \in inpc(u,v)$ ถ้าและเพียงแต่ถ้า $in_i(u) \neq in_i(v) \vee in(v) = *$ นั่นคือ x_i ถูกกำหนดให้เป็น x_i^+ ในเงื่อนไขการเปลี่ยนสภาวะจากสภาวะ u ไปยังสภาวะ v ถ้าและเพียงแต่ถ้า $in_i(v)=1 \wedge in_i(u) \neq 1$ และเช่นเดียวกัน x_i ถูกกำหนดให้เป็น x_i^- ถ้าและเพียงแต่ถ้า $in_i(v)=0 \wedge in_i(u) \neq 0$ และ x_i ถูกกำหนดให้เป็น x_i^* ถ้าและเพียงแต่ถ้า $in_i(v)=*$ การกำหนดเอาต์พุตที่มีการเปลี่ยนแปลงในเงื่อนไขการเปลี่ยนสภาวะ $z_j \in outpv(u,v)$ ถ้าและเพียงแต่ถ้า $out_j(u) \neq out_j(v)$ นั่นคือ z_j ถูกกำหนดให้เป็น z_j^+ ในเงื่อนไขการเปลี่ยนสภาวะจากสภาวะ u ไปยังสภาวะ v ถ้าและเพียงแต่ถ้า $out_j(v)=1 \wedge out_j(u)=0$ และเช่นเดียวกัน z_j ถูกกำหนดให้เป็น z_j^- ถ้าและเพียงแต่ถ้า $out_i(v)=0 \wedge out_i(u)= 1$

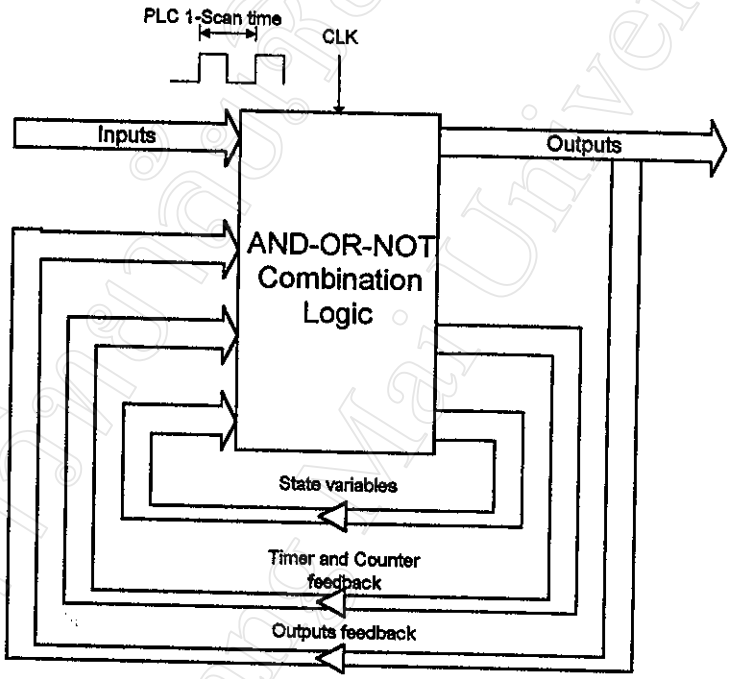
3.3.1 ข้อกำหนดและข้อจำกัดในการกำหนด

กราฟอธิบายการเปลี่ยนแปลงสภาวะดังรูปที่ 3.7 มีข้อจำกัดในการสร้างกราฟเพื่อควบคุมการกำหนดที่อาจจะเป็นการกำหนดที่เกินความสามารถของซอฟต์แวร์ ข้อจำกัดทำให้การสังเคราะห์ทำได้อย่างถูกต้องและระบบควบคุมสามารถทำงานได้ตามฟังก์ชันที่ต้องการ

1) จำนวนตัวแปรอินพุตได้ไม่เกิน 24 ตัวแปร
2) จำนวนตัวแปรเอาต์พุตได้ไม่เกิน 16 ตัวแปร
3) การเปลี่ยนแปลงของสัญญาณอินพุตที่เป็นเงื่อนไขของการเปลี่ยนแปลงสภาวะกำหนดเฉพาะสัญญาณที่มีการเปลี่ยนแปลงเท่านั้น
4) สภาวะเริ่มต้นของระบบควบคุมต้องกำหนดค่าของตัวแปรทุก ๆ ตัว
5) สัญญาณอินพุตที่เป็น don't care ต้องกำหนดลงในเงื่อนไขของการเปลี่ยนแปลงสภาวะทุก ๆ ตัว
6) การเปลี่ยนแปลงสภาวะทุกๆ ครั้งเกิดขึ้นเมื่อสัญญาณอินพุตมีการเปลี่ยนแปลงค่าที่แน่นอนอย่างน้อยที่สุด 1 สัญญาณ
7) ในสภาวะเดียวกันเซตของอินพุตที่ถูกกำหนดในการเปลี่ยนแปลงสภาวะจะต้องไม่เป็นเซตย่อยของอินพุตของการเปลี่ยนแปลงสภาวะอื่น ๆ

3.4 แบบจำลองของแลดเดอร์ไดอะแกรมสำหรับพีแอลซี

วงจรตรรกเชิงลำดับที่ได้กล่าวไว้ในบทที่ 2 สัญญาณป้อนกลับของวงจรสร้างมาจากเอาต์พุตที่เป็นตัวแปรสภาวะเท่านั้น แต่เมื่อเรานำเอาทฤษฎีการออกแบบวงจรตรรกมาใช้เพื่อออกแบบแลดเดอร์ไดอะแกรมสำหรับพีแอลซี เอาต์พุตของวงจรจะถูกนำเข้ามาเป็นตัวแปรสภาวะด้วย ซึ่งจะทำให้จำนวนของตัวแปรสภาวะลดลงและผลที่ตามมาคือจำนวนคำสั่งที่ใช้ในการป้อนโปรแกรมให้กับพีแอลซีก็จะลดลง ในแลดเดอร์ไดอะแกรมในขณะที่พีแอลซีทำงานอยู่นั้นสัญญาณป้อนกลับ (Feedback Line) จะมีค่าหน่วงเวลาอยู่เท่ากับ 1-Scan Time ดังแสดงในรูปที่ 3.8



รูปที่ 3.8 แสดงโมเดลของแลดเดอร์ไดอะแกรมสำหรับพีแอลซี

ดังนั้นการทำงาน ของวงจรตรรกจึงเป็นแบบ Synchronous Sequential Logic Circuit จากโมเดลของแลดเดอร์ไดอแกรมสำหรับพีแอลซีสามารถอธิบายให้อยู่ในรูปของเซตและความสัมพันธ์ได้คือ (X,Y,Z,δ)

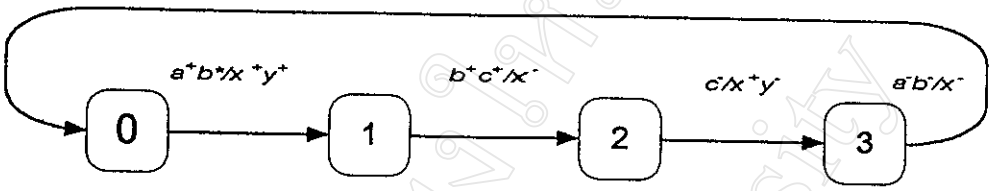
X เป็นเซตของสัญลักษณ์ทางอินพุตโดยมีเอาต์พุตของตัวตั้งเวลาและตัวนับของพีแอลซีเป็นสมาชิกเซตรวมอยู่ด้วย

Y เป็นเซตของสัญลักษณ์ทางเอาต์พุตโดยมีอินพุตของตัวตั้งเวลาและตัวนับเป็นสมาชิกเซตรวมอยู่ด้วย

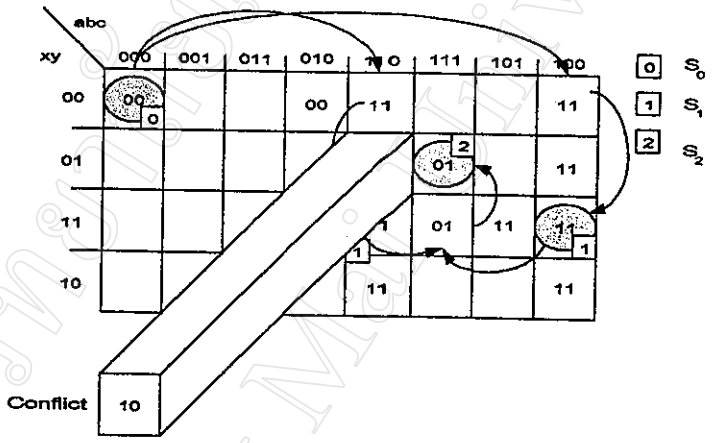
Z เป็นเซตของตัวแปรสภาวะภายใน

$\delta: X \times Y \times Z \rightarrow Y \times Z$ เป็นฟังก์ชันของสภาวะต่อไป (Next-State Function)

ตัวอย่างการสังเคราะห์วงจร



รูปที่ 3.9 ตัวอย่างการกำหนดพฤติกรรมของระบบควบคุม



รูปที่ 3.10 แสดงการชนกันของเอาต์พุต

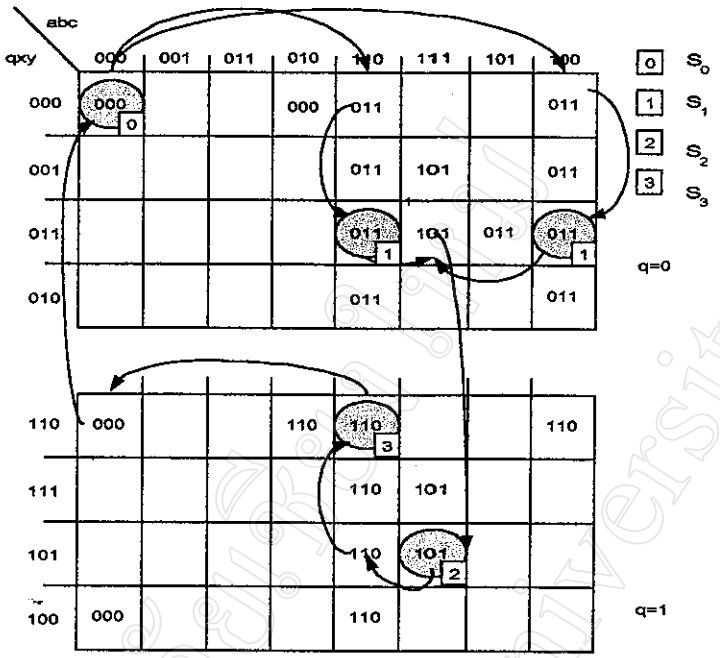
ตัวอย่างการอธิบายพฤติกรรมการควบคุมดังรูปที่ 3.9 สามารถอธิบายการสังเคราะห์วงจรและการทำงานของตัวควบคุม (PLC) ที่ตรงกันกับพฤติกรรมการทำงานของระบบควบคุมที่กำหนดขึ้นโดยใช้กราฟอธิบายการเปลี่ยนแปลงสภาวะ และอธิบายขั้นตอนการสร้างตารางสภาวะและการกำหนดค่าต่างๆ ที่จำเป็นลงในตารางเพื่อให้ได้วงจรการทำงานของระบบควบคุมเป็นไปตามที่ได้กำหนดโดยกราฟอธิบายการเปลี่ยนแปลงสภาวะดังรูปที่ 3.9 ช่องตารางสภาวะที่ไม่ถูกกำหนดค่าใดเป็นช่องที่เป็น don't care และจากตารางสภาวะที่ได้จะสามารถหาสมการทางตรรกของระบบควบคุมได้โดยตรงเนื่องจากตารางสภาวะเป็นตารางที่อธิบายสภาวะต่อไป ของค่าอินพุต เอาต์พุต ของทุก ๆ สภาวะ

การทำงานของระบบควบคุมเริ่มต้นการทำงานที่สภาวะเริ่มต้น (Initial State) คือ S_0 ที่สภาวะเริ่มต้นระบบรอเงื่อนไขอินพุต $a+b^*$ ในระหว่างการเกิดขึ้นของอินพุตตามเงื่อนไขของ abc คือ $a+b^*$ นั้นการเปลี่ยนแปลงอาจจะเปลี่ยนจาก 000 ไปเป็น 100 ถ้าสัญญาณอินพุต $a+$ เกิดขึ้นก่อน หรือ

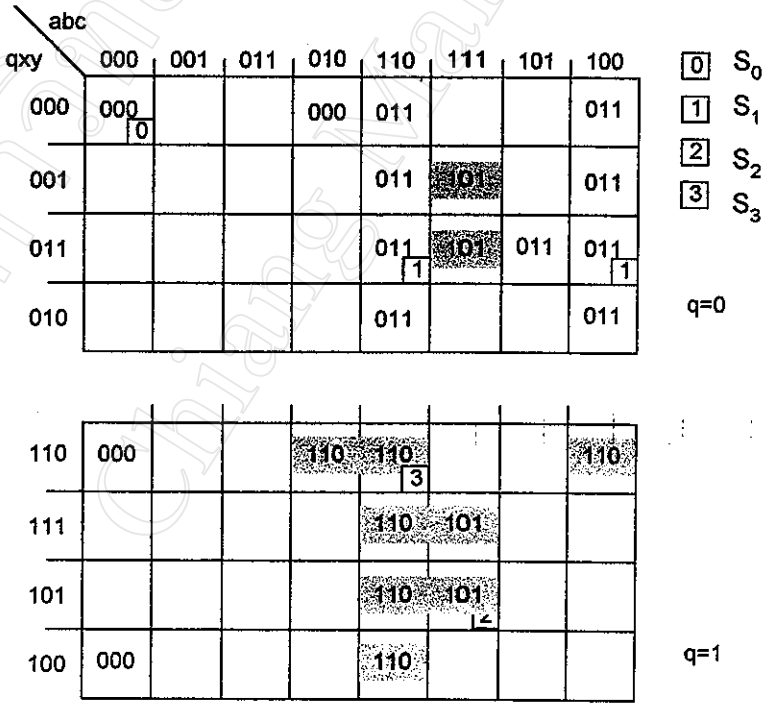
abc อาจจะมีการเปลี่ยนจาก *000* ไปเป็น *110* ผ่าน *010* ถ้าเกิด *b+* ขึ้นก่อน *a+* เอาต์พุตจะไม่สามารถเปลี่ยนแปลงได้เลยถ้าไม่มีสัญญาณอินพุต *a+* เกิดขึ้น ในกรณีนี้เนื่องจากอินพุต *b* เป็น don't care ดังนั้นเอาต์พุตของ *XY* ที่ *abcxy=00000* และที่ *abcxy=01000* คือ *00* และเมื่อเงื่อนไขเป็นไปตามที่กำหนดเอาต์พุตของ *XY* จึงเปลี่ยนไปเป็น *11* ดังนั้น ในช่องที่ *abcxy = 1x0xx* จึงเป็น *11* ทุกช่องและ *XY* เท่ากับ *11* สภาวะจะคงที่อยู่ตรงตำแหน่ง *xy = 11* คือสภาวะ S_1 ซึ่งสังเกตได้ว่า *b* จะเป็น *0* หรือ *1* ก็ได้

ในสภาวะ S_1 ระบบควบคุมรออินพุต *b+c+* ซึ่งการเปลี่ยนแปลงของสัญญาณอินพุตจากสภาวะ S_1 อาจจะมีการเปลี่ยนแปลงจาก *abc* ที่มีค่าเท่ากับ *100* ไปเป็น *111* ผ่าน *101* ถ้าอินพุต *b* มีค่าอยู่ที่ *0* และอาจจะเปลี่ยนแปลงจาก *110* ไปเป็น *111* ถ้ากรณีที่ *b* มีค่าเท่ากับ *0* อยู่แล้ว เอาต์พุตต่อไปของ *XY* ที่ *abcxy* เท่ากับ *10011* , *10111* และ *11011* (*xy* ยังเป็น *11* อยู่แล้ว) หลังจากอินพุตตรงตามเงื่อนไขเอาต์พุต *XY* จึงเปลี่ยนมาเป็น *01* ทำให้สภาวะของระบบควบคุมอยู่ที่สภาวะ S_2 ในสภาวะนี้ระบบควบคุมจะรอการเกิดขึ้นของอินพุต *c-* และเมื่อ *c-* เกิดขึ้นตรงตามเงื่อนไขที่กำหนดไว้เอาต์พุต *XY* จะต้องเปลี่ยนไปเป็น *10* ปัญหาเกิดขึ้นเนื่องจากในช่องตารางสภาวะที่มี *abcxy = 110xx* เอาต์พุตถูกกำหนดให้เป็น *11* อยู่ก่อนแล้วในระหว่างการเปลี่ยนสภาวะจาก S_2 ไปเป็น S_3 ในการหลีกเลี่ยงปัญหานี้คือต้องเพิ่มตัวแปรสภาวะเพิ่มขึ้นอีกหนึ่งตัว นั่นคือการเพิ่มตารางอีกหนึ่งตาราราง ดังรูปที่ 3.10 โดยการย้อนสภาวะกลับไปยังสภาวะก่อนการเกิดการชนกันของเอาต์พุตในช่องตารางเดียวกัน หลังจากนั้นทำการเปลี่ยนสภาวะภายในและสัญญาณเอาต์พุตพร้อม ๆ กันไปยังอีกตารางที่เพิ่มเติมเข้ามารองรับ ดังนั้นสภาวะ S_2 เข้าสู่สภาวะคงที่ในตารางที่เพิ่มเข้ามาใหม่ กระบวนการสร้างตารางสภาวะจาก สภาวะ S_2 ไปยังสภาวะ S_3 เกิดขึ้นในตารางที่เพิ่มเข้ามาเช่นกัน จนกระทั่งสะภาวะของระบบควบคุมเข้าสู่สภาวะ S_0 เป็นการสิ้นสุดการสร้างตารางสภาวะ

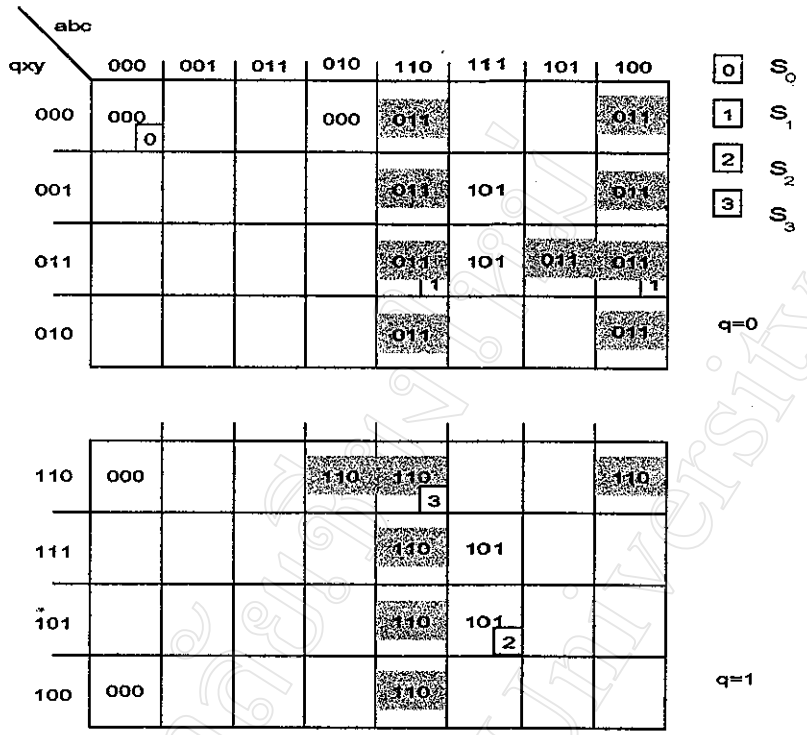
สมการบูลลีนจากตารางสภาวะอยู่ในเทอมของผลบวกของเทอมผลคูณ เอาต์พุตฟังก์ชันแต่ละเอาต์พุตได้มาโดยการเลือกออนเซต (On-Set) ของเอาต์พุตแต่ละตัวในตารางสภาวะที่สมบูรณ์มาจัดให้อยู่ในเทอมของผลบวกของเทอมผลคูณ และลดรูปสมการ โดยใช้วิธีการของ Quine-McClusky [6]



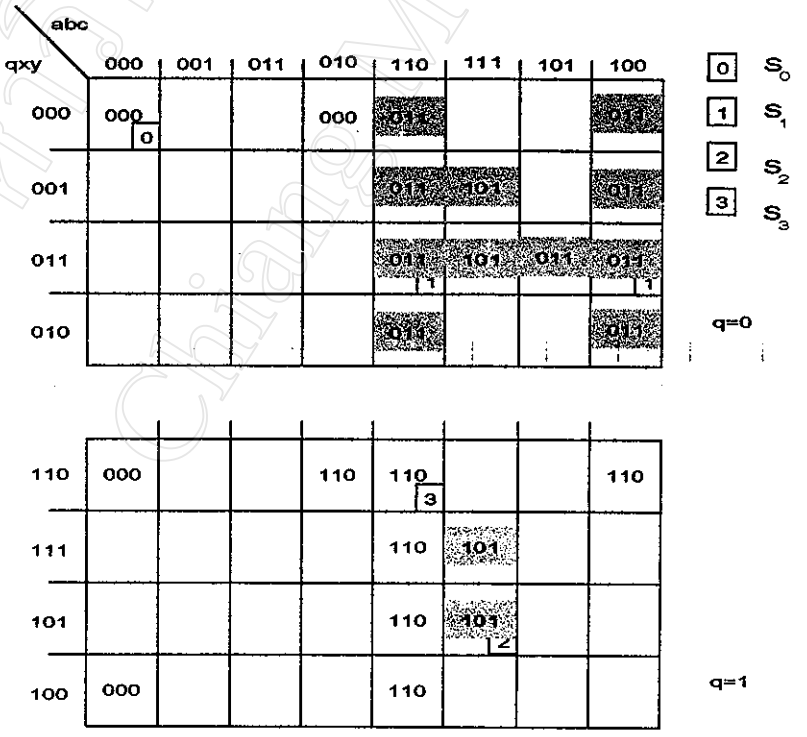
รูปที่ 3.11 ตารางสภาวะที่สมบูรณ์



รูปที่ 3.12 แสดงออนเซตของตัวแปรสภาวะ q



รูปที่ 3.13 แสดงออนเซตของเอาต์พุต X



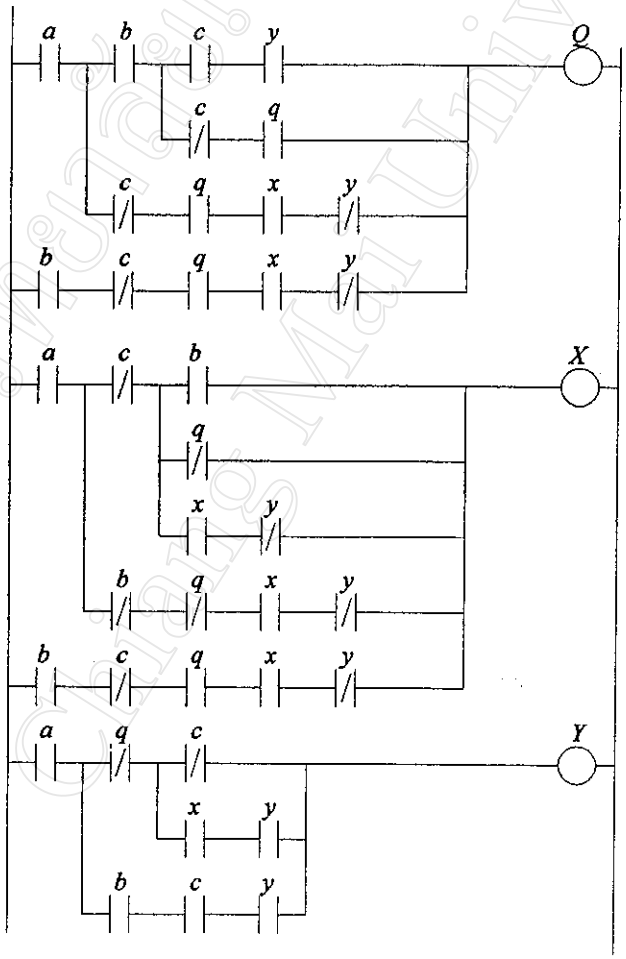
รูปที่ 3.14 แสดงออนเซตของเอาต์พุต Y

$$q = a(b(cy+c'q)+c'qxy')+bc'qxy'$$

$$x = a(c'(b+q'+xy')+b'q'xy)+bc'qxy'$$

$$y = a(q'(c'+xy)+bcy)$$

สมการที่ได้จากการสังเคราะห์เป็นแบบสมการตรรกสองระดับ (Two Level Logic Function) เมื่อนำไปแปลเป็นรหัสคำสั่งของพีแอลซีจะทำให้มีจำนวนคำสั่งมากเกินความจำเป็นจึงต้องทำ สมการตรรกสองระดับ ให้เป็นสมการตรรกแบบหลายระดับ (Multi Level Logic Function) ก่อนการแปลเป็นรหัสคำสั่งของพีแอลซี สำหรับการสร้างสมการตรรกแบบหลายระดับจากสมการตรรกแบบสองระดับ ได้อธิบายไว้ในตอนท้ายของบทนี้



รูปที่ 3.15 แสดงแลดเดอร์ไดอะแกรมจากตัวอย่างการสังเคราะห์

3.5 การสังเคราะห์วงจรแบบอัตโนมัติ (Automatic Circuit Synthesis)

หัวข้อก่อนนี้ได้แสดงกระบวนการสังเคราะห์สมการตรรกจากกราฟอธิบายการเปลี่ยนแปลงสภาวะด้วยตัวอย่างง่าย ๆ ในหัวข้อนี้อธิบายกระบวนการการสังเคราะห์วงจรแบบอัตโนมัติ ซึ่งเป็นขั้นตอนการนำเอาข้อมูลจากการกำหนดพฤติกรรมของระบบควบคุม ซึ่งจัดให้อยู่ในรูปของโครงสร้างข้อมูลที่คอมพิวเตอร์สามารถรับรู้ได้ (ใช้โครงสร้างข้อมูลแบบกราฟแบบมีทิศทาง) ร่วมกับข้อกำหนดและนิยามต่างๆ ที่ได้กล่าวแล้วในบทนี้ และที่จะกล่าวถึงต่อไป นำไปผ่านกระบวนการทางซอฟต์แวร์ เพื่อให้ได้สมการทางตรรกที่ทำให้ระบบควบคุมสามารถทำงานได้ถูกต้องตามที่กำหนดในการกำหนดคุณสมบัติ วิธีการเริ่มจากการสร้างตารางสภาวะต่อไป (Next State Table) และแทนค่าของเอาต์พุตลงในช่องตารางให้ครบทุกช่องที่จำเป็น หลังจากนั้นทำการหาสมการโดยการนำเอาออนเซต (On-Set) ของแต่ละเอาต์พุตซึ่งจัดให้อยู่ในรูปแบบที่เรียกว่า "ผลบวกของเทอมผลคูณ" (SUM-OF-PRODUCT หรือ SOP) จากการปฏิบัติตามข้อกำหนดพฤติกรรมของระบบควบคุม และต่อมาเป็นการค้นหาสมการทางตรรกของเอาต์พุตแต่ละตัวรวมถึงสมการทางตรรกของตัวแปรสภาวะด้วย โดยได้จากตารางสภาวะต่อไปที่สร้างขึ้น ต่อจากนั้นจะทำการลดรูปสมการเพื่อให้มีขนาดที่เหมาะสมที่สุด และท้ายสุดจะทำการจัดรูปแบบของโครงสร้างข้อมูลที่ง่ายสำหรับการดำเนินการกับสมการที่ได้มาจากการสังเคราะห์

3.5.1 นิยาม

นิยามต่อไปนี้เป็นแนวความคิดที่อ้างอิงจากเอกสารอ้างอิงต่อไปนี้ [6,9,10,13] ซึ่งได้นำเอาแนวความคิดดังกล่าวมาใช้ในการศึกษาและดำเนินการในหัวข้อต่อไปนี้ เพื่อทำให้เข้าใจการทำงานของซอฟต์แวร์ตามที่โครงการวิจัยนี้ได้สร้างขึ้นได้มากยิ่งขึ้น

ฟังก์ชันตรรก (Logic Function) f คือความสัมพันธ์ระหว่าง $\{0,1\}^n$ ไปยัง $\{0,1,*\}$ เมื่อ n แทนจำนวนเอาต์พุตของฟังก์ชัน เขียนแทนด้วย $f:\{0,1\}^n \rightarrow \{0,1,*\}$

มินเทอม (Minterm) คือ การจัดเรียง n สิ่ง (n-tuple) $[x_1,x_2,...,x_n]$ เมื่อ x_i เป็นค่าของอินพุตบิตที่ i ของฟังก์ชัน f มีค่าเป็น "0" หรือ "1"

ออน-เซต (On-Set) เป็นเซตของมินเทอมที่ทำให้เอาต์พุตเป็น "1" เช่นออน-เซตของ Y หมายถึงเซตของมินเทอมที่เอาต์พุต Y เป็น "1"

ออฟ-เซต (Off-Set) เป็นเซตของมินเทอมที่ทำให้เอาต์พุตเป็น "0" เช่นออป-เซตของ Y หมายถึงเซตของมินเทอมที่เอาต์พุต Y เป็น "0"

คิวบ์ (Cube) c แทนด้วย $[c_1,c_2,...,c_n]$ เป็นเวคเตอร์ใน $\{0,1,*\}^n$ คิวบ์เป็นมินเทอม $[x_1,x_2,...,x_n]$ ถ้าทุก $i \in 1,...,n, x_i \neq *$

คิวบ์ $[a_1,a_2,...,a_n]$ เป็นคิวบ์ที่ *บรรจุ* อยู่ใน (Contain) คิวบ์ $[b_1,b_2,...,b_n]$ ถ้าและเพียงแต่ถ้า (*iff*) $a_i = b_i$ หรือ $a_i = *$ เมื่อ $i=1,...,n$

คิวบ์ $[a_1,a_2,...,a_n]$ เป็นคิวบ์ที่มี *สมาชิกร่วม* กันกับ (Intersec) คิวบ์อื่น $[b_1,b_2,...,b_n]$ ถ้าและเพียงแต่ถ้า $a_i = b_i$ หรือ $a_i = *$ หรือ $b_i = *$ เมื่อ $i=1,...,n$

สัญพจน์ (Literal) คือตัวแปรและคอมพลีเมนท์ (Complement) ของตัวแปร *เทอมผลคูณ* (Product Term) คือการคูณกันแบบบูลลีนของสัญพจน์ และ *เทอมผลบวกของเทอมผลคูณ* (Sum of Product Term) คือการบวกกันแบบบูลลีนของเทอมผลคูณ ข้อจำกัดของเทอมผลคูณคือ *ไม่มีเทอมผลคูณใดที่มีทั้งตัวแปรและคอมพลีเมนท์ของตัวแปรเดียวกันอยู่ในเทอมผลคูณเดียวกัน* โดยข้อจำกัดนี้ทำให้เทอมผลคูณและคิวบ์เหมือนกัน ดังนั้นเทอมผลคูณ $x_1x_3'x_4$ ตรงกับคิวบ์ $[1,*,0,1,*,...,*]$

อิมพลิแคนท์ (Implicant) ของฟังก์ชันตรรก f คือเทอมผลคูณซึ่งไม่มีมินเทอมที่เป็นออฟ-เซตของ f บรรจุอยู่ด้วย (อิมพลิแคนท์ไม่ใช่ออน-เซต)

คัฟเวอร์ (Cover) C ของฟังก์ชันตรรก f คือเซตของอิมพลิแคนท์ของ f ดังนั้นมินเทอมที่เป็น ออน-เซตของ f ทั้งหมดถูกครอบคลุมด้วยบางคิวบ์ของ C แต่ไม่มีมินเทอมที่เป็นออฟ-เซตรวมอยู่ด้วย คัฟเวอร์จัดเป็นเทอมผลบวกของเทอมผลคูณเพื่อสร้างเป็นฟังก์ชันตรรกของ f

ถ้า $A = [a_1,a_2,...,a_n]$ และ $B = [b_1,b_2,...,b_n]$ *ทรานซิชั่นคิวบ์* (Transition Cube) $C = [c_1,c_2,...,c_n]$ หาได้จาก $c_i = a_i = b_i$ ถ้า $a_i = b_i$ และ $c_i = *$ ถ้า $a_i \neq b_i$ เมื่อ $i = 1,...,n$ ทรานซิชั่นคิวบ์แทนด้วย $[A,B]$ เป็นคิวบ์เล็กที่สุดที่มี A และ B อยู่ในคิวบ์ทั้งคู่

ทราเจคทอรี (Trajectory) ในทรานซิชั่นคิวบ์ $[A,B]$ คือเวกเตอร์ของมินเทอมในคิวบ์ $[A,B]$ เขียนแทนด้วย $[m_1,m_2,...,m_p]$ เมื่อ j ใน $1,...,p-1$ มินเทอม m_j และ m_{j+1} มีค่าไบนารีบิตของตัวแปรแตกต่างกันเพียง 1 ตำแหน่งเท่านั้น

3.5.2 ตารางสภาวะต่อไป (Next State Table) และการกำหนดค่าลงในตาราง

กราฟ $G = (V,E,v_0,I,O,in,out)$ เมื่อ $I = \{x_1,x_2,...,x_n\}$ และ $O = \{y_1,y_2,...,y_m\}$ เป็นเซตของอินพุตและเอาต์พุต กำหนดให้ X เป็นเซตของอินพุตบิตเวคเตอร์ $\{(x_1,x_2,...,x_n) \mid x_j \in \{0,1\}, j \in 1,2,...,n\}$ และ Y เป็นเซตของเอาต์พุตบิตเวคเตอร์ $\{(y_1,y_2,...,y_m) \mid y_j \in \{0,1\}, k \in 1,2,...,m\}$ ตารางสภาวะคือ $T = (V,I,O,\delta,\lambda)$ เมื่อ V เป็นเซตของสภาวะและ $\delta: V \times X \times O \rightarrow V \cup \{*\}$ เป็นฟังก์ชันของสภาวะต่อไปและ $\lambda: V \times X \times O \rightarrow \{0,1,*\}$ เป็นฟังก์ชันของเอาต์พุต

การกำหนดของเอาต์พุตและตัวแปรสภาวะลงในตารางสภาวะต่อไป แต่ละช่องตารางของตารางสภาวะต่อไปคือมินเทอมที่เกิดจากการรวมกันของ อินพุตบิตเวกเตอร์ เอาต์พุตบิตเวกเตอร์ และบิตเวกเตอร์ของตัวแปรสภาวะ นั่นก็คือแต่ละมินเทอมก็จะประกอบด้วยตัวแปรทั้งหมด $m+n+r$ โดยที่ m คือจำนวนตัวแปรอินพุต n คือจำนวนตัวแปรเอาต์พุต และ r คือจำนวนตัวแปรสภาวะซึ่งค่าที่แน่นอนของ r จะเกิดขึ้นหลังจากเสร็จสิ้นกระบวนการสร้างตารางสภาวะแล้วเท่านั้น

การกำหนดเอาต์พุตลองในช่องตารางขณะอินพุตเปลี่ยนแปลงให้ M เป็นมินเทอมในทราเจคทอรีของทรานซิชั่นคิวบ์ $[A,B]$ เมื่อ $A = [in(u),L(u),out(u)]$ และ $B = [in(v),L(u),out(u)]$ และ $C = [in(v),*,*]$ เมื่อ $k \in 1,\ldots,n$

$\lambda_k(M)$ $= out_k(v)$ iff $[M,C] = C$ และ

$\lambda_k(M)$ $= out_k(u)$ otherwise

$\delta(M)$ $= u$

เมื่อสัญญาณอินพุตเปลี่ยนแปลงและตรงกับเงื่อนไขของการเปลี่ยนแปลงสภาวะ (u,v) ดังนั้นทำให้เอาต์พุตเปลี่ยนแปลงสภาวะจาก out(u) ไปเป็น out(v) ซึ่งเป็นสถาวะชั่วขณะก่อนการเข้าสู่สภาวะคงที่ของระบบควบคุม และระบบการควบคุมจะเข้าสู่สภาวะคงที่หลังจากตัวแปรสภาวะ เปลี่ยนแปลงจาก L(u) ไปเป็น L(v) การเปลี่ยนแปลงสภาวะของตัวแปรสภาวะกำหนดดังต่อไปนี้

ให้ M เป็นมินเทอมในทราเจคทอรีของทรานซิชั่นคิวบ์ $[A,B]$ เมื่อ $A = [in(v),L(u),out(u)]$ และ $B = [in(v),L(v),out(v)]$ และ $C = [in(v),*,out(v)]$ เมื่อ $k \in 1,\ldots,n$

$\lambda_k(M)$ $= out_k(v)$

$\delta(M)$ $= v$ iff $[M,C] = C$ และ

$\delta(M)$ $= u$ otherwise (สภาวะชั่วขณะ)

สำหรับช่องตารางที่ไม่มีการกำหนดเอาต์พุตและสภาวะใด ๆ $\lambda_k(M) = *$ และ

$\delta(M)$ $= *$

การเข้ารหัสแต่ละชั้นของตาราง (Layer Encoding) ชั้น (Layer) สร้างขึ้นมาสำหรับตารางสภาวะต่อไป แต่ละชั้นประกอบด้วยกลุ่มของสภาวะที่มีรูปแบบ (Pattern) อินพุตบิตเวคเตอร์และเอาต์พุตบิตเวคเตอร์ ของสัญญาณที่แตกต่างกัน ถ้าสภาวะ 2 สภาวะใดมีรูปแบบของอินพุตบิตเวคเตอร์และเอาต์พุตบิตเวคเตอร์ที่ตรงกันจะไม่รวมอยู่ในชั้นเดียวกัน กำหนดให้ L แทนจำนวนชั้นทั้งหมดในตารางสภาวะต่อไป ดังนั้นความสัมพันธ์ $L^{(r-1)} \leq L \leq 2^r$ ทำให้สามารถหาจำนวนบิตที่นำมาสร้างรหัสให้แต่ละชั้นได้ ตัวแปร r จากความสัมพันธ์คือจำนวนบิตที่ต้องใช้สำหรับการเข้ารหัส กระบวนการเข้ารหัสชั้นด้วยค่าไบนารี่ สามารถกำหนดได้โดยง่าย เนื่องจากการสังเคราะห์สมการเพื่อนำมาแปลเป็นโปรแกรมสำหรับพีแอลซี ไม่ต้องคำนึงถึงความผิดพลาดที่เรียกว่าฮาซาร์ด (Hazard) และคริติคัลเรซ (Critical Race) ที่อาจจะเกิดขึ้นได้ถ้าเป็นวงจรดิจิตัล เพราะเอาต์พุตฟังก์ชันของพีแอลซีได้มาจากการแทนค่าของอินพุตลงในสมการทางตรรกโดยระบบปฏิบัติการของพีแอลซี การกำหนดรหัสให้กับชั้นสามารถเริ่มต้น

ที่ค่าไบนารี่ *0...0* สำหรับชั้นที่ประกอบด้วยสภาวะเริ่มต้น และชั้นต่อมาการกำหนดจะต้องไม่มีซ้ำกันจนครบทุกชั้นไปจนถึงชั้นสุดท้าย *1...1*

3.5.3 ฟังก์ชันเอาต์พุต

ฟังก์ชันเอาต์พุตสามารถแยกออกมาได้โดยตรงจากตารางสภาวะต่อไปที่สมบูรณ์ เนื่องจากตารางสภาวะได้รวมเอาอินพุต เอาต์พุต และสภาวะทั้งหมด เมื่อเซตของมินเทอม M ที่กำหนดขึ้นมาในระหว่างการสร้างตารางสภาวะ

ฟังก์ชันเอาต์พุต Y_k ได้จากเซตของมินเทอม M ที่ $\lambda_k(M) = 1$ (เป็นออน-เซตของฟังก์ชัน Y_k) โดยจัดอยู่ในรูปของผลบวกของเทอมผลคูณเมื่อ $k \in 1,...,n$ และเช่นเดียวกันฟังก์ชันของตัวแปรสภาวะ $Z_1,...,Z_r$ คือเซตของมินเทอม M ที่ $\lambda_j(M) = 1$ (เป็นออน-เซตของฟังก์ชัน Z_j) โดยจัดอยู่ในรูปของผลบวกของเทอมผลคูณเมื่อ $j \in 1,...,r$ โดย r เป็นจำนวนของตัวแปรสภาวะที่ได้จากความสัมพันธ์ $L^{(r-1)} \le L \le 2^r$ และ L คือจำนวนชั้นของตารางสภาวะ

3.5.4 การลดทอนของสมการทางตรรก

สมการตรรกของเอาต์พุตที่ได้จากออนเซตในตารางสภาวะ จัดอยู่ในรูปของ ผลบวกของเทอมผลคูณ ซึ่งเทอมผลคูณทั้งหมดที่แยกออกมาจากตารางสภาวะต่อไป เป็นมินเทอมทั้งหมด ดังนั้นจึงสามารถนำมาลดขนาดของสมการใช้วิธีการของ Quine-McCluskey ได้โดยตรงเนื่องจากการลดรูปสมการด้วยวิธีนี้ทุก ๆ เทอมต้องเป็นมินเทอมเท่านั้น ขั้นตอนของการลดรูปสมการมีอยู่ 2 ขั้นตอนคือ

1) การค้นหา Prime Cube (PC) โดยวิธีตารางเปรียบเทียบ
2) เลือก PC ที่ครอบคลุมมากที่สุดโดยวิธีตารางเปรียบเทียบ

น้ำหนัก (Weight) ของมินเทอมหมายถึงจำนวนของ Uncomplement ภายในมินเทอม ตัวอย่างเช่น มินเทอมที่ประกอบด้วย 5 ตัวแปร มินเทอมที่มีน้ำหนักเท่ากับ *0* คือ [*00000*] และมินเทอมที่มีน้ำหนักเท่ากับ 2 คือ [*01100*] มินเทอมหรือเรียกอีกอย่างว่า *0*-คิวบ์ คือเทอมที่มีตัวแปรครบทุกตัวแปร และถ้าเทอมที่มีตัวแปรขาดไปบางตัวแปรเช่น [*0*010*] เรียกว่า *1*-คิวบ์ และ [*0***0*] เรียกว่า *3*-คิวบ์ นั่นคือถ้ามินเทอมประกอบด้วย n ตัวแปร *3*-คิวบ์ คือเทอมผลคูณที่มีตัวแปรปรากฏอยู่แค่ $n-3$ ตัวจากตัวแปร n ตัว

การลดรูปสมการโดยใช้ Quine-McCluskey เป็นการสร้างตารางเปรียบเทียบระหว่างมินเทอมที่มีน้ำหนักของเทอมแตกต่างกัน 1 ระดับ มินเทอมทั้ง 2 เทอมที่นำมาเปรียบเทียบกันสามารถรวม (Combine) กันได้ถ้าค่าของตัวแปรภายในมินเทอมแตกต่างกันเท่ากับ 1 ตำแหน่งเช่น มินเทอม [*01011*] และ [*01001*] สามารถรวมกันได้และได้ผลลัพท์เป็น [*010*1*] โดยใช้กฎ $A.B+A'.B = B$ ดังนั้นมินเทอมที่

รวมกันได้จะสร้างเทอมที่เป็นเทอมผลคูณ *1*-คิวบ์ และถ้าเทอมผลคูณ *1*-คิวบ์ รวมกันได้ก็จะสร้างเทอมผลคูณ *2*-คิวบ์ และสามารถรวมต่อเนื่องไปจนถึง (*n-1*)-คิวบ์ กำหนดให้ M เป็นออนเซตของเอาต์พุตใด ๆ เท่ากับ $\{m_0, m_1, \ldots m_k\}$ และ T_0 คือตารางที่ประกอบด้วยเซตของออนเซตที่มีน้ำหนักเท่ากันคือ $(w_0, w_2, \ldots w_l)$ โดยที่ w_i เป็นเซตย่อยของ M ที่สมาชิกเซตมีน้ำหนักเท่ากับ i และ $i \in 0,1,\ldots,l$

ตาราง T_1 สร้างจากตาราง T_0 โดยการนำมินเทอม m_i เปรียบเทียบกับเทอม $m_{(i+1)}$ ถ้าทั้งสองเทอมรวมกันได้ก็จะเป็นเทอม *1*-คิวบ์ในตาราง T_1 มินเทอมในตาราง T_0 ที่ไม่สามารถรวมกับเทอมใดได้เลยคือ Prime Cube การสร้างตาราง T_2 มีขั้นตอนเหมือนกันกับการสร้างตาราง T_1 ทำการสร้างตารางจนกระทั่งไม่มีเทอมของผลคูณที่สามารถรวมกันได้อีก

ตาราง Prime Cube สร้างจาก Prime Cube ที่ได้ในตาราง $T_0, \ldots, T_{(n-1)}$ โดยกำหนดให้ P เป็นเซตของ Prime Cube นั่นคือ $P = \{p_0, p_1, \ldots, p_r\}$ ถ้า m_i เป็นมินเทอมที่รวมกันได้ Prime Cube p_j และ m_i ไม่มีรวมอยู่ใน Prime Cube ใด ดังนั้น p_j คือ Essential Prime Cube

p_i ที่มีสมาชิกมินเทอมเป็นเซตย่อยของเทอมอื่นเป็น Redundant Prime Cube (RPC) ตัดทิ้งและที่เหลือเป็น Selective Prime Cube (SPC) สมการบูลลีนของเอาต์พุตคือ EPC + SPC

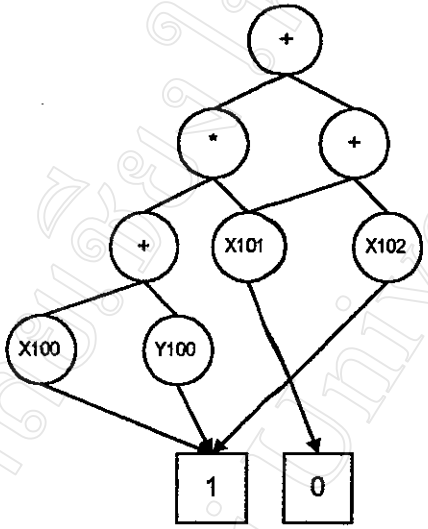
3.5.5 โครงสร้างข้อมูลบูลลีนเอ็กซ์เพรสชันทรี (Boolean Expression Tree)

โครงสร้างข้อมูล บูลลีนเอ็กซ์เพรสชันทรีเป็นทรี ใช้ในการเก็บสมการทางตรรกเพื่อประโยชน์ในการดำเนินการที่เกี่ยวกับสมการตรรก เช่นการแปลสมการตรรกเป็นคำสั่งสำหรับโปรแกรมพีแอลซี บูลลีนเอ็กซ์เพรสชันทรีเป็นทรีเป็นโครงสร้างข้อมูลแบบต้นไม้ (Tree) ที่เพิ่มเวอร์เท็กซ์พิเศษเข้าไปดังนิยามต่อไปนี้

นิยาม 3.1 บูลลีนเอ็กซ์เพรสชันทรีเป็นทรี (Tree) ที่ประกอบด้วยเวอร์เท็กซ์ (Vertex) ซึ่งแยกออกเป็น 3 ชนิดคือ เวอร์เท็กซ์ปลายทาง (Terminal Vertex) เวอร์เท็กซ์ตัวดำเนินการ (Operator Vertex) และเวอร์เท็กซ์ตัวแปร (*Variable Vertex*) แต่ละชนิดมีคุณสมบัติที่แตกต่างกันดังนี้

- เวอร์เท็กซ์ v เป็นเวอร์เท็กซ์ปลายทาง $\inf o(v) \in \{0,1\}$ ไม่มีเวอร์เท็กซ์ลูก (Children Vertex) $left(v), right(v) = \{\phi\}$ มีเวอร์เท็กซ์แม่ (Parent Vertex) ที่เป็นเวอร์เท็กซ์ตัวแปรเท่านั้น $\inf o(parent(v)) \in \{a, b, c, \ldots\}$
- เวอร์เท็กซ์ v เป็นเวอร์เท็กซ์ตัวดำเนินการ $\inf o(v) \in \{AND, OR\}$ มีเวอร์เท็กซ์ลูกทั้งทางซ้ายและทางขวาคือ $left(v), right(v) \in V$ ซึ่งมีคุณสมบัติ $\inf o(left(v)), \inf o(right(v)) \in \{AND, OR\} \cup \{a, b, c, \ldots.\}$ มีเวอร์เท็กซ์แม่หรือไม่มีก็ได้ถ้ามี $parent(v) \in V$ ถ้า $\inf o(parent(v)) \in \{\phi\}$ แสดงว่าเวอร์เท็กซ์ v เป็นรูทเวอร์เท็กซ์ของทรี ถ้าไม่เช่นนั้น $\inf o(parent(v)) \in \{AND, OR\}$

- เวอร์เท็กซ์ v เป็นเวอร์เท็กซ์ตัวแปร $\inf o(v) \in \{a,b,c,...\}$ มีเวอร์เท็กซ์ลูกที่เป็นเวอร์เท็กซ์ปลายทางเท่านั้น $\inf o(left(v)), \inf o(right(v)) \in \{0,1\}$ มีเวอร์เท็กซ์แม่ที่เป็นเวอร์เท็กซ์ตัวดำเนินการเท่านั้น $\inf o(parent(v)) \in \{AND, OR\}$



รูปที่ 3.16 บูลลีนเอ็กซ์เพรสชันทรี

3.5.6 การสร้างสมการตรรกแบบหลายระดับจากสมการตรรกสองระดับ

สมการที่ได้จากการสังเคราะห์เป็นสมการตรรกแบบสองระดับ ในขั้นตอนนี้จะทำการแปลงเป็นสมการตรรกแบบหลายระดับ ความสำคัญของการเปลี่ยนจากสมการตรรกแบบสองระดับเป็นสมการตรรกแบบหลายระดับคือ เพื่อลดจำนวนคำสั่งเมื่อสมการตรรกถูกแปลไปเป็นคำสั่งสำหรับพีแอลซี จากตารางแสดงให้เห็นจำนวนคำสั่งที่ลดลงเมื่อเปรียบเทียบระหว่างสมการตรรกแบบหลายระดับและสมการตรรกแบบสองระดับ

ตารางที่ 3.1 เปรียบเทียบรหัสนีโมนิกส์ของพีแอลซีระหว่างสมการตรรกสองระดับกับสมการตรรกแบบหลายระดับ

สมการตรรกแบบสองระดับ (Two Level Logic Function)	สมการตรรกแบบหลายระดับ (Multi Level Logic Function)
Y = A.B + A.C+B.C	Y = A.(B+C)+B.C
คำสั่ง	คำสั่ง
LD A	LD A
LD B	LD B
AND A	LD C
LD A	OR B
LD C	ANDLD
AND A	LD B
ORLD	LD C
LD B	AND B
LD C	ORLD
AND B	OUT Y
ORLD	END
OUT Y	
END	

การสร้างสมการตรรกแบบหลายระดับใช้กฎการกระจายของพีชคณิตบูลลีน [6]

$A.B+A.C = A.(B+C)$

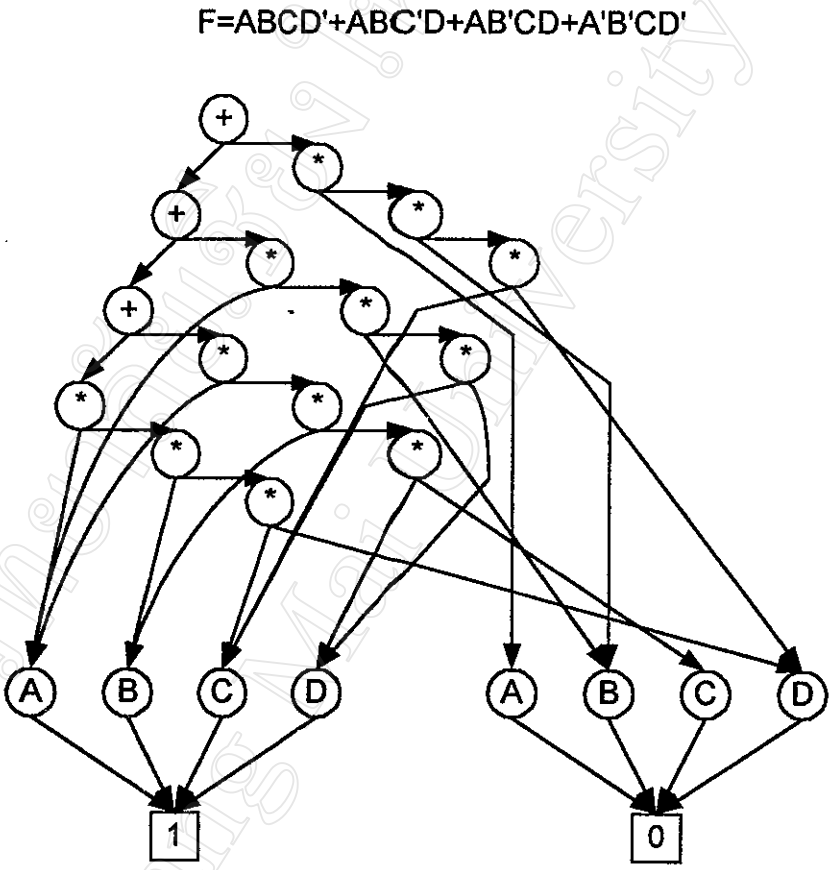
$A+1 = 1$

$A.1 = A$

โดยการประยุกต์ใช้บูลลีนเอ็กซ์เพรสชันทรีทำการปรับโครงสร้างเพื่อให้ได้ตามกฎการกระจายดังนิยามการปรับโครงสร้างดังนี้

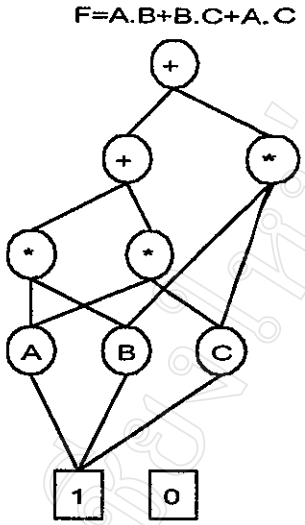
กำหนดให้ เทอมผลคูณในบูลลีนเอ็กซ์เพรสชันทรี คือซับบูลลีนเอ็กซ์เพรสชันทรี ที่มีรูทเวอร์เท็กซ์ v เป็นเวอร์เท็กซ์ตัวดำเนินการ AND นั่นคือ $\inf o(v) = AND$ และภายใต้ซับบูลลีนเอ็กซ์เพรสชั่นทรี เวอร์เท็กซ์ตัวดำเนินการทั้งหมดมีคุณสมบัติ $\inf o(v) = AND$

บูลลีนเอ็กซ์เพรสชันทรีของสมการตรรกสองระดับ คือบูลลีนเอ็กซ์เพรสชันทรีที่มีรูทเวอร์เท็กซ์ *v* มีคุณสมบัติ *info(v)=OR* และมีเวอร์เท็กซ์ลูกทางด้านขวาเป็นซับบูลลีนเอ็กซ์เพรสชันทรีของเทอมผลคูณ เวอร์เท็กซ์ลูกทางด้านซ้ายเป็นซับบูลลีนเอ็กซ์เพรสชันทรีของเทอมผลคูณเหมือนกันหรือเป็นซับบูลลีนเอ็กซ์เพรสชันทรีของสมการตรรกสองระดับเท่านั้น



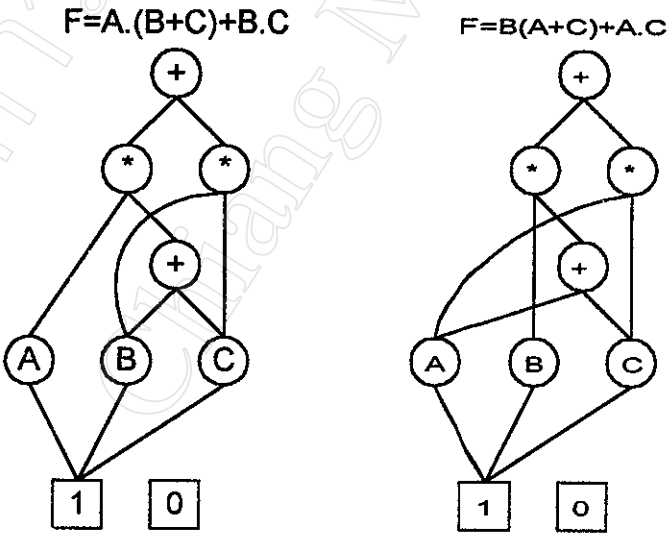
รูปที่ 3.17 แสดงบูลลีนเอ็กเพรสชั่นทรีของสมการตรรกสองระดับ

การสร้างสมการตรรกแบบหลายระดับจากสมการตรรกแบบสองระดับ เพื่อให้ได้คำสั่งบูลลีนจำนวนน้อย อาศัยความสัมพันธ์ F = D.Q+R [12] เมื่อ D คือเซตของตัวแปรที่แยกออกมาจากสมการเป็นเทอมผลคูณ Q คือเซตของเทอมผลคูณที่ถูกแยก D ออกไปจัดอยู่ในรูปของเทอมผลบวกของเทอมผลคูณ และ R เป็นเซตของเทอมผลคูณที่ยังคงอยู่และจัดอยู่แบบผลบวกของเทอมผลคูณ F = D+R ถ้าในเซตของเทอมผลคูณประกอบด้วยเทอมที่เท่ากับ D ใน F = D.Q + R



รูปที่ 3.18 แสดงบูลลีนเอ็กซ์เพรสชันทรีของฟังชัน F=A.B+B.C+A.C

การสร้างบูลลีนเอ็กซ์เพรสชันทรีของสมการตรรกหลายระดับจากบูลลีนเอ็กซ์เพรสชันทรีของสมการตรรกสองระดับอธิบายในบทที่ 4 หัวข้อที่ 4.2.3.2 รูปที่ 3.17 และ 3.18 แสดงบูลลีนเอ็กซ์เพรสชั่นทรีของสมการตรรกสองระดับ และบูลลีนเอ็กซ์เพรสชันทรีของสมการตรรกหลายระดับ



รูปที่ 3.19 แสดงสองทางเลือกของการจัดวางสมการหลายระดับ

3.5.7 การแปลสมการตรรกเป็นคำสั่งและรหัสคำสั่งของพีแอลซี

รหัสนีโมนิกส์ (Mnemonic Code) หรือรหัสคำสั่งบูลลีน (Boolean Instruction Code) ที่ใช้ในการโปรแกรมพีแอลซี ได้จากการท่อง (Travel) เข้าไปในโครงสร้างข้อมูลโดยการใช้ฟังก์ชันแบบการเรียกซ้ำ (Recursive Function) เข้าไปยังทุก ๆ เวอร์เท็กซ์ในบูลลีนเอ็กซ์เพรสชันทรีโดยเริ่มต้นที่รูทเวอร์เท็กซ์ (Root Vertex) เวอร์เท็กซ์ทุกตัวที่ผ่านเข้าไป ด้วยนิยามต่อไปนี้จะสามารถทำให้เราได้รหัสนีโมนิกส์สำหรับการโปรแกรมพีแอลซี และรหัสคำสั่ง (Instruction Code) สำหรับพีแอลซีรุ่น CMU-1 โดยวิธีเปิดตาราง (Lookup Table) ซึ่งจะกล่าวถึงในหัวข้อต่อไปจากนี้

นิยามที่ 3.2 คำสั่งบูลลีน I_v เกี่ยวข้องกับเวอร์เท็กซ์ v ในบูลลีนเอ็กซ์เพรสชันทรีโดยขึ้นอยู่กับชนิดของเวอร์เท็กซ์ดังต่อไปนี้

- กรณีที่ v เป็นเวอร์เท็กซ์ตัวแปร

$$I_v = \begin{cases} \text{info(v)} \; if \; \text{info(child(v))} = 1 \\ inverse(\text{info(v)}) \; if \; \text{info}(child(v)) = 0 \end{cases}$$

- กรณีที่ v เป็นเวอร์เท็กซ์ตัวดำเนินการ

เวอร์เท็กซ์ชนิดนี้จะมีเวอร์เท็กซ์ลูก (Children Vertex) อยู่ 2 ชนิดคือ เวอร์เท็กซ์ตัวดำเนินการด้วยกัน และเวอร์เท็กซ์ตัวแปร มีทางเลือก I_v ที่อาจจะเป็นไปได้ทั้งหมด 4 ทางเลือก ตามคณสมบัติของ v ดังนี้

เมื่อเวอร์เท็กซ์ v มี $left(v), right(v) \in \{AND, OR\}$

$$I_V = \begin{cases} I_{left}(v) \; I_{right}(v) \; ORL, & if \; \text{info}(v) = OR \\ I_{left}(v) \; I_{right}(v) \; ANDL, & if \; \text{info}(v) = AND \end{cases}$$

เมื่อเวอร์เท็กซ์ v มี $left(v) \in \{AND, OR\}, right(v) \in \{x_1, x_2, \ldots, x_n\}$

$$I_V = \begin{cases} I_{left(v)} \; OR \; \text{info}((right(v)), & if \; \text{info}(v) = OR \\ I_{left(v)} \; AND \; \text{info}((right(v)), & if \; \text{info}(v) = AND \end{cases}$$

เมื่อเวอร์เท็กซ์ v มี $left(v) \in \{x_1, x_2, \ldots, x_n\}, right(v) \in \{AND, OR\}$

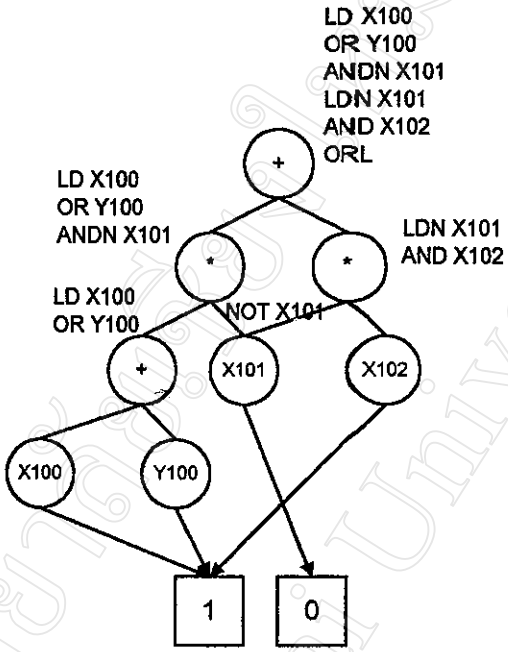
$$I_V = \begin{cases} LD \; I_{left}(v) \; I_{right}(v) \; ORL, & if \; \text{info}(v) = OR \\ LD \; I_{left}(v) \; I_{right}(v) \; ANDL, & if \; \text{info}(v) = AND \end{cases}$$

เมื่อเวอร์เท็กซ์ v มี $left(v), right(v) \in \{x_1, x_2, \ldots, x_n\}$

$$I_V = \begin{cases} LD \; I_{left}(v) \; OR \; I_{right}(v), & if \; \text{info}(v) = OR \\ LD \; I_{left}(v) \; AND \; I_{right}(v), & if \; \text{info}(v) = AND \end{cases}$$

เมื่อเวอร์เท็กซ์ v มี $left(v) \in \{x_1, x_2, \ldots, x_n\}, parent(v) = AND$

$$I_V = AND\, I_{left(v)}\ I_{right(v)}$$



รูปที่ 3.20 ตัวอย่างของบูลลีนเอ็กซ์เพรสชันทรีและคำสั่งในแต่ละเวอร์เท็กซ์

3.5.8 การกำหนดอุปกรณ์เสมือนภายในพีแอลซี

กระบวนการสังเคราะห์วงจรตรรกตามลำดับขั้นตอนต่าง ๆ ตัวแปรอินพุต เอาต์พุต สุดท้ายแล้ว ตัวแปรทุกตัวจะต้องกำหนดให้ตรงกับอุปกรณ์ภายในพีแอลซี โดยกำหนดดังนี้

1) ตัวแปรอินพุตกำหนดเป็น อินพุตภายนอกของพีแอลซี
2) ตัวแปรเอาต์พุตกำหนดเป็นเอาต์พุตภายนอกของพีแอลซี
3) ตัวแปรสภาวะกำหนดเป็นรีเลย์ช่วยภายในพีแอลซี
4) ตัวแปรตัวตั้งเวลากำหนดเป็นตัวตั้งเวลาภายในพีแอลซี
5) ตัวแปรตัวนับกำหนดเป็นตัวนับภายในพีแอลซี