การสร้างแบบจำลองด้วยโพรเมลาเพื่อทวนสอบซิกแนลแทรนซิชันกราฟในการสร้างวงจรอสมวาร

นายคณุตม์ บุญเรืองขาว



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2560

# A MODEL CONSTRUCTION BY PROMELA FOR SIGNAL TRANSITION GRAPH VERIFICATION I N ASYNCHRONOUS CIRCUIT IMPLEMENTATION

Mr. Kanut Boonroeangkaow



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2017

| | |
|---|---|
| หัวข้อวิทยานิพนธ์ | การสร้างแบบจำลองด้วยโพรเมลาเพื่อทวนสอบซิกแนลแทรนซิชันกราฟในการสร้างวงจรอสมวาร |
| โดย | นายคณุตม์ บุญเรืองขาว |
| สาขาวิชา | วิศวกรรมคอมพิวเตอร์ |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก | ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์ |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม | รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ |

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร. สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

(รองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ)

.....กรรมการ

(รองศาสตราจารย์ ดร.สมชาย ประสิทธิ์จูตระกูล)

.....กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร.เด่นดวง ประดับสุวรรณ)

คณุตม์ บุญเรืองขาว : การสร้างแบบจำลองด้วยโพรเมลาเพื่อทวนสอบซิกแนลแทรนซิชันกราฟในการสร้างวงจรอสมวาร (A MODEL CONSTRUCTION BY PROMELA FOR SIGNAL TRANSITION GRAPH VERIFICATION IN ASYNCHRONOUS CIRCUIT IMPLEMENTATION) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร.อาทิตย์ ทองทักษ์, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม: รศ. ดร.วิวัฒน์ วัฒนาวุฒิ, 152 หน้า.

การทวนสอบวงจรอสมวารนั้นมีความจำเป็นอย่างยิ่งในขั้นตอนการออกแบบเพื่อความถูกต้องในการทำงานของสัญญาณ โดยวงจรจะถูกออกแบบในขั้นต้นด้วยซิกแนลแทรนซิชันกราฟ วิทยานิพนธ์ใช้ประโยชน์จากเทคนิคการตรวจสอบแบบจำลองเพื่อทวนสอบซิกแนลแทรนซิชันกราฟในคุณสมบัติซึ่งประกอบด้วย คุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะสมบูรณ์ ซึ่งซิกแนลแทนซิชันกราฟประกอบด้วยประเภทวัฏจักรเชิงเดี่ยว และประเภทวัฏจักรหลากหลาย

ในขั้นแรกซิกแนลแทรนซิชันกราฟจะถูกแปลงเป็นรหัสโพรเมลาแบบจำลองโครงสร้าง และแบบจำลองวัฏจักรที่มีจุดยอดไม่ซ้ำกัน จากนั้นจึงนำซิกแนลแทรนซิชันกราฟไปแปลงเป็นตรรกะเวลาเชิงเส้นซึ่งประกอบด้วยคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์ จากนั้นคุณสมบัติความปลอดภัยจะทวนสอบโดยนำรหัสโพรเมลาแบบจำลองโครงสร้าง และตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัยไปทวนสอบโดยเครื่องมือสปินจะได้ผลการทวนสอบของคุณสมบัติ คุณสมบัติไลฟ์เนสจะทวนสอบโดยนำรหัสโพรเมลาแบบจำลองโครงสร้าง และตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนสไปทวนสอบโดยเครื่องมือสปินจะได้ผลการทวนสอบของคุณสมบัติ คุณสมบัติความทนทานจะทวนสอบโดยนำรหัสโพรเมลาแบบจำลองโครงสร้าง และตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทานไปทวนสอบโดยเครื่องมือสปินจะได้ผลการทวนสอบของคุณสมบัติ คุณสมบัติความต้องกันจะทวนสอบโดยนำรหัสโพรเมลาแบบแบบจำลองวัฏจักรที่มีจุดยอดไม่ซ้ำกัน และตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันไปทวนสอบโดยเครื่องมือสปินจะได้ผลการทวนสอบของคุณสมบัติ ในขั้นสุดท้ายคุณสมบัติการกำหนดสถานะที่สมบูรณ์จะนำรหัสโพรเมลาแบบแบบจำลองวัฏจักรที่มีจุดยอดไม่ซ้ำกัน และตรรกะเวลาเชิงเส้นของการกำหนดสถานะที่สมบูรณ์มาเพื่อหาความสัมพันธ์เชิงล็อคและทวนสอบโดยเครื่องมือสปิน จากนั้นจึงนำผลที่ได้จากการจำลองมาตรวจสอบในเครื่องมือที่พัฒนาขึ้นจึงได้คำตอบของการทวนสอบคุณสมบัตินี้ อย่างไรก็ตามเทคนิคของงานวิจัยนี้ยังไม่เป็นอัตโนมัติในบางคุณสมบัติ

| | |
|---|---|
| ภาควิชา วิศวกรรมคอมพิวเตอร์ | ลายมือชื่อนิสิต ..... |
| สาขาวิชา วิศวกรรมคอมพิวเตอร์ | ลายมือชื่อ อ.ที่ปรึกษาหลัก ..... |
| ปีการศึกษา 2560 | ลายมือชื่อ อ.ที่ปรึกษาร่วม ..... |

# # 5870390321 : MAJOR COMPUTER ENGINEERING

KEYWORDS: PROMELA, VERIFICATION, SIGNAL TRANSITION GRAPH, ASYNCHRONOUS CIRCUIT

KANUT BOONROEANGKAOW: A MODEL CONSTRUCTION BY PROMELA FOR SIGNAL TRANSITION GRAPH VERIFICATION IN ASYNCHRONOUS CIRCUIT IMPLEMENTATION. ADVISOR: ASST. PROF. ARTHIT THONGTAK, Ph.D., CO-ADVISOR: ASSOC. PROF. WIWAT VATANAWOOD, Ph.D., 152 pp.

Verification of asynchronous circuit is necessary in design phase, for the correctness of the signal operation. At First, the circuit will be designed by signal transition graph. This thesis exploits the model checking technique to verify signal transition graph in properties are safety liveness persistency consistency and Complete state coding (CSC).The types of signal transition graph are single- cycle and multi-cycle.

At First, Signal transition graph is converted to Promela code type model structure and model simple-cycle, then Signal transition graph is converted Linear temporal logic (LTL) that consist of Safety, liveness, persistency, consistency and CSC. Then, safety property will check by Promela type model structure and LTL of safety property, verified by SPIN, the result will be shown. Then, liveness property will check by Promela type model structure and LTL of liveness property, verified by SPIN, the result will be shown. Then, persistency property will check by Promela type model structure and LTL of persistency property, verified by SPIN, the result will be shown. Then, consistency property will check by Promela type model simple-cycle and LTL of consistency property, verified by SPIN, the result will be shown. At last, CSC property will check by Promela type model simple-cycle and LTL of CSC property for lock relation checking and verified by SPIN, Then , the simulation result from SPIN will be investigate in the developed tool. The result will answered by the tool. However, the limitation of this thesis technique is not automatic in some properties.



| | | |
|---|---|---|
| Department: | Computer Engineering | Student's Signature ..... |
| Field of Study: | Computer Engineering | Advisor's Signature ..... |
| Academic Year: | 2017 | Co-Advisor's Signature ..... |

# กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สาเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความกรุณาและความช่วยเหลือเป็นอย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.อาทิตย์ ทองทักษ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์หลักและรองศาสตราจารย์ ดร.วิวัฒน์ วัฒนาวุฒิ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ซึ่งท่านทั้งสองได้กรุณาให้ความรู้ คำแนะนำ คำปรึกษาและข้อคิดเห็นต่างๆ แก่ผู้วิจัยด้วยความตั้งใจและเอาใจใส่เป็นอย่างดีมาโดยตลอด จึงขอขอบพระคุณเป็นอย่างสูง ณ ที่นี้ด้วย

ขอขอบพระคุณรองศาสตราจารย์ ดร.ธาราทิพย์ สุวรรณศาสตร์ ประธานกรรมการสอบ รองศาสตราจารย์ ดร.สมชาย ประสิทธิ์จูตระกูล และผู้ช่วยศาสตราจารย์ ดร.เด่นดวง ประดับสุวรรณ กรรมการสอบที่กรุณาสละเวลา ให้คำแนะนำและชี้ให้เห็นถึงข้อบกพร่องต่างๆ ของงานวิจัยนี้

ขอขอบคุณ คุณปฏิมากร จริยฐิติพงศ์ คุณชานนท์ เดชสุภา และคุณปาณิสรา ดำจันทร์ นิสิตปริญญาเอกและปริญญาโท สังกัดห้องปฏิบัติการวิศวกรรมดิจิตอล ที่ได้ให้คำปรึกษาและความช่วยเหลือต่างๆ แก่ผู้วิจัยเป็นอย่างดีมาโดยตลอด

ขอขอบคุณสมาชิกห้องปฏิบัติการวิศวกรรมดิจิตอล ทุกท่านที่ไม่ได้กล่าวถึง การสัมมนารายสัปดาห์ ของห้องปฏิบัติการวิศวกรรมระบบดิจิตอล เป็นโอกาสที่ดีในการแลกเปลี่ยนความรู้ซึ่งกันและกัน ทำให้เกิดแนวคิดดีๆ ในการทำวิจัยมากขึ้น

ขอขอบคุณบุคคลที่เกี่ยวข้องทุกท่านที่ได้ให้ความช่วยเหลือ และคอยอำนวยความสะดวกในด้านต่างๆ ทำให้การทำงานวิจัยสำเร็จลุล่วงด้วยดี

ท้ายที่สุดนี้ผู้วิจัยขอขอบคุณบิดา มารดา ที่คอยสนับสนุน เป็นกำลังใจให้แก่ผู้วิจัยเป็นอย่างดีมาโดยตลอดจนงานวิจัยสามารถสำเร็จลุล่วงไปได้ด้วยดี

**สารบัญ**

หน้า

หน้า

หน้า

หน้า

# สารบัญภาพ

หน้า

# สารบัญตาราง

หน้า

# บทที่ 1
# บทนำ

## 1.1 ที่มาและความสำคัญของปัญหา

วงจรอสมวาร (Asynchronous Circuit) เป็นวงจรหนึ่งที่คาดว่าจะสามารถนำมาใช้ทดแทนวงจรสมวารได้ในอนาคต [1] เนื่องจากมีคุณสมบัติที่เป็นประโยชน์ต่อการใช้งาน เช่น การใช้พลังงานที่น้อยกว่า การรบกวนของสัญญาณที่มีน้อย และการทำงานที่ไม่ขึ้นกับสัญญาณนาฬิกาของระบบ เป็นต้น ซึ่งข้อดีเหล่านี้เป็นส่วนที่สามารถทำงานได้ดีกว่าวงจรสมวาร (Synchronous circuit) แต่ก็พบปัญหาบางอย่างตามมา เช่น การเปลี่ยนแปลงของสัญญาณที่อาจผิดพลาดได้ เนื่องจากไม่ต้องรอสัญญาณนาฬิกาของระบบ เป็นต้น การทวนสอบวงจรจึงมีความจำเป็นและเมื่อเทคโนโลยีการออกแบบวงจรอสมวารมีความซับซ้อนมากขึ้นทำให้การทวนสอบที่มีประสิทธิภาพมีความสำคัญมากขึ้น [2]

การออกแบบวงจรอสมวารนั้นจะเริ่มจากการออกแบบโดยแสดงความสัมพันธ์ของสัญญาณในรูปของซิกแนลแทรนซิชันกราฟ จากนั้นจึงสร้างกราฟแสดงสถานะจากซิกแนลแทรนซิชันกราฟและสังเคราะห์ความสัมพันธ์ที่ได้ไปเป็นวงจรอสมวาร [3, 4] การทวนสอบวงจรอสมวารสามารถทำได้โดยการทวนสอบระหว่างข้อกำหนดความต้องการ (Requirement Specification) และการออกแบบ (Design) หรือระหว่างข้อกำหนดความต้องการและวงจรที่ได้สังเคราะห์แล้ว (Implementation) การทวนสอบสามารถแบ่งได้เป็น 2 วิธี [5] คือ วิธีการทวนสอบแบบจำลองการทำงาน (Simulation Method) และวิธีการทวนสอบแบบรูปนัย (Formal Verification) การทวนสอบแบบรูปนัยเป็นวิธีที่สามารถครอบคลุมได้มากกว่า (Coverage) สำหรับการออกแบบวงจรอสมวาร ในปัจจุบันเครื่องมือที่ใช้ในการทวนสอบมีอยู่จำนวนหนึ่ง เช่น เครื่องมือสปิน (SPIN) เครื่องมือนัสเอ็มวี (NusMV) เครื่องมือแบลสต์ (Blast) [6] เป็นต้น แต่เครื่องมือที่สามารถกำหนดคุณสมบัติต่างๆ ในการทวนสอบได้อย่างมีประสิทธิภาพคือ เครื่องมือสปิน [6] โดยสามารถกำหนดคุณสมบัติในรูปของตรรกะเวลาเชิงเส้น (Linear Temporal Logic : LTL) โดยได้มีงานวิจัยได้เสนอการสังเคราะห์วงจรอสมวาร [4] และมีงานวิจัยได้เสนอวิธีการทวนสอบคุณสมบัติความทนทานโดยการเขียนเป็นภาษาโพรเมลา [7] และทวนสอบด้วยเครื่องมือสปินซึ่งสร้างแบบจำลองโครงสร้างภาษาโพรเมลาเป็นแบบสถิตทำให้การจำลองด้วยวิธีนี้ยังมีข้อจำกัดในการทวนสอบซิกแนลแทรนซิชันกราฟที่ซับซ้อน เช่น ซิกแนลแทรนซิชันกราฟประเภทวัฏจักรหลากหลาย เป็นต้น เนื่องจากวิธีการยังไม่เป็นอัตโนมัติและไม่สามารถแสดงการทำงานในช่วงเวลาต่างๆ ได้ ซึ่งงานวิจัยดังกล่าวเป็นการทวนสอบระหว่างข้อกำหนดความต้องการและวงจรที่ได้สังเคราะห์แล้ว อีกงานวิจัยหนึ่งได้นำเสนอการทวนสอบการทำงานของระบบเครือข่าย

ไร้สายซึ่งอธิบายระบบด้วยแผนภาพเพทริเนตและแปลงแผนภาพเพทริเนตเป็นรหัสโพรเมลา [8] โดยรหัสโพรเมลาที่ได้เสนอนั้นสามารถจำลองการทำงานแบบพลวัตได้ แต่เนื่องด้วยงานวิจัยไม่ได้เสนอการทำงานของระบบวงจรอสมวารซึ่งมีเงื่อนไขต่างกัน การทวนสอบจึงมีความต่างกัน

ดังนั้นงานวิจัยนี้จึงมุ่งเน้นไปที่การเสนอวิธีสร้างแบบจำลองด้วยโพรเมลาเพื่อทวนสอบซิกแนลแทรนซิชันกราฟโดยใช้เครื่องมือสปิน วิธีการสร้างแบบจำลองเป็นการแปลงซิกแนลแทรนซิชันกราฟเป็นโพรเมลาที่ครอบคลุมการทำงานแบบพลวัต พร้อมกับเสนอวิธีการสร้างตรรกะเวลาเชิงเส้นของการทวนสอบซึ่งใช้วิธีการทวนสอบแบบรูปนัยซึ่งประกอบด้วย 5 คุณสมบัติ คือ คุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน คุณสมบัติการกำหนดสถานะที่สมบูรณ์ และพัฒนาเครื่องมือสนับสนุนตามวิธีที่นำเสนอ โดยในบทที่ 2 เสนอทฤษฎีที่เกี่ยวข้องและงานวิจัยที่เกี่ยวข้อง บทที่ 3 เสนอวิธีการแปลงซิกแนลแทรนซิชันกราฟเป็นรหัสโพรเมลาและตรรกะเวลาเชิงเส้น บทที่ 4 เสนอการพัฒนาเครื่องมือและการใช้งาน บทที่ 5 การทวนสอบตัวอย่างวงจรอสมวาร บทที่ 6 เสนอสรุปผลวิจัยและข้อเสนอแนะ

## 1.2 วัตถุประสงค์ของงานวิจัย

เพื่อเสนอวิธีการสร้างแบบจำลองด้วยโพรเมลาและเสนอกฎการสร้างตรรกะเวลาเชิงเส้นในการทวนสอบเชิงรูปนัยสำหรับซิกแนลแทรนซิชันกราฟด้วยเครื่องมือสปิน

## 1.3 ขอบเขตการวิจัย

1) ซิกแนลแทรนซิชันกราฟประกอบด้วยประเภทวัฏจักรเชิงเดี่ยว และวัฏจักรหลากหลาย
2) เครื่องมือที่ใช้ในการทวนสอบคือ Spin version 6.4.5 ขึ้นไป
3) สร้างเครื่องมือในการแปลงซิกแนลแทรนซิชันกราฟเป็นภาษาโพรเมลาและเครื่องมือในการสร้างตรรกะเชิงเวลา
4) การทวนสอบซิกแนลแทรนซิชันกราฟประกอบด้วย 5 คุณสมบัติ คือ คุณสมบัติไลฟ์เนส คุณสมบัติความปลอดภัย คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์
5) คุณสมบัติที่ทวนสอบนำมาแปลงเป็นตรรกะเชิงเวลาที่สามารถนำเข้าเครื่องมือสปินได้อย่างน้อย 4 คุณสมบัติ ประกอบด้วยคุณสมบัติไลฟ์เนส คุณสมบัติความปลอดภัย คุณสมบัติความทนทาน และคุณสมบัติความต้องกัน
6) กรณีศึกษาของซิกแนลแทรนซิชันกราฟที่ศึกษามี 3 ตัวอย่างขึ้นไป

## 1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1) ศึกษางานที่เกี่ยวข้อง
   - ศึกษาและทำความเข้าใจทฤษฎีเกี่ยวกับซิกแนลแทรนซิชันกราฟ
   - ศึกษาทฤษฎีการทวนสอบและเครื่องมือที่ใช้ในการทวนสอบ
   - ศึกษางานวิจัยที่เกี่ยวข้อง
2) กำหนดแนวคิดและความสามารถ
   - แนวคิดวิธีการทวนสอบคุณสมบัติและซิกแนลแทรนซิชันกราฟแต่ละประเภท
   - แนวคิดแบบโพรเมลาของแต่ละคุณสมบัติและซิกแนลแทรนซิชันกราฟแต่ละประเภท
   - แนวคิดตรรกะเวลาเชิงเส้นของแต่ละคุณสมบัติและซิกแนลแทรนซิชันกราฟแต่ละประเภท
3) ออกแบบเครื่องมือสนับสนุน
   - ออกแบบส่วนต่อประสานกับผู้ใช้
   - ออกแบบโครงสร้างเครื่องมือ
4) พัฒนาเครื่องมือสำหรับการสร้างแบบโพรเมลา ตรรกะเวลาเชิงเส้นของแต่ละคุณสมบัติจากข้อมูลนำเข้าการอธิบายซิกแนลแทรนซิชันกราฟมาตรฐาน
5) ทดสอบเครื่องมือที่ถูกพัฒนาขึ้น
6) ปรับปรุงแก้ไขเครื่องมือ
7) สรุปและประเมินผลการทดสอบ
8) จัดทำวิทยานิพนธ์นำเสนอบทความวิชาการ



## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) วิธีการสร้างแบบแบบจำลองด้วยโพรเมลาและกฎการสร้างตรรกะเวลาเชิงเส้นในการทวนสอบเชิงรูปนัยสำหรับซิกแนลแทรนซิชันกราฟ
2) เครื่องมือสำหรับสร้างแบบโพรเมลาและตรรกะเวลาเชิงเส้นสำหรับการทวนสอบด้วยเครื่องมือสปิน

**1.6 บทความวิชาการที่ได้รับการตีพิมพ์**

1) งานวิจัยนี้ได้รับคัดเลือกและตีพิมพ์เป็นบทความวิชาการเรื่อง “Formal Modeling for Persistence Checking of Signal Transition Graph Specification with Promela” โดย คณุตม์ บุญเรืองขาว, อาทิตย์ ทองทักษ์ และวิวัฒน์ วัฒนาวุฒิ ในการประชุมวิชาการ “*The 25th International MultiConference of Engineers and Computer Scientists* (IMECS 2017)” ระหว่างวันที่ 15-17 มีนาคม 2560 ณ โรงแรม เดอะ รอยัล การ์เดน เมืองเกาลูน เขตบริหารพิเศษฮ่องกงแห่งสาธารณรัฐประชาชนจีน

2) งานวิจัยนี้ได้รับคัดเลือกและตีพิมพ์เป็นบทความวิชาการเรื่อง “Formal Modeling for Consistency Checking of Signal Transition Graph” โดย คณุตม์ บุญเรืองขาว, อาทิตย์ ทองทักษ์ และ วิวัฒน์ วัฒนาวุฒิ ในการประชุมวิชาการ *“The 11th International Conference on Telecommunication Systems, Services, and Applications (TSSA) 2017”* ระหว่างวันที่ 26-27 ตุลาคม 2560 ณ โรงแรม คีล่า เซงกิกิ บีช ลอมบอก เมืองลอมบอก ประเทศอินโดนีเซีย

# บทที่ 2
# ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

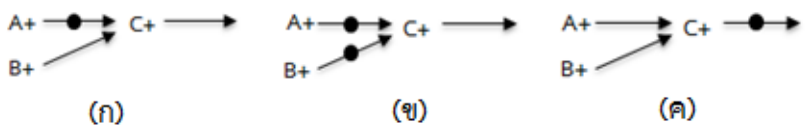
## 2.1 ทฤษฎีที่เกี่ยวข้อง

### 2.1.1 ซิกแนลแทรนซิชันกราฟ (Signal Transition Graphs) [3, 4, 7]

ซิกแนลแทรนซิชันกราฟ เป็นกราฟประเภทหนึ่งที่ถูกลดรูปลงจากเพทริเนตเพื่อให้สามารถใช้ในการจำลองพฤติกรรมของระบบ อาทิเช่น เครือข่ายพันธุกรรม และวงจรอสมวาร เป็นต้น ซิกแนลแทรนซิชันกราฟถูกเสนอโดย Tam-Anh Chu ซิกแนลแทรนซิชันกราฟเป็นข้อกำหนดพฤติกรรมแบบเป็นทางการของวงจรที่เกี่ยวข้องกับการทำงานที่เป็นระเบียบ การทำงานพร้อมกันและการทำงานที่ขัดกัน ในวงจรดิจิตอลนั้นจะประกอบด้วยขาเข้าและขาออกซึ่งมีการเชื่อมกัน เซตของการเชื่อมต่อกันนั้นเรียกว่า เซตของสัญญาณ ให้ J แทนเซตของสัญญาณ J สามารถแบ่งได้เป็นสัญญาณขาเข้าแทนด้วย $J_i$ และสัญญาณขาออกแทนด้วย $J_o$ และสัญญาณภายในแทนด้วย $J_n$

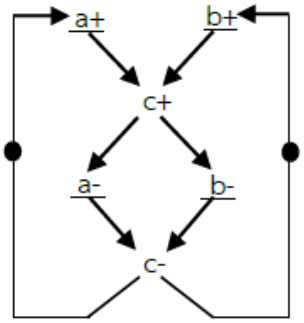
เซตของแทรนซิชันแทนด้วย T กำหนดโดย $T = J \times \{+, -\}$ โดยทุกๆสัญญาณ $j \in J$ และมีคู่ของการเปลี่ยนแปลงสัญญาณ {j+, j-} เกี่ยวข้อง แทรนซิชัน T สามารถแบ่งได้เป็น $T_i = J_i \times \{+, -\}$, $T_n = J_n \times \{+, -\}$, $T_o = J_o \times \{+, -\}$ ในการจำแนกความแตกต่างของสัญญาณขาเข้า $T_i$ จะใช้การขีดเส้นใต้

ซิกแนลแทรนซิชันกราฟประกอบด้วยสัญญาณที่มีการเปลี่ยนแปลงหรือแทรนซิชันและลูกศรที่เชื่อมต่อแทรนซิชันแสดงเป็นโครงสร้างเชิงสถิตของระบบพร้อมกับสามารถแสดงพฤติกรรมเชิงพลวัตด้วยการวางตำแหน่งโทเค็น (จุดดำ) ลงบนลูกศรซึ่งเรียกว่า มาร์กกิ้ง (M) โดยการกำหนดมาร์กกิ้งตั้งต้น ($M_0$) เป็นสถานะเริ่มต้นของซิกแนลแทรนซิชันกราฟ จากนั้นพฤติกรรมของซิกแนลแทรนซิชันกราฟจะดำเนินการโดยพิจารณาสถานะเชิงพลวัตที่แสดงความพร้อมในการเปลี่ยนแปลงสัญญาณโดยดูจากโทเค็นที่ปรากฏบนลูกศรขาเข้าของสัญญาณที่จะเปลี่ยนแปลงซึ่งสามารถแสดงได้ 3 ประเภทดังรูปที่ 2-1




รูปที่ 2-1 สถานะการเปลี่ยนแปลงสัญญาณ (ก) สถานะไม่พร้อมเปลี่ยนสัญญาณ (Disabled) (ข) สถานะพร้อมเปลี่ยนสัญญาณ (Enabled) (ค) สถานะหลังเปลี่ยนสัญญาณ (Firing)

สัญญาณที่มีโทเค็นไม่ครบจะอยู่ในสถานะไม่พร้อมเปลี่ยนแปลงสัญญาณดังรูปที่ 2-1(ก) จะไม่เกิดการเปลี่ยนแปลงสัญญาณ สัญญาณที่มีโทเค็นในสายสัญญาณขาเข้าครบทุกเส้นดังรูปที่ 2-1(ข) จะมีความพร้อมในการเปลี่ยนสัญญาณ หลังจากนั้นโทเค็นจะเคลื่อนย้ายจากเส้นสายสัญญาณขาเข้ากลายเป็นสถานะหลังเปลี่ยนสัญญาณดังรูปที่ 2-1(ค)



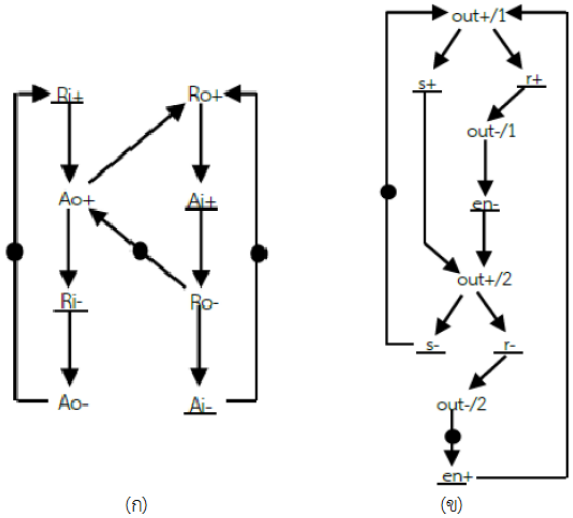

รูปที่ 2-2 ตัวอย่างซิกแนลแทรนซิชันกราฟ

จากรูปที่ 2-2 เซตของสัญญาณประกอบด้วย J = {a, b, c}, $J_i$ = {a, b}, $J_n$ = {c} และ เซตของแทรนซิชัน T = {a+, a-, b+, b-, c+, c-}, $T_i$ = {a+, a-, b+, b-}, $T_n$ = {c+, c-} และมาร์กกิ้งตั้งต้นอยู่ที่ c- ซึ่งมี 2 โทเค็น

ซิกแนลแทรนซิชันกราฟที่ใช้ในงานวิจัยนี้แบ่งเป็นซิกแนลแทรนซิชันกราฟประเภทต่างๆ ดังนี้

1) วัฏจักรเชิงเดี่ยว (Single-cycle) คือ ซิกแนลแทรนซิชันกราฟที่ประกอบด้วยสัญญาณต่างๆ ที่มีค่าบวกและลบและสามารถเกิดขึ้นได้เพียงครั้งเดียว ตามตัวอย่างดังรูปที่ 2-3 (ก)

2) วัฏจักรหลากหลาย (Multi-cycle) คือ ซิกแนลแทรนซิชันกราฟที่ประกอบด้วยสัญญาณต่างๆที่มีค่าบวกและลบและเกิดขึ้นมากกว่าหนึ่งครั้ง ตามตัวอย่างดังรูปที่ 2-3 (ข)




รูปที่ 2-3 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟประเภทวัฏจักรเชิงเดี่ยว

(ข) ตัวอย่างซิกแนลแทรนซิชันกราฟประเภทวัฏจักรหลากหลาย

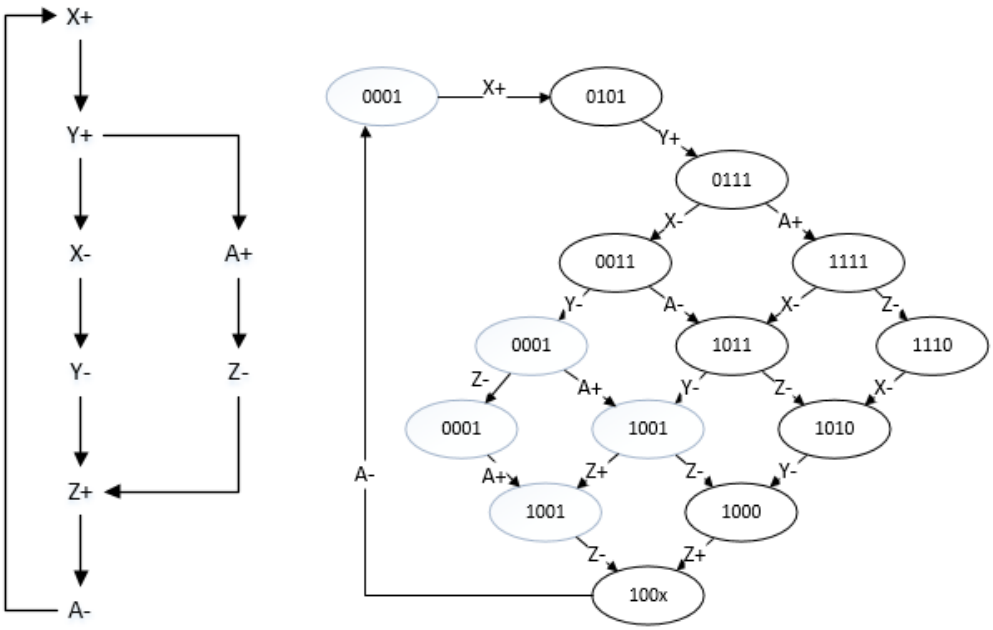
### 2.1.2 กราฟสถานะ (State Diagram) [3]

กราฟสถานะเป็นกราฟที่ใช้แสดงสถานะของสัญญาณในซิกแนลแทรนซิชันกราฟด้วยการแสดงแบบไบนารีและแสดงเป็นเวกเตอร์ในวงจร โดยกราฟสถานะประกอบด้วยสถานะแทนด้วย s โดยในทุกๆสัญญาณที่ j $\in$ J ให้ s(j) เป็นค่าของสัญญาณ j ในสถานะ s และ j แสดงแทรนซิชันของ j (j+, j-) และให้ s[t>s' ยืนยันว่าการเกิดขึ้นของแทรนซิชัน t ในสถานะ s ทำให้ระบบมีการเปลี่ยนสถานะ สามารถบอกได้ว่า t ทำงานได้ในสถานะ s และ s' เป็นสถานะถัดมาของ s

กราฟสถานะสามารถนิยามได้ดังนี้ ให้ J = {$j_1$, $j_2$, ..., $j_n$} เป็นเซตของสัญญาณของวงจร กราฟสถานะสามารถนิยามได้บน J โดยให้ $\Phi$j = (S, T, $\sigma$, $s_0$ )

โดยให้ S เป็นเซตของสถานะ กำหนดโดย S = { s | s : J -> {0, 1}} ทุก s $\in$ S

- $s_0$ $\in$ S เป็นสถานะเริ่มต้นของวงจร
- T = J x {+, -} เป็นเซตของแทรนซิชัน
- $\sigma$: S x T -> S เป็นฟังก์ชันแทรนซิชัน มีคุณสมบัติ $\forall$s, s' $\in$ S, $\forall$t $\in$ T โดยที่ $\sigma$(s, t) = s'
- ถ้า t= j+ แล้ว s(j) = 0 และ s' (j) = 1
- ถ้า t= j- แล้ว s(j) = 1 และ s' (j) = 0
- s’ เป็นสถานะต่อไปของ s




รูปที่ 2-4 ตัวอย่างซิกแนลแทรนซิชันกราฟและกราฟสถานะ

จากรูปที่ 2-4 รูปด้านซ้ายมือแสดงตัวอย่างของซิกแนลแทรนซิชันกราฟและรูปด้านขวามือแสดงกราฟสถานะจากการเปลี่ยนแปลงสัญญาณของซิกแนลแทรนซิชันกราฟ

**2.1.3 คุณสมบัติของซิกแนลแทรนซิชันกราฟสำหรับการสร้างวงจรอสมวารได้อย่างถูกต้อง [3]**

ซิกแนลแทรนซิชันกราฟมีคุณสมบัติที่ใช้ในการรับรองความถูกต้องมีจำนวน 5 ข้อดังนี้

1) คุณสมบัติไลฟ์เนส (Liveness) หมายถึง การที่ทุกๆจุดของกราฟจะต้องสามารถเข้าถึงได้เมื่อมีการเริ่มต้นกราฟใหม่ และไลฟ์เนสยังรวมถึงทุกๆ แทรนซิชันจะต้องสามารถเปลี่ยนแปลงได้ตลอด เพราะมีอย่างน้อย 1 สถานะที่สามารถเข้าถึงได้ โดยสามารถสรุปเงื่อนไขของไลฟ์เนสได้ดังนี้

- เซตของสัญญาณ J, $\Phi j = (S, T, \sigma, s_0)$ มีคุณสมบัติไลฟ์เนสก็ต่อเมื่อมีการเชื่อมต่อที่แข็งแกร่ง และทุกๆ $t \in T$ มี $s \in s_0$ ที่ทำให้ s(t)

ในลูกศรทุกๆ เส้นจะต้องสามารถใช้งานได้ตลอดเวลาเมื่อมีการเรียกใช้ ดังตัวอย่างจากรูปที่ 2-5(ก) ลูกศรที่มาจากสัญญาณ r+ เมื่อวิ่งผ่านไปรอบหนึ่งแล้วจะไม่สามารถเรียกใช้ได้อีกจึงไม่มีคุณสมบัติไลฟ์เนส หากซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติไลฟ์เนสแล้ว เมื่อวงจรทำงานจะเกิดการหยุดนิ่งไม่ทำงาน (Deadlock)

2) คุณสมบัติความปลอดภัย (Safety) หมายถึง การที่ในแต่ละลูกศรของซิกแนลแทรนซิชันกราฟจะต้องมีโทเค็นไม่เกินหนึ่งโทเค็น ดังตัวอย่างจากรูปที่ 2-5 (ข) ลูกศรจากสัญญาณ r+ และสัญญาณ x+ มีสองโทเค็นอยู่จึงไม่มีคุณสมบัติความปลอดภัย หากซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติความปลอดภัยแล้ว เมื่อวงจรทำงานจะเกิดการเปลี่ยนแปลงสัญญาณมากเกินกำหนด

3) คุณสมบัติความทนทาน (Persistency) หมายถึง การที่สัญญาณตัวต่อไปสามารถเปลี่ยนได้เมื่อครบตามเงื่อนไขที่กำหนด โดยมีเงื่อนไขดังนี้

- ทุกๆ $t \in Ti$ มีคุณสมบัติความทนทาน
- $t \in Tn$ มีคุณสมบัติความทนทาน ถ้า $\forall s, s' \in S, \forall t' \in T : s[t> \wedge s[t'>s' \rightarrow s'[t>$
- ทุกๆแทรนซิชันใน T มีคุณสมบัติความทนทาน

ตัวอย่างจากรูปที่ 2-5 (ค) สัญญาณ y+ ไม่สามารถเปลี่ยนได้ตามเงื่อนไขหากว่าสัญญาณ x- เกิดขึ้นก่อน เพราะค่าของ x จะกลายเป็น 0 ทำให้โทเค็นที่ x+ หายไป จึงไม่มีคุณสมบัติความทนทาน หากซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติความทนทานแล้วเมื่อวงจรทำงาน การเปลี่ยนแปลงสัญญาณจะไม่เป็นไปตามเงื่อนไขที่กำหนด

4) คุณสมบัติความต้องกัน (Consistency) หมายถึง การที่เมื่อมีการเคลื่อนย้ายโทเค็นในซิกแนลแทรนซิชันกราฟที่เกี่ยวข้องกับสัญญาณนั้นๆ แล้วค่าของสัญญาณนั้นต้องเปลี่ยนแปลง มีเงื่อนไขดังนี้

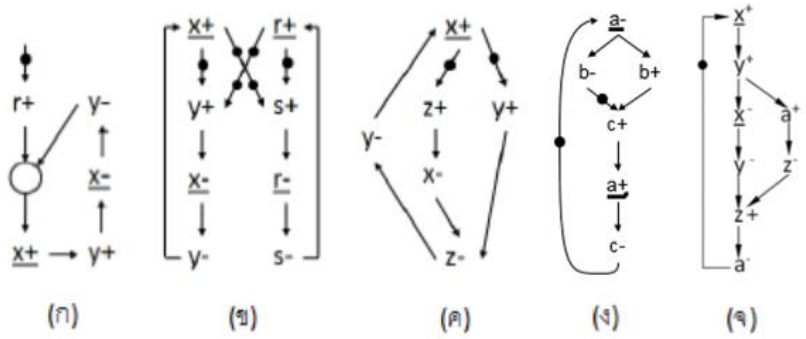
- ให้ ∑j เป็น ซิกแนลแทรนซิชันกราฟ และ $\Phi_j$ เป็นกราฟสถานะของซิกแนลแทรนซิชันกราฟ $\Phi_j$ จะมีคุณสมบัติความต้องกันถ้าทุกๆ คู่ของแทรนซิชัน t, t' อยู่ใน ∑j

ดังตัวอย่างจากรูปที่ 2-5 (ง) เมื่อเกิดการเคลื่อนย้ายโทเค็นที่สัญญาณ b+ แล้วค่าของ b ไม่สามารถเปลี่ยนเป็น b- ได้เนื่องจากไม่มี b และ b’ ใน ∑j จึงไม่มีคุณสมบัติความต้องกัน หากซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติความต้องกันแล้ว เมื่อวงจรทำงานสัญญาณบางสัญญาณจะไม่สามารถเปลี่ยนค่าได้

5) คุณสมบัติการกำหนดสถานะที่สมบูรณ์ (Complete State Code, CSC) หมายถึง การที่ทุกๆ เพลสในซิกแนลแทรนซิชันกราฟจะต้องไม่มีสัญลักษณ์ซ้ำกัน มีเงื่อนไขดังนี้

- ทุกสถานะในกราฟจะต้องมีรหัสไบนารีที่แตกต่างกัน
- เมื่อสองสถานะหรือมากกว่านั้นมีค่าเหมือนกัน สัญญาณภายในและสัญญาณขาออกทั้งหมดของแทรนซิชันที่ถูกใช้งานในสถานะนี้จะต้องเหมือนกัน

ดังตัวอย่างจากรูปที่ 2-5 (จ) มีสถานะ (a, x, y, z) เป็นค่า 1001 และ 0001 ซ้ำกันมากกว่า 1 ครั้ง จึงไม่มีคุณสมบัติการกำหนดสถานะที่สมบูรณ์ หากซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติการกำหนดสถานะที่สมบูรณ์แล้ว เมื่อวงจรทำงานและเกิดการเปลี่ยนแปลงสัญญาณสถานะของวงจรจะไม่ถูกต้องตามที่กำหนดไว้




รูปที่ 2-5 ตัวอย่างซิกแนลแทรนซิชันกราฟที่ไม่มีคุณสมบัติ (ก) Liveness (ข) Safety (ค) Persistency (ง) Consistency (จ) Complete State Coding

### 2.1.4 เครื่องมือสปิน (SPIN) [6, 9]

เครื่องมือสปินเป็นเครื่องมือโอเพนซอร์สที่เป็นที่นิยมในด้านการทวนสอบซอฟต์แวร์ถูกใช้อย่างแพร่หลาย โดยสามารถใช้สำหรับการทวนสอบอย่างมีแบบแผนและแบบมัลติเธรด (Multithread) เครื่องมือสปินถูกพัฒนาขึ้นที่เบลล์แล็บ โดยทีมซึ่งนำโดย Gerard J. Holzmann ภาษาที่ใช้ในเครื่องมือสปิน คือ ภาษาโพรเมลา (Promela) ซึ่งคล้ายกับภาษาที่ใช้เขียนโปรแกรม สปินสามารถตรวจสอบคุณสมบัติด้วยตรรกะเวลาเชิงเส้น

### 2.1.5 ภาษาโพรเมลา (Promela) [6, 9]

ภาษาโพรเมลาเป็นภาษาที่ใช้สำหรับการเขียนแบบจำลองเพื่อการทวนสอบแบบจำลอง ภาษาโพรเมลา สามารถวิเคราะห์ได้โดยใช้เครื่องมือสปิน โปรแกรมที่เขียนโดยภาษาโพรเมลาประกอบด้วย กระบวนการ (Process), ตัวแปร (Variable) และช่องทางการส่งผ่านข้อมูล (Message channel) กระบวนการเป็นวัตถุที่สามารถเรียกได้จากทุกส่วนของโปรแกรมที่แทนการทำงานเป็นเอกลักษณ์แบบพร้อมกันของระบบ ช่องทางการส่งผ่านข้อมูลและตัวแปรสามารถประกาศเป็นทั่วไปหรือเฉพาะภายในก็ได้ กระบวนการจะเป็นตัวกำหนดพฤติกรรม ช่องทางการส่งผ่านข้อมูลและตัวแปรกำหนดสภาพแวดล้อมที่กระบวนการทำงาน

**1) ชนิดของข้อมูลในภาษาโพรเมลา**

ตัวแปรในภาษาโพรเมลาสามารถกำหนดได้หลายประเภท และสามารถประกาศตัวแปรสามารถประกาศเป็นอาร์เรย์ได้ ดังแสดงในตารางที่ 2-1 และ 2-2

ตารางที่ 2-1 ชนิดของข้อมูลในภาษาโพรเมลา

| ชื่อ | ขนาด(บิต) | Usage | ค่าของข้อมูล |
|---|---|---|---|
| bit | 1 | unsigned | 0, 1 |
| bool | 1 | unsigned | 0, 1 |
| byte | 8 | unsigned | 0-255 |
| mtype | 8 | unsigned | 0-255 |
| short | 16 | signed | $-2^{15}$ - $2^{15}$ -1 |
| int | 32 | signed | $-2^{31}$ - $2^{31}$ -1 |



ตารางที่ 2-2 ตัวอย่างการประกาศอาร์เรย์

| คำสั่ง | คำอธิบาย |
|---|---|
| int x[10]; | สร้างอาร์เรย์ชนิด int ขนาด 10 ช่อง |
| int x[3]; | สร้างอาร์เรย์ชนิด int ขนาด 3 ช่อง |
| x[0] = 1; | ช่องแรกกำหนดค่า 1 |
| x[1] = 2; | ช่องที่สองกำหนดค่า 2 |
| x[2] = 3; | ช่องที่สามกำหนดค่า 3 |

**2) กระบวนการ (processes)**

พฤติกรรมของ processes กำหนดโดยใช้ proctype ตามตัวอย่างดังตารางที่ 2-3

ตารางที่ 2-3 ตัวอย่างการสร้างกระบวนการ

| คำสั่ง | คำอธิบาย |
|---|---|
| proctype A()<br>{<br>byte state;<br>state = 3;<br>} | สร้าง proctype ที่มีชื่อ A โดยภายในประกาศตัวแปรชื่อ state และกำหนดค่าเป็น 3 |

กระบวนการที่แสดงนี้เป็นเพียงการกำหนดพฤติกรรมเท่านั้นไม่สามารถดำเนินการได้ โดยโปรแกรมจะดำเนินการจากกระบวนการชื่อ init

**3) โครงสร้างอะตอมมิกซ์ (atomic construct)**

โครงสร้างอะตอมมิกซ์เป็นโครงสร้างที่มีลักษณะพิเศษในการทำงานดังอธิบายในตารางที่ 2-4

ตารางที่ 2-4 ตัวอย่างโครงสร้างอะตอมมิกซ์

| คำสั่ง | คำอธิบาย |
|---|---|
| Atomic<br>{<br>statements;<br>} | สร้างโครงสร้างอะตอมมิกซ์และคำสั่งการทำงานภายใน โดยระบบจะทำคำสั่งในอะตอมมิกซ์ให้เสร็จก่อน โดยจะไม่ข้ามไปทำคำสั่งหรือกระบวนการอื่นจนกว่าจะทำในอะตอมมิกซ์เสร็จ |



**4) โครงสร้างควบคุมทิศทางการไหลของโปรแกรม (Control flow constructs)**

การควบคุมทิศทางการทำงานของโปรแกรมสามารถทำได้สองแบบ คือ โครงสร้างทางเลือกและการทำซ้ำ ดังตารางที่ 2-5 และ 2-6

ตารางที่ 2-5 ตัวอย่างการใช้งานตัวควบคุมแบบโครงสร้างทางเลือก

| คำสั่ง | คำอธิบาย |
|---|---|
| If<br>:: (A == true) -> option1;<br>:: (B == true) -> option2;<br>:: else -> falthrough_option;<br>fi | โปรแกรมจะทำ option1 ถ้า A เป็นจริง และจะทำ option2 ถ้า B เป็นจริงและจะทำ falthrough_option ถ้า A และ B ไม่จริง โดยการใช้งานเงื่อนไขแบบนี้นั้นถ้าโปรแกรมไม่สามารถทำอันใดได้เลยโปรแกรมจะไม่ไปต่อ |

ตารางที่ 2-6 ตัวอย่างการใช้งานตัวควบคุมแบบทำซ้ำ

| คำสั่ง | คำอธิบาย |
|---|---|
| Do<br>:: count = count +1<br>:: a = b + 2<br>:: (count == 0) -> break<br>Od | โปรแกรมจะคำสั่งไปเรื่อยๆ ตาม guard จนกว่าจะทำใน guard ไม่ได้หรือเจอคำสั่ง break |

**5) การตรวจสอบข้อผิดพลาด**

การตรวจสอบข้อผิดพลาดวิธีหนึ่งที่สามารถทำได้ในภาษาโพรเมลาโดย ไม่ได้ใช้ตรรกะเวลาเชิงเส้นคือการใช้คำสั่ง assert ดังตัวอย่างตามตารางที่ 2-7

ตารางที่ 2-7 ตัวอย่างการใช้งานคำสั่ง assert

| คำสั่ง | คำอธิบาย |
|---|---|
| assert(any_boolean_condition) | คำสั่งนี้จะถูกกระทำเสมอโดยถ้าเงื่อนไขไม่เป็นจริง โปรแกรมจะหยุดการทำงานและแจ้งข้อผิดพลาด |

**6) คำสงวนในภาษาโพรเมลา**

ภาษาโพรเมลามีคำสงวนดังนี้

| | | | | | |
|---|---|---|---|---|---|
| active | do | init | printf | unsigned | atomic |
| else | int | priority | xr | bit | empty |
| len | proctype | xs | bool | enabled | mtype |
| provided | break | fi | run | byte | full |
| never | short | chan | goto | nfull | skip |
| d_step | hidden | od | timeout | if | of |
| typedef | d_proctype | inline | pc_value | unless | |

**7) ตรรกะเวลาเชิงเส้น [6]**

ใช้ในการตรวจสอบคุณสมบัติของแบบจำลองที่ถูกจำลองขึ้นในภาษาโพรเมลาเพื่อตรวจสอบในเครื่องมือสปิน โดยสามารถเรียกใช้ในเครื่องมือสปินตามกฎดังนี้

ไวยากรณ์ : ltl ::= opd | ( ltl ) | ltl binop ltl | unop ltl

การดำเนินการ (opd) : true, false ชื่อที่ตั้งโดยผู้ใช้ด้วยตัวอักษรเล็ก, การทำงานภายในปีกกา เช่น {a+b > n} ตัวดำเนินการเดี่ยว (unop) และตัวดำเนินการคู่ (binop) สามารถเขียนได้ดังตารางที่ 2-8 และตารางที่ 2-9

ตารางที่ 2-8 ตัวดำเนินการเดี่ยว

| ตัวดำเนินการ | คำอธิบาย |
|---|---|
| [] | ตัวดำเนินการในเงื่อนไขตลอดไป (always) |
| <> | ตัวดำเนินการในเงื่อนไขในที่สุด (eventually) |
| ! | ตัวดำเนินการในเงื่อนไขปฏิเสธ (negation) |

ตารางที่ 2-9 ตัวดำเนินการคู่

| ตัวดำเนินการ | คำอธิบาย |
|---|---|
| U | ตัวดำเนินการในเงื่อนไขแข็งแกร่งจนกว่า |
| W | ตัวดำเนินการในเงื่อนไขอ่อนจนกว่า |
| V | คู่ของ U : (p V q ) หมายถึง !(!p U !q) |
| && | ตัวดำเนินการค่าความจริง และ |
| \|\| | ตัวดำเนินการค่าความจริง หรือ |
| /\ | การเขียน && อีกแบบ |
| \/ | การเขียน \|\| อีกแบบ |
| -> | ตัวดำเนินการค่าความจริง ถ้าแล้ว |
| <-> | ตัวดำเนินการค่าความจริง ก็ต่อเมื่อ |

จากกฎที่ได้กล่าวมาข้างต้นสามารถนำมาสร้างเป็นโครงสร้างของตรรกะเวลาเชิงเส้นเพื่อใช้ในสปินได้ดังตัวอย่างดังตารางที่ 2-10

ตารางที่ 2-10 ตัวอย่างการเขียนตรรกะเวลาเชิงเส้นในเครื่องมือสปิน

| คำสั่ง | คำอธิบาย |
|---|---|
| ltl [ name ] '{' formula '}' | เป็นวิธีการเขียนตรรกะเวลาเชิงเส้นในสปิน |
| ltl p1 { []<> p } | ให้สปินตรวจสอบตรรกะเวลาเชิงเส้น p1 โดยใช้เงื่อนไข always eventually p |

### 2.1.6 ความสัมพันธ์เชิงล็อค (Lock Relation) [4, 10, 11]

ความสัมพันธ์เชิงล็อคถูกเสนอขึ้นเพื่อใช้ในการแก้ปัญหาสถานะที่มากมายของกราฟสถานะ ซึ่งหากพบความสัมพันธ์บางลักษณะสามารถบ่งบอกได้ถึงการพบคุณสมบัติบางคุณสมบัติของซิกแนลแทรนซิชันกราฟดังที่ได้มีงานวิจัยได้เสนอไว้ ซึ่งความสัมพันธ์เชิงล็อคเกิดจากการหาความสัมพันธ์ของสัญญาณบนวัฏจักรที่มีจุดยอดไม่ซ้ำกัน (Simple cycle) ของซิกแนลแทรนซิชันกราฟ
วัฏจักรที่มีจุดยอดไม่ซ้ำกันมีคำนิยามว่า "ลำดับของการเปลี่ยนแปลงที่ตำแหน่ง $t_1 \ldots t_i t_j \ldots t_1$ โดย $t_i \neq t_j$ สำหรับทุกๆ i และ j ($1 \leq i \leq j$)" [4, 10][4][12]

ความสัมพันธ์เชิงล็อคได้ถูกแบ่งออกเป็น 5 ประเภท ดังนี้

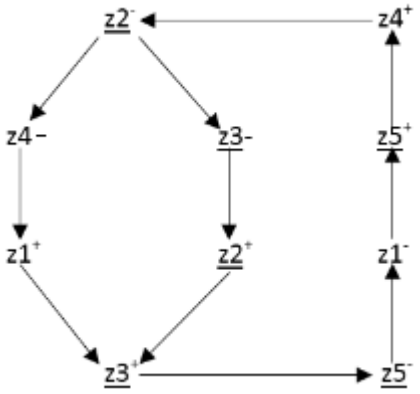
1) ฟูลล็อค (Full-lock) ถ้าสองสัญญาณ a และ b ที่พบการเกิดสัญญาณ $a \rightarrow b \rightarrow a' \rightarrow b'$ บนวัฏจักรที่มีจุดยอดไม่ซ้ำกัน แสดงว่าสองสัญญาณนั้นถูกเรียกว่าเป็น **ฟูลล็อค**

2) เซมิล็อค (Semi-lock) ถ้าสองสัญญาณ a และ b ที่พบการเกิดสัญญาณ $a \rightarrow b \rightarrow a'$ หรือ $b \rightarrow a' \rightarrow b'$ บนวัฏจักรที่มีจุดยอดไม่ซ้ำกัน แสดงว่าสองสัญญาณนั้นถูกเรียกว่าเป็น **เซมิล็อค**

3) อะโซซิเอทล็อค (Associate-lock) เมื่อเซตสัญญาณฟูลล็อค A และสัญญาณ b ที่พบการเกิดสัญญาณ $\exists a1, a2 \in A : a1 \rightarrow b \rightarrow a2 \rightarrow b'$ บนวัฏจักรที่มีจุดยอดไม่ซ้ำกัน ดังนั้น A และ b เป็น**อะโซซิเอทล็อค** และ $A \cup b$ เป็นเซตของแทรนซิทีฟล็อคระดับที่ 0

4) แทรนซิทีฟล็อค (Transitive-lock) เมื่อสัญญาณ b และเซตของแทรนซิทีฟล็อคระดับที่ i ซึ่งใช้ชื่อ A ที่พบการเกิดสัญญาณ $\exists a1, a2 \in A : a1 \rightarrow b \rightarrow a2 \rightarrow b'$ บนวัฏจักรที่มีจุดยอดไม่ซ้ำกัน ดังนั้น A และ b เป็น**แทรนซิทีฟล็อค** และ $A \cup b$ เป็นเซตของแทรนซิทีฟล็อคระดับที่ (i+1)

5) ซูเปอร์ล็อค (Super-lock) เซตสัญญาณฟูลล็อค A และสัญญาณ t เป็นอะโซซิเอทล็อค และสัญญาณ t' เป็นสัญญาณที่เกิดขึ้นพร้อมกัน (Concurrent) กับการเปลี่ยนแปลงสัญญาณในเซต A ดังนั้น t' เป็น**ซูเปอร์ล็อค**กับเซต A




รูปที่ 2-6 ตัวอย่างซิกแนลแทรนซิชันกราฟเพื่ออธิบายความสัมพันธ์เชิงล็อค

จากรูปที่ 2-6 พบ 2 วัฏจักรที่มีจุดยอดไม่ซ้ำกัน คือ $z2-\rightarrow z4+\rightarrow z1+\rightarrow z3+\rightarrow z5-\rightarrow z1-\rightarrow z5+\rightarrow z4+$ และ $z2-\rightarrow z3-\rightarrow z2+\rightarrow z3+\rightarrow z5-\rightarrow z1-\rightarrow z5+\rightarrow z4+$ และพบความสัมพันธ์เชิงล็อคดังนี้

1) สัญญาณ z1 มีความสัมพันธ์แบบเซมิล็อคกับสัญญาณ z2, z3, z4 (วัฏจักรที่มีจุดยอดไม่ซ้ำกัน 2)

2) สัญญาณ z2 มีความสัมพันธ์แบบฟูลล็อคกับสัญญาณ z3 (วัฏจักรที่มีจุดยอดไม่ซ้ำกัน 2)

3) สัญญาณ z1 มีความสัมพันธ์แบบฟูลล็อคกับสัญญาณ z5 (วัฏจักรที่มีจุดยอดไม่ซ้ำกัน 1)

4) สัญญาณ z4 มีความสัมพันธ์แบบอะโซซิเอทล็อคกับเซต (z2, z3) (วัฏจักรที่มีจุดยอดไม่ซ้ำกัน 1)

5) สัญญาณ z4-มีความสัมพันธ์แบบซูเปอร์ล็อคกับเซต (z2, z3) เนื่องจาก z4- เกิดขึ้นพร้อมกันกับ z3-

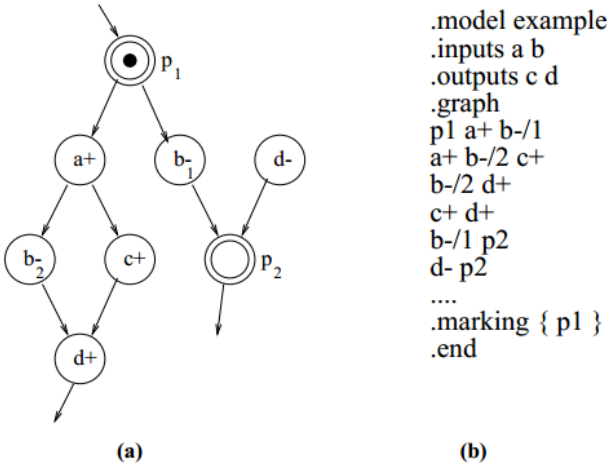
6) สัญญาณ z1 มีความสัมพันธ์แบบแทรนซิทีฟล็อคระดับที่ 1 กับเซต (z2, z3, z4) ซึ่งเป็นอะโซซิเอทล็อค (วัฏจักรที่มีจุดยอดไม่ซ้ำกัน 1)

7) สัญญาณ z5 มีความสัมพันธ์แบบแทรนซิทีฟล็อคระดับที่ 2 กับเซต (z1, z2, z3, z4) ซึ่งเป็นอะโซซิเอทล็อค ดังนั้นเซตของแทรนซิทีฟล็อคระดับ 2 ประกอบด้วยทุกสัญญาณในซิกแนลแทรนซิชันกราฟ

จากความสัมพันธ์เชิงล็อคที่พบในรูปที่ 2-6 พบว่าความสัมพันธ์แบบแทรนซิทีฟล็อคระดับที่ 2 ประกอบด้วยทุกสัญญาณในซิกแนลแทรนซิชันกราฟ จึงสามารถสรุปได้ว่าพบคุณสมบัติการกำหนดสถานะที่สมบูรณ์

### 2.1.7 เน็ตลิสต์ (Net list) [12]

ซิกแนลแทรนซิชันกราฟสามารถแสดงเป็นรูปแบบข้อความ เรียกว่า เน็ตลิสต์ เมื่อใช้เป็นตัวนำเข้าเครื่องมือวิเคราะห์แทนรูปแบบกราฟิกดังรูปที่ 2-7




รูปที่ 2-7 ตัวอย่างซิกแนลแทรนซิชันกราฟ (a) เน็ตลิสต์ (b)

จากรูปที่ 2-7 เป็นตัวอย่างการเขียนเน็ตลิสต์เพื่ออธิบายซิกแนลแทรนซิชันกราฟโดยประกอบด้วยทั้งหมด 6 ส่วนดังนี้

1) ส่วนของชื่อซิกแนลแทรนซิชันกราฟใช้คำขึ้นต้นว่า “.model” และตามด้วยชื่อ

2) ส่วนของสัญญาณขาเข้าใช้คำขึ้นต้นว่า “.inputs” และตามด้วยชื่อสัญญาณคั่นด้วยวรรค

3) ส่วนของสัญญาณขาออกใช้คำขึ้นต้นว่า “.outputs” และตามด้วยชื่อสัญญาณคั่นด้วยวรรค

4) ส่วนของการอธิบายกราฟใช้คำขึ้นต้นว่า “.graph” และขึ้นบรรทัดถัดมา จะเขียนอธิบายโดยสัญญาณตัวแรก คือ สัญญาณที่สนใจและสัญญาณต่อมาคือสัญญาณขาออกจากสัญญาณที่สนใจ เช่น ที่ p1 สัญญาณขาออก คือ a+ และ b- จะเขียนว่า “p1 a+ b-” หมายถึงลูกศร “p1 a+” และลูกศร “p1 b-”

5) ส่วนของมาร์กกิ้งตั้งต้นใช้คำขึ้นต้นว่า “.marking” และตามด้วยวงเล็บปีกกา โดยในวงเล็บประกอบด้วยมาร์กกิ้งตั้งต้นที่อยู่ในวงเล็บ เช่น มาร์กกิ้งตั้งต้นที่ลูกศร p⟶a+ จะใช้คำอธิบายว่า “<p,a+>”

6) วนสุดท้ายใช้คำว่า “.end” เป็นการปิดเน็ตลิสต์

## 2.2 งานวิจัยที่เกี่ยวข้อง

### 2.2.1 Signal Persistence Checking of Asynchronous System Implementation using SPIN โดย Weerasak Lawsunnee และ Arthit Thongtak และ Wiwat Vatanawood ปี ค.ศ. 2015 [7]

งานวิจัยนี้ได้นำเสนอวิธีการทวนสอบวงจรอสมการ โดยมีเป้าหมายเพื่อทวนสอบคุณสมบัติความทนทานของวงจร และใช้เครื่องมือสปินในการทวนสอบ ในงานวิจัยได้นำเสนอแบบจำลองวงจรอสมวารในภาษาโพรเมลา ซึ่งจำลองโครงสร้างซิกแนลแทรนซิชันกราฟแบบสถิต ในการทวนสอบนั้นมีวิธีการทวนสอบโดยการนำเอาซิกแนลแทรนซิชันกราฟมาเพื่อแปลงเป็นภาษาโพรเมลาตามวิธีที่ได้เสนอไว้ เมื่อได้ภาษาโพรเมลาแล้วจึงนำมาจำลองด้วยเครื่องมือสปิน และจากนั้นจึงนำผลจากการจำลองมาทำการเปรียบเทียบเพื่อหาคุณสมบัติไลฟ์เนสจากหลักการความสัมพันธ์เชิงล็อค และการใช้โปรแกรมไมโครซอฟท์เอ็กซ์เซลตรวจหาความผิดพลาด หากว่ามีคุณสมบัติไลฟ์เนสแล้วจึงนำไปเปรียบเทียบเพื่อหาคุณสมบัติความทนทาน ถ้าไม่มีคุณสมบัติไลฟ์เนสก็สามารถสรุปได้ว่าไม่สามารถตรวจหาคุณสมบัติความทนทานได้ โดยการจำลองแบบสถิตนี้ยังมีข้อจำกัดในการทวนสอบกับซิกแนลแทรนซิชันกราฟที่ซับซ้อน เช่น ซิกแนลแทรนซิชันกราฟประเภทวัฏจักรหลากหลาย เป็นต้น ซึ่งวิธีการทวนสอบใช้การจำลองที่ไม่อัตโนมัติ และทวนสอบได้เฉพาะคุณสมบัติไลฟ์เนสและคุณสมบัติความทนทานเท่านั้น

### 2.2.2 Model Checking Techniques for Verification of an Encryption Scheme for Wireless Sensor Networks โดย Zohra Sbai และ Mohamed Escheikh ปี ค.ศ. 2013 [8]



งานวิจัยนี้ได้นำเสนอวิธีการเขียนแบบจำลองภาษาโพรเมลาจากแผนภาพเพทริเนตของเครือข่ายไร้สายที่มีการเข้ารหัส เพื่อใช้ในการทวนสอบคุณสมบัติต่างๆ ของวงจรโดยใช้เครื่องมือสปิน โดยการทำงานของเครือข่ายไร้สายที่มีการเข้ารหัสนี้ประกอบด้วยสถานะต่างๆ ในการทำงาน เช่น การรับข้อมูล การส่งข้อมูล การเข้ารหัสข้อมูล การถอดรหัสข้อมูล เป็นต้น ซึ่งการทำงานของระบบนี้จะทำงานเป็นสถานะต่างๆ ต่อกัน ทำให้สามารถแสดงพฤติกรรมโดยใช้การจำลองเป็นเพลสและแทรนซิชันด้วยแผนภาพเพทริเนตได้ งานวิจัยได้นำเสนอการเขียนโครงสร้างภาษาโพรเมลาจำลองการทำงานของแผนภาพเพทริเนตในรูปแบบพลวัต ซึ่งสามารถจำลองการทำงานในแต่ละช่วงเวลาได้ โดยอ้างอิงเพลสและแทรนซิชันในรูปของอาร์เรย์ในภาษาโพรเมลา และแสดงการเขียนความสัมพันธ์ของเพลสต่างๆ โดยอ้างอิงตามจำนวนโทเค็นในอาร์เรย์ ทำให้สามารถเห็นการทำงานของแผนภาพแบบพลวัตได้ งานวิจัยนี้ได้แสดงการทวนสอบคุณสมบัติความปลอดภัยในรูปแบบของตรรกะเวลาเชิงเส้น

**2.2.3 Synthesis of Asynchronous VLSI Circuits from Signal Transition Graph Specifications โดย Sung-Bum Park และ Takashi Nanya ปี 1996 [4]**

งานวิจัยนี้ได้นำเสนอการสังเคราะห์วงจรจากซิกแนลแทรนซิชันกราฟ โดยในการสังเคราะห์วงจรอสมวารนั้นจะต้องประกอบด้วยซิกแนลแทรนซิชันกราฟที่มีคุณสมบัติต่างๆ เพื่อป้องกันการสร้างวงจรที่ทำงานผิดพลาดและวงจรที่สร้างขึ้นต้องทำให้มีขนาดเล็กที่สุด งานวิจัยนี้ครอบคลุมซิกแนลแทรนซิชันกราฟประเภทวัฏจักรเชิงเดี่ยว, วัฏจักรหลากหลาย และ free choice ซึ่งใช้ความสัมพันธ์เชิงล็อคมาทวนสอบคุณสมบัติเพื่อป้องกันการทำงานที่ผิดพลาด เมื่อได้ซิกแนลแทรนซิชันกราฟที่ผ่านการทวนสอบแล้ว จึงได้นำซิกแนลแทรนซิชันกราฟไปสังเคราะห์เป็นวงจรอสมวาร แต่ยังพบว่าวงจรอสมวารที่ได้ยังมีขนาดใหญ่จึงได้นำเสนอวิธีการลดขนาดของวงจร จากนั้นจึงนำวงจรอสมวารที่ได้ไปสร้างวงจร และทดสอบว่าวงจรทำงานได้ถูกต้อง งานวิจัยนี้ได้เสนอวงจรอสมวารที่ได้จากการสังเคราะห์ซิกแนลแทรนซิชันกราฟ และผ่านการทดสอบว่าวงจรทำงานได้ถูกต้องไว้หลายวงจร

# บทที่ 3
# วิธีการแปลงซิกแนลแทรนซิชันกราฟเป็นรหัสโพรเมลาและตรรกะเวลาเชิงเส้น

ในบทนี้จะอธิบายขั้นตอนและวิธีการแปลงซิกแนลแทรนซิชันกราฟซึ่งประกอบด้วยกฎการแปลงซิกแนลแทรนซิชันกราฟเป็นรหัสโพรเมลาและกฎการสร้างตรรกะเวลาเชิงเส้นจากซิกแนลแทรนซิชันกราฟ โดยรหัสโพรเมลาที่ได้จากการแปลงมี 2 แบบ คือ รหัสโพรเมลาแบบ A1 และ A2 และตรรกะเวลาเชิงเส้นที่ได้จากการแปลงประกอบด้วย คุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์

## 3.1 การแปลงซิกแนลแทรนซิชันกราฟเป็นรหัสโพรเมลา

คุณสมบัติของซิกแนลแทรนซิชันกราฟสำหรับการตรวจสอบวงจรอสมวารต้องใช้รหัสโพรเมลาที่แตกต่างกัน โดยการทวนสอบคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส และคุณสมบัติความทนทานนั้นจะดูจากการเคลื่อนย้ายโทเค็นในซิกแนลแทรนซิชันกราฟ แต่การทวนสอบคุณสมบัติความต้องกันและคุณสมบัติการกำหนดสถานะที่สมบูรณ์จะดูจากวัฏจักรที่มีจุดยอดไม่ซ้ำกัน ดังนั้นรหัสโพรเมลาที่ได้จากการแปลงซิกแนลแทรนซิชันกราฟจึงมี 2 แบบ คือ

1) รหัสโพรเมลาแบบ A1 เป็นรหัสโพรเมลาที่เน้นการจำลองโครงสร้างของซิกแนลแทรนซิชันกราฟให้ทำงานตามกฎการเคลื่อนย้ายโทเค็น

2) รหัสโพรเมลาแบบ A2 เป็นรหัสโพรเมลาที่เน้นการสร้างแบบจำลองเพื่อให้สามารถค้นหาวัฏจักรที่มีจุดยอดไม่ซ้ำกันในซิกแนลแทรนซิชันกราฟได้



### 3.1.1 รหัสโพรเมลาแบบ A1

รหัสโพรเมลาแบบ A1 เป็นรหัสโพรเมลาที่เน้นการจำลองโครงสร้างของซิกแนลแทรนซิชันกราฟให้ทำงานตามกฎการเคลื่อนย้ายโทเค็น สามารถนำไปใช้ในการทวนสอบคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส และคุณสมบัติความทนทาน ซึ่งรหัสโพรเมลาแบบ A1 มีวิธีการแปลง 4 ส่วน ดังนี้

ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรแทรนซิชัน

ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น

ส่วนที่ 3 การกำหนดมาร์กกิ้งตั้งต้น

ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ

**1) การกำหนดตัวแปรลูกศรและตัวแปรแทรนซิชัน**

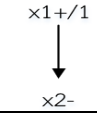
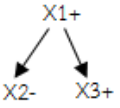
การกำหนดตัวแปรลูกศรและตัวแปรแทรนซิชันเป็นรหัสโพเมลาแบบ A1 ส่วนที่ 1 โดยตัวแปรลูกศรใช้ในการบันทึกจำนวนโทเค็นในลูกศรนั้นๆ และตัวแปรแทรนซิชันใช้บันทึกจำนวนครั้งที่โทเค็นเคลื่อนที่ผ่านลูกศรนั้นๆ ซึ่งมีประโยชน์ในการทวนสอบคุณสมบัติของซิกแนลแทรนซิชันกราฟ

การกำหนดตัวแปรลูกศรกำหนดจากลูกศรทุกลูกศรในซิกแนลแทรนซิชันกราฟ โดยกำหนดชื่อลูกศรจากสัญญาณต้นลูกศรและสัญญาณปลายลูกศร และสัญญาณที่ลงท้ายด้วยเครื่องหมาย “+” แทนด้วย “p” และเครื่องหมาย “-” แทนด้วย “m” เช่น สัญญาณ “a+” ให้แทนชื่อด้วย “ap” เป็นต้น ถ้ากรณีที่เป็นประเภทวัฏจักรหลากหลายให้เติมเลขหลังเครื่องหมาย “/” หลัง p หรือ m เช่น สัญญาณ “x+/1” ให้แทนชื่อด้วย “xp1” เป็นต้น

ส่วนการกำหนดตัวแปรแทรนซิชันให้กำหนดให้เท่ากับจำนวนตัวแปรลูกศร โดยกำหนดชื่อ “t” และตามด้วยเลขลำดับ

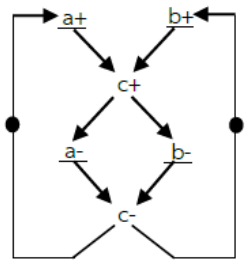
ตัวอย่างการกำหนดชื่อตัวแปรลูกศรและตัวแปรแทรนซิชันแสดงในตารางที่ 3-1

ตารางที่ 3-1 ตัวอย่างการกำหนดชื่อตัวแปรลูกศรและตัวแปรแทรนซิชัน

| ลักษณะลูกศร | ชื่อตัวแปรลูกศรและตัวแปรแทรนซิชัน |
|---|---|
|  | “x1px2m” และ “t1” |
|  | “x1p1x2m” และ “t1” |
|  | “x1px2m”,“x1px3p” และ “t1”, “t2” |



จากรูปที่ 3-1 เป็นตัวอย่างการกำหนดตัวแปรลูกศรและตัวแปรแทรนซิชันในการแปลงซิกแนลแทรนซิชันกราฟเป็นรหัสโพรเมลาแบบ A1 ส่วนที่ 1 พบว่า ซิกแนลแทรนซิชันกราฟในรูปที่ 3-1(ก) มีลูกศรทั้งหมด 8 ลูกศร จึงสามารถกำหนดตัวแปรลูกศรได้ 8 ตัวแปร ได้แก่ apcp, bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp และกำหนดตัวแปรแทรนซิชันได้ 8 ตัวแปร ได้แก่ t1, t2, t3, t4, t5, t6, t7, t8 ดังแสดงในรูปที่ 3-1(ข)




(ก)

```
byte apcp, bpcp, cpam, cpbm, amcm, bmcm,
cmap, cmbp;
int t1, t2, t3, t4, t5, t6, t7, t8;
```

(ข)

รูปที่ 3-1 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A1 ส่วนที่ 1

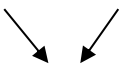
**2) กฎการเคลื่อนย้ายโทเค็น**

กฎการเคลื่อนย้ายโทเค็นเป็นรหัสโพเมลาแบบ A1 ส่วนที่ 2 ซึ่งประกอบด้วย กฎการเคลื่อนย้ายโทเค็นขาเข้าและกฎการเคลื่อนย้ายโทเค็นขาออก

กฎการเคลื่อนย้ายขาเข้าให้ทำการลบโทเค็นออกจากทุกตัวแปรลูกศรขาเข้า 1 โทเค็น และเพิ่มโทเค็นในทุกตัวแปรลูกศรขาออก 1 โทเค็น ซึ่งกฎนี้ช่วยให้เกิดสถานะต่างๆ ของซิกแนลแทรนซิชันกราฟ เช่น สถานะพร้อมเปลี่ยนแปลงสัญญาณ สถานะหลังเปลี่ยนแปลงสัญญาณ เป็นต้น โดยการสร้างกฎกระทำโดยหาจำนวนลูกศรขาเข้าและจำนวนลูกศรขาออกที่มากที่สุด จากนั้นจึงประกาศคำสั่ง “#define inp1(x,t) x = x-1; t = t+1” โดย x คือ ตัวแปรลูกศร และ t คือ ตัวแปรแทรนซิชัน โดยกฎการเคลื่อนย้ายโทเค็นขาเข้าจะต้องมีเท่ากับจำนวนลูกศรขาเข้ามากที่สุด เช่น มีลูกศรขาเข้ามากที่สุด 2 ลูกศร จะต้องมีกฎการเคลื่อนย้ายโทเค็นขาเข้า 2 กฎ คือ ขาเข้า 1 ลูกศร และขาเข้า 2 ลูกศร

กฎการเคลื่อนย้ายโทเค็นขาออกกระทำโดยประกาศคำสั่ง “#define outp1(x) x=x+1” โดย x แทนตัวแปรลูกศรเมื่อเข้าเงื่อนไขให้เพิ่มโทเค็นในลูกศร 1 โทเค็น ถ้ามีลูกศรขาออกมากที่สุด 2 ลูกศร จะต้องสร้างกฎการเคลื่อนย้ายโทเค็นขาออก 2 กฎ คือ ขาออก 1 ลูกศร และขาออก 2 ลูกศร ในการสร้างกฎโดยการประกาศ “#define” นั้นเสมือนการสร้างฟังก์ชันในภาษาอื่นๆ สามารถแสดงตัวอย่างการสร้างกฎการเคลื่อนย้ายโทเค็นได้ในตารางที่ 3-2

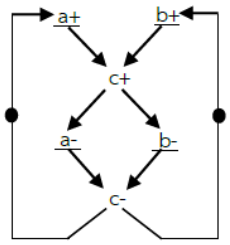
ตารางที่ 3-2 ตัวอย่างกฎการเคลื่อนย้ายโทเค็น

| รูปแบบของลูกศรขาเข้า/ขาออก | กฎการเคลื่อนย้ายโทเค็น |
|---|---|
|   | #define inp1(x, t) (x>0) -> x=x-1; t=t+1 |
|   | #define inp2(x1, x2, t1, t2) (x1>0 && x2>0) -> x1=x-1; x2=x2-1; ta=ta+1; tb=tb+1 |

ตารางที่ 3-2 ตัวอย่างกฎการเคลื่อนย้ายโทเค็น (ต่อ)

| **รูปแบบของลูกศรขาเข้า/ขาออก** | **กฎการเคลื่อนย้ายโทเค็น** |
|---|---|
| จำนวนลูกศรขาเข้าเท่ากับ n>2 | #define inpn(x1, .., xn, t1, ..., tn) (x1>0 &&...&& xn>0)-> x1=x1-1; ...; xn=xn-1; ta=ta+1; tn=tn+1 |
|  | #define outp1(x) x=x+1 |
|  | #define outp2(x1, x2) x1=x1+1; x2=x2+1 |
| จำนวนลูกศรขาออกเท่ากับ n>2 | #define outpn(x1, .., xn) x1=x1+1;..;xn=xn+1 |

จากรูปที่ 3-2(ก) พบว่ามีจำนวนลูกศรขาเข้ามากที่สุด 2 ลูกศรที่สัญญาณ c+ และ c- จึงสามารถสร้างกฎการเคลื่อนย้ายโทเค็นขาเข้าได้ 2 กฎ ได้แก่ inp1, inp2 และพบว่ามีลูกศรขาออกมากที่สุด 2 ลูกศรที่สัญญาณ c+ และ c- จึงสามารถสร้างกฎการเคลื่อนย้ายโทเค็นขาออกได้ 2 กฎ ได้แก่ outp1, outp2 ดังแสดงในรูปที่ 3-2(ข)




(ก)

```
#define inp1(x, t) (x>0) -> x=x-1; t=t+1
#define inp2(x1, x2, ta, tb) (x1>0 && x2>0) -> x1=x-1; x2=x2-1;
ta=ta+1; tb=tb+1
#define outp1(x) x = x+1
#define outp2(x1,x2) x1=x1+1;x2=x2+1
```

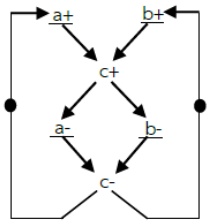
(ข)

รูปที่ 3-2 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A1 ส่วนที่ 2

**3) การกำหนดมาร์กกิ้งตั้งต้น**

การกำหนดมาร์กกิ้งตั้งต้นเป็นรหัสโพเมลาแบบ A1 ส่วนที่ 3 โดยดูจากมาร์กกิ้งตั้งต้นว่าอยู่บนลูกศรใดแล้วให้ตัวแปรลูกศรของลูกศรนั้นเท่ากับ 1 โดยกำหนดทุกลูกศรที่มีโทเค็น และการกำหนดนั้นจะต้องอยู่ภายในฟังก์ชัน init ซึ่งเป็นฟังก์ชันที่ใช้ในการเริ่มการทำงานในภาษาโพรเมลา

จากรูปที่ 3-3(ก) พบว่า 2 มาร์กกิ้งตั้งต้นบนลูกศร “cmbp” และลูกศร “cmap” จึงสามารถกำหนดมาร์กกิ้งตั้งต้นในรหัสโพรเมลาแบบ A1 ส่วนที่ 3 ได้ดังรูปที่ 3-3(ข)




(ก)

```
init{
cmap = 1;
cmbp = 1;
```

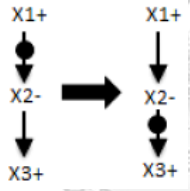
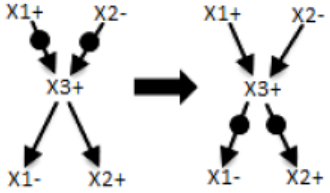
(ข)

รูปที่ 3-3 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A1 ส่วนที่ 3

**4) การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ**

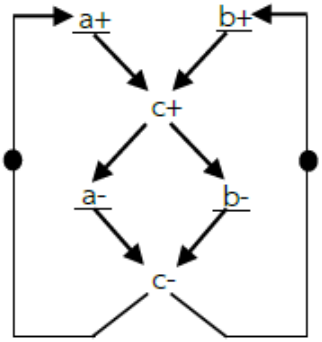
การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟเป็นรหัสโพเมลาแบบ A1 ส่วนที่ 4 โดยดูจากลูกศรขาเข้าและลูกศรขาออกของซิกแนลแทรนซิชันกราฟโดยจะต้องทำให้ครบทุกคู่ ในการกำหนดโครงสร้างให้เลือกสัญญาณอ้างอิงเพื่อดูลูกศรขาเข้าและขาออกดังตัวอย่างในตารางที่ 3-3

ตารางที่ 3-3 ตัวอย่างกฎการกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ

| ตัวอย่างที่ | รูปแบบ | รหัสโพรเมลา |
|---|---|---|
| 1 |   | “::atomic{inp1(x1px2m, t1) -> outp1(x2mx3p)}” |
| 2 |   | “::atomic{inp2(x1px3p, x2mx3p, t1, t2) -> outp2(x3px1m, x3px2p)}” |

จากตารางที่ 3-3 ตัวอย่างที่ 1 มีลูกศรขาเข้าและลูกศรขาออกซึ่งเป็นตัวอย่างแสดงสถานะพร้อมเปลี่ยนแปลงสัญญาณและหลังเปลี่ยนแปลงสัญญาณ โดยใช้สัญญาณ “x2-” เป็นสัญญาณอ้างอิง พบว่ามีสัญญาณขาเข้า 1 ลูกศรและสัญญาณขาออก 1 ลูกศร จึงกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟโดยใช้กฎการเคลื่อนย้ายขาเข้า 1 ลูกศรและแทนตัวแปรลูกศรเป็นลูกศรขาเข้าสัญญาณ “x2-” และใช้กฎการเคลื่อนย้ายลูกศรขาออก 1 ลูกศรเป็นตัวแปรลูกศรขาออกจากสัญญาณ “x2-” ตัวอย่างที่ 2 อ้างอิงจากสัญญาณ “x3+” พบว่ามี 2 ลูกศรขาเข้าและ 2 ลูกศรขาออก จึงกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟโดยใช้กฎการเคลื่อนย้ายขาเข้า 2 ลูกศรและ

แทนตัวแปรลูกศรเป็นลูกศรขาเข้าสัญญาณ “x3+” และใช้กฎการเคลื่อนย้ายลูกศรขาออก 2 ลูกศร เป็นตัวแปรลูกศรขาออกจากสัญญาณ “x3+”




(ก)

do

::atomic{ inp1(cmap, t1) -> outp1(apcp)}

::atomic{ inp2(bpcp,apcp,t2,t3) -> outp2(cpam,cpbm)}

::atomic{ inp1(cpam,t4) -> outp1(amcm)}

::atomic{ inp1(cpbm,t5) -> outp1(bmcm)}

::atomic{ inp2(amcm,bmcm,t6,t7) -> outp2(cmap, cmbp)}

::atomic{ inp1(cmbp,t8) -> outp1(bpcp)}

od}

(ข)

รูปที่ 3-4 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A1 ส่วนที่ 4

จากซิกแนลแทรนซิชันกราฟในรูปที่ 3-4(ก) สามารถกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟได้ดังรูปที่ 3-4(ข) ดังนี้

- ใช้สัญญาณ a+ เป็นสัญญาณอ้างอิง พบว่ามี 1 ลูกศรขาเข้าและ 1 ลูกศรขาออกโดยตัวแปรลูกศรขาเข้า คือ “cmap” และตัวแปรลูกศรขาออก คือ “apcp” ในกรณีเป็นสัญญาณอ้างอิงสัญญาณแรกให้เริ่มต้นด้วย “do” จึงสามารถกำหนดโครงสร้างได้ดังนี้

“do ::atomic{inp1(cmap, t1) -> outp1(apcp)}”

- ใช้สัญญาณ c+ เป็นสัญญาณอ้างอิงพบว่ามีลูกศรขาเข้า 2 ลูกศร คือ “apcp, bpcp” และลูกศรขาออก 2 ลูกศร คือ “cpam, cpbm” สามารถกำหนดโครงสร้างได้ดังนี้

“::atomic{inp2(bpcp, apcp, t2, t3) -> outp2(cpam, cpbm)}”

- ใช้สัญญาณ a- เป็นสัญญาณอ้างอิงพบว่ามีลูกศรขาเข้า 1 ลูกศร คือ “cpam” และลูกศรขาออก 1 ลูกศร คือ “amcm” สามารถกำหนดโครงสร้างได้ดังนี้

“::atomic{inp1(cpam, t4) -> outp1(amcm)}”

- ใช้สัญญาณ b- เป็นสัญญาณอ้างอิงพบว่ามีลูกศรขาเข้า 1 ลูกศร คือ “cpbm” และลูกศรขาออก 1 ลูกศร คือ “bmcm” สามารถกำหนดโครงสร้างได้ดังนี้

“::atomic{inp1(cpbm, t5) -> outp1(bmcm)}”

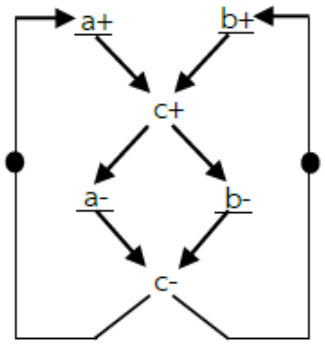
- ใช้สัญญาณ c- เป็นสัญญาณอ้างอิง พบว่ามีลูกศรขาเข้า 2 ลูกศร คือ “amcm, bmcm” และลูกศรขาออก 2 ลูกศร คือ “cmap, cmbp” สามารถกำหนดโครงสร้างได้ดังนี้

“::atomic{inp2(amcm, bmcm, t6, t7) -> outp2(cmap, cmbp)}”

- ใช้สัญญาณ b+ เป็นสัญญาณอ้างอิง พบว่ามีลูกศรขาเข้า 1 ลูกศร คือ “cmbp” และลูกศรขาออก 1 ลูกศร คือ “bpcp” และเป็นกรณีสัญญาณสุดท้ายให้เติม “od}” สามารถกำหนดโครงสร้างได้ดังนี้

“::atomic{inp1(cmbp,t8) -> outp1(bpcp)} od}”

จากวิธีการแปลงรหัสโพรเมลาแบบ A1 ทั้ง 4 ส่วน สามารถแปลงซิกแนลแทรนซิชันกราฟในรูปที่ 3-5(ก) เป็นรหัสโพรเมลาแบบ A1 ได้ดังรูปที่ 3-5(ข)




```
/*ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรแทรนซิชัน*/
byte bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp, apcp;
byte t1,t2,t3,t4,t5,t6,t7,t8;

/*ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น*/
#define inp1(x1,t1)  (x1>0) -> x1 = x1-1; t1 = t1+1
#define inp2(x1,x2,t1,t2)       (x1>0 && x2>0) -> x1 = x1-1; x2 = x2-1; t1 =
t1+1; t2 = t2+1
#define outp1(x1)   x1 = x1+1
#define outp2(x1,x2)            x1 = x1+1; x2 = x2+1

/*ส่วนที่ 3 การกำหนดมาร์กกิ้งตั้งต้น*/
init {
cmap=1;cmbp=1;
/*ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ*/
do
        ::atomic{ inp1(cmap,t1) -> outp1(apcp)}
        ::atomic{ inp2(bpcp,apcp,t2,t3) -> outp2(cpam,cpbm)}
        ::atomic{ inp1(cpam,t4) -> outp1(amcm)}
        ::atomic{ inp1(cpbm,t5) -> outp1(bmcm)}
        ::atomic{ inp2(amcm,bmcm,t6,t7) -> outp2(cmap,cmbp)}
        ::atomic{ inp1(cmbp,t8) -> outp1(bpcp)}
od}
```

(ก) (ข)

รูปที่ 3-5 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A1

### 3.1.2 รหัสโพรเมลาแบบ A2

รหัสโพรเมลาแบบ A2 เป็นรหัสโพรเมลาที่เน้นการสร้างแบบจำลองเพื่อให้สามารถค้นหาวัฏจักรที่มีจุดยอดไม่ซ้ำกันในซิกแนลแทรนซิชันกราฟได้ สามารถนำไปใช้ในการทวนสอบคุณสมบัติความต้องกันและคุณสมบัติการกำหนดสถานะที่สมบูรณ์ ซึ่งรหัสโพรเมลาแบบ A2 มีวิธีการแปลง 5 ส่วน ดังนี้

ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรสัญญาณ

ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น

ส่วนที่ 3 การกำหนดชุดค่าตั้งต้น

ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ

ส่วนที่ 5 การกำหนดลำดับการทำงานของการตรวจสอบ

#### 1) การกำหนดตัวแปรลูกศรและตัวแปรสัญญาณ

การกำหนดตัวแปรลูกศรและตัวแปรสัญญาณเป็นรหัสโพเมลาแบบ A2 ส่วนที่ 1 โดยตัวแปรลูกศรใช้ในการบันทึกจำนวนโทเค็นในลูกศรนั้นๆ และตัวแปรสัญญาณใช้บันทึกจำนวนครั้งที่โทเค็นเคลื่อนที่ผ่านสัญญาณนั้นๆ

ในการกำหนดตัวแปรลูกศรกำหนดจากลูกศรทุกลูกศรในซิกแนลแทรนซิชันกราฟ โดยกำหนดชื่อลูกศรจากสัญญาณต้นลูกศรและสัญญาณปลายลูกศร และสัญญาณที่ลงท้ายด้วยเครื่องหมาย “+” แทนด้วย “p” และเครื่องหมาย “-” แทนด้วย “m” เช่น สัญญาณ “a+” ให้แทนชื่อด้วย “ap” เป็นต้น ถ้ากรณีที่เป็นประเภทวัฏจักรหลากหลายให้เติมเลขหลังเครื่องหมาย “/” หลัง p หรือ m เช่น สัญญาณ “x+/1” ให้แทนชื่อด้วย “xp1” เป็นต้น



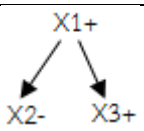
ส่วนตัวแปรสัญญาณให้กำหนดจากสัญญาณขาเข้าและสัญญาณขาออกของซิกแนลแทรนซิชันกราฟ โดยกำหนดให้แต่ละสัญญาณมีสองตัวแปรโดยเติม “p” และ “m” ต่อจากชื่อสัญญาณ

ตัวอย่างการกำหนดชื่อตัวแปรลูกศรและตัวแปรสัญญาณดังตัวอย่างในตารางที่ 3-4

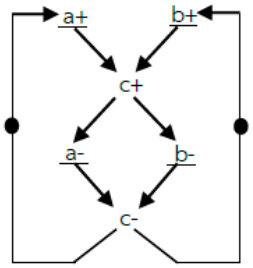
ตารางที่ 3-4 ตัวอย่างการกำหนดชื่อตัวแปรลูกศรและตัวแปรสัญญาณ

| ลักษณะลูกศร | ชื่อตัวแปรลูกศรและตัวแปรแทรนซิชัน |
|---|---|
| X1+ ↓ X2- | “x1px2m” และ “x1p,x1m,x2p,x2m” |
| x1+/1 ↓ x2- | “x1p1x2m” และ “x1p,x1m,x2p,x2m” |

ตารางที่ 3-4 ตัวอย่างการกำหนดชื่อตัวแปรลูกศรและตัวแปรสัญญาณ (ต่อ)

| ลักษณะลูกศร | ชื่อตัวแปรลูกศรและตัวแปรแทรนซิชัน |
|---|---|
|  | “x1px2m, x1px3p” และ “x1p,x1m,x2p,x2m,x3p,x3m” |

จากรูปที่ 3-6(ก) พบว่า ซิกแนลแทรนซิชันกราฟมีลูกศรทั้งหมด 8 ลูกศร จึงสามารถกำหนดตัวแปรลูกศรได้ 8 ตัวแปร ได้แก่ apcp, bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp และซิกแนลแทรนซิชันกราฟมี 3 สัญญาณ คือ a, b และ c จึงกำหนดได้ 6 ตัวแปรสัญญาณ ได้แก่ ap, am, bp, bm, cp, cm ดังแสดงในรูปที่ 3-6(ข)




```
byte apcp, bpcp, cpam, cpbm, amcm, bmcm,
cmap, cmbp;
byte ap, am, bp, bm, cp, cm;
```

(ก) (ข)

รูปที่ 3-6 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A2 ส่วนที่ 1

**2) กฎการเคลื่อนย้ายโทเค็น**

เนื่องจากรหัสโพรเมลาแบบ A2 เน้นการค้นหาวัฏจักรที่มีจุดยอดไม่ซ้ำกันจึงทำให้การเดินทางของโทเค็นต้องเกิดการเดินที่เป็นวัฏจักรและกลับมาที่จุดเดิม ทำให้ในการเคลื่อนย้ายโทเค็นจึงไม่ต้องสนใจว่าจะมีโทเค็นครบทุกลูกศรขาเข้าหรือไม่และโทเค็นจะต้องไม่เคลื่อนกลับมาซ้ำที่ลูกศรเดิมด้วย ดังนั้นการกำหนดกฎการเคลื่อนย้ายขาเข้าจะมีสำหรับ 1 ลูกศรขาเข้าเท่านั้น ส่วนการกำหนดกฎการเคลื่อนย้ายขาออกจะมีจำนวนกฎตามจำนวนลูกศรขาออกมากที่สุด กฎจะนิยามให้สามารถเลือกเดินทางได้เพียงลูกศรเดียว ตัวอย่างกฎการเคลื่อนย้ายโทเค็นดังตัวอย่างในตารางที่ 3-5

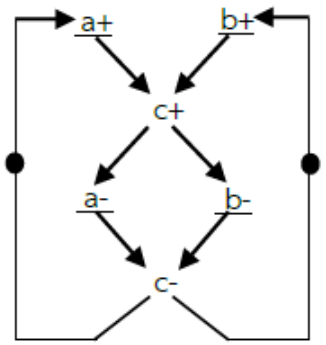
ตารางที่ 3-5 ตัวอย่างกฎการเคลื่อนย้ายโทเค็นของรหัสโพรเมลาแบบ A2

| ตัวอย่างที่ | รูปแบบของลูกศรขาเข้า/ขาออก | กฎการเคลื่อนย้ายโทเค็น |
|---|---|---|
| 1 |   | #define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++; y++; fi |
| 2 |   | #define outp1(x) if ::(x!=10) -> x++; :: else -> break; fi |
| 3 |   | #define outp2(x1, x2) if :: (x1!=10) -> x++; :: (x2!=10) -> x2++; :: else -> break; fi |
| 4 | จำนวนลูกศรขาออกเท่ากับ n>2 | #define outpn(x1, .., xn) if :: (x1!=10) -> x++; :: (x2!=10) -> x2++; ... :: (xn!=10) -> xn++; :: else -> break; fi |

จากตารางที่ 3-5 กฎการเคลื่อนย้ายโทเค็นขาเข้าในตัวอย่างที่ 1 จะกำหนดการตรวจสอบเงื่อนไข x=1 และ x=2 ซึ่งมีการดำเนินการในฟังก์ชันต่างกัน โดยตัวแปร x แทนตัวแปรลูกศรและตัวแปร y แทนตัวแปรสัญญาณ ส่วนกฎการเคลื่อนย้ายโทเค็นขาออกในตัวอย่างที่ 2 ถึง 4 ใช้คำสั่ง “if” เพื่อทำให้โทเค็นเลือกเดินทางได้ทางเดียว โดยตัวแปร x1, x2 แทนตัวแปรลูกศร

จากรูปที่ 3-7(ก) พบว่ามีลูกศรขาออกมากที่สุด 2 ลูกศรที่สัญญาณ c+ และ c- จึงสามารถสร้างกฎการเคลื่อนย้ายโทเค็นขาออกได้ 2 กฎ ได้แก่ outp1, outp2 และสามารถสร้างกฎการเคลื่อนย้ายโทเค็นขาเข้าได้แก่ inp ดังแสดงในรูปที่ 3-7(ข)






(ก)

```
#define inp(x,y) if ::(x==1) -> x=10; y++;
::(x==2) -> x++;y++; fi

#define outp1(x)   if ::(x!=10) -> x++; :: else -> break; fi
#define outp2(x1, x2)   if :: (x1!=10) -> x++;
:: (x2!=10) -> x2++; :: else -> break; fi
```

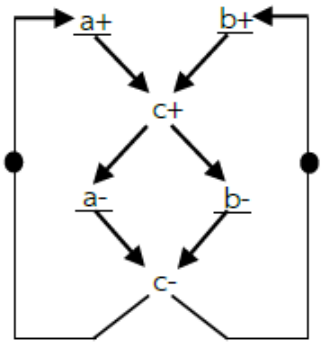
(ข)

รูปที่ 3-7 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A2 ส่วนที่ 2

**3) การกำหนดชุดค่าตั้งต้น**

การกำหนดชุดค่าตั้งต้นเป็นรหัสโพเมลาแบบ A2 ส่วนที่ 3 โดยจำนวนชุดค่าตั้งต้นจะมีจำนวนเท่ากับจำนวนมาร์กกิ้งตั้งต้นและในแต่ละชุดจะสนใจมาร์กกิ้งตั้งต้น 1 โทเค็น และกำหนดชุดค่าตั้งต้นในคำสั่ง if ใน “proctype ชื่อ(byte k)” ซึ่ง proctype เป็นการกำหนดฟังก์ชันและรับตัวแปร k ในชนิด byte และใช้คำสั่ง if เพื่อให้เกิดการเลือกทำทีละชุด

จากรูปที่ 3-8(ก) พบว่ามี 2 มาร์กกิ้งตั้งต้นบนลูกศร “cmap” และ “cmbp” จึงสามารถกำหนดชุดค่าตั้งต้นได้ 2 ชุด โดยชุดที่ 1 สนใจโทเค็นบนลูกศร “cmap” และชุดที่ 2 สนใจโทเค็นบนลูกศร “cmbp” และกำหนดค่าให้ตัวแปรที่มีโทเค็น โดยให้ตัวแปรลูกศรที่มีโทเค็นที่สนใจมีค่าเท่ากับ 2 และตัวแปรลูกศรอื่นที่มีโทเค็นเท่ากับ 10 ส่วนตัวแปรที่เหลือให้มีค่าเท่ากับ 0 และให้ k มีค่าตามเลขลำดับของชุดที่สนใจ จึงสามารถกำหนดชุดค่าตั้งต้นชุดที่ 1 เป็น “cmap = 2; cmbp = 10” และกำหนดชุดค่าตั้งต้นชุดที่ 2 เป็น “cmap = 10; cmbp = 2” ดังแสดงในรูปที่ 3-8(ข)




(ก)

```
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> cmap=2; cmbp=10; bpcp=0; cpam=0; cpbm=0;
amcm=0; bmcm=0; apcp=0; ap=0; am=0; bp=0; bm=0;
cp=0; cm=0;
   :: k==2 -> cmbp=2; cmap=10; bpcp=0; cpam=0; cpbm=0;
amcm=0; bmcm=0; apcp=0; ap=0; am=0; bp=0; bm=0;
cp=0; cm=0;
  fi
```

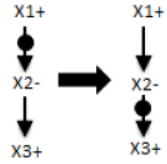
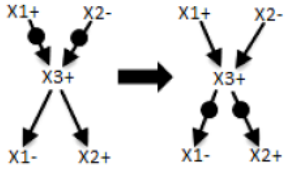
(ข)

รูปที่ 3-8 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A2 ส่วนที่ 3

**4) การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ**

การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟเป็นรหัสโพเมลาแบบ A2 ส่วนที่ 4 โดยดูจากลูกศรขาเข้าและลูกศรขาออกของซิกแนลแทรนซิชันกราฟโดยจะต้องทำให้ครบทุกสัญญาณ ในการกำหนดโครงสร้างให้เลือกสัญญาณอ้างอิงเพื่อดูลูกศรขาเข้าและขาออกดังตัวอย่างในตารางที่ 3-6

ตารางที่ 3-6 ตัวอย่างกฎการกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ

| ตัวอย่างที่ | รูปแบบ | รหัสโพรเมลา |
|---|---|---|
| 1 |  | “::atomic{inp(x1px2m,x1p) ;<br>printf("printf(x1p=%d)",x1p) -> outp2(x2mx3p)}” |
| 2 |  | “::atomic{inp(x1px3p,x1p) ;<br>printf("printf(x1p=%d)", x1p) -> outp2(x3px1m, x3px2p)}”<br>“::atomic{inp(x2mx3p,x2m) ;<br>printf("printf(x2m=%d)", x2m) -> outp2(x3px1m, x3px2p)}” |

จากตารางที่ 3-6 ตัวอย่างที่ 1 มีลูกศรขาเข้าและลูกศรขาออกซึ่งตัวอย่างแสดงสถานะพร้อมเปลี่ยนแปลงสัญญาณและหลังเปลี่ยนแปลงสัญญาณ ใช้สัญญาณ “x2-” เป็นสัญญาณอ้างอิง พบว่า มีสัญญาณขาเข้า 1 ลูกศรและสัญญาณขาออก 1 ลูกศร จึงกำหนดโครงสร้างโดยใช้กฎการเคลื่อนย้ายขาเข้า 1 ลูกศรและแทนตัวแปรลูกศรเป็นลูกศรขาเข้าสัญญาณ “x2-” ซึ่งตัวแปรสัญญาณให้ใช้สัญญาณที่ต้นลูกศรขาเข้า “x2-” พบว่าเป็นสัญญาณ “x1+” จึงแทนค่าเป็น “x1p” และพิมพ์ค่าตัวแปรสัญญาณ และใช้กฎการเคลื่อนย้ายลูกศรขาออก 1 ลูกศรเป็นตัวแปรลูกศรขาออกจากสัญญาณ “x2-” ตัวอย่างที่ 2 อ้างอิงจากสัญญาณ “x3+” พบว่ามี 2 ลูกศรขาเข้าและ 2 ลูกศรขาออกทำให้ได้โครงสร้าง 2 ส่วน คือ ส่วนที่ 1 ใช้สัญญาณ “x1+” เป็นตัวแปรลูกศรขาเข้าและส่วนที่ 2 ใช้สัญญาณ “x2-” เป็นตัวแปรลูกศรขาเข้า

จากซิกแนลแทรนซิชันกราฟในรูปที่ 3-9(ก) สามารถกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟได้ดังรูปที่ 3-9(ข) ดังนี้

- ใช้สัญญาณ a+ เป็นสัญญาณอ้างอิงพบว่ามี 1 ลูกศรขาเข้าและ 1 ลูกศรขาออกโดยตัวแปรลูกศรขาเข้า คือ “cmap” และตัวแปรลูกศรขาออก คือ “apcp” ในกรณีเป็นสัญญาณอ้างอิงสัญญาณแรกให้เริ่มต้นด้วย “do” จึงสามารถกำหนดโครงสร้างได้ดังนี้

“do ::atomic{inp1(cmap,cm); printf("printf(cm=%d)",cm) -> outp1(apcp)}”

- ใช้สัญญาณ c+ เป็นสัญญาณอ้างอิงพบว่ามีลูกศรขาเข้า 2 ลูกศร คือ “apcp, bpcp” และลูกศรขาออก 2 ลูกศร คือ “cpam, cpbm” จึงสามารถกำหนดโครงสร้างได้ดังนี้

“::atomic{ inp(bpcp,bp);printf("printf(bp=%d)",bp) -> outp2(cpam,cpbm)}

::atomic{ inp(apcp,ap);printf("printf(ap=%d)",ap) -> outp2(cpam,cpbm)}”

- ใช้สัญญาณ a- เป็นสัญญาณอ้างอิงพบว่ามีลูกศรขาเข้า 1 ลูกศร คือ “cpam” และลูกศรขาออก 1 ลูกศร คือ “amcm” จึงสามารถกำหนดโครงสร้างได้ดังนี้

“::atomic{ inp(cpam,cp);printf("printf(cp=%d)",cp) -> outp1(amcm)}”

- ใช้สัญญาณ b- เป็นสัญญาณอ้างอิงพบว่ามีลูกศรขาเข้า 1 ลูกศร คือ “cpbm” และลูกศรขาออก 1 ลูกศร คือ “bmcm” จึงสามารถกำหนดโครงสร้างได้ดังนี้

“::atomic{ inp(cpbm,cp);printf("printf(cp=%d)",cp) -> outp1(bmcm)}”

- ใช้สัญญาณ c- เป็นสัญญาณอ้างอิงพบว่ามีลูกศรขาเข้า 2 ลูกศร คือ “amcm, bmcm” และลูกศรขาออก 2 ลูกศร คือ “cmap, cmbp” จึงสามารถกำหนดโครงสร้างได้ดังนี้

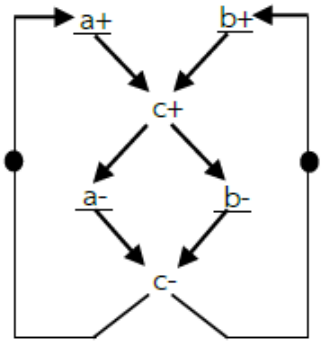
“::atomic{ inp(amcm,am);printf("printf(am=%d)",am) -> outp2(cmap,cmbp)}

::atomic{ inp(bmcm,bm);printf("printf(bm=%d)",bm) -> outp2(cmap,cmbp)}”



- ใช้สัญญาณ b+ เป็นสัญญาณอ้างอิงพบว่ามีลูกศรขาเข้า 1 ลูกศร คือ “cmbp” และลูกศรขาออก 1 ลูกศร คือ “bpcp”และเป็นกรณีสัญญาณสุดท้ายให้เติม “od” จึงสามารถกำหนดโครงสร้างได้ดังนี้

“::atomic{ inp(cmbp,cm);printf("printf(cm=%d)",cm) -> outp1(bpcp)}od}}”




(ก)

```
do
        ::atomic{ inp(cmap,cm);printf("printf(cm=%d)",cm) ->
outp1(apcp)}
        ::atomic{ inp(bpcp,bp);printf("printf(bp=%d)",bp) ->
outp2(cpam,cpbm)}
        ::atomic{ inp(apcp,ap);printf("printf(ap=%d)",ap) ->
outp2(cpam, cpbm)}
        ::atomic{ inp(cpam,cp);printf("printf(cp=%d)",cp) ->
outp1(amcm)}
        ::atomic{ inp(cpbm,cp);printf("printf(cp=%d)",cp) ->
outp1(bmcm)}
        ::atomic{ inp(amcm,am);printf("printf(am=%d)",am) ->
outp2(cmap, cmbp)}
        ::atomic{ inp(bmcm,bm);printf("printf(bm=%d)",bm) ->
outp2(cmap, cmbp)}
        ::atomic{ inp(cmbp,cm);printf("printf(cm=%d)",cm) ->
outp1(bpcp)}
od}}
```

(ข)

รูปที่ 3-9 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A2 ส่วนที่ 4

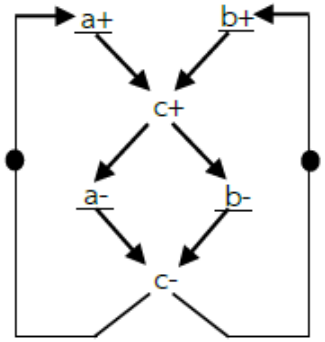
**5) การกำหนดลำดับการทำงานของการตรวจสอบ**

การกำหนดลำดับการทำงานของการตรวจสอบกระทำโดยสร้างฟังก์ชัน “init” แล้วใช้คำสั่ง “run ชื่อฟังก์ชัน(ค่า k)” โดยจำนวนคำสั่ง run ให้กระทำให้ครบตามจำนวนโทเค็น

จากรูปที่ 3-9(ก) พบว่ามี 2 โทเค็น จึงสามารถกำหนดลำดับการทำงานของการตรวจสอบได้ดังนี้

“init{run P3(1); run P3(2);}”

จากวิธีการแปลงรหัสโพรเมลาแบบ A2 ทั้ง 5 ส่วน สามารถแปลงซิกแนลแทรนซิชันกราฟรูปที่ 3-10(ก) เป็นรหัสโพรเมลาแบบ A2 ได้ดังรูปที่ 3-10(ข)




(ก)

```
/*ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรสัญญาณ*/
byte bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp, apcp;
byte ap, am, bp, bm, cp, cm;

/*ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น*/
#define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++;y++; fi
#define outp1(x1)   if :: (x1 != 10) -> x1++; :: else -> break; fi
#define outp2(x1,x2)            if :: (x1 != 10) -> x1++; :: ( x2 != 10) -> x2++; ::
else -> break; fi

/*ส่วนที่ 3 การกำหนดชุดค่าตั้งต้น*/
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> cmap=2;cmbp=10;bpcp=0;cpam=0;cpbm=0;amcm=0;bmcm=0;
apcp=0;ap=0;am=0;bp=0;bm=0;cp=0;cm=0;
   :: k==2 -> cmbp=2;cmap=10;bpcp=0;cpam=0;cpbm=0;amcm=0;bmcm=0;
apcp=0;ap=0;am=0;bp=0;bm=0;cp=0;cm=0;
  fi

/*ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ*/
  do
            ::atomic{ inp(cmbp,cm);printf("printf(cm=%d)",cm) -> outp1(bpcp)}
            ::atomic{ inp(bpcp,bp);printf("printf(bp=%d)",bp) ->
outp2(cpam,cpbm)}
            ::atomic{ inp(apcp,ap);printf("printf(ap=%d)",ap) ->
outp2(cpam,cpbm)}
            ::atomic{ inp(cpam,cp);printf("printf(cp=%d)",cp) -> outp1(amcm)}
            ::atomic{ inp(cpbm,cp);printf("printf(cp=%d)",cp) -> outp1(bmcm)}
            ::atomic{ inp(amcm,am);printf("printf(am=%d)",am) ->
outp2(cmap,cmbp)}
            ::atomic{ inp(bmcm,bm);printf("printf(bm=%d)",bm) ->
outp2(cmap,cmbp)}
            ::atomic{ inp(cmap,cm);printf("printf(cm=%d)",cm) -> outp1(apcp)}
  od}}

/*ส่วนที่ 5 การกำหนดลำดับการทำงานของการตรวจสอบ*/
init{run P3(1);  run P3(2);}
```

(ข)

รูปที่ 3-10 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) รหัสโพรเมลาแบบ A2

### 3.2 วิธีการแปลงซิกแนลแทรนซิชันกราฟเป็นตรรกะเวลาเชิงเส้น

ตรรกะเวลาเชิงเส้นเป็นตรรกะที่ใช้ในการทวนสอบด้วยเครื่องมือสปิน ซึ่งช่วยให้สามารถทวนสอบเงื่อนไขนั้นๆ ได้ตลอดเวลาหรือในเวลาที่ต้องการ โดยเงื่อนไขที่ตั้งขึ้นนั้นเพื่อทวนสอบคุณสมบัติต่างๆ ที่สนใจตามนิยามของคุณสมบัตินั้นๆ ซึ่งจะต้องทวนสอบร่วมกับแบบจำลองในรหัสโพรเมลาที่เกี่ยวข้อง

ในการสร้างตรรกะเวลาเชิงเส้นสามารถแบ่งได้เป็น 2 ส่วน คือ เงื่อนไขและการกำหนดตรรกะเวลาเชิงเส้น โดยประกาศคำสั่ง “#define ชื่อคุณสมบัติ (เงื่อนไข) ”และกำหนดตรรกะเวลาเชิงเส้นโดยประกาศคำสั่ง “ltl ชื่อตรรกะเวลาเชิงเส้น {ตัวดำเนินการ ชื่อคุณสมบัติ}”

ส่วนนี้จะกล่าวถึงกฎการสร้างตรรกะเวลาเชิงเส้น 5 คุณสมบัติประกอบด้วย

1) คุณสมบัติความปลอดภัย จากนิยาม “ในแต่ละลูกศรของซิกแนลแทรนซิชันกราฟจะต้องมีโทเค็นไม่เกินหนึ่งโทเค็น” จึงตรวจสอบว่าไม่มีโทเค็นในตัวแปรลูกศรเกิน 1 โทเค็นตลอดเวลา

2) คุณสมบัติไลฟ์เนส จากนิยาม “ทกๆ จุดของกราฟจะต้องสามารถเข้าถึงได้เมื่อมีการเริ่มต้นกราฟใหม่ และไลฟ์เนสยังรวมถึงทุกๆ แทรนซิชันจะต้องสามารถเปลี่ยนแปลงได้ตลอด เพราะมีอย่างน้อย 1 สถานะที่สามารถเข้าถึงได้” จึงตรวจสอบว่าในที่สุดแล้วทุกแทรนซิชันมีการเคลื่อนผ่านของโทเค็นมากกว่า 1 ครั้ง

3) คุณสมบัติความทนทาน จากนิยาม “ทุกๆสัญญาณจะสามารถเปลี่ยนได้เมื่อครบกำหนด” ซึ่งมักจะพบปัญหาที่สัญญาณที่มีลูกศรขาออกมากกว่า 1 ลูกศร เนื่องจากสัญญาณนี้เป็นจุดเปลี่ยนของสัญญาณต่อมา ซึ่งสามารถทำงานได้พร้อมกันจึงต้องตรวจสอบว่าหากเกิดการเปลี่ยนค่าดิจิทัลของสัญญาณนี้แล้วสัญญาณอื่นๆ จะต้องสามารถเปลี่ยนแปลงได้ถูกต้องดังนิยาม

4) คุณสมบัติความต้องกัน จากนิยามว่า “เมื่อมีการเคลื่อนย้ายโทเค็นในซิกแนลแทรนซิชันกราฟที่เกี่ยวข้องกับสัญญาณนั้นๆ แล้วค่าของสัญญาณนั้นต้องเปลี่ยนแปลง” สำหรับคุณสมบัตินี้นั้นปกติแล้วจะต้องทวนสอบโดยกราฟสถานะซึ่งแสดงค่าดิจิทัลของแต่ละสัญญาณในทุกสถานะ แต่เมื่อการทำงานของวงจรซับซ้อนขึ้นการสร้างกราฟสถานะจึงเป็นไปได้ยาก จึงได้มีงานวิจัยเสนอวิธีการทวนสอบโดยทวนสอบจากวัฏจักรที่มีจุดยอดไม่ซ้ำกันและได้สรุปว่าถ้าพบวัฏจักรที่มีจุดยอดไม่ซ้ำกันที่มีสัญญาณนั้นๆในค่าดิจิทัล 1 และ 0 แสดงว่าสัญญาณนั้นสามารถเปลี่ยนแปลงได้ดังนิยามของคุณสมบัติความต้องกัน [3, 13]

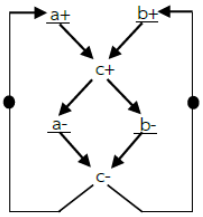
5) คุณสมบัติการกำหนดสถานะที่สมบูรณ์ จากนิยาม “ทุกๆ เพลสในซิกแนลแทรนซิชันกราฟจะต้องไม่มีสัญลักษณ์ซ้ำกัน” ซึ่งปกติแล้วจะทวนสอบจากกราฟสถานะ [13-15] แต่เช่นเดียวกับคุณสมบัติความต้องกันจึงได้มีงานวิจัยเสนอคุณสมบัติเชิงล็อคเพื่อทวนสอบคุณสมบัตินี้ [16-18] ซึ่ง

สามารถสรุปได้ว่า ถ้าพบแทรนซิทีฟล็อคที่ประกอบด้วยทุกสัญญาณที่ไม่ใช่อินพุตของซิกแนลแทรนซิชันกราฟแสดงว่าพบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ [18]

### 3.2.1 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย

คุณสมบัติความปลอดภัย สามารถทวนสอบได้โดยการตรวจสอบว่า “จำนวนโทเค็นในทุกเวลาว่าไม่มีโทเค็นในลูกศรใดๆ มีมากกว่า 1 โทเค็น” ซึ่งจะใช้รหัสโพรเมลาแบบ A1 ซึ่งจะใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ

ในการสร้างกฎนั้นกระทำโดย (1) สร้างเงื่อนไขในการทวนสอบ คือ ให้ตัวแปรลูกศรทุกลูกศรมีค่าน้อยกว่า 2 และ (2) สร้างตรรกะเวลาเชิงเส้นด้วยตัวดำเนินการ “ตลอดไป” จากรูปที่ 3-11(ก) สามารถสร้างตรรกะเวลาเชิงเส้นได้ดังรูปที่ 3-11(ข)




```
/*สร้างเงื่อนไขและการกำหนดตรรกะเวลาเชิงเส้น*/
#define safety (apcp<2 && bpcp<2 && cpam<2 && cpbm<2
&& amcm<2 && bmcm<2  && cmap<2 && cmbp<2)
/*สร้างตรรกะเวลาเชิงเส้น*/
ltl p1{[] safety}
```

(ก) (ข)

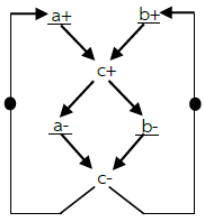
รูปที่ 3-11 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย

### 3.2.2 ตรรกะเวลาเชิงเส้นคุณสมบัติไลฟ์เนส



คุณสมบัติไลฟ์เนส สามารถทวนสอบได้โดยการตรวจสอบว่า “ทุกๆ แทรนซิชันมีโทเค็นเคลื่อนผ่านมากกว่า 1 ครั้งหรือไม่” ซึ่งจะใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ

ในการสร้างกฎกระทำโดย (1) สร้างเงื่อนไขในการทวนสอบ คือ ให้ตัวแปรแทรนซิชันทุกตัวแปรมีค่ามากกว่า 1 และ (2) สร้างตรรกะเวลาเชิงเส้นด้วยตัวดำเนินการ “ในที่สุด” จากรูปที่ 3-12(ก) สามารถสร้างตรรกะเวลาเชิงเส้นได้ดังรูปที่ 3-12(ข)




```
/*สร้างเงื่อนไขและการกำหนดตรรกะเวลาเชิงเส้น*/
#define live (t1>1 && t2>1 && t3>1 && t4>1 && t5>1 &&
t6>1 && t7>1 && t8>1)
/*สร้างตรรกะเวลาเชิงเส้น*/
ltl p2 {<> live}
```

(ก) (ข)

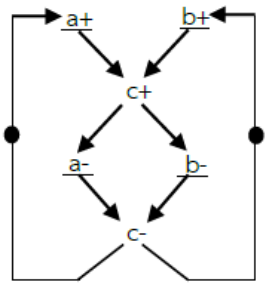
รูปที่ 3-12 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) ตรรกะเวลาเชิงเส้นคุณสมบัติไลฟ์เนส

### 3.2.3 ตรรกะเวลาเชิงเส้นคุณสมบัติความทนทาน

คุณสมบัติความทนทาน สามารถทวนสอบได้โดยการตรวจสอบ “ในทุกๆ สัญญาณที่มีลูกศรขาออกมากกว่า 1 ลูกศรถ้ามีโทเค็นในลูกศรขาออกของสัญญาณตรงข้ามสัญญาณนั้นแล้วลูกศรขาออกของสัญญาณนั้นจะต้องไม่มีโทเค็นเหลืออยู่” ซึ่งจะใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ

ในการสร้างกฎนั้นกระทำโดย (1) สร้างเงื่อนไขในการทวนสอบ โดยขั้นตอนแรกให้หาว่ามีสัญญาณใดบ้างที่มีลูกศรขาออกมากกว่า 1 ลูกศรจากนั้นให้นำตัวแปรลูกศรขาออกจากสัญญาณนั้นทุกตัวมีค่าเท่ากับ 0 เชื่อมตัวแปรแต่ละตัวด้วยเครื่องหมาย “&&” จากนั้นหาสัญญาณตรงข้ามและให้ตัวแปรลูกศรขาออกจากสัญญาณตรงข้ามทุกตัวมีค่าเท่ากับ 1 ถ้ามีมากกว่า 1 ลูกศรให้เชื่อมกันด้วยเครื่องหมาย “||” จากนั้นนำมาเชื่อมกันด้วยเครื่องหมาย “->” และ (2) สร้างตรรกะเวลาเชิงเส้นด้วยตัวดำเนินการ “ตลอดไป”

จากรูปที่ 3-13(ก) พบว่า มีสัญญาณที่มีตัวแปรที่มีสัญญาณขาออกมากกว่า 2 ลูกศร 2 สัญญาณ คือ c+ และ c- จึงนำตัวแปรลูกศรขาออกของทั้งสองสัญญาณมาเท่ากับ 0 และลูกศรขาออกจากสัญญาณตรงข้ามเท่ากับ 1 และเชื่อมกันด้วยเครื่องหมาย “->” สามารถสร้างตรรกะเวลาเชิงเส้นได้ดังรูปที่ 3-13(ข)




(ก)

```
/*สร้างเงื่อนไขและการกำหนดตรรกะเวลาเชิงเส้น*/
#define persist (((cmap==1 || cmbp==1)  -> (cpam==0 &&
cpbm==0))  && ((cpam==1 || cpbm==1)  -> (cmap==0 &&
cmbp==0)))
/*สร้างตรรกะเวลาเชิงเส้น*/
ltl p3 {[] persist}
```

(ข)

รูปที่ 3-13 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) ตรรกะเวลาเชิงเส้นคุณสมบัติความทนทาน

### 3.2.4 ตรรกะเวลาเชิงเส้นคุณสมบัติความต้องกัน

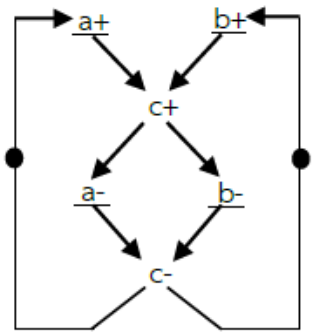
คุณสมบัติความต้องกัน สามารถทวนสอบได้โดยการตรวจสอบว่า “แต่ละสัญญาณจะต้องพบค่าดิจิทัลของสัญญาณนั้นๆ เป็น 0 และ 1 ในวัฏจักรที่มีจุดยอดไม่ซ้ำกันอย่างน้อยหนึ่งวัฏจักร” ซึ่งจะใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ

การสร้างเงื่อนไขในการทวนสอบคุณสมบัติความต้องกัน โดยหาว่ามีสัญญาณใดบ้างในซิกแนลแทรนซิชันกราฟจากนั้น และกำหนดให้ “สัญญาณบวกและสัญญาณลบมีค่าเท่ากับ 1 และ

ตัวแปรลูกศรที่มีมาร์กกิ้งตั้งต้นตัวใดตัวหนึ่งมีค่าเท่ากับ 4” เช่น สัญญาณ a ที่มีมาร์กกิ้งตั้งต้นที่ตัวแปรลูกศร “apxp” และ “bpxp” จะได้เงื่อนไขว่า “#define consist(ap==1 && am==1 && (apxp==4 || bpxp==4))” การตั้งค่าตัวแปรที่มีมาร์กกิ้งตั้งต้นเท่ากับ 4 เพื่อเป็นการตรวจสอบว่าโทเค็นเดินทางกลับมาที่จุดเริ่มต้น การตั้งกฎการเคลื่อนย้ายขาเข้าและการกำหนดชุดค่าตั้งต้นในส่วนรหัสโพรเมลาแบบ A2 ที่กำหนดให้ตัวแปรลูกศรมีค่าเท่ากับ 2 ก็เพื่อการตรวจสอบวัฏจักรที่มีจุดยอดไม่ซ้ำกัน

จากนั้นสร้างตรรกะเวลาเชิงเส้นคุณสมบัติความต้องกันด้วยตัวดำเนินการ “!<>” ซึ่งจะตรวจว่าในทุกๆ กรณีของเงื่อนไขนั้นเป็นเท็จเสมอหรือไม่ จึงทำให้พบความผิดพลาดเมื่อพบกรณีที่ตรงตามเงื่อนไข

จากรูปที่ 3-14(ก) พบว่า มีสัญญาณในซิกแนลแทรนซิชันกราฟ 3 สัญญาณ คือ a, b และ c และมีตัวแปรลูกศรที่มีมาร์กกิ้งตั้งต้น 2 ตัวแปร คือ “cmap” และ “cmbp” จึงสามารถสร้างเงื่อนไขในการทวนสอบได้ 3 เงื่อนไข และ 3 ตรรกะเวลาเชิงเส้นตามเงื่อนไขดังรูปที่ 3-14(ข)




(ก)

```
/*สร้างเงื่อนไขและการกำหนดตรรกะเวลาเชิงเส้น*/
#define consist1 (ap>=1 && am>=1 && (cmap==4 || cmbp==4))
#define consist2 (bp>=1 && bm>=1 && (cmap==4 || cmbp==4))
#define consist3 (cp>=1 && cm>=1 && (cmap==4 || cmbp==4))
/*สร้างตรรกะเวลาเชิงเส้น*/
ltl p1 {!<> consist1}
ltl p2 {!<> consist2}
ltl p3 {!<> consist3}
```

(ข)



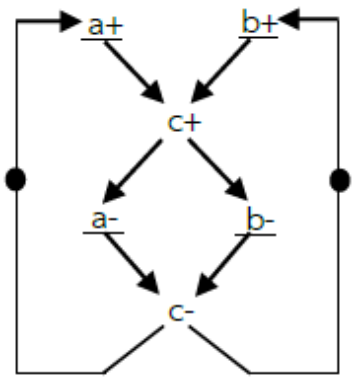
รูปที่ 3-14 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) ตรรกะเวลาเชิงเส้นคุณสมบัติความต้องกัน

### 3.2.5 ตรรกะเวลาเชิงเส้นคุณสมบัติการกำหนดสถานะที่สมบูรณ์

คุณสมบัติการกำหนดสถานะที่สมบูรณ์ สามารถทวนสอบได้โดยการตรวจสอบหา “แทรนซิทีฟล็อคที่ประกอบด้วยทุกสัญญาณที่ไม่ใช่อินพุตของซิกแนลแทรนซิชันกราฟ” การหาแทรนซิทีฟล็อคนั้นจะต้องพบฟูลล็อคก่อน เนื่องจากแทรนซิทีฟล็อคจะต้องประกอบด้วยสัญญาณในเซตของฟูลล็อค โดยความสัมพันธ์เชิงล็อคนั้นจะต้องหาจากวัฏจักรที่มีจุดยอดไม่ซ้ำกัน ดังนั้นในการทวนสอบคุณสมบัติจึงต้องทวนสอบ 2 ขั้นตอน โดยขั้นตอนที่ 1 สร้างตรรกะเวลาเชิงเส้นของฟูลล็อคเพื่อหาฟูลล็อค และขั้นตอนที่ 2 นำเซตสัญญาณที่ได้จากฟูลล็อคมาสร้างตรรกะเวลาเชิงเส้นของแทรนซิทีฟล็อคเพื่อหาแทรนซิทีฟล็อค และเมื่อนำไปทวนสอบก็จะได้ผลของคุณสมบัติการกำหนดสถานะที่สมบูรณ์

ขั้นตอนที่ 1 การสร้างตรรกะเวลาเชิงเส้นของฟูลล็อคกระทำโดย (1) สร้างเงื่อนไขในการทวนสอบ โดยการหาว่าทุกคู่สัญญาณที่สนใจพบค่าดิจิทัล 1 และ 0 ในวัฏจักรที่มีจุดยอดไม่ซ้ำกัน เช่น คู่สัญญาณ a และ b และมีตัวแปรลูกศรที่มีมาร์กกิ้งตั้งต้นที่ “apxp” และ “bpxp” จะได้เงื่อนไขว่า “#define fullab (ap==1 && am==1 && bp==1 && bm==1 && (apxp==4 || bpxp==4))” โดยเงื่อนไขในวงเล็บสุดท้ายเพื่อตรวจว่าพบวัฏจักรที่มีจุดยอดไม่ซ้ำกัน และ (2) สร้างตรรกะเวลาเชิงเส้นโดยตัวดำเนินการ “!<>” ซึ่งในขั้นตอนที่ 1 นี้จะต้องทวนสอบให้ครบทุกคู่สัญญาณของซิกแนลแทรนซิชันกราฟจึงได้เงื่อนไขเท่ากับจำนวนคู่สัญญาณที่จับได้และได้ตรรกะเวลาเชิงเส้นเท่ากับจำนวนเงื่อนไขที่ตั้ง

จากรูปที่ 3-15(ก) พบว่า มีสัญญาณในซิกแนลแทรนซิชันกราฟ 3 สัญญาณ คือ a, b และ c จึงสามารถจับคู่สัญญาณได้ 3 คู่สัญญาณ คือ คู่สัญญาณ a, b คู่สัญญาณ a, c คู่สัญญาณ b, c จึงสามารถสร้างเงื่อนไขและตรรกะเวลาเชิงเส้นของฟูลล็อคได้ดังรูปที่ 3-15(ข)




(ก)

```
/*สร้างเงื่อนไขและการกำหนดตรรกะเวลาเชิงเส้น*/
#define fullab (ap==1 && am==1 && bp==1 && bm==1 &&
(cmap==4 || cmbp==4))
#define fullac (ap==1 && am==1 && cp==1 && cm==1 &&
(cmap==4 || cmbp==4))
#define fullbc (bp==1 && bm==1 && cp==1 && cm==1 &&
(cmap==4 || cmbp==4))
/*สร้างตรรกะเวลาเชิงเส้น*/
ltl c1 {!<> fullab}
ltl c2 {!<> fullac}
ltl c3 {!<> fullbc}
```

(ข)

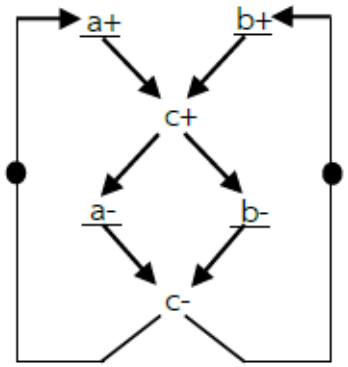


รูปที่ 3-15 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) ตรรกะเวลาเชิงเส้นฟูลล็อค

ขั้นตอนที่ 2 เป็นการนำเซตสัญญาณที่ได้จากฟูลล็อคมาสร้างตรรกะเวลาเชิงเส้นของแทรนซิทีฟล็อค กระทำโดย (1) สร้างเงื่อนไขในการทวนสอบจากคู่สัญญาณที่อยู่ในเซตของฟูลล็อคและสัญญาณที่ไม่ใช่อินพุตทั้งหมด โดยตรวจหาค่าดิจิทัล 1 และ 0 ของทุกสัญญาณในวัฏจักรที่มีจุดยอกไม่ซ้ำกันและต้องทวนสอบให้ครบทุกคู่สัญญาณที่พบฟูลล็อค เช่น พบฟูลล็อคที่คู่สัญญาณ a, b และสัญญาณ x เป็นสัญญาณที่ไม่ใช่อินพุต โดยมีตัวแปรลูกศรที่มีมาร์กกิ้งตั้งต้นคือ “apxp” และ “bpxp” จะได้เงื่อนไขว่า “#define transitive ((ap==1 || am==1) && (bp==1 || bm==1) &&

(cp==1 && cm==1) && (apxp==4 || bpxp==4))” จากกรณีนี้สัญญาณจากเซตของฟูลล็อคจะเชื่อมด้วยเครื่องหมาย “||” และ (2) สร้างตรรกะเวลาเชิงเส้นโดยตัวดำเนินการ “!<>”

จากรูปที่ 3-16(ก) พบว่า คู่สัญญาณ a, c และคู่สัญญาณ b, c เป็นฟูลล็อค จึงสามารถสร้างเงื่อนไขและตรรกะเวลาเชิงเส้นของฟูลล็อคได้ดังรูปที่ 3-16(ข)




(ก)

```
/*สร้างเงื่อนไขและการกำหนดตรรกะเวลาเชิงเส้น*/
#define transitive1((ap==1 || am==1) && (cp==1 || cm==1) &&
(cpam==4 || cpbm==4))
#define transitive2((bp==1 || bm==1) && (cp==1 || cm==1) &&
(cpam==4 || cpbm==4))
/*สร้างตรรกะเวลาเชิงเส้น*/
ltl p4 {!<>transitive1}
ltl p5 {!<>transitive2}
```

(ข)

รูปที่ 3-16 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) ตรรกะเวลาเชิงเส้นของแทรนซิทีฟล็อค

# บทที่ 4
# การพัฒนาเครื่องมือและการใช้งาน

สำหรับบทนี้จะนำแนวคิดและวิธีการที่ได้อธิบายไว้แล้วในบทที่ 3 มาพัฒนาเป็นเครื่องมือสนับสนุนซิกแนลแทรนซิชันกราฟและตรรกะเวลาเชิงเส้น ซึ่งในบทนี้ผู้วิจัยนำเสนอการออกแบบโปรแกรมเชิงวัตถุด้วยภาษายูเอ็มแอลเพื่ออธิบายฟังก์ชันการทำงานของเครื่องมือประกอบไปด้วยแผนภาพกิจกรรม (Activity Diagram) และแผนภาพคลาส (Class Diagram) นำเสนอวิธีการแปลงเน็ตลิสต์เป็นรหัสโพรเมลา นำเสนอสภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือและโครงสร้างส่วนต่อประสานกับผู้ใช้งานของเครื่องมือ และนำเสนอวิธีการทวนสอบคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์ ซึ่งมีรายละเอียดดังนี้

## 4.1 การออกแบบเครื่องมือสนับสนุน

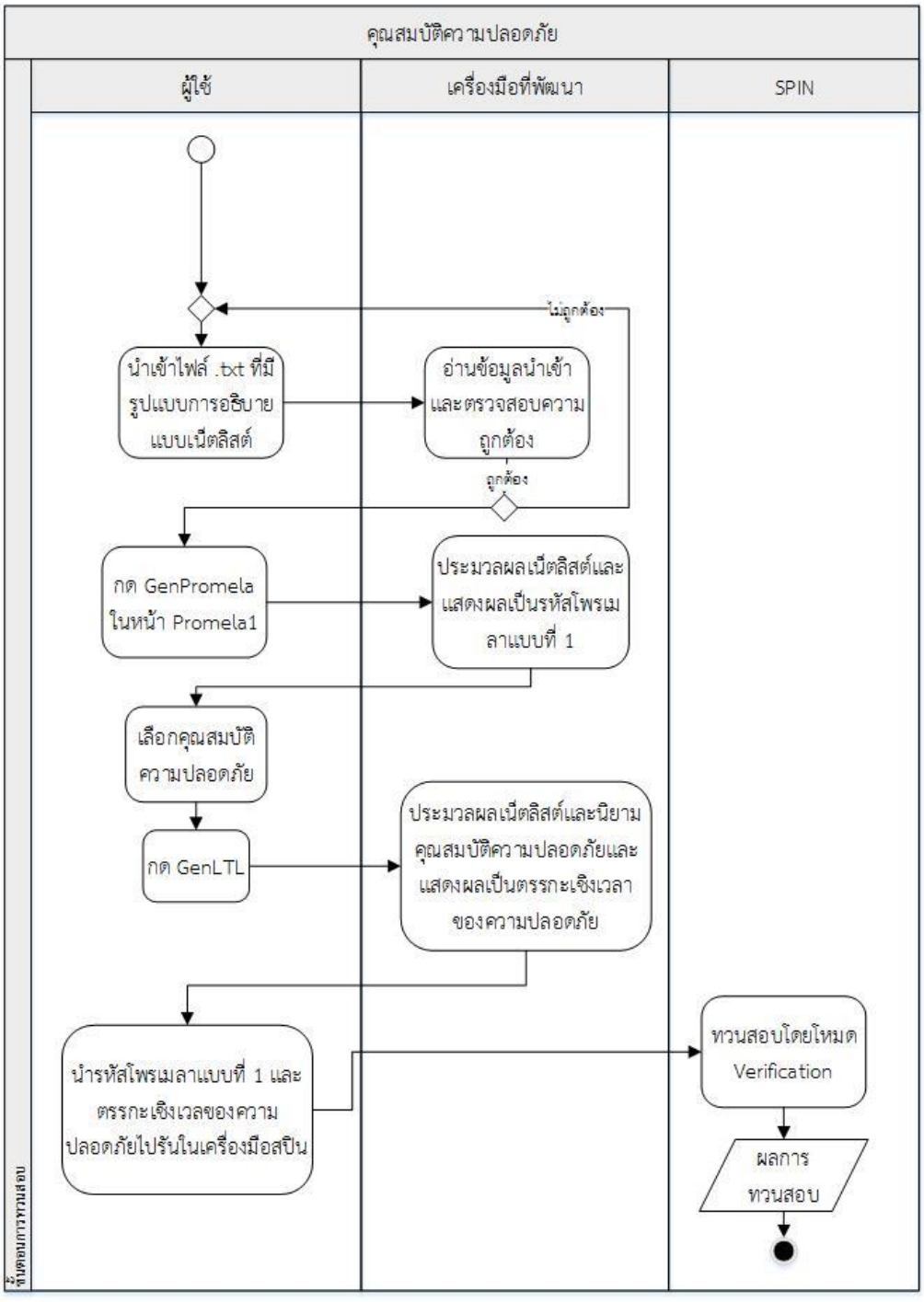
การออกแบบเครื่องมือสนับสนุน จะนำใช้แผนภาพกิจกรรมอธิบายขั้นตอนกระบวนการทำงานของเครื่องมือในการทวนสอบคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์ และใช้แผนภาพคลาสแสดงโครงสร้างของเครื่องมือ

### 4.1.1 แผนภาพกิจกรรม



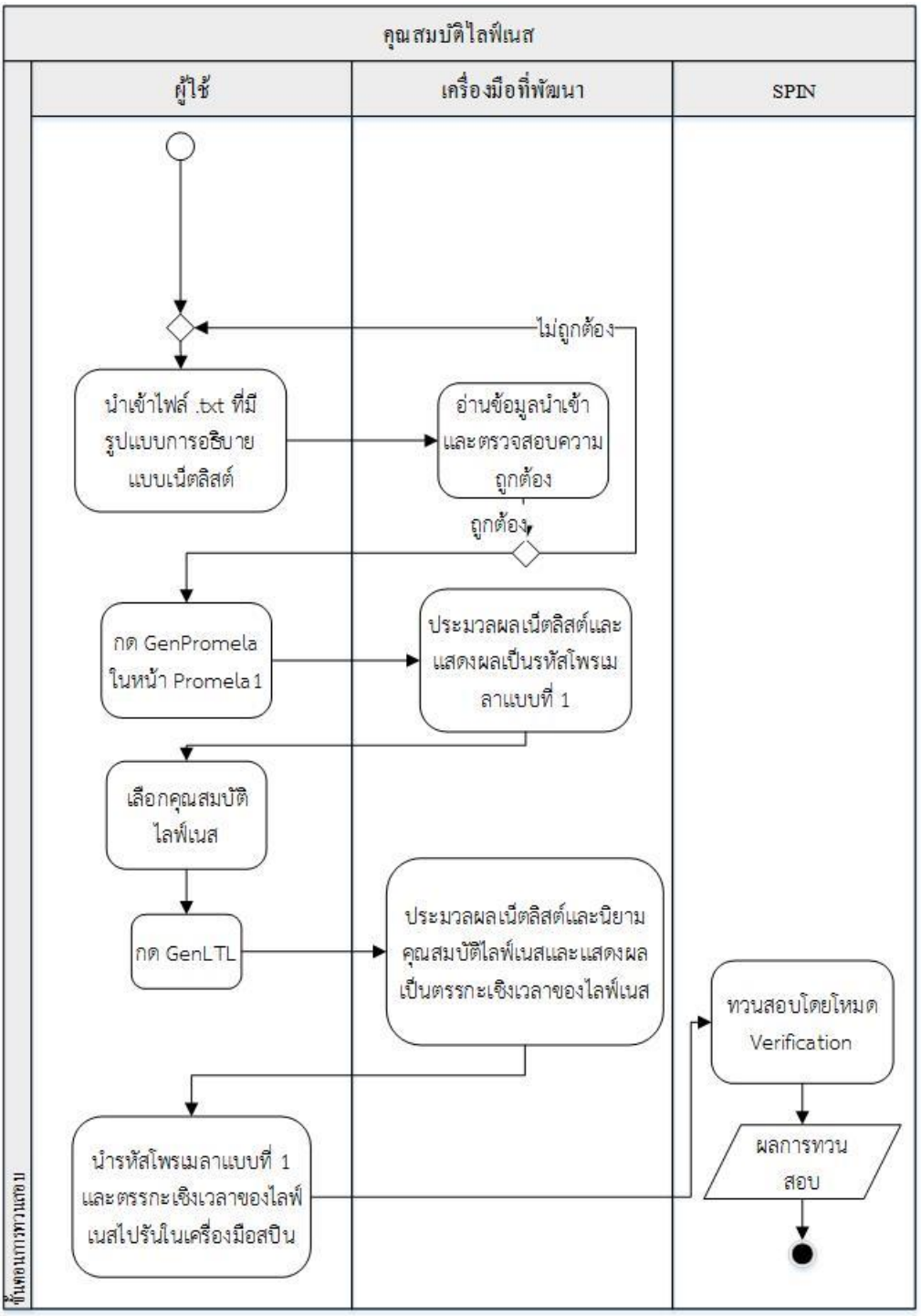
แผนภาพกิจกรรมเป็นแผนภาพที่ใช้แสดงขั้นตอนกระบวนการทำงานของเครื่องมือเพื่ออธิบายกิจกรรมที่เกิดขึ้นในลักษณะกระแสการไหลของการทำงาน ซึ่งในส่วนนี้ได้อธิบายกระบวนการทำงานของเครื่องมือในการทวนสอบคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์ ดังรูปที่ 4-1 ถึง 4-5 ตามลำดับ

1) ทวนสอบคุณสมบัติความปลอดภัย



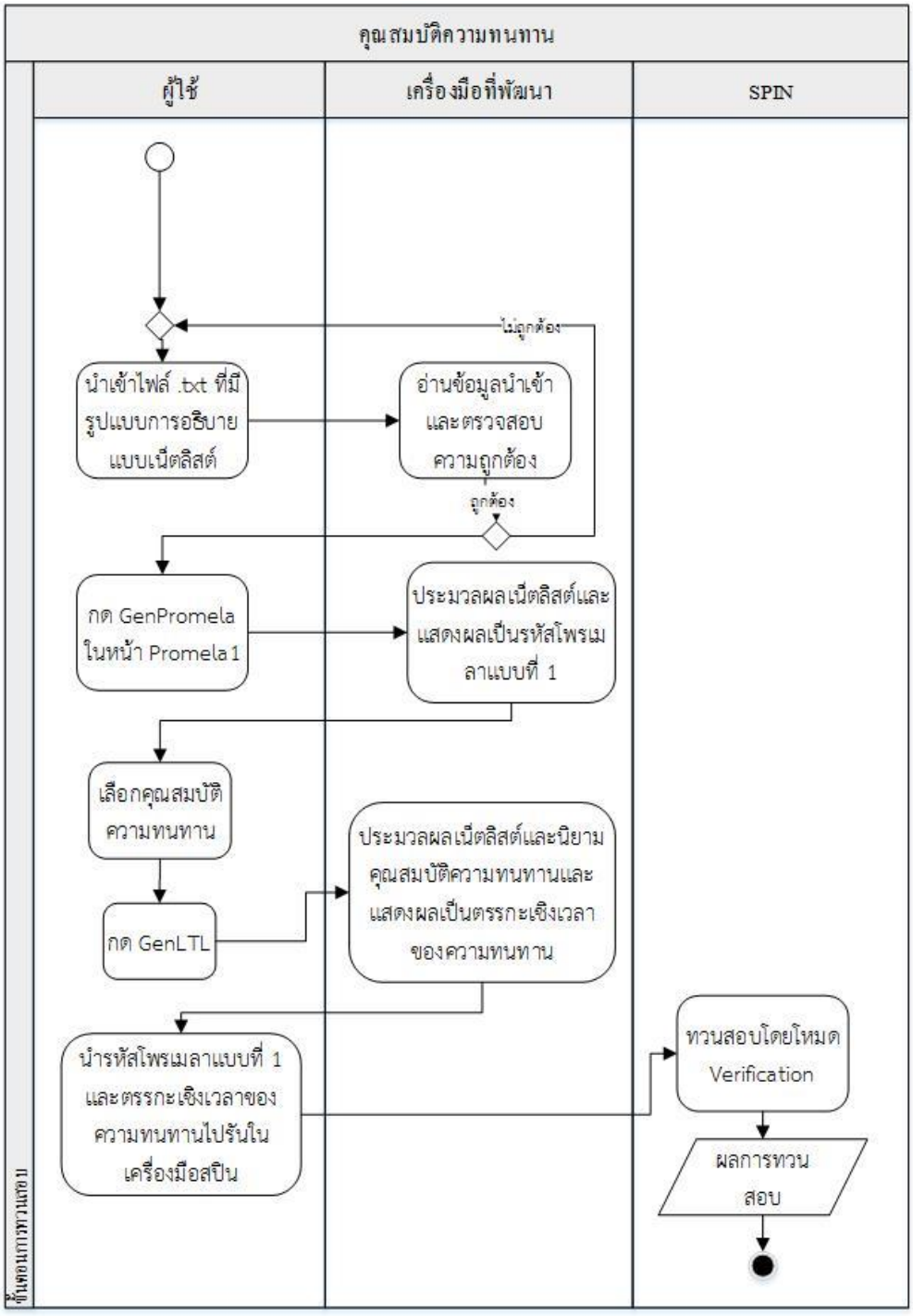

รูปที่ 4-1 แผนภาพกิจกรรมการทวนสอบคุณสมบัติความปลอดภัย

2) ทวนสอบคุณสมบัติไลฟ์เนส



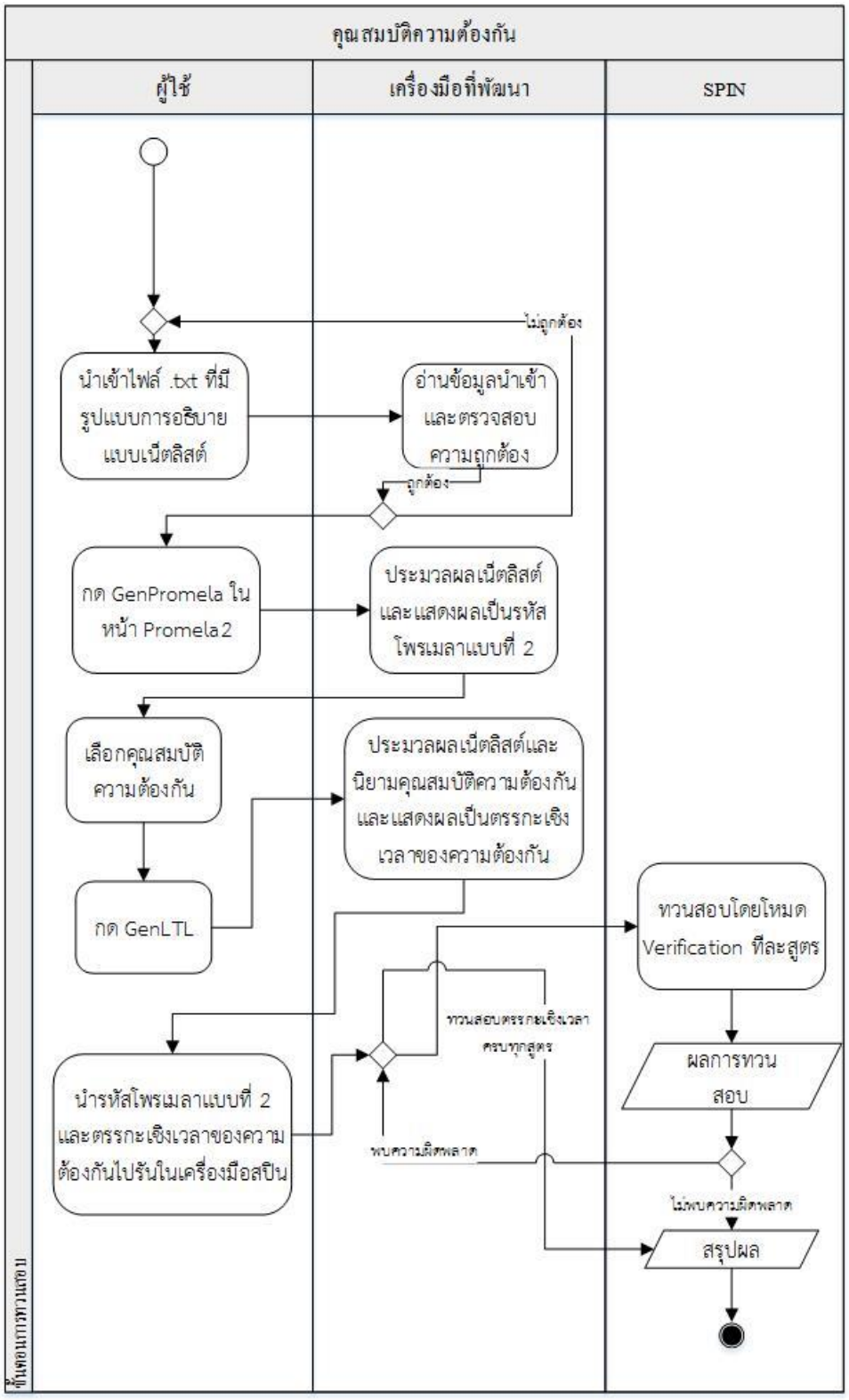

รูปที่ 4-2 แผนภาพกิจกรรมการทวนสอบคุณสมบัติไลฟ์เนส

3) ทวนสอบคุณสมบัติความทนทาน



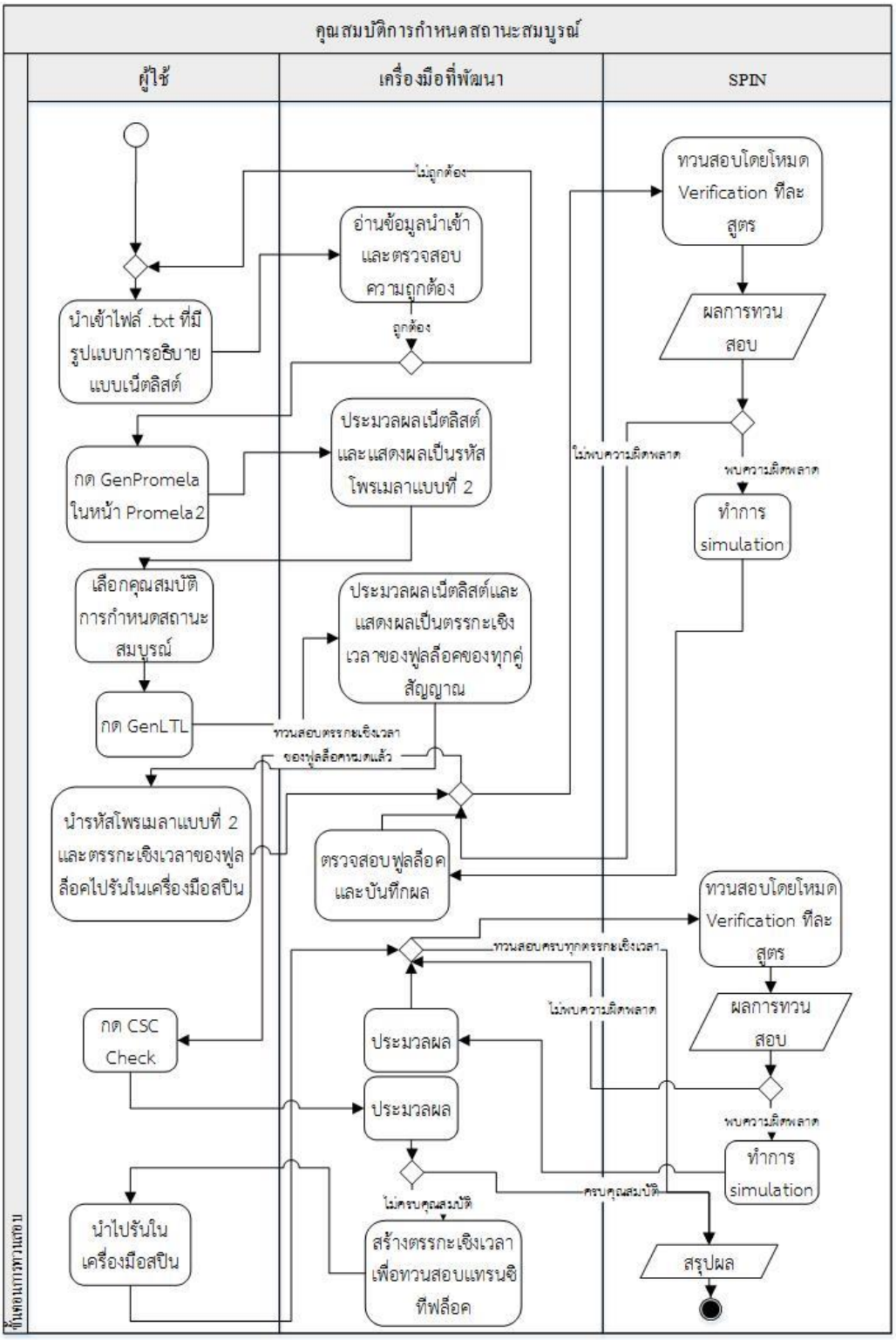

รูปที่ 4-3 แผนภาพกิจกรรมการทวนสอบคุณสมบัติความทนทาน

4) ทวนสอบคุณสมบัติความต้องกัน



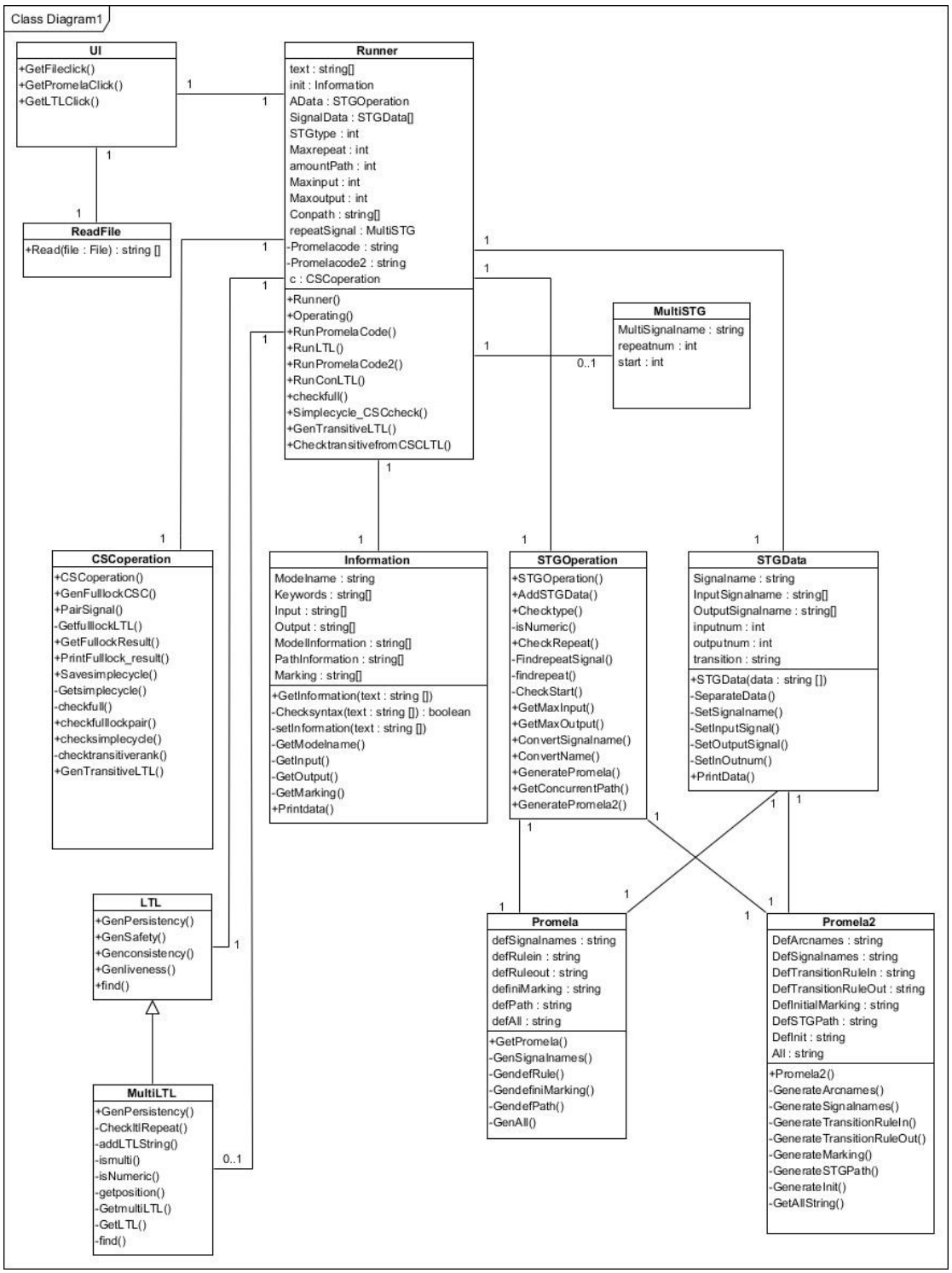

รูปที่ 4-4 แผนภาพกิจกรรมการทวนสอบคุณสมบัติความต้องกัน

5) ทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์




รูปที่ 4-5 แผนภาพกิจกรรมการทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์

### 4.1.2 แผนภาพคลาส




รูปที่ 4-6 แผนภาพคลาสแสดงโครงสร้างของเครื่องมือ

จากแผนภาพคลาสรูปที่ 4-6 สามารถจำแนกหน้าที่ของคลาสต่างๆ ได้ดังนี้

1) คลาส UI ทำหน้าที่เป็นส่วนติดต่อกับผู้ใช้ทั้งรับค่าและแสดงผลโดยโปรแกรมจะปฏิบัติการที่นี่

2) คลาส ReadFile ทำหน้าที่เรียกตำแหน่งที่เก็บไฟล์มาแสดงต่อผู้ใช้และรับข้อมูลจากไฟล์

3) คลาส Runner ทำหน้าที่เป็นตัวกลางในการทำงานทั้งหมดของโปรแกรม

4) คลาส Information ทำหน้าที่จำแนกข้อมูลจากไฟล์ที่อยู่ในรูปแบบมาตรฐานเน็ตลิสต์เพื่อแปลงเป็นข้อมูลของซิกแนลแทรนซิชันกราฟ

5) คลาส STGData ทำหน้าที่เป็นชนิดข้อมูลที่เก็บข้อมูลของสัญญาณขาเข้าและขาออกของซิกแนลแทรนซิชันกราฟเพื่อใช้ในการแปลงเป็นรหัสโพรเมลาและตรรกะเวลาเชิงเส้นต่อไป

6) คลาส STGOperation ทำหน้าที่เป็นตัวนำเนินการแปลงข้อมูลที่มีเป็นรหัสโพรเมลาโดยจะเรียกใช้คลาสอื่นๆและมีส่วนที่คืนค่าเป็นรหัสโพรเมลาแบบ A1 และแบบ A2

7) คลาส Promela ทำหน้าที่เป็นคลาสที่แปลงรหัสโพรเมลาแบบ A1 ซึ่งมี 4 ส่วน จะถูกเรียกใช้โดยคลาส STGOperation และคืนค่าเป็นรหัสโพรเมลาแบบ A1 ทั้ง 4 ส่วน

8) คลาส Promela2 ทำหน้าที่เป็นคลาสที่แปลงรหัสโพรเมลาแบบ A2 ซึ่งมี 5 ส่วน จะถูกเรียกใช้โดยคลาส STGOperation และคืนค่าเป็นรหัสโพรเมลาแบบ A2 ทั้ง 5 ส่วน

9) คลาส MultiSTG ทำหน้าที่เป็นชนิดข้อมูลเพื่อเก็บค่าเฉพาะบางส่วนของซิกแนลแทรนซิชันกราฟประเภทวัฏจักรหลากหลาย

10) คลาส LTL ทำหน้าที่แปลงข้อมูลที่มีเป็นตรรกะเวลาเชิงเส้นของซิกแนลแทรนซิชันกราฟประเภทวัฏจักรเชิงเดี่ยว สำหรับคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน และคุณสมบัติความต้องกัน

11) คลาส MultiLTL ทำหน้าที่แปลงข้อมูลที่มีเป็นตรรกะเวลาเชิงเส้นของซิกแนลแทรนซิชันกราฟประเภทวัฏจักรหลากหลาย สำหรับคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน และคุณสมบัติความต้องกัน

12) คลาส CSCoperation ทำหน้าที่แปลงข้อมูลที่มีเป็นตรรกะเวลาเชิงเส้นสำหรับคุณสมบัติการกำหนดสถานะที่สมบูรณ์ และทำหน้าที่ตรวจสอบฟูลล็อคและแทรนซิทีฟล็อคด้วย

เมื่อผู้ใช้เปิดโปรแกรมส่วนติดต่อผู้ใช้จะแสดงโดยคลาส UI และเมื่อผู้ใช้เรียกไฟล์คลาส ReadFile จะถูกนำมาใช้งานเพื่ออ่านข้อมูลจากไฟล์เมื่อข้อมูลถูกอ่านแล้วจะถูกส่งต่อไปยังคลาส Runner เพื่อเรียกใช้คลาส Information เพื่อการจำแนกข้อมูลจากข้อความที่อ่านได้เป็นข้อมูลสัญญาณประกอบต่างๆ ของซิกแนลแทรนซิชันกราฟ

เมื่อผู้ใช้กดปุ่มเพื่อสร้างรหัสโพรเมลาคลาส Runner จะเรียกคลาส STGData เพื่อเก็บข้อมูลแต่ละลูกศรของซิกแนลแทรนซิชันกราฟ จากนั้นคลาส Runner จะเรียกคลาส STGOperation เพื่อสร้างรหัสโพรเมลาโดย STGOperation จะเรียกคลาส Promela และคลาส Promela2 เพื่อสร้างรหัสโพรเมลาและส่งกลับไปแสดงผลที่ UI ผ่านคลาส Runner

เมื่อผู้ใช้กดสร้างตรรกะเวลาเชิงเส้นคลาส Runner จะเรียกคลาส LTL เพื่อสร้างตรรกะเวลาเชิงเส้นหากเป็นซิกแนลแทรนซิชันกราฟประเภทวัฏจักรเชิงเดี่ยว และ MultiLTL หากเป็นประเภทวัฏจักรหลากหลาย เมื่อผู้ใช้เลือกคุณสมบัติการกำหนดสถานะที่สมบูรณ์คลาส Runner จะเรียกคลาส CSCoperation เพื่อสร้างตรรกะเวลาเชิงเส้นและการหาตรวจหาฟูลล็อคและแทรนซิทีฟล็อคและแสดงผลผ่านคลาส UI

**4.2 การแปลงเน็ตลิสต์เป็นรหัสโพรเมลา**

ในส่วนนี้จะเป็นการแสดงวิธีการแปลงเน็ตลิสต์เป็นรหัสโพรเมลาแบบ A1 และ A2 ซึ่งจะอธิบายในรูปแบบรหัสเทียม การแปลงเน็ตลิสต์นี้เป็นส่วนหนึ่งของเครื่องมือที่พัฒนาโดยจะอธิบายโดยไม่ลงรายละเอียดระดับล่าง

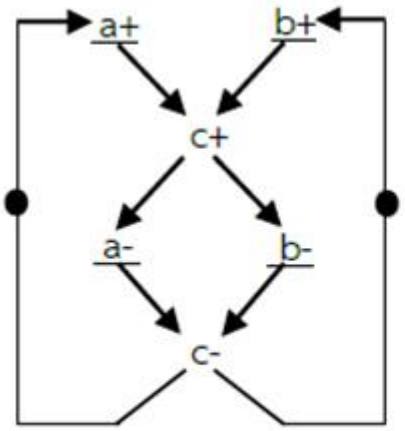
เน็ตลิสต์ประกอบด้วย 3 ส่วน

1) ส่วนข้อมูลซึ่งประกอบด้วยบรรทัด .model .inputs .outputs

2) ส่วนโครงสร้างของซิกแนลแทรนซิชันกราฟประกอบด้วยบรรทัดระหว่าง .graph และ .marking

3) ส่วนกำหนดมาร์กกิ้งตั้งต้นประกอบด้วยบรรทัด .marking

ในการเขียนอธิบายถึงแต่ละสัญญาณของเน็ตลิสต์นั้นจะใช้ช่องว่างเป็นตัวแบ่ง ในบรรทัด .inputs และ .outputs ตัวอักษรที่อยู่ติดกันและขั้นด้วยช่องว่าง หมายถึง แต่ละสัญญาณ เช่น ".inputs a b" หมายถึงสัญญาณ a และ b เป็นสัญญาณขาเข้า และในส่วนโครงสร้างจะใช้ช่องว่างเป็นตัวแบ่งเช่นกัน แต่สัญญาณซ้ายสุดหมายถึงสัญญาณที่เป็นจุดบนซิกแนลแทรนซิชันกราฟ และสัญญาณถัดมา หมายถึง จุดปลายของลูกศร ในกรณีที่บรรทัดนั้นๆ มีมากกว่า 2 สัญญาณจะหมายถึงการมีมากกว่า 1 ลูกศร เช่น "c+ a- b-" สัญญาณ c+ ซึ่งเป็นสัญญาณซ้ายสุดเป็นจุดบนซิกแนลแทรนซิชันกราฟ และสัญญาณ a- คือลูกศรหนึ่งระหว่าง c+ $\longrightarrow$ a- และลูกศรหนึ่งระหว่าง c+ $\longrightarrow$ b- สามารถแสดงตัวอย่างเน็ตลิสต์ที่อธิบายซิกแนลแทรนซิชันกราฟได้ดังรูปที่ 4-7




(ก)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ข)

รูปที่ 4-7 (ก) ตัวอย่างซิกแนลแทรนซิชันกราฟ (ข) ตัวอย่างเน็ตลิสต์

### 4.2.1 การแปลงเน็ตลิสต์เป็นรหัสโพรเมลา A1

การแปลงเน็ตลิสต์เป็นรหัสโพรเมลาแบบ A1 นั้นแปลงตามวิธีการแปลงซิกแนลแทรนซิชันกราฟเป็นรหัสโพรเมลาแบบ A1 ในหัวข้อ 3.1.1 ซึ่งการแปลงเน็ตลิสต์เป็นรหัสโพรเมลาแบบ A1 ประกอบด้วยรหัสเทียม 5 รหัสเทียม ดังนี้

1) การกำหนดตัวแปรของลูกศรและตัวแปรแทรนซิชัน
2) กฎการเคลื่อนย้ายโทเค็น
3) การกำหนดมาร์กกิ้งตั้งต้น
4) การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ
5) การรวมรหัสโพรเมลาแบบ A1



**1) การกำหนดตัวแปรของลูกศรและตัวแปรแทรนซิชัน**

รหัสโพรเมลาแบบ A1 ส่วนที่ 1 เป็นการกำหนดตัวแปรของลูกศรและตัวแปรแทรนซิชัน โดยนับจำนวนลูกศรที่อยู่ในบรรทัดตั้งแต่ .graph ถึง .marking ในเน็ตลิสต์เพื่อนำมาสร้างเป็นตัวแปรลูกศรและตัวแปรแทรนซิชัน ซึ่งจำนวนตัวแปรลูกศรและตัวแปรแทรนซิชันจะเท่ากับจำนวนลูกศรบนซิกแนลแทรนซิชันกราฟ สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-8

```
Algorithm Promela A1 Part1
Variables : Arcname, Transitionname, TotalArc, Totalline ,line
READ line between .graph to .marking
Totalline = line.amount
For i=1 to Totalline
        Arcname = Arcname + Arc(line i)
Next
TotalArc = Arcname.amount
For i=1 to TotalArc
        Transitionname = Transitionname + “t+i”
Next
End Promela A1 Part 1
```

รูปที่ 4-8 รหัสเทียมของโพรเมลาแบบ A1 ส่วนที่ 1

จากรูปที่ 4-9(ก) พบว่าเน็ตลิสต์มีจำนวนลูกศร 8 เส้น ที่อยู่ในบรรทัดระหว่าง .graph ถึง .marking จึงสามารถกำหนดตัวแปรในรหัสโพรเมลาแบบ A1 ส่วนที่ 1 คือ ตัวแปรลูกศร 8 ตัวแปร ได้แก่ apcp, bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp และตัวแปรแทรนซิชัน 8 ตัวแปร ได้แก่ t1, t2, t3, t4, t5, t6, t7, t8 ดังแสดงในรูปที่ 4-9(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
byte apcp, bpcp, cpam, cpbm, amcm, bmcm,
cmap, cmbp;
int t1, t2, t3, t4, t5, t6, t7, t8;
```

(ข)

รูปที่ 4-9 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A1 ส่วนที่ 1

**2) กฎการเคลื่อนย้ายโทเค็น**

รหัสโพรเมลาแบบ A1 ส่วนที่ 2 เป็นการสร้างกฎการเคลื่อนย้ายโทเค็นขาเข้าและกฎการเคลื่อนย้ายโทเค็นขาออก โดยจำนวนกฎการเคลื่อนย้ายโทเค็นขาเข้าและขาออกจะเท่ากับจำนวนลูกศรขาเข้าที่มากที่สุดและลูกศรขาออกที่มากที่สุดตามลำดับ สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-10

```
Algorithm Promela A1 Part2
Variables : InputTransitionRule, OutputTransitionRule, MaxinputArc, MaxoutputArc,
InputTransitionrule i Arc, OutputTransitionrule i Arc
READ MaxinputArc
READ MaxoutputArc
For i=1 to MaxinputArc
        InputTransitionrule i Arc = delete 1 token from all input arc and Increase
transition
        InputTransitionRule = InputTransitionRule + InputTransitionrule i Arc
Next
For i=1 to MaxoutputArc
        OutputTransitionrule i Arc = increase 1 token for all output arc
        OutputTransitionRule = OutputTransitionRule + OutputTransitionrule i Arc
Next
End Promela A1 Part 2
```

รูปที่ 4-10 รหัสเทียมของโพรเมลาแบบ A1 ส่วนที่ 2

จากรูปที่ 4-11(ก) พบว่าเน็ตลิสต์มีลูกศรขาออกมากที่สุด 2 ลูกศร และลูกศรขาเข้ามากที่สุด 2 ลูกศร ดังนั้นสามารถสร้างกฎการเคลื่อนย้ายโทเค็นขาเข้า 2 กฎ คือ inp1 และ inp2 โดยแต่ละกฎจะลบโทเค็นออก 1 โทเค็นจากทุกลูกศรขาเข้าและเพิ่มค่าขึ้น 1 สำหรับแทรนซิชัน และสร้างกฎการเคลื่อนย้ายโทเค็นขาออก 2 กฎ คือ outp1 และ outp2 โดยแต่ละกฎจะเพิ่มโทเค็นในทุกลูกศรขาออก 1 โทเค็น ดังรูปที่ 4-11(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
#define inp1(x,t) (x>0) -> x=x-1; t=t+1
#define inp2(x1, x2, ta, tb) (x1>0 && x2>0) -
>x1=x-1;x2=x2-1;ta=ta+1;tb=tb+1

#define outp1(x) x = x+1
#define outp2(x1,x2) x1=x1+1;x2=x2+1
```

(ข)

รูปที่ 4-11 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A1 ส่วนที่ 2

**3) การกำหนดมาร์กกิ้งตั้งต้น**

รหัสโพรเมลาแบบ A1 ส่วนที่ 3 เป็นการกำหนดมาร์กกิ้งตั้งต้นให้ซิกแนลแทรนซิชันกราฟ โดยให้ลูกศรที่มีมาร์กกิ้งตั้งต้นมีค่าเป็น 1 ซึ่งสามารถดูได้จากบรรทัด .marking ในเน็ตลิสต์ สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-12

```
Algorithm Promela A1 Part3
Variables : InitialMarking, TotalInitialArc, Arc, line
READ line .marking
InitialMarking = “init{”
TotalInitialArc = amount of arc in .marking line
For i=1 to TotalInitialArc
        Arc = “Arcname in line number i = 1”
        InitialMarking = InitialMarking + Arc
Next
End Promela A1 Part 3
```

รูปที่ 4-12 รหัสเทียมของโพรเมลาแบบ A1 ส่วนที่ 3

จากรูปที่ 4-13(ก) พบว่าเน็ตลิสต์มีมาร์กกิ้งตั้งต้น 2 โทเค็น ในการแปลงเป็นภาษาโพรเมลานี้ต้องประกาศฟังก์ชัน init และกำหนดค่าตัวแปรลูกศรมีมาร์กกิ้งตั้งต้นเท่ากับ 1 ได้ดังรูปที่ 4-13(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
init{
cmap = 1; cmbp = 1;
```

(ข)

รูปที่ 4-13 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A1 ส่วนที่ 3

**4) การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ**

รหัสโพรเมลาแบบ A1 ส่วนที่ 4 เป็นการกำหนดโครงสร้างให้ซิกแนลแทรนซิชันกราฟ โดยนำข้อมูลส่วนอธิบายโครงสร้างมาระบุลูกศรขาเข้าและลูกศรขาออก ซึ่งสามารถดูได้จากจำนวนบรรทัดระหว่าง .graph ถึง .marking ในเน็ตลิสต์ แล้วนำลูกศรขาเข้าและขาออกของบรรทัดนั้นมาสร้างโครงสร้างของซิกแนลแทรนซิชันกราฟ สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-14

```
Algorithm Promela A1 Part4
Variables : Structure, line, InputArc, OutputArc
READ line between .graph to .marking
Structure = “do\n”
Totalline = amount of line
For i=1 to Totalline
        READ InputArc for line i
        READ OutputArc for line i
        Structure = Structure + “atomic{”
        InputTransitionRule(InputArc,Transitionname)
        ⟶ OutputTransitionRule(OutputArc)}
Next
Structure = Structure + “od}”
End Promela A1 Part 4
```

รูปที่ 4-14 รหัสเทียมของโพรเมลาแบบ A1 ส่วนที่ 4

จากรูปที่ 4-15(ก) พบว่าเน็ตลิสต์มีโครงสร้าง 6 บรรทัด ซึ่งหมายถึง 6 จุดบนซิกแนลแทรนซิชันกราฟนำมาหาลูกศรขาเข้าและขาออกของทุกจุดและกำหนดโครงสร้าง และแปลงเป็นรหัสโพรเมลาแบบ A1 ส่วนที่ 4 ได้ดังรูปที่ 4-15(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
do
    ::atomic{ inp1(cmbp,t1) -> outp1(bpcp)}
    ::atomic{ inp2(bpcp,apcp,t2,t3) -> outp2(cpam,cpbm)}
    ::atomic{ inp1(cpam,t4) -> outp1(amcm)}
    ::atomic{ inp1(cpbm,t5) -> outp1(bmcm)}
    ::atomic{ inp2(amcm,bmcm,t6,t7) ->
    outp2(cmap,cmbp)}
    ::atomic{ inp1(cmap,t8) -> outp1(apcp)}
od}
```

(ข)

รูปที่ 4-15 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A1 ส่วนที่ 4

**5) การรวมรหัสโพรเมลาแบบ A1**

แบบจำลองรหัสโพรเมลาแบบ A1 เป็นการรวมรหัสโพรเมลาแบบ A1 ทั้ง 4 ส่วนที่ได้จากการแปลงเน็ตลิสต์เป็นรหัสโพรเมลาแบบ A1 ได้แก่ ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรแทรนซิชัน, ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น, ส่วนที่ 3 การกำหนดมาร์กกิ้งตั้งต้น และส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ ตามลำดับ สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-16



```
Algorithm Promela A1 All
Variables : All, Structure, InitialMarking, InputTransitionRule, OutputTransitionRule, Arcname,
Transitionname
Structure = Part4.Structure
InitialMarking = Part3.InitialMarking
InputTransitionRule = Part2.InputTransitionRule
OutputTransitionRule = Part2.OutputTransitionRule
Arcname = Part1.Arcname
Transitionname = Part1.Transitionname
All = Arcname + Transitionname + InputTransitionRule + OutputTransitionRule + InitialMarking +
Structure
End Promela A1 All
```

รูปที่ 4-16 รหัสเทียมของโพรเมลาแบบ A1 ทั้ง 4 ส่วน

จากการแปลงเน็ตลิสต์ในรูปที่ 4-17(ก) เป็นรหัสโพรเมลาแบบ A1 ทั้ง 4 ส่วน สามารถรวมทุกส่วนเป็นแบบจำลองรหัสโพรเมลาแบบ A1 ได้ดังรูปที่ 4-17(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
/*ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรแทรนซิชัน*/
byte bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp,
apcp;
byte t1,t2,t3,t4,t5,t6,t7,t8;

/*ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น*/
#define inp1(x1,t1)          (x1>0) -> x1 = x1-1; t1 = t1+1
#define inp2(x1,x2,t1,t2)  (x1>0 && x2>0) -> x1 = x1-1; x2
= x2-1; t1 = t1+1; t2 = t2+1
#define outp1(x1)            x1 = x1+1
#define outp2(x1,x2)         x1 = x1+1; x2 = x2+1

/*ส่วนที่ 3 การกำหนดมาร์กกิ้งตั้งต้น*/
init {
cmap=1;cmbp=1;

/*ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ*/
do
        ::atomic{ inp1(cmbp,t1) -> outp1(bpcp)}
        ::atomic{ inp2(bpcp,apcp,t2,t3) ->
outp2(cpam,cpbm)}
        ::atomic{ inp1(cpam,t4) -> outp1(amcm)}
        ::atomic{ inp1(cpbm,t5) -> outp1(bmcm)}
        ::atomic{ inp2(amcm,bmcm,t6,t7) ->
outp2(cmap,cmbp)}
        ::atomic{ inp1(cmap,t8) -> outp1(apcp)}
od}
```

(ข)

รูปที่ 4-17 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A1 ทั้งหมด

### 4.2.2 การแปลงเน็ตลิสต์เป็นรหัสโพรเมลาแบบ A2

การแปลงเน็ตลิสต์เป็นรหัสโพรเมลาแบบ A2 นั้นแปลงตามวิธีการแปลงซิกแนลแทรนซิชันกราฟเป็นรหัสโพรเมลาแบบ A2 ในหัวข้อ 3.1.2 ซึ่งการแปลงเน็ตลิสต์เป็นรหัสโพรเมลาแบบ A2 ประกอบด้วยรหัสเทียม 5 รหัสเทียม ดังนี้

1) การกำหนดตัวแปรของลูกศรและตัวแปรสัญญาณ

2) กฎการเคลื่อนย้ายโทเค็น

3) การกำหนดชุดค่าตั้งต้น

4) การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ และ การกำหนดลำดับการทำงานของการตรวจสอบ

5) การรวมรหัสโพรเมลาแบบ A2

**1) การกำหนดตัวแปรลูกศรและตัวแปรสัญญาณ**

รหัสโพรเมลาแบบ A2 ส่วนที่ 1 เป็นการกำหนดตัวแปรของลูกศรและตัวแปรสัญญาณ โดยนับจำนวนลูกศรที่อยู่ในบรรทัดตั้งแต่ .graph ถึง .marking ในเน็ตลิสต์ และนับจำนวนสัญญาณที่อยู่ในบรรทัด .inputs และ .outputs ซึ่งจำนวนตัวแปรลูกศรเท่ากับจำนวนลูกศรบนซิกแนลแทรนซิชันกราฟ และจำนวนตัวแปรสัญญาณเท่ากับสองเท่าของผลรวมของจำนวนสัญญาณขาเข้าและสัญญาณขาออกบนซิกแนลแทรนซิชันกราฟ สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-18

```
Algorithm Promela A2 Part1
Variables : Arcname, Signalname, TotalArc, Totalline ,line, Signal
READ line between .graph to .marking
Totalline = line.amount
For i=1 to Totalline
        Arcname = Arcname + Arc(line i)
Next
TotalArc = Arcname.amount
Read line .inputs and .outputs
For i=1 to 2
        Signal = Signal in line(i) + “p” and  Signal in line(i) + “m”
        Signalname = Signalname + Signal
Next
End Promela A2 Part 1
```

รูปที่ 4-18 รหัสเทียมของโพรเมลาแบบ A2 ส่วนที่ 1

จากรูปที่ 4-19(ก) พบว่าเน็ตลิสต์มีจำนวนลูกศร 8 เส้น และมีสัญญาณ inputs และ outputs 3 สัญญาณ คือ a, b และ c จึงสามารถกำหนดตัวแปรในรหัสโพรเมลาแบบ A2 ส่วนที่ 1 คือ ตัวแปรลูกศร 8 ตัวแปร ได้แก่ apcp, bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp และตัวแปรสัญญาณ 6 ตัวแปร ได้แก่ ap, am, bp, bm, cp, cm ดังแสดงในรูปที่ 4-19(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
byte apcp, bpcp, cpam, cpbm, amcm, bmcm,
cmap, cmbp;
byte ap, am, bp, bm, cp, cm;
```

(ข)

รูปที่ 4-19 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A2 ส่วนที่ 1

**2) กฎการเคลื่อนย้ายโทเค็น**

รหัสโพรเมลาแบบ A2 ส่วนที่ 2 เป็นการสร้างกฎการเคลื่อนย้ายโทเค็นขาเข้าและกฎการเคลื่อนย้ายโทเค็นขาออก โดยจำนวนกฎการเคลื่อนย้ายโทเค็นขาเข้าจะมีเพียงกฎเดียว และจำนวนกฎการเคลื่อนย้ายโทเค็นขาออกจะเท่ากับจำนวนลูกศรลูกศรขาออกที่มากที่สุดตามลำดับ สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-20



```
Algorithm Promela A2 Part2
Variables : InputTransitionRule, OutputTransitionRule, MaxoutputArc,
OutputTransitionrule i Arc
READ MaxoutputArc
InputTransitionRule = InputTransitionRule condition and result
For i=1 to MaxoutputArc
        OutputTransitionrule i Arc = if condition and increase 1 to arcname
        OutputTransitionRule = OutputTransitionRule + OutputTransitionrule i Arc
Next
End Promela A2 Part 2
```

รูปที่ 4-20 รหัสเทียมของโพรเมลาแบบ A2 ส่วนที่ 2

จากรูปที่ 4-21(ก) พบว่าเน็ตลิสต์มีลูกศรขาออกมากที่สุดที่ 2 ลูกศร ดังนั้นสามารถสร้างกฎการเคลื่อนย้ายโทเค็นขาเข้า 1 กฎ คือ inp โดยจะเลือกทำกรณีที่เป็นจริง 1 กรณีจากกรณีที่ตัวแปรลูกศรมีค่าเท่ากับ 1 ให้เพิ่มค่าตัวแปรลูกศรเป็น 10 และเพิ่มค่าตัวแปรสัญญาณ และกรณีที่ตัวแปรลูกศรมีค่าเท่ากับ 2 ให้เพิ่มค่าตัวแปรลูกศรและเพิ่มค่าตัวแปรสัญญาณ และสร้างกฎการเคลื่อนย้ายโทเค็นขาออก 2 กฎ คือ outp1 และ outp2 โดยแต่ละกฎจะเลือกทำกรณีที่เป็นจริง 1 กรณีจากกรณีที่ตัวแปรลูกศรตัวใดตัวหนึ่งมีค่าไม่เท่ากับ 10 ให้เพิ่มโทเค็นในตัวแปรลูกศรทุกลูกศรขาออก 1 โทเค็น ดังแสดงในรูปที่ 4-21(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
#define inp(x,y) if ::(x==1) -> x=10; y++;
::(x==2) -> x++;y++; fi

#define outp1(x)   if ::(x!=10) -> x++;
:: else -> break; fi
#define outp2(x1, x2)   if :: (x1!=10) -> x++;
:: (x2!=10) -> x2++; :: else -> break; fi
```

(ข)

รูปที่ 4-21 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A2 ส่วนที่ 2

**3) การกำหนดชุดค่าตั้งต้น**

รหัสโพรเมลาแบบ A2 ส่วนที่ 3 เป็นการกำหนดมาร์กกิ้งตั้งต้นให้ซิกแนลแทรนซิชันกราฟโดยจะกำหนดเป็นชุดข้อมูลให้สามารถเข้าถึงได้โดยค่าที่ป้อนเข้ามาผ่านตัวแปร k ซึ่งจำนวนชุดค่าตั้งต้นและมาร์กกิ้งตั้งต้นที่สนใจสามารถดูได้จากบรรทัด .marking ในเน็ตลิสต์ และกำหนดให้มาร์กกิ้งตั้งต้นที่สนใจมีค่าเป็น 2 มาร์กกิ้งตั้งต้นอื่นๆ มีค่าเป็น 10 และตัวแปรลูกศรและตัวแปรสัญญาณอื่นๆ ที่ไม่มีมาร์กกิ้งตั้งต้นมีค่าเป็น 0 สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-22

Algorithm Promela A2 Part3

Variables : InitialMarking, TotalInitialArc, Signalname, Arcname, line, Marking

READ line .marking

Signalname = Part1.Signalname

Arcname = Part1.Arcname

InitialMarking = “Proctype P3(byte k){\n atomic{\nif\n”

TotalInitialArc = amount of arc in .marking line

For i=1 to TotalInitialArc

Marking = “:: k ==”+ i + -> marking(i) = 2; other marking = 10 and other

Arcname,signalname = 0 + “\n”

InitialMarking = InitialMarking + marking

Next

InitialMarking = InitialMarking + “fi”

End Promela A2 Part 3

รูปที่ 4-22 รหัสเทียมของโพรเมลาแบบ A2 ส่วนที่ 3

จากรูปที่ 4-23(ก) พบว่าเน็ตลิสต์มีมาร์กกิ้งตั้งต้น 2 โทเค็น จึงกำหนดชุดค่าตั้งต้นได้ 2 ชุด ดังรูปที่ 4-23(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
proctype P3 (byte k){
 atomic{
 if
   :: k==1 -> cmap=2; cmbp=10; bpcp=0; cpam=0; cpbm=0;
amcm=0; bmcm=0; apcp=0; ap=0; am=0; bp=0; bm=0; cp=0;
cm=0;
   :: k==2 -> cmbp=2; cmap=10; bpcp=0; cpam=0; cpbm=0;
amcm=0; bmcm=0; apcp=0; ap=0; am=0; bp=0; bm=0; cp=0;
cm=0;
  fi
```

(ข)

รูปที่ 4-23 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A2 ส่วนที่ 3

**4) การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟและการกำหนดลำดับการทำงานของการตรวจสอบ**

รหัสโพรเมลาแบบ A2 ส่วนที่ 4 เป็นการกำหนดโครงสร้างให้ซิกแนลแทรนซิชันกราฟ โดยนำข้อมูลส่วนอธิบายโครงสร้างมาระบุลูกศรขาเข้าและลูกศรขาออก ซึ่งสามารถดูได้จากจำนวนบรรทัดระหว่าง .graph ถึง .marking ในเน็ตลิสต์ แล้วนำลูกศรขาเข้าและขาออกของสัญญาณในบรรทัดนั้นมาสร้างโครงสร้างของซิกแนลแทรนซิชันกราฟ และรหัสโพรเมลาแบบ A2 ส่วนที่ 5 เป็นการกำหนดลำดับการทำงานของการตรวจสอบกระทำ โดยสั่งให้ฟังก์ชันที่สร้างขึ้นในการกำหนดชุดค่าตั้งต้นทำงานให้ครบตามจำนวนชุดค่าตั้งต้น สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-24

```
Algorithm Promela A2 Part4,5
Variables : Structure, line, InputArc, OutputArc, Runner, Markingnumber
READ line between .graph to .marking
Structure = “do\n”
Totalline = amount of line
For i=1 to Totalline
        READ InputArc for signal in line i
        READ OutputArc for line i
        Structure = Structure + “atomic{”
        InputTransitionRule(InputArc,Signalname);printf(signalname) ⟶
        OutputTransitionRule(OutputArc)}
Next
Structure = Structure + “od}”
Markingnumber = Part3.TotalInitialArc
Runner = “init{”
For i=1 to Markingumber
        Runner = Runner + “Run P3(i);”
Next
End Promela A2 Part 4,5
```

รูปที่ 4-24 รหัสเทียมของโพรเมลาแบบ A2 ส่วนที่ 4 และ 5

จากรูปที่ 4-25(ก) พบว่าเน็ตลิสต์มีโครงสร้าง 6 บรรทัด ซึ่งหมายถึง 6 จุดบนซิกแนลแทรนซิชันกราฟ และนำมาหาลูกศรขาเข้าและขาออกของทุกจุดและกำหนดโครงสร้างซึ่งถ้าบรรทัดใดที่มีลูกศรขาเข้ามากกว่า 1 ลูกศรจะถูกแยกโครงสร้างเป็น 2 ส่วน และนำมาแปลงเป็นรหัสโพรเมลาแบบ A2 ส่วนที่ 4 และพบว่ามีมาร์กกิ้งตั้งต้น 2 ชุด จึงแปลงเป็นรหัสโพรเมลาแบบ A2 ส่วนที่ 5 ดังแสดงในรูปที่ 4-25(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 4*/
do
        ::atomic{ inp(cmap,cm);
printf("printf(cm=%d)",cm) -> outp1(apcp)}
        ::atomic{ inp(bpcp,bp);printf("printf(bp=%d)",bp)
-> outp2(cpam,cpbm)}
        ::atomic{ inp(apcp,ap);printf("printf(ap=%d)",ap) -
> outp2(cpam,cpbm)}
        ::atomic{ inp(cpam,cp);printf("printf(cp=%d)",cp)
-> outp1(amcm)}
        ::atomic{ inp(cpbm,cp);printf("printf(cp=%d)",cp)
-> outp1(bmcm)}
        ::atomic{ inp(amcm,am);
printf("printf(am=%d)",am) -> outp2(cmap,cmbp)}
        ::atomic{ inp(bmcm,bm);
printf("printf(bm=%d)",bm) -> outp2(cmap,cmbp)}
        ::atomic{ inp(cmbp,cm);
printf("printf(cm=%d)",cm) -> outp1(bpcp)}
od}}

/*รหัสโพรเมลาแบบ A2 ส่วนที่ 5*/
init{run P3(1); run P3(2);}
```

(ข)

รูปที่ 4-25 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A2 ส่วนที่ 4 และ 5

**5) การรวมรหัสโพรเมลาแบบ A2**

แบบจำลองรหัสโพรเมลาแบบ A2 เป็นการรวมรหัสโพรเมลาแบบ A2 ทั้ง 5 ส่วนที่ได้จากการแปลงเน็ตลิสต์เป็นรหัสโพรเมลาแบบ A2 ได้แก่ ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรสัญญาณ, ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น, ส่วนที่ 3 การกำหนดชุดค่าตั้งต้น, ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ และส่วนที่ 5 การกำหนดลำดับการทำงานของการตรวจสอบตามลำดับ สามารถแสดงรหัสเทียมได้ดังรูปที่ 4-26

```
Algorithm Promela A2 All
Variables : All, Runner, Structure, InitialMarking, InputTransitionRule,
OutputTransitionRule, Arcname, Signalname
Runner = Part4,5.Runner
Structure = Part4,5.Structure
InitialMarking = Part3.InitialMarking
InputTransitionRule = Part2.InputTransitionRule
OutputTransitionRule = Part2.OutputTransitionRule
Arcname = Part1.Arcname
Signalname = Part1.Transitionname
All = Arcname + Signalname + InputTransitionRule + OutputTransitionRule +
InitialMarking + Structure + Runner
End Promela A2 All
```

รูปที่ 4-26 รหัสเทียมของโพรเมลาแบบ A2 ทั้ง 5 ส่วน

จากการแปลงเน็ตลิสต์ในรูปที่ 4-27(ก) เป็นรหัสโพรเมลาแบบ A2 ทั้ง 5 ส่วน สามารถรวมทุกส่วนเป็นแบบจำลองรหัสโพรเมลาแบบ A2 ได้ดังรูปที่ 4-27(ข)

```
.model sampleSTG
.inputs a,b
.outputs c
.graph
a+ c+
b+ c+
c+ a- b-
a- c-
b- c-
c- a+ b+
.marking {<c-,a+>,<c-,b+>}
.end
```

(ก)

```
/*ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรสัญญาณ*/
byte bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp, apcp;
byte ap, am, bp, bm, cp, cm;
/*ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น*/
#define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++;y++; fi
#define outp1(x1)   if :: (x1 != 10) -> x1++; :: else -> break; fi
#define outp2(x1,x2)  if :: (x1 != 10) -> x1++; :: ( x2 != 10) -> x2++;    ::
else -> break; fi
/*ส่วนที่ 3 การกำหนดชุดค่าตั้งต้น*/
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> cmap=2; cmbp=10; bpcp=0; cpam=0; cpbm=0; amcm=0;
bmcm=0; apcp=0; ap=0; am=0; bp=0; bm=0; cp=0; cm=0;
   :: k==2 -> cmbp=2; cmap=10; bpcp=0; cpam=0; cpbm=0; amcm=0;
bmcm=0; apcp=0; ap=0; am=0; bp=0; bm=0; cp=0; cm=0;
  fi
/*ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ*/
 do
          ::atomic{ inp(cmbp,cm);printf("printf(cm=%d)",cm) ->
outp1(bpcp)}
          ::atomic{ inp(bpcp,bp);printf("printf(bp=%d)",bp) ->
outp2(cpam,cpbm)}
          ::atomic{ inp(apcp,ap);printf("printf(ap=%d)",ap) ->
outp2(cpam,cpbm)}
          ::atomic{ inp(cpam,cp);printf("printf(cp=%d)",cp) ->
outp1(amcm)}
          ::atomic{ inp(cpbm,cp);printf("printf(cp=%d)",cp) ->
outp1(bmcm)}
          ::atomic{ inp(amcm,am);printf("printf(am=%d)",am) ->
outp2(cmap,cmbp)}
          ::atomic{ inp(bmcm,bm);printf("printf(bm=%d)",bm) ->
outp2(cmap,cmbp)}
          ::atomic{ inp(cmap,cm);printf("printf(cm=%d)",cm) ->
outp1(apcp)}
  od}}
/*ส่วนที่ 5 การกำหนดลำดับการทำงานของการตรวจสอบ*/
init{run P3(1);  run P3(2);}
```

(ข)

รูปที่ 4-27 (ก) ตัวอย่างเน็ตลิสต์ (ข) รหัสโพรเมลาแบบ A2 ทั้งหมด

## 4.3 การพัฒนาเครื่องมือสนับสนุน

### 4.3.1 สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือ

สภาพแวดล้อมที่ใช้ในการพัฒนาเครื่องมือมีรายละเอียดดังนี้

1) ฮาร์ดแวร์ (Hardware)

- เครื่องคอมพิวเตอร์แบบพกพา หน่วยประมวลผลอินเทลคอร์ไอไฟว์ 2.5 กิกะเฮิรตซ์
- หน่วยความจำของคอมพิวเตอร์หรือแรม (Ram) 4.0 กิกะไบต์ (4 GB)
- ฮาร์ดดิสก์ (Hard disk) 320 กิกะไบต์ (320 GB)

2) ซอฟต์แวร์ (Software)

- ระบบปฏิบัติการ (Operating System) ไมโครซอฟท์วินโดวส์ 7 (64 บิต)
- อีคลิปส์นีออน (Eclipse Neon) เป็นโปรแกรมที่ใช้สำหรับพัฒนาเครื่องมือโดยภาษาจาวา

### 4.3.2 ข้อกำหนดการใช้งานและความสามารถของเครื่องมือ

1) ความสามารถของเครื่องมือ

1.1) สร้างรหัสโพรเมลาตามที่นำเสนอแบบ A1 และ A2

1.2) สร้างตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์

1.3) ไม่สามารถแสดงผลคำตอบว่าพบคุณสมบัติในซิกแนลแทรนซิชันกราฟนั้นๆ หรือไม่ยกเว้นคุณสมบัติการกำหนดสถานะที่สมบูรณ์

1.4) สิ่งที่ได้จากเครื่องมือจะต้องนำไปทวนสอบในเครื่องมือสปินเพื่อทวนสอบคุณสมบัติทั้ง 5 คุณสมบัติ

2) ข้อกำหนดการใช้งานเครื่องมือ

2.1) ข้อมูลนำเข้า

- ไฟล์ .txt ซึ่งมีโครงสร้างมาตรฐานแบบเน็ตลิสต์
- สามารถเป็นได้ทั้งซิกแนลแทรนซิชันกราฟแบบวัฏจักรเชิงเดี่ยว และวัฏจักรหลากหลาย
- สำหรับคุณสมบัติการกำหนดสถานะที่สมบูรณ์เมื่อได้ผลการจำลองจากการทวนสอบฟูลล็อคแล้วให้นำผลมาใส่ในเครื่องมือ

- สำหรับคุณสมบัติการกำหนดสถานะที่สมบูรณ์เมื่อได้ผลการจำลองจากการทวนสอบแทรนซิทีฟล็อคแล้วให้นำผลมาใส่ในเครื่องมือ

2.2) ข้อมูลส่งออกที่ได้จากเครื่องมือ

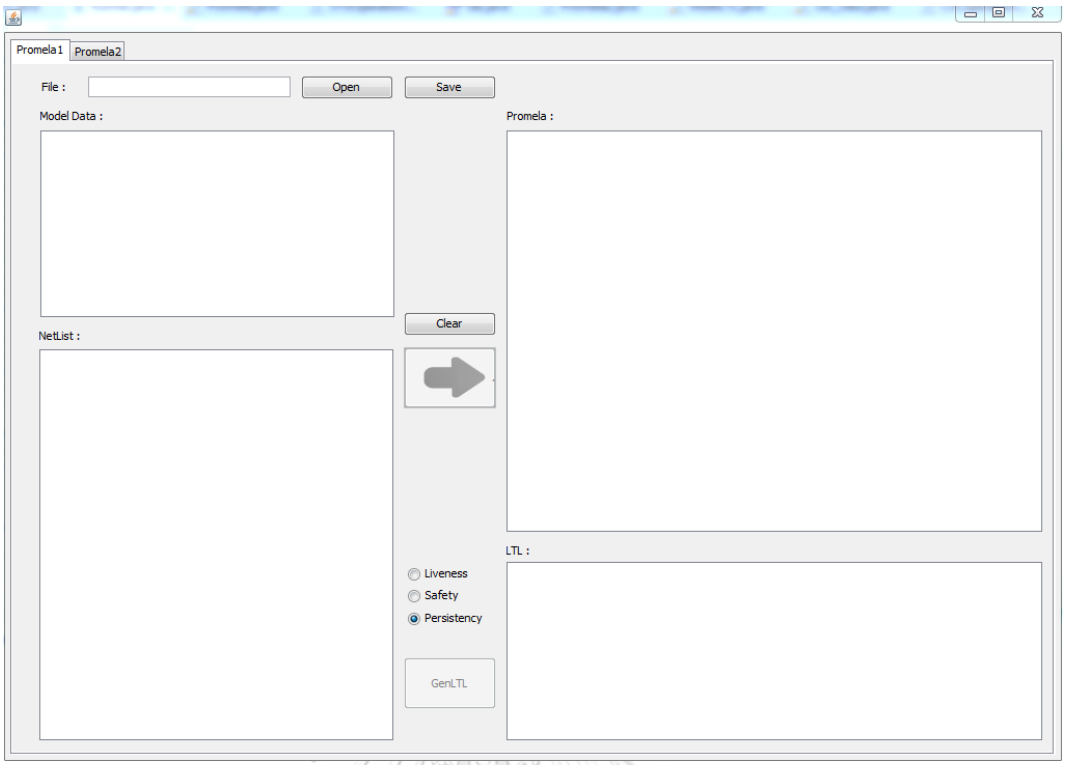
- รหัสโพรเมลาแบบ A1 หากเลือกคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน
- รหัสโพรเมลาแบบ A2 หากเลือกคุณสมบัติความต้องกันและคุณสมบัติการกำหนดสถานะที่สมบูรณ์
- ตรรกะเวลาเชิงเส้นของคุณสมบัติที่เลือก
- ถ้าเป็นคุณสมบัติการกำหนดสถานะที่สมบูรณ์จะได้ตรรกะเวลาเชิงเส้น 2 ครั้ง คือ ตรรกะเวลาเชิงเส้นสำหรับการทวนสอบฟูลล็อค และตรรกะเวลาเชิงเส้นสำหรับการทวนสอบแทรนซิทีฟล็อค
- สำหรับคุณสมบัติการกำหนดสถานะที่สมบูรณ์เครื่องมือจะบอกว่าคู่ใดเป็นฟูลล็อคบ้าง
- สำหรับคุณสมบัติการกำหนดสถานะที่สมบูรณ์เครื่องมือจะบอกว่าคู่ใดพบแทรนซิทีฟล็อคหรือไม่
- สำหรับคุณสมบัติการกำหนดสถานะที่สมบูรณ์เมื่อทำตามขั้นตอนครบแล้วเครื่องมือจะบอกว่าพบคุณสมบัตินี้หรือไม่

### 4.3.3 โครงสร้างส่วนต่อประสานกับผู้ใช้ของเครื่องมือที่พัฒนาขึ้น

เครื่องมือที่พัฒนาขึ้นประกอบด้วยส่วนต่อประสานกับผู้ใช้ 2 หน้าต่าง คือ หน้าต่างหลักและหน้าต่างรอง โดยมีรายละเอียดดังนี้

1) หน้าต่างหลักของเครื่องมือ

เป็นหน้าต่างเริ่มต้นของเครื่องมือซึ่งจะประกอบด้วย ส่วนแสดงเส้นทางของไฟล์ ส่วนแสดงข้อมูลของซิกแนลแทรนซิชันกราฟ ส่วนแสดงข้อมูลที่อยู่ในไฟล์ในรูปแบบเน็ตลิสต์ ส่วนแสดงรหัสโพรเมลาแบบ A1 และส่วนแสดงตรรกะเวลาเชิงเส้น ดังแสดงในรูปที่ 4-28



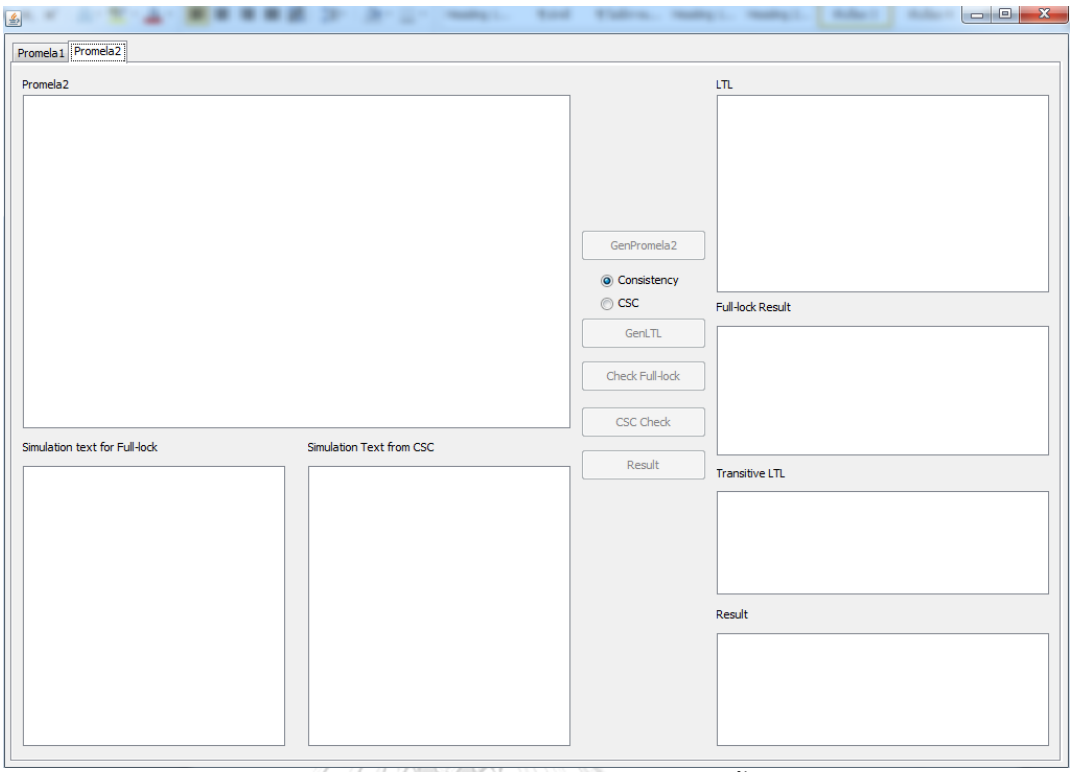

รูปที่ 4-28 หน้าต่างหลักของเครื่องมือที่พัฒนาขึ้น

ในหน้าต่างหลักของเครื่องมือจะประกอบด้วยปุ่มต่างๆ ดังนี้

- ปุ่ม “Open” เพื่อเลือกไฟล์และนำเข้าสู่การประมวลผลของเครื่องมือ
- ปุ่ม “⟶” เครื่องมือแสดงรหัสโพรเมลาในส่วนแสดงรหัสโพรเมลา
- ปุ่ม “GenLTL” เครื่องมือแสดงตรรกะเวลาเชิงเส้นตามที่เลือกโดยเลือกได้ครั้งละ 1 คุณสมบัติ โดยจะแสดงในส่วนแสดงตรรกะเวลาเชิงเส้น
- ปุ่ม “Clear” เครื่องมือลบข้อมูลทั้งหมดที่แสดงในหน้าต่างหลักและหน้าต่างรอง

2) หน้าต่างรองของเครื่องมือ

เป็นหน้าต่างซึ่งประกอบด้วย ส่วนแสดงรหัสโพรเมลาแบบ A2 ส่วนแสดงตรรกะเวลาเชิงเส้น ส่วนแสดงผลของฟูลล็อค ส่วนแสดงตรรกะเวลาเชิงเส้นของแทรนซิทีฟล็อค ส่วนแสดงผลของคุณสมบัติการกำหนดสถานะที่สมบูรณ์ ส่วนใส่ค่าการจำลองฟูลล็อค และส่วนใส่ค่าการจำลอง CSC ดังแสดงในรูปที่ 4-29




รูปที่ 4-29 หน้าต่างรองของเครื่องมือที่พัฒนาขึ้น

ในหน้าต่างรองของเครื่องมือจะประกอบด้วยปุ่มต่างๆ ดังนี้

- ปุ่ม “GenPromela2” เครื่องมือแสดงรหัสโพรเมลาแบบ A2 ในส่วนแสดงรหัสโพรเมลาแบบ A2
- ปุ่ม “GenLTL” เครื่องมือแสดงตรรกะเวลาเชิงเส้นของคุณสมบัติที่เลือกในส่วนแสดงตรรกะเวลาเชิงเส้น
- คลิกปุ่ม “Check Full-lock” เครื่องมือแสดงผลการตรวจสอบฟูลล็อคในส่วนแสดงผลของฟูลล็อค
- คลิกปุ่ม “CSC Check” เครื่องมือแสดงผลว่าพบคุณสมบัติการกำหนดสถานะที่สมบูรณ์หรือไม่ ถ้าพบเครื่องมือแสดงตรรกะเชิงเวลาเพื่อหาแทรนซิทีฟล็อคในส่วนแสดงตรรกะเวลาเชิงเส้นของแทรนซิทีฟล็อค
- ปุ่ม “Result” เครื่องมือตรวจสอบผลของคุณสมบัติการกำหนดสถานะที่สมบูรณ์จากข้อมูลในส่วนใส่ค่าการจำลอง CSC

## 4.4 วิธีการทวนสอบ

ในส่วนนี้จะนำรหัสโพรเมลาและตรรกะเวลาเชิงเส้นที่ได้เครื่องมือมาทวนสอบคุณสมบัติของซิกแนลแทรนซิชันกราฟ 5 คุณสมบัติ คือ คุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์ โดยในการทวนสอบนั้นจะต้องนำเครื่องมือสปินมาใช้ในการทวนสอบ

### 4.4.1 การรวมรหัสโพรเมลาและตรรกะเวลาเชิงเส้น

ในการทวนสอบคุณสมบัติของซิกแนลแทรนซิชันกราฟด้วยเครื่องมือสปินจะต้องประกอบด้วยแบบจำลองและตรรกะเวลาเชิงเส้น แต่เครื่องมือที่พัฒนาขึ้นจะสร้างแบบจำลองและตรรกะเวลาเชิงเส้นแยกส่วนกัน ดังนั้นจึงต้องรวมแบบจำลองรหัสโพลาและตรรกะเวลาเชิงเส้นเป็นไฟล์เดียว โดยนำตรรกะเวลาเชิงเส้นของคุณสมบัติที่ต้องการทวนสอบไปแทรกหลังรหัสโพรเมลาส่วนกฎการเคลื่อนย้ายโทเค็น ดังแสดงในรูปที่ 4-30

```
/*รหัสโพรเมลา ส่วนที่ 1 */
...
/*รหัสโพรเมลา ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น*/
...

/*เงื่อนไขตรรกะเวลาเชิงเส้น*/
/*ตรรกะเวลาเชิงเส้น*/
...
...

/*รหัสโพรเมลา ส่วนที่ 3 – 5*/
...
```

รูปที่ 4-30 การรวมรหัสโพรเมลาและตรรกะเวลาเชิงเส้น

ในการทวนสอบคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส และคุณสมบัติความทนทานนั้นจะใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ จากรูปที่ 4-30 สามารถแทรกตรรกะเวลาเชิงเส้นในรหัสโพรเมลาแบบ A1 ระหว่างส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น กับส่วนที่ 3 การกำหนดมาร์กกิ้งตั้งต้น ดังรูปที่ 4-31

```
/*รหัสโพรเมลาแบบ A1 ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรแทรนซิชัน*/
byte bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp, apcp;
byte t1,t2,t3,t4,t5,t6,t7,t8;
/*รหัสโพรเมลาแบบ A1 ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น*/
#define inp1(x1,t1)         (x1>0) -> x1 = x1-1; t1 = t1+1
#define inp2(x1,x2,t1,t2) (x1>0 && x2>0) -> x1 = x1-1; x2 = x2-1; t1 = t1+1; t2 = t2+1
#define outp1(x1)           x1 = x1+1
#define outp2(x1,x2)        x1 = x1+1; x2 = x2+1
/*ตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัย*/
#define safe ((bpcp <= 1) && (cpam <= 1) && (cpbm <= 1) && (amcm <= 1) && (bmcm <= 1) && (cmap
<= 1) && (cmbp <= 1) && (apcp <= 1) )
ltl p1 { [] safe }
/*ตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนส*/
#define live ((t1>1) && (t2>1) && (t3>1) && (t4>1) && (t5>1) && (t6>1) && (t7>1) && (t8>1))
ltl p2 { <> live }
/*ตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทาน*/
#define persist ((((cmap == 1) || (cmbp == 1)) -> ((cpam == 0) && (cpbm == 0))) && (((cpam == 1) ||
(cpbm == 1)) -> ((cmap == 0) && (cmbp == 0))))
ltl p3 { [] persist }
/*รหัสโพรเมลาแบบ A1 ส่วนที่ 3 การกำหนดมาร์กกิ้งตั้งต้น*/
init {
cmap=1;cmbp=1;
/*รหัสโพรเมลาแบบ A1 ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ*/
do
    ::atomic{ inp1(cmbp,t1) -> outp1(bpcp)}
    ::atomic{ inp2(bpcp,apcp,t2,t3) -> outp2(cpam,cpbm)}
    ::atomic{ inp1(cpam,t4) -> outp1(amcm)}
    ::atomic{ inp1(cpbm,t5) -> outp1(bmcm)}
    ::atomic{ inp2(amcm,bmcm,t6,t7) -> outp2(cmap,cmbp)}
    ::atomic{ inp1(cmap,t8) -> outp1(apcp)}
od
}
```

รูปที่ 4-31 การรวมรหัสโพรเมลาแบบ A1 และตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส และคุณสมบัติความทนทาน

ในการทวนสอบคุณสมบัติความต้องกันนั้นจะใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ จากรูปที่ 4-30 สามารถแทรกตรรกะเวลาเชิงเส้นในรหัสโพรเมลาแบบ A2 ระหว่างส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น กับส่วนที่ 3 การกำหนดชุดค่าตั้งต้น ดังรูปที่ 4-32

```
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรสัญญาณ*/
byte bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp, apcp;
byte ap, am, bp, bm, cp, cm;
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น*/
#define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++;y++; fi
#define outp1(x1)          if :: (x1 != 10) -> x1++; :: else -> break; fi
#define outp2(x1,x2)       if :: (x1 != 10) -> x1++; :: ( x2 != 10) -> x2++; :: else -> break; fi
/*ตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกัน*/
#define consist1 (ap>=1 && am>=1 && (cmap==4 || cmbp==4))
#define consist2 (bp>=1 && bm>=1 && (cmap==4 || cmbp==4))
#define consist3 (cp>=1 && cm>=1 && (cmap==4 || cmbp==4))
ltl p4{!<>consist1}
ltl p5{!<>consist2}
ltl p6{!<>consist3}
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 3 การกำหนดชุดค่าตั้งต้น*/
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> cmap=2;cmbp=10;bpcp=0;cpam=0;cpbm=0;amcm=0;bmcm=0;apcp=0;ap=0;am=0;bp=0;bm=0;cp=0;cm=0;
   :: k==2 -> cmbp=2;cmap=10;bpcp=0;cpam=0;cpbm=0;amcm=0;bmcm=0;apcp=0;ap=0;am=0;bp=0;bm=0;cp=0;cm=0;
  fi
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ*/
  do
       ::atomic{ inp(cmbp,cm);printf("printf(cm=%d)",cm) -> outp1(bpcp)}
       ::atomic{ inp(bpcp,bp);printf("printf(bp=%d)",bp) -> outp2(cpam,cpbm)}
       ::atomic{ inp(apcp,ap);printf("printf(ap=%d)",ap) -> outp2(cpam,cpbm)}
       ::atomic{ inp(cpam,cp);printf("printf(cp=%d)",cp) -> outp1(amcm)}
       ::atomic{ inp(cpbm,cp);printf("printf(cp=%d)",cp) -> outp1(bmcm)}
       ::atomic{ inp(amcm,am);printf("printf(am=%d)",am) -> outp2(cmap,cmbp)}
       ::atomic{ inp(bmcm,bm);printf("printf(bm=%d)",bm) -> outp2(cmap,cmbp)}
       ::atomic{ inp(cmap,cm);printf("printf(cm=%d)",cm) -> outp1(apcp)}
  od}}
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 5 การการกำหนดลำดับการทำงานของการตรวจสอบ*/
init{run P3(1);  run P3(2);}
```

รูปที่ 4-32 การรวมรหัสโพรเมลาแบบ A2 และตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกัน

ในการทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์นั้นจะใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ จากรูปที่ 4-30 สามารถแทรกตรรกะเวลาเชิงเส้นในรหัสโพรเมลาแบบ A2 ระหว่างส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น กับส่วนที่ 3 การกำหนดชุดค่าตั้งต้น ดังรูปที่ 4-33

```
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 1 การกำหนดตัวแปรของลูกศรและตัวแปรสัญญาณ*/
byte bpcp, cpam, cpbm, amcm, bmcm, cmap, cmbp, apcp;
byte ap, am, bp, bm, cp, cm;
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 2 กฎการเคลื่อนย้ายโทเค็น*/
#define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++;y++; fi
#define outp1(x1)           if :: (x1 != 10) -> x1++; :: else -> break; fi
#define outp2(x1,x2)        if :: (x1 != 10) -> x1++; :: ( x2 != 10) -> x2++; :: else -> break; fi
/*ตรรกะเวลาเชิงเส้นของคุณสมบัติการกำหนดสถานะที่สมบูรณ์*/
#define csc_full_ab (ap==1 && am==1 && bp==1 && bm==1 && (cmap==4 || cmbp==4))
#define csc_full_ac (ap==1 && am==1 && cp==1 && cm==1 && (cmap==4 || cmbp==4))
#define csc_full_bc (bp==1 && bm==1 && cp==1 && cm==1 && (cmap==4 || cmbp==4))
ltl c1 {!<> csc_full_ab}
ltl c2 {!<> csc_full_ac}
ltl c3 {!<> csc_full_bc}
#define transitive1((ap==1 || am==1) && (cp==1 || cm==1) && (cpam==4 || cpbm==4))
#define transitive2((bp==1 || bm==1) && (cp==1 || cm==1) && (cpam==4 || cpbm==4))
ltl p4 {!<>transitive1}
ltl p5 {!<>transitive2}
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 3 การกำหนดชุดค่าตั้งต้น*/
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> cmap=2;cmbp=10;bpcp=0;cpam=0;cpbm=0;amcm=0;bmcm=0;apcp=0;ap=0;am=0;bp=0;bm=0;cp=0;cm=0;
   :: k==2 -> cmbp=2;cmap=10;bpcp=0;cpam=0;cpbm=0;amcm=0;bmcm=0;apcp=0;ap=0;am=0;bp=0;bm=0;cp=0;cm=0;
  fi
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 4 การกำหนดโครงสร้างของซิกแนลแทรนซิชันกราฟ*/
  do
       ::atomic{ inp(cmbp,cm);printf("printf(cm=%d)",cm) -> outp1(bpcp)}
       ::atomic{ inp(bpcp,bp);printf("printf(bp=%d)",bp) -> outp2(cpam,cpbm)}
       ::atomic{ inp(apcp,ap);printf("printf(ap=%d)",ap) -> outp2(cpam,cpbm)}
       ::atomic{ inp(cpam,cp);printf("printf(cp=%d)",cp) -> outp1(amcm)}
       ::atomic{ inp(cpbm,cp);printf("printf(cp=%d)",cp) -> outp1(bmcm)}
       ::atomic{ inp(amcm,am);printf("printf(am=%d)",am) -> outp2(cmap,cmbp)}
       ::atomic{ inp(bmcm,bm);printf("printf(bm=%d)",bm) -> outp2(cmap,cmbp)}
       ::atomic{ inp(cmap,cm);printf("printf(cm=%d)",cm) -> outp1(apcp)}
  od}}
/*รหัสโพรเมลาแบบ A2 ส่วนที่ 5 การการกำหนดลำดับการทำงานของการตรวจสอบ*/
init{run P3(1);  run P3(2);}
```

รูปที่ 4-33 การรวมรหัสโพรเมลาแบบ A2 และตรรกะเวลาเชิงเส้นของคุณสมบัติการกำหนดสถานะที่สมบูรณ์

### 4.4.2 การทวนสอบคุณสมบัติความปลอดภัย

ในการทวนสอบคุณสมบัติความปลอดภัยนั้นจะนำแบบจำลองรหัสโพรเมลาแบบ A1 และตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัยมาทวนสอบในเครื่องมือสปิน ถ้าเครื่องมือสปินทวนสอบพบความผิดพลาด แสดงว่า มีกรณีที่เงื่อนไขความปลอดภัยไม่เป็นจริง จึงสามารถสรุปได้ว่าซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติความปลอดภัย สามารสรุปขั้นตอนการทวนสอบได้ดังนี้

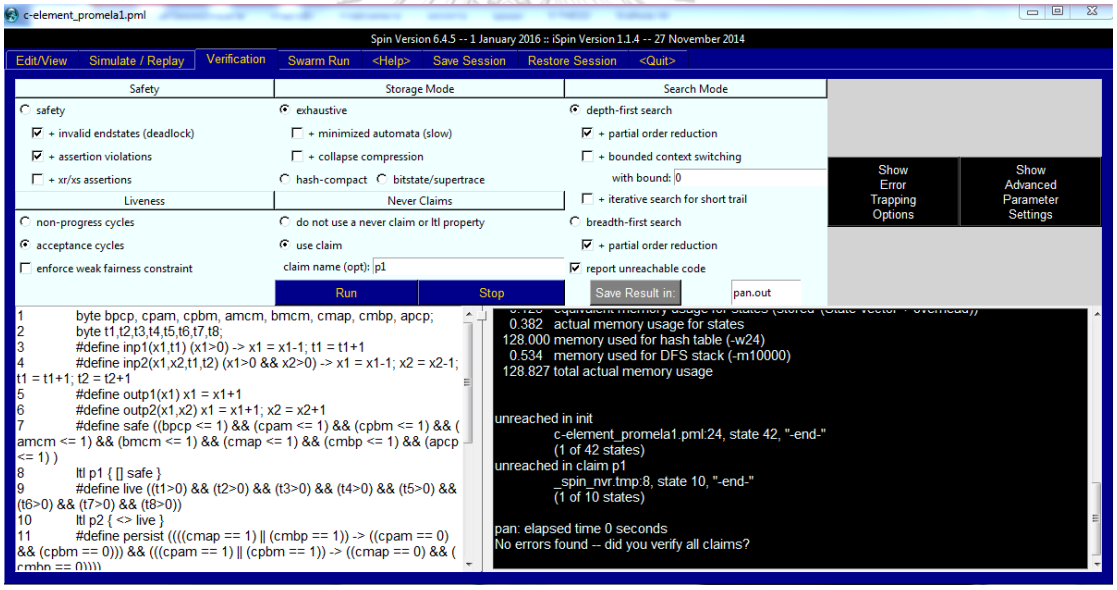
1) รวมรหัสโพรเมลาแบบ A1 และตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัย ในโหมด “Edit/View”

2) เลือกโหมด “Verification”

3) เลือก “use claim” แล้วใส่ชื่อตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัย

4) กดปุ่ม “run”

5) ดูผลในช่องสีดำ ถ้าพบ “error” สรุปว่าไม่มีคุณสมบัติความปลอดภัย แต่ถ้า “No errors” สรุปว่ามีคุณสมบัติความปลอดภัย




รูปที่ 4-34 ผลการทวนสอบคุณสมบัติความปลอดภัยของตัวอย่างซิกแนลแทรนซิชันกราฟ

เมื่อทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัย “p1” จะได้ผล “No errors” ดังรูปที่ 4-34 จึงสรุปได้ว่าตัวอย่างซิกแนลแทรนซิชันกราฟที่นำมาทวนสอบมีคุณสมบัติความปลอดภัย

### 4.4.3 การทวนสอบคุณสมบัติความไลฟ์เนส

ในการทวนสอบคุณสมบัติความไลฟ์เนสนั้นจะนำแบบจำลองรหัสโพรเมลาแบบ A1 และตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนสมาทวนสอบในเครื่องมือสปิน ถ้าเครื่องมือสปินทวนสอบพบความผิดพลาด แสดงว่า มีกรณีที่เงื่อนไขไลฟ์เนสไม่เป็นจริง จึงสามารถสรุปได้ว่าซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติไลฟ์เนส สามารสรุปขั้นตอนการทวนสอบได้ดังนี้

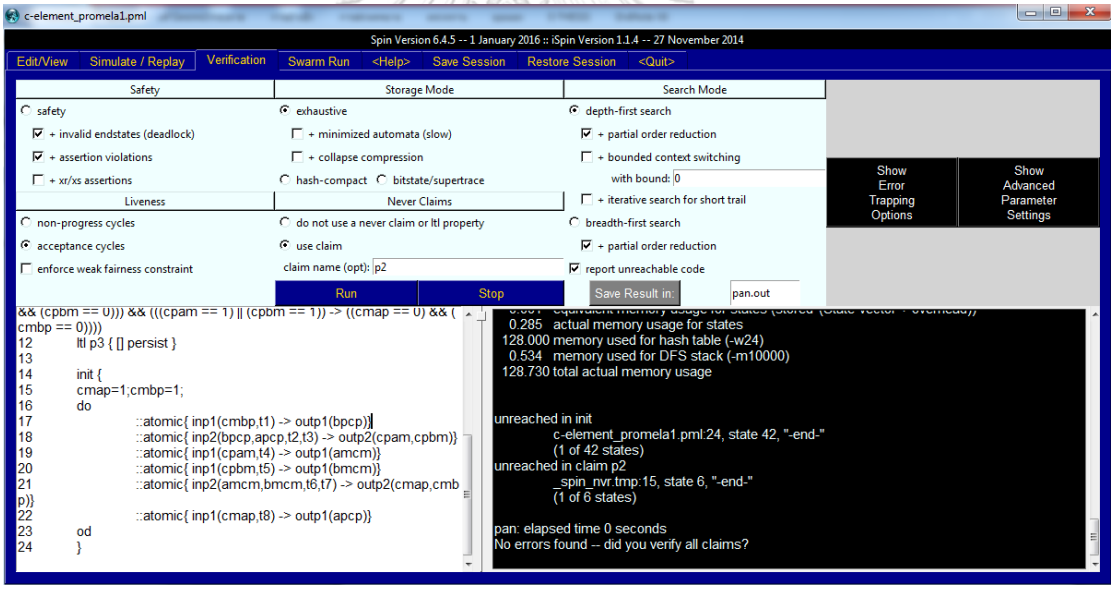
1) รวมรหัสโพรเมลาแบบ A1 และตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนสในโหมด “Edit/View”

2) เลือกโหมด “Verification”

3) เลือก “use claim” แล้วใส่ชื่อตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนส

4) กดปุ่ม “run”

5) ดูผลในช่องสีดำ ถ้าพบ “error” สรุปว่าไม่มีคุณสมบัติไลฟ์เนส แต่ถ้า “No errors” สรุปว่ามีคุณสมบัติไลฟ์เนส




รูปที่ 4-35 ผลการทวนสอบคุณสมบัติไลฟ์เนสของตัวอย่างซิกแนลแทรนซิชันกราฟ

เมื่อทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนส “p2” จะได้ผล “No errors” ดังรูปที่ 4-35 จึงสรุปได้ว่าตัวอย่างซิกแนลแทรนซิชันกราฟที่นำมาทวนสอบมีคุณสมบัติไลฟ์เนส

### 4.4.4 การทวนสอบคุณสมบัติความทนทาน

ในการทวนสอบคุณสมบัติความทนทานนั้นจะนำแบบจำลองรหัสโพรเมลาแบบ A1 และตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทานมาทวนสอบในเครื่องมือสปิน ถ้าเครื่องมือสปินทวนสอบพบความผิดพลาด แสดงว่า มีกรณีที่เงื่อนไขความทนทานไม่เป็นจริง จึงสามารถสรุปได้ว่าซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติความทนทาน สามารสรุปขั้นตอนการทวนสอบได้ดังนี้

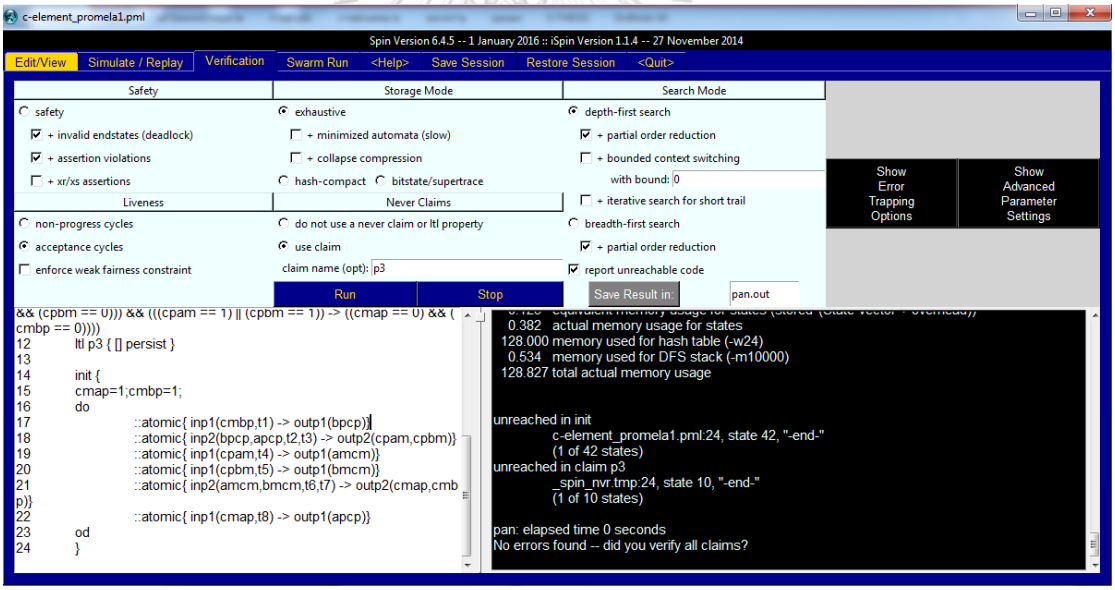
1) รวมรหัสโพรเมลาแบบ A1 และตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทานในโหมด “Edit/View”

2) เลือกโหมด “Verification”

3) เลือก “use claim” แล้วใส่ชื่อตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทาน

4) กดปุ่ม “run”

5) ดูผลในช่องสีดำ ถ้าพบ “error” สรุปว่าไม่มีคุณสมบัติความทนทานแต่ถ้า “No errors” สรุปว่ามีคุณสมบัติความทนทาน




รูปที่ 4-36 ผลการทวนสอบคุณสมบัติความทนทานของตัวอย่างซิกแนลแทรนซิชันกราฟ

เมื่อทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทาน “p3” จะได้ผล “No errors” ดังรูปที่ 4-36 จึงสรุปได้ว่าตัวอย่างซิกแนลแทรนซิชันกราฟที่นำมาทวนสอบมีคุณสมบัติความทนทาน

### 4.4.5 การทวนสอบคุณสมบัติความต้องกัน

ในการทวนสอบคุณสมบัติความต้องกันนั้นจะนำแบบจำลองรหัสโพรเมลาแบบ A2 และตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันมาทวนสอบในเครื่องมือสปิน ถ้าเครื่องมือสปินทวนสอบพบความผิดพลาด แสดงว่า มีกรณีที่เงื่อนไขความต้องกันเป็นจริง จึงสามารถสรุปได้ว่าซิกแนลแทรนซิชันกราฟมีความต้องกันในสัญญาณที่ทวนสอบ คุณสมบัติความต้องกันนี้ประกอบด้วยตรรกะเวลาเชิงเส้นหลายชุดและต้องทวนสอบให้ครบทุกชุดเพื่อยืนยันว่าทุกสัญญาณพบค่าดิจิทัล 1 และ 0 ในวัฏจักรที่มีจุดยอดไม่ซ้ำกัน การสรุปผลว่าซิกแนลแทรนซิชันกราฟมีคุณสมบัติความต้องกันจะสรุปได้ก็ต่อเมื่อทวนสอบตรรกะเวลาเชิงเส้นครบทุกชุดแล้วพบ “error” ในทุกชุด แต่ถ้าพบว่าตรรกะเวลาเชิงเส้นบางชุด “No errors” สามารถสรุปได้ทันทีว่าซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติความต้องกัน สามารสรุปขั้นตอนการทวนสอบได้ดังนี้

1) รวมรหัสโพรเมลาแบบ A2 และตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันในโหมด “Edit/View”

2) เลือกโหมด “Verification”

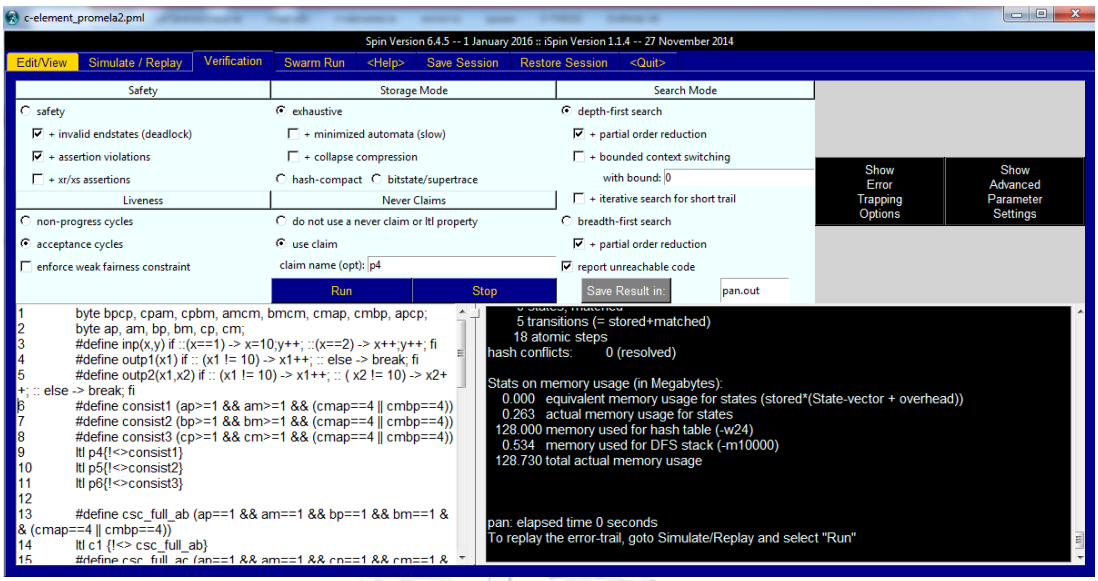
3) เลือก “use claim” แล้วใส่ชื่อตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกัน

4) กดปุ่ม “run”

5) ดูผลในช่องสีดำถ้าพบ “No errors” สรุปว่าไม่มีคุณสมบัติความต้องกัน ถ้า “error” ทวนสอบชุดตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันชุดถัดไป กรณีครบทุกชุดแล้วสรุปผลว่ามีคุณสมบัติความต้องกัน

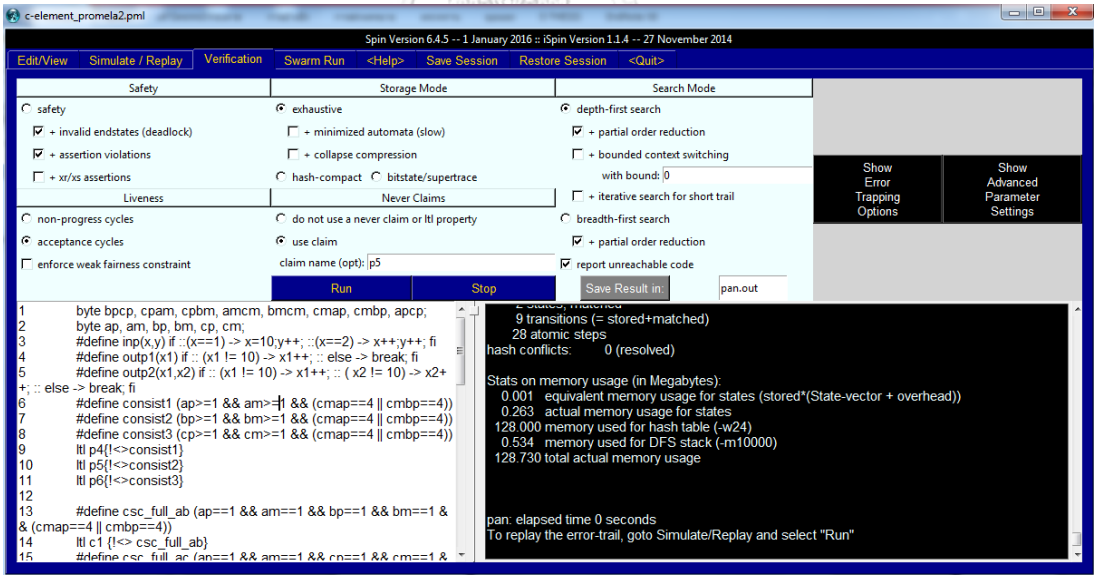
ในการทวนสอบคุณสมบัติความต้องกัน โดยใช้แบบจำลองโพรเมลาในรูปที่ 4-32 นั้นจะพบว่า มีสัญญาณ 3 สัญญาณ คือ a, b และ c และมีตรรกะเชิงเวลาของคุณสมบัติความต้องกัน 3 ชุด คือ p4, p5 และ p6 ซึ่งต้องทำการทวนสอบให้ครบทุกชุด และเมื่อนำมาทวนสอบจะได้ผลดังนี้

เมื่อทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันของสัญญาณ a “p4” พบว่าสัญญาณ a ที่มีค่าดิจิทัล 1 และ 0 ถูกพบในวัฏจักรที่มีจุดยอดไม่ซ้ำกัน ดังรูปที่ 4-37 จึงทวนสอบสัญญาณถัดไป



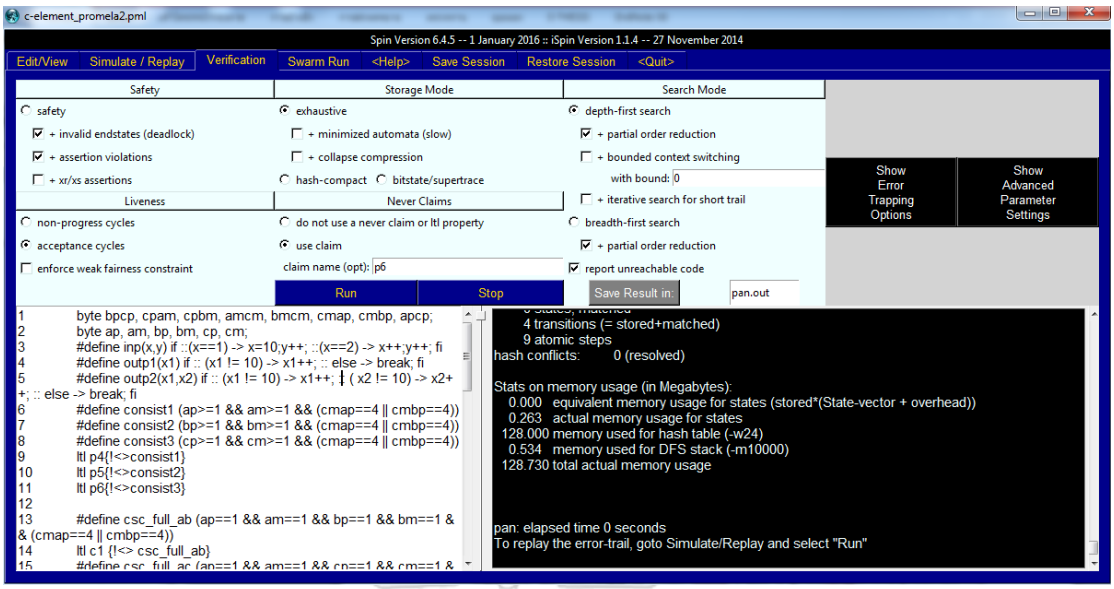

รูปที่ 4-37 ผลการทวนสอบความต้องกันของสัญญาณ a

เมื่อทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันของสัญญาณ b “p5” พบว่าสัญญาณ b ที่มีค่าดิจิทัล 1 และ 0 ถูกพบในวัฏจักรที่มีจุดยอดไม่ซ้ำกัน ดังรูปที่ 4-38 จึงทวนสอบสัญญาณถัดไป




รูปที่ 4-38 ผลการทวนสอบความต้องกันของสัญญาณ b

เมื่อทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันของสัญญาณ c “p6” พบว่าสัญญาณ c ที่มีค่าดิจิทัล 1 และ 0 ถูกพบในวัฏจักรที่มีจุดยอดไม่ซ้ำกัน ดังรูปที่ 4-39 ซึ่งพบว่ามีการจึงทวนสอบทวนสอบครบทุกสัญญาณแล้วจึงสรุปได้ว่าตัวอย่างซิกแนลแทรนซิชันกราฟที่ทวนสอบมีคุณสมบัติความต้องกัน




รูปที่ 4-39 ผลการทวนสอบความต้องกันของสัญญาณ c

### 4.4.6 การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์

ในการทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์นั้นจะนำแบบจำลองรหัสโพรเมลาแบบ A2 และตรรกะเวลาเชิงเส้นของคุณสมบัติการกำหนดสถานะที่สมบูรณ์มาทวนสอบในเครื่องมือสปิน โดยในขั้นตอนที่ 1 ทวนสอบตรรกะเวลาเชิงเส้นในส่วนฟูลล็อค ถ้าเครื่องมือสปินทวนสอบพบความผิดพลาด แสดงว่า มีกรณีที่เงื่อนไขฟูลล็อคเป็นจริง ให้ทำการจำลองการทำงานแล้วนำผลการจำลองไปตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนา โดยต้องทวนสอบให้ครบทุกชุด และในขั้นตอนที่ 2 เมื่อได้ผลฟูลล็อคแล้วจึงนำผลฟูลล็อคไปสร้างตรรกะเวลาเชิงเส้นในส่วนแทรนซิทีฟล็อค ทวนสอบตรรกะเวลาในส่วนแทรนซิทีฟล็อค ให้ทำการจำลองการทำงานแล้วนำผลการจำลองไปตรวจสอบแทรนซิทีฟล็อคในเครื่องมือที่พัฒนา เครื่องมือจะให้ผลว่าพบคุณสมบัติการกำหนดสถานะที่สมบูรณ์หรือไม่ สามารสรุปขั้นตอนการทวนสอบได้ดังนี้

1) รวมรหัสโพรเมลาแบบ A2 และตรรกะเวลาเชิงเส้นของคุณสมบัติการกำหนดสถานะที่สมบูรณ์ในโหมด “Edit/View”

2) เลือกโหมด “Verification”

3) เลือก “use claim” แล้วใส่ชื่อตรรกะเวลาเชิงเส้นในส่วนฟูลล็อค

4) กดปุ่ม “run”

5) ดูผลในช่องสีดำถ้าพบ “No errors” ให้ทวนสอบตรรกะเวลาเชิงเส้นของฟูลล็อคชุดถัดไป ถ้า “error” ให้เลือกโหมด “Simulation” แล้วเลือก “Guided with trail” แล้วกด “Run” นำผล

การจำลองที่ได้ไปตรวจสอบในเครื่องมือที่พัฒนา แล้วทวนสอบคุณสมบัติฟูลล็อคชุดถัดไปทวนสอบให้ครบทุกชุด

6) เครื่องมือที่พัฒนาจะให้ผลว่าสัญญาณคู่ใดเป็นฟูลล็อค ให้สร้างตรรกะเวลาเชิงเส้นของแทรนซิทีฟล็อคแล้วนำมาทวนสอบในเครื่องมือสปิน

7) ทวนสอบในเครื่องมือสปินทีละชุดโดยโหมด “Verification”

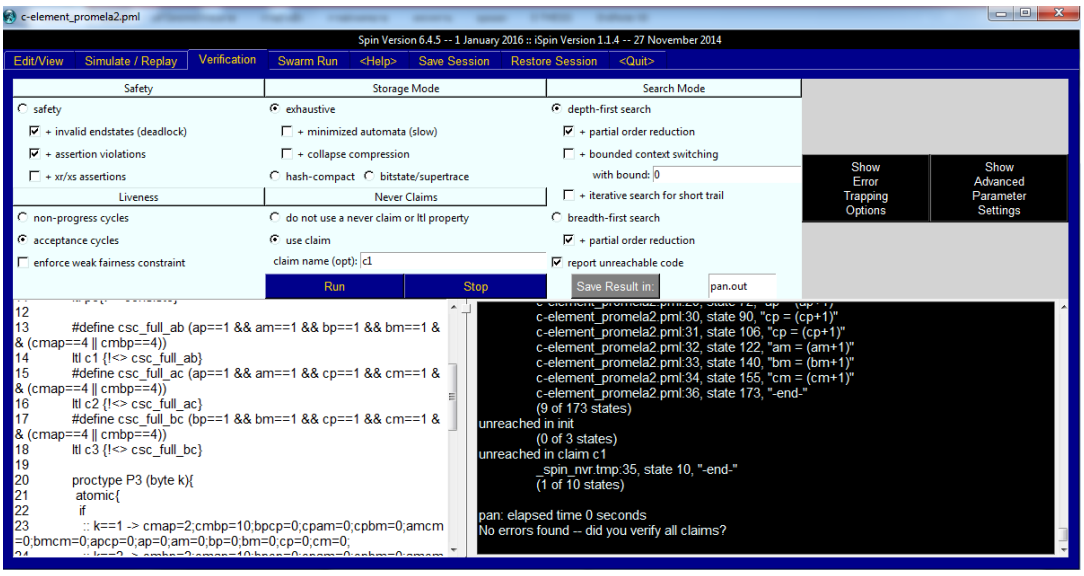
8) เลือก “use claim” แล้วใส่ชื่อตรรกะเวลาเชิงเส้นในส่วนแทรนซิทีฟล็อค

9) กดปุ่ม “run”

10) ดูผลในช่องสีดำถ้าพบ “No errors” ให้ทวนสอบตรรกะเวลาเชิงเส้นของแทรนซิทีฟล็อคชุดถัดไป ถ้า “error” ให้เลือกโหมด “Simulation” แล้วเลือก “Guided with trail” แล้วกด “Run” นำผลการจำลองที่ได้ไปตรวจสอบในเครื่องมือที่พัฒนา ถ้าเครื่องมือแสดงผลว่าพบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ให้จบการทำงาน แต่ถ้าแสดงผลว่าไม่พบให้ทวนสอบคุณสมบัติแทรนซิทีฟล็อคชุดถัดไป ในกรณีที่ทวนสอบครบทุกชุดแล้วไม่พบ “error” หรือเครื่องมือที่พัฒนาแสดงผลว่าไม่พบให้สรุปว่าซิกแนลแทรนซิชันกราฟไม่มีคุณสมบัติการกำหนดสถานะที่สมบูรณ์

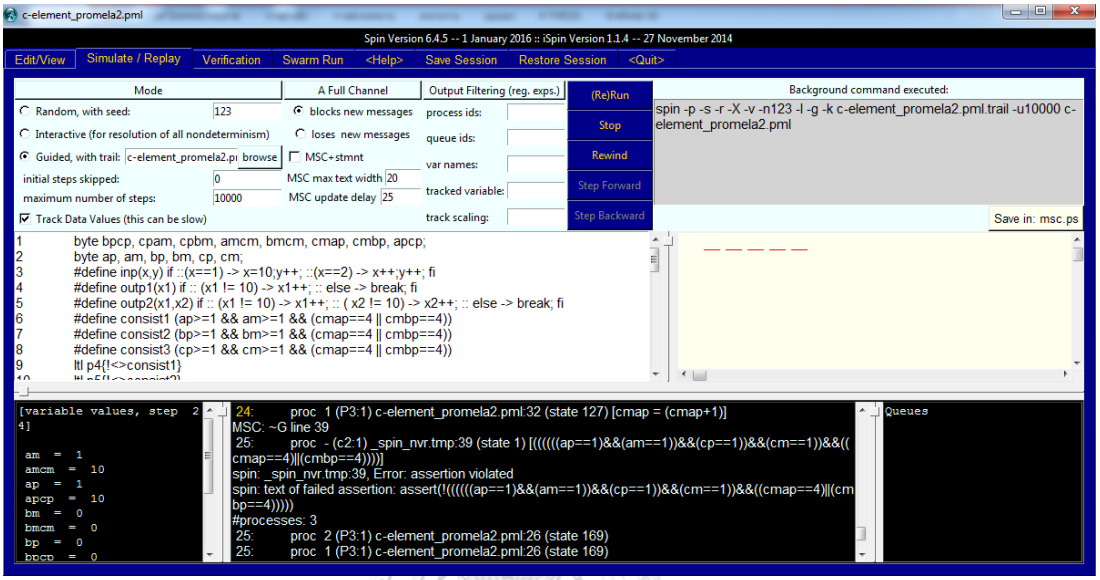
ในการทวนสอบคุณสมบัติคุณสมบัติการกำหนดสถานะที่สมบูรณ์ โดยใช้แบบจำลองโพรเมลาในรูปที่ 4-33 นั้นจะพบว่า มีคู่สัญญาณฟูลล็อคมี 3 คู่ คือ คู่สัญญาณ a, b คู่สัญญาณ a, c และคู่สัญญาณ b, c เมื่อนำมาทวนสอบจะได้ผลดังนี้

เมื่อทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นในส่วนฟูลล็อคคู่สัญญาณ a, b “c1” พบว่า “No errors” ดังรูปที่ 4-40 จึงทวนสอบสัญญาณถัดไป แสดงว่าคู่สัญญาณ a, b ไม่เป็นฟูลล็อคซึ่งกันและกัน จึงทวนสอบตรรกะเวลาเชิงเส้นชุดถัดไป




รูปที่ 4-40 ผลการทวนสอบตรรกะเวลาเชิงเส้น c1

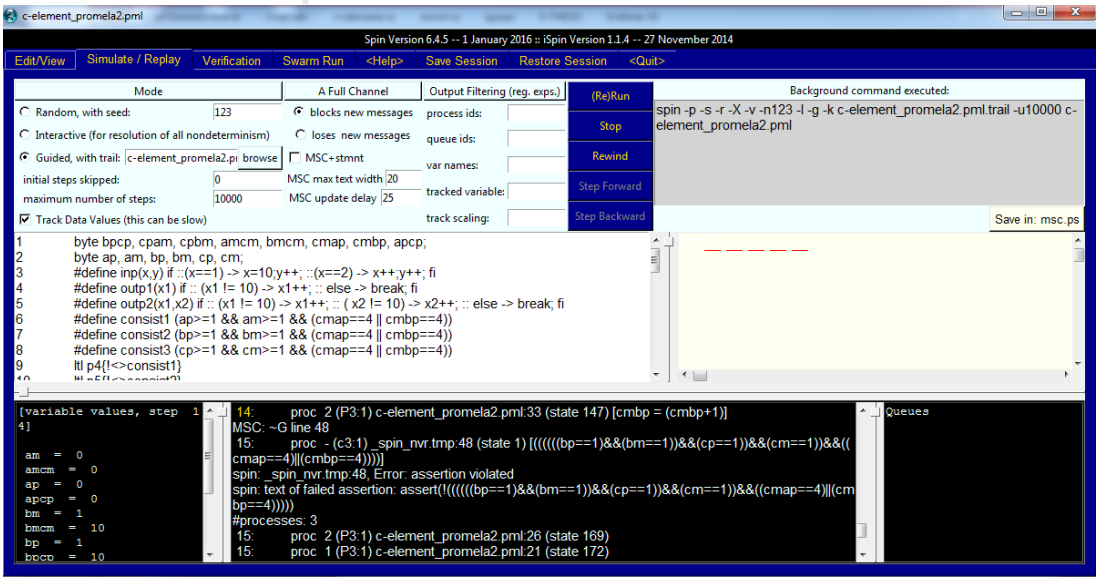
เมื่อทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นในส่วนฟูลล็อคคู่สัญญาณ a, c “c2” พบ “error” ดังรูปที่ 4-41 จึงทำการจำลองการทำงานแล้วนำผลไปตรวจสอบอีกครั้งในเครื่องมือที่พัฒนา พบว่าคู่สัญญาณ a, c เป็นฟูลล็อคซึ่งกันและกัน จึงทวนสอบตรรกะเวลาเชิงเส้นชุดถัดไป




รูปที่ 4-41 ผลการทวนสอบตรรกะเวลาเชิงเส้น c2

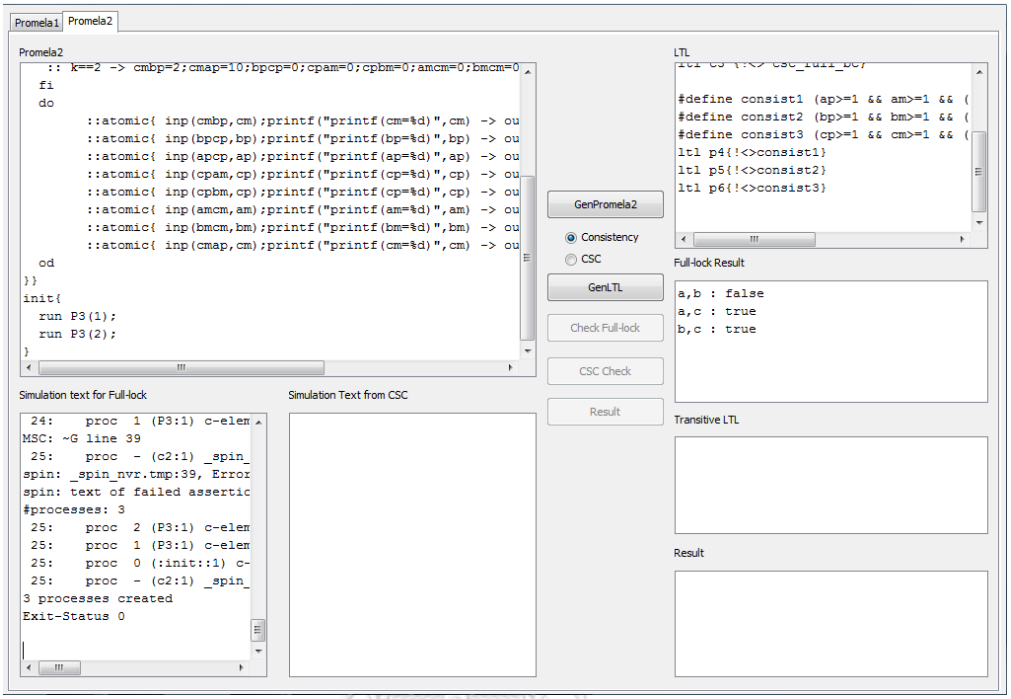
เมื่อทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นในส่วนฟูลล็อคคู่สัญญาณ b, c “c3” พบ “error” ดังรูปที่ 4-42 จึงทำการจำลองการทำงานแล้วนำผลไปตรวจสอบอีกครั้งในเครื่องมือที่พัฒนา พบว่าคู่สัญญาณ b, c เป็นฟูลล็อคซึ่งกันและกัน





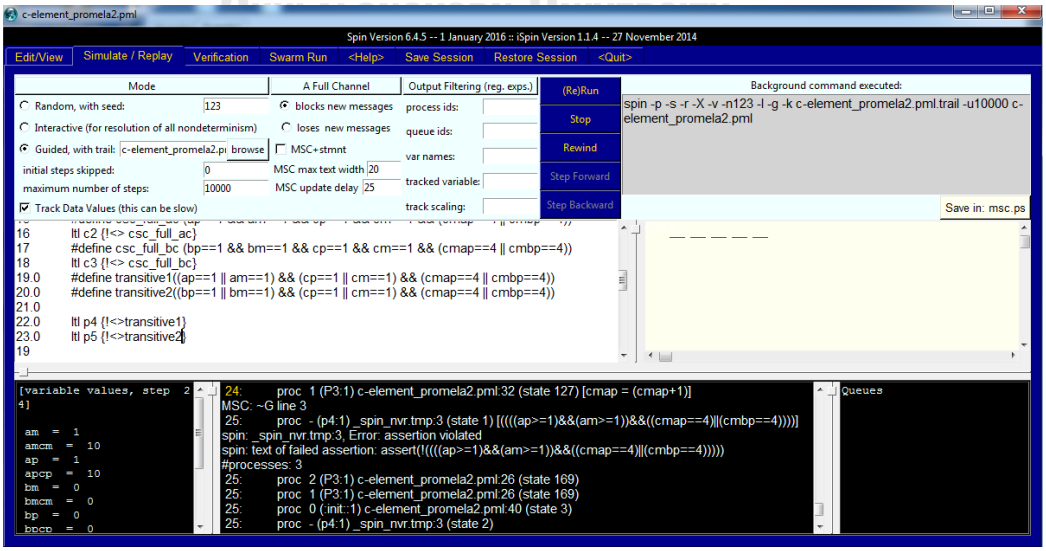

รูปที่ 4-42 ผลการทวนสอบตรรกะเวลาเชิงเส้น c3

เมื่อทำการจำลองการทำงานแล้วนำผลไปตรวจสอบอีกครั้งในเครื่องมือที่พัฒนา พบว่าคู่สัญญาณ a, b ไม่เป็นฟูลล็อคซึ่งกันและกัน ส่วนคู่สัญญาณ a, c และคู่สัญญาณ b, c เป็นฟูลล็อคซึ่งกันและกัน ซึ่งผลการตรวจสอบฟูลล็อคแสดงในเครื่องมือดังรูปที่ 4-43



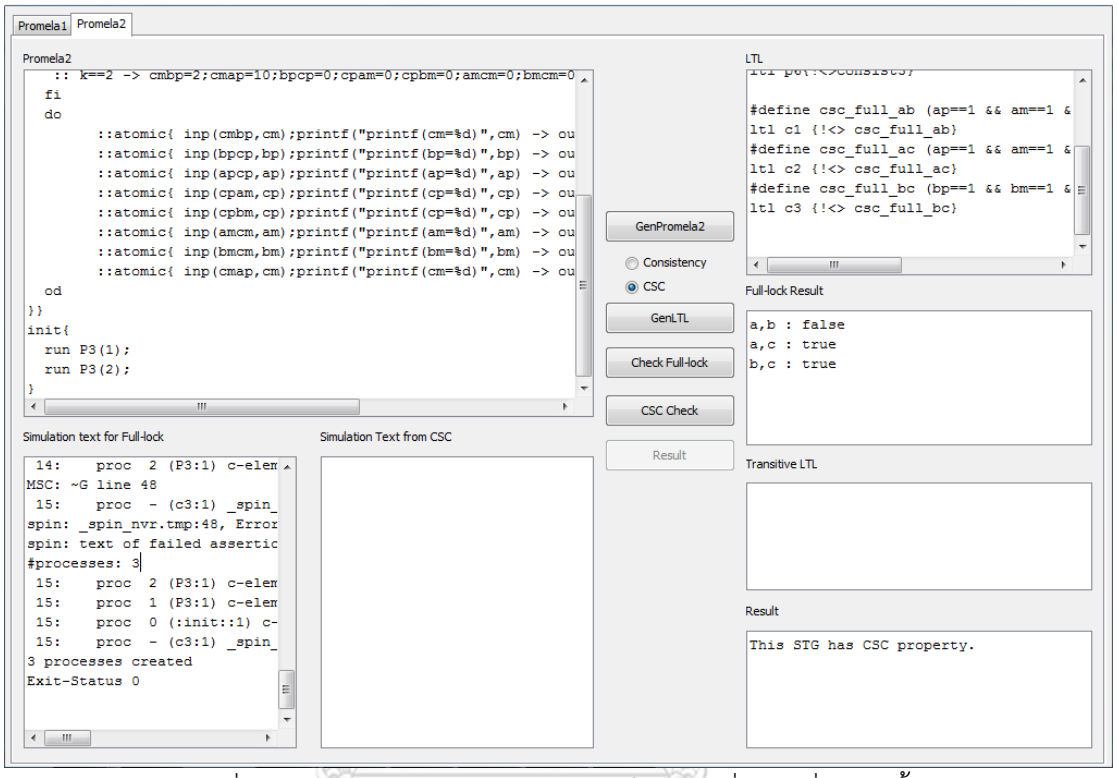

รูปที่ 4-43 ผลการตรวจสอบฟูลล็อค

เมื่อทวนสอบฟูลล็อคครบทุกคู่สัญญาณแล้ว จึงทวนสอบด้วยเครื่องมือสปิน โดยใช้ตรรกะเวลาเชิงเส้นของแทรนซิทีฟล็อค “p4” พบว่า “errors” ดังรูปที่ 4-44 จึงทำการจำลองการทำงานแล้วนำผลไปตรวจสอบอีกครั้งในเครื่องมือที่พัฒนา




รูปที่ 4-44 ผลการทวนสอบตรรกะเชิงเวลา p4

เมื่อทำการจำลองการทำงานแล้วนำผลไปตรวจสอบอีกครั้งในเครื่องมือที่พัฒนาดังรูปที่ 4-45 นั้นเครื่องมือแสดงผล “พบคุณสมบัติการกำหนดสถานะที่สมบูรณ์” จึงสรุปได้ว่าซิกแนลแทรนซิชันกราฟที่ทวนสอบมีคุณสมบัติการกำหนดสถานะที่สมบูรณ์




รูปที่ 4-45 ผลการตรวจสอบแทรนซิทีฟล็อคในเครื่องมือที่พัฒนาขึ้น

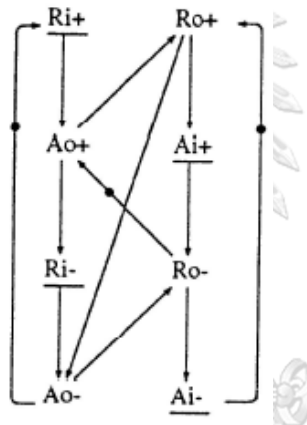
# บทที่ 5
# การทวนสอบตัวอย่างวงจรอสมวาร

ในบทนี้เสนอตัวอย่างการทวนสอบวงจรอสมวารด้วยตัวอย่างวงจรอสมวารทั้งหมด 6 วงจร ประกอบด้วยวงจรอสมวารฟูล (full) วงจรอสมวารไทรมอสเซนท์ (trimos-send) วงจรอสมวารอีเบอร์เจ็น (ebergen) วงจรอสมวารอินพุต (input) วงจรอสมวารทดลอง 1 และวงจรอสมวารทดลอง 2

## 5.1 การทวนสอบวงจรอสมวาร

### 5.1.1 กรณีศึกษาที่ 1 การทวนสอบตัวอย่างวงจรอสมวารฟูล

วงจรอสมวารฟูลมีซิกแนลแทรนซิชันกราฟดังรูป 5-1




รูปที่ 5-1 ซิกแนลแทรนซิชันกราฟของวงจรอสมวารฟูล



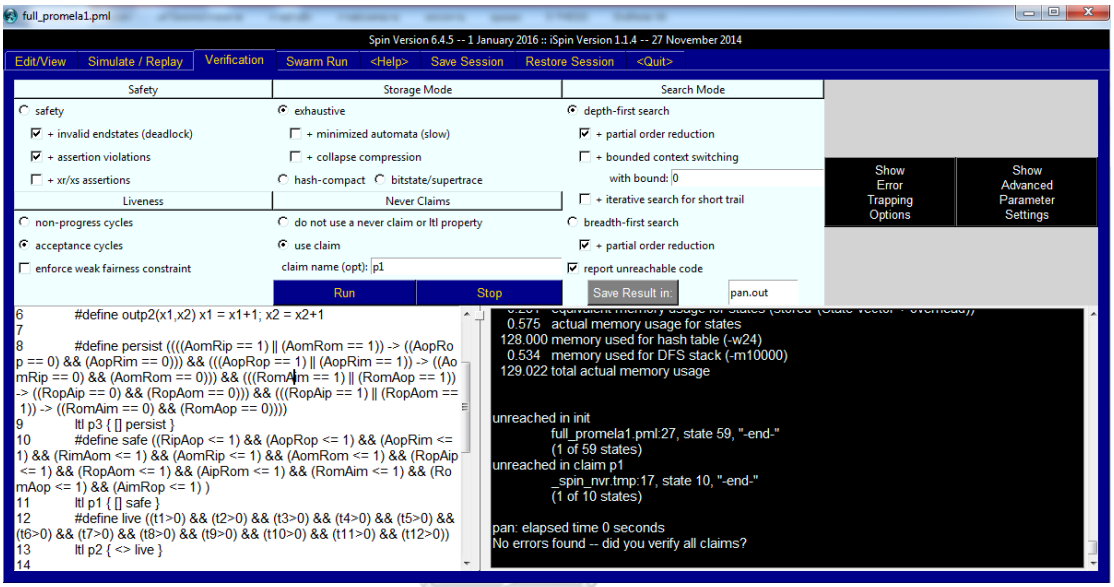
1) การทวนสอบคุณสมบัติความปลอดภัย

การทวนสอบคุณสมบัติความปลอดภัยใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยวงจรอสมวารฟูลสามารถแสดงเน็ตลิสต์ได้ดังรูปที่ 5-2

| | |
|---|---|
| .model full | Ao- Ri+ Ro- |
| .input Ri Ai | Ro+ Ai+ Ao- |
| .output Ro Ao | Ai+ Ro- |
| .graph | Ro- Ai- Ao+ |
| Ri+ Ao+ | Ai- Ro+ |
| Ao+ Ro+ Ri- | .marking {<Ao-,Ri+>,<Ro-,Ao+>,<Ai-,Ro+>} |
| Ri- Ao- | .end |

รูปที่ 5-2 เน็ตลิสต์ของวงจรอสมวารฟูล

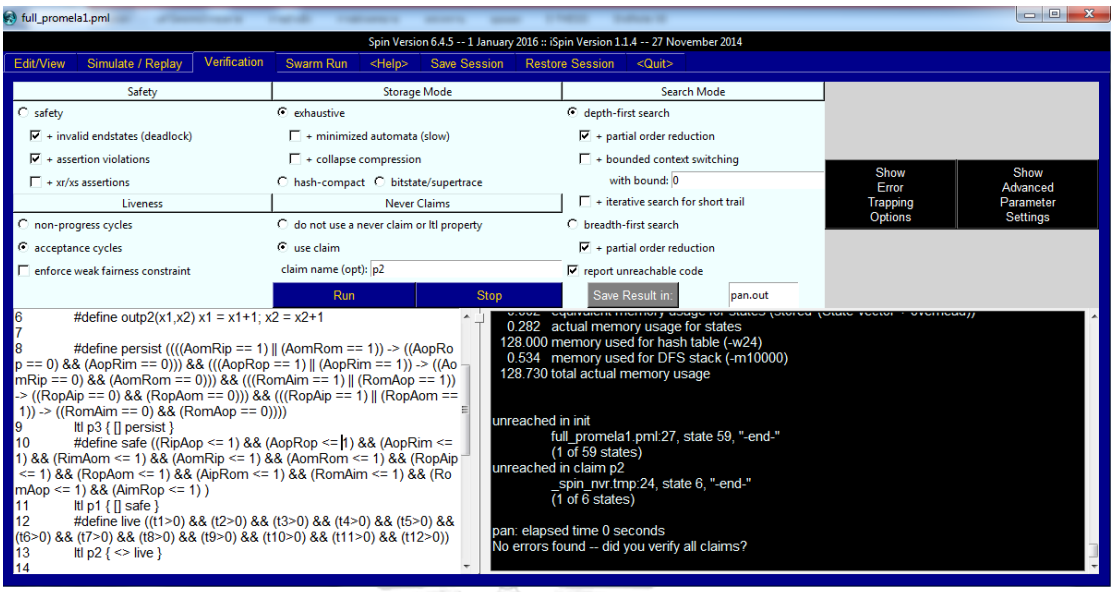
รหัสโพรเมลาแบบ A1 ของวงจรอสมวารฟูล แสดงในภาคผนวก ก.1.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความปลอดภัยตามหัวข้อ 4.4.2 ได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัยซึ่งแสดงในภาคผนวก ก.1.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-3 ซึ่งสปินไม่พบความผิดพลาดแสดงว่าวงจรอสมวารฟูล มีคุณสมบัติความปลอดภัย




รูปที่ 5-3 ผลการทวนสอบคุณสมบัติความปลอดภัยจากวงจรอสมวารฟูล

2) การทวนสอบคุณสมบัติไลฟ์เนส

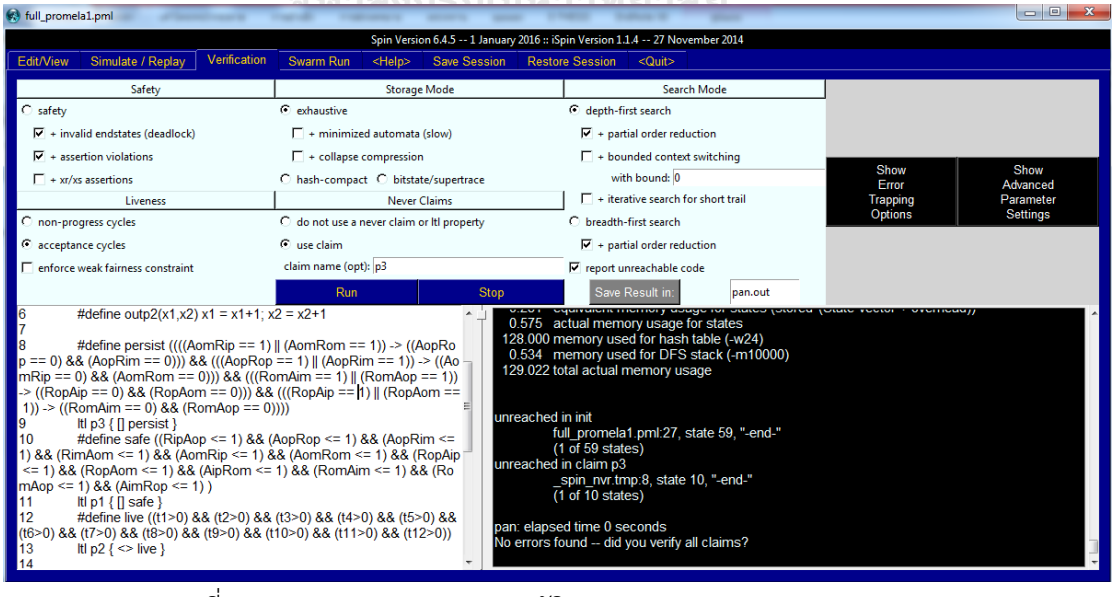
การทวนสอบคุณสมบัติไลฟ์เนสใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบโดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารฟูล แสดงในภาคผนวก ก.1.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติไลฟ์เนสตามหัวข้อ 4.4.3 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนสซึ่งแสดงในภาคผนวก ก.1.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-4 ซึ่งสปินไม่พบความผิดพลาดแสดงว่าวงจรอสมวารฟูล มีคุณสมบัติไลฟ์เนส




รูปที่ 5-4 ผลการทวนสอบคุณสมบัติความไลฟ์เนสจากวงจรอสมวารฟูล

3) การทวนสอบคุณสมบัติความทนทาน

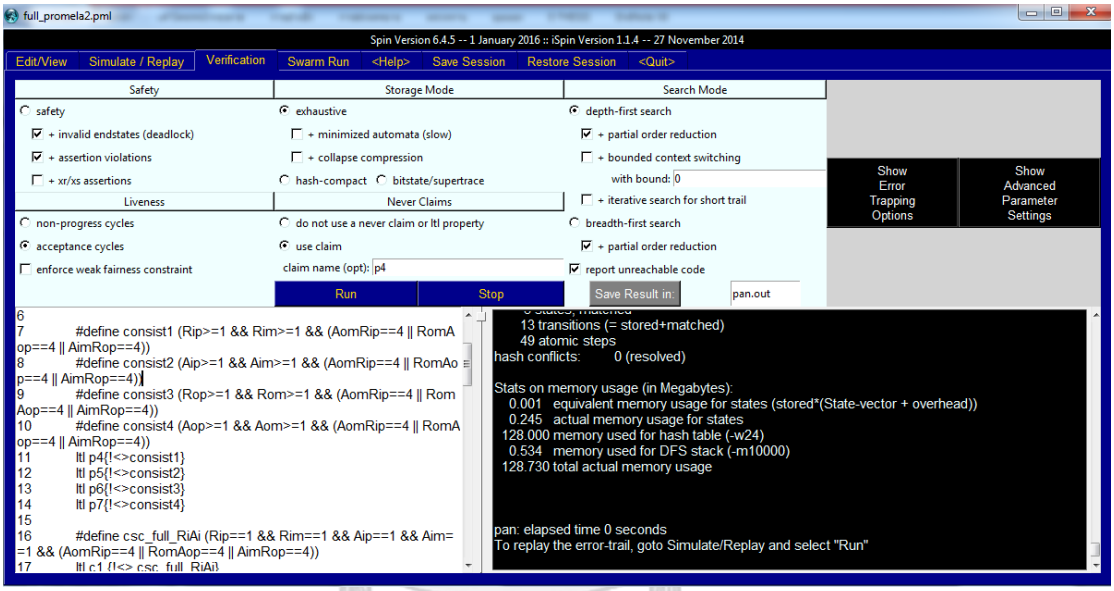
การทวนสอบคุณสมบัติความทนทานใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารฟูล แสดงในภาคผนวก ก.1.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความทนทานตามหัวข้อ 4.4.4 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทานซึ่งแสดงในภาคผนวก ก.1.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-5 ซึ่งสปินไม่พบความผิดพลาดแสดงว่าวงจรอสมวารฟูล มีคุณสมบัติความทนทาน



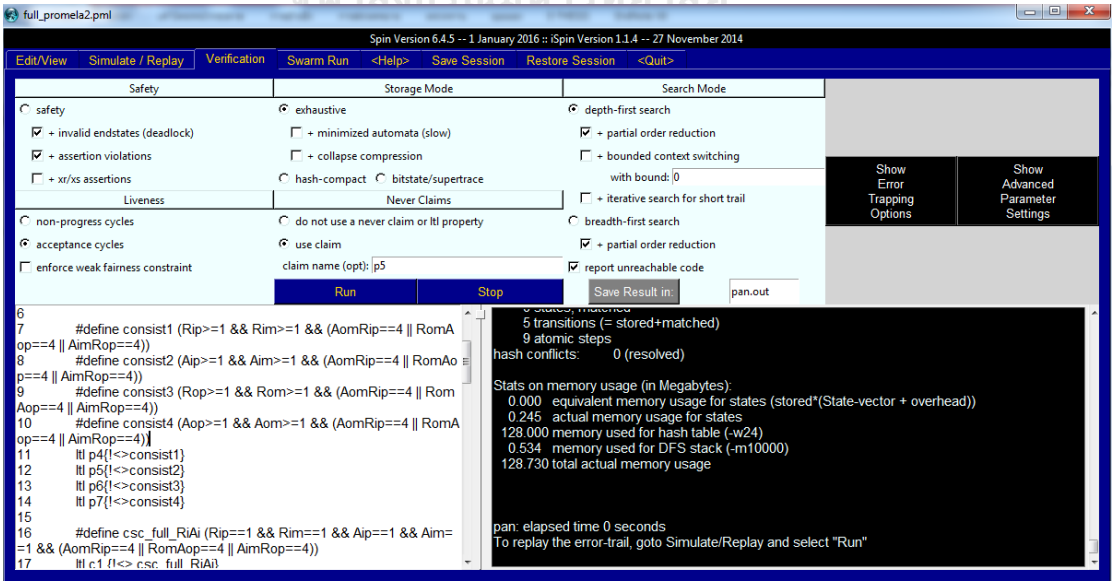

รูปที่ 5-5 ผลการทวนสอบคุณสมบัติความทนทานจากวงจรอสมวารฟูล

4) การทวนสอบคุณสมบัติความต้องกัน

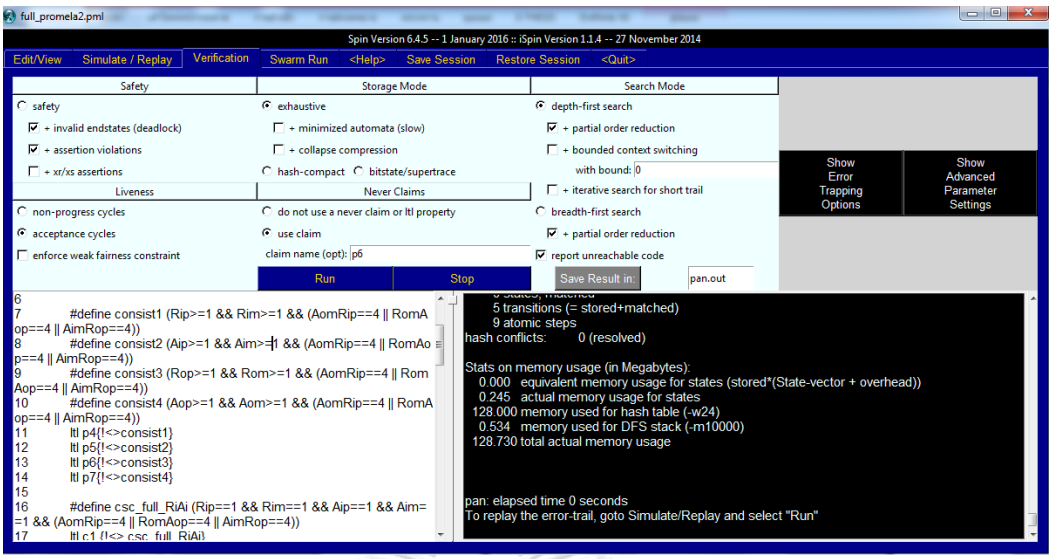
การทวนสอบคุณสมบัติความต้องกันใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารฟูล แสดงในภาคผนวก ก.1.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความต้องกันตามหัวข้อ 4.4.5 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันซึ่งแสดงในภาคผนวก ก.1.4 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-6 ถึงรูปที่ 5-9 ซึ่งสปินพบความผิดพลาดของทุกสัญญาณแสดงว่าวงจรอสมวารฟูล มีคุณสมบัติความต้องกัน



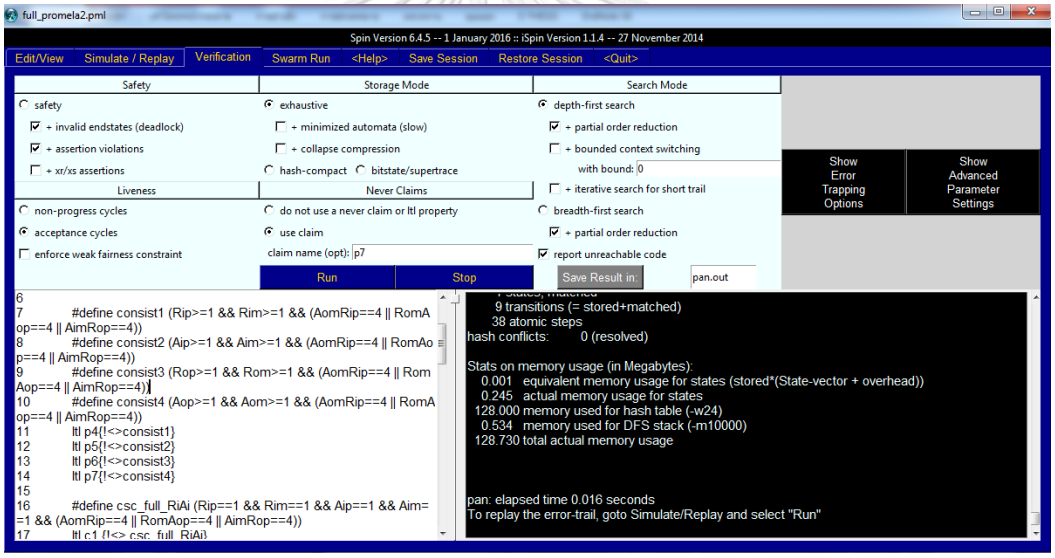

รูปที่ 5-6 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Ri




รูปที่ 5-7 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Ai



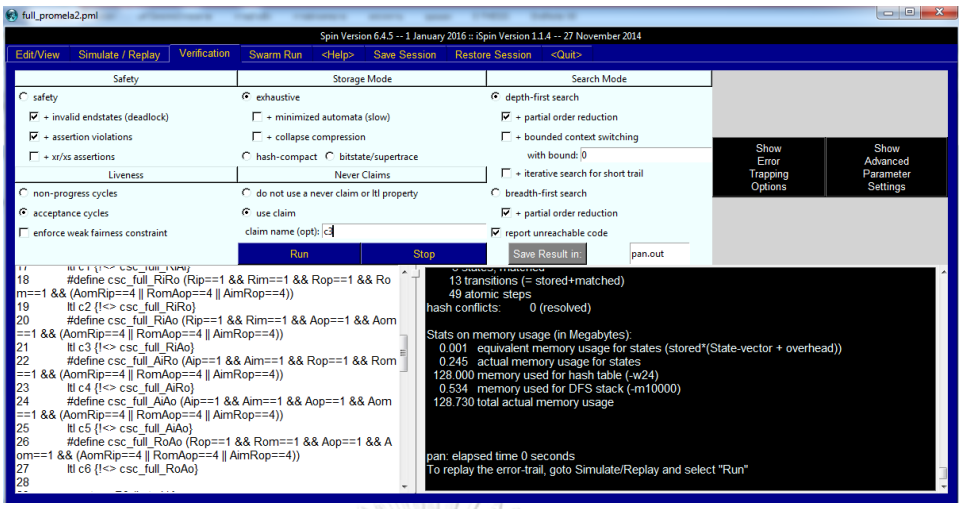

รูปที่ 5-8 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Ao



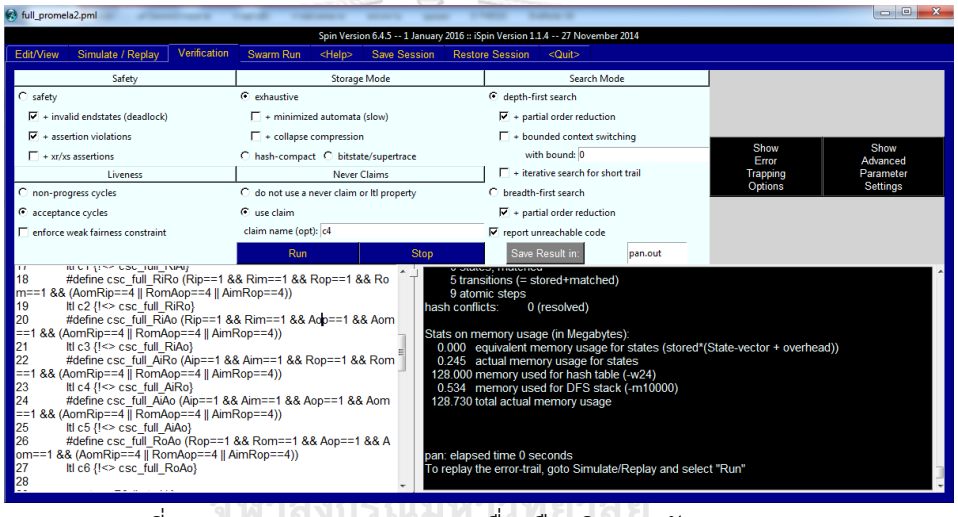

รูปที่ 5-9 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Ro

5) การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์

การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารฟูล แสดงในภาคผนวก ก.1.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ตามหัวข้อ 4.4.6 จะได้ตรรกะเวลาเชิงเส้นของฟูลล็อคซึ่งแสดงในภาคผนวก ก.1.5 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินพบว่าจากคู่สัญญาณทั้งหมดทวนสอบพบความผิดพลาดดังรูปที่ 5-10 ถึงรูปที่ 5-12 ซึ่งเครื่องมือสปินพบความผิดพลาดจึงนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น

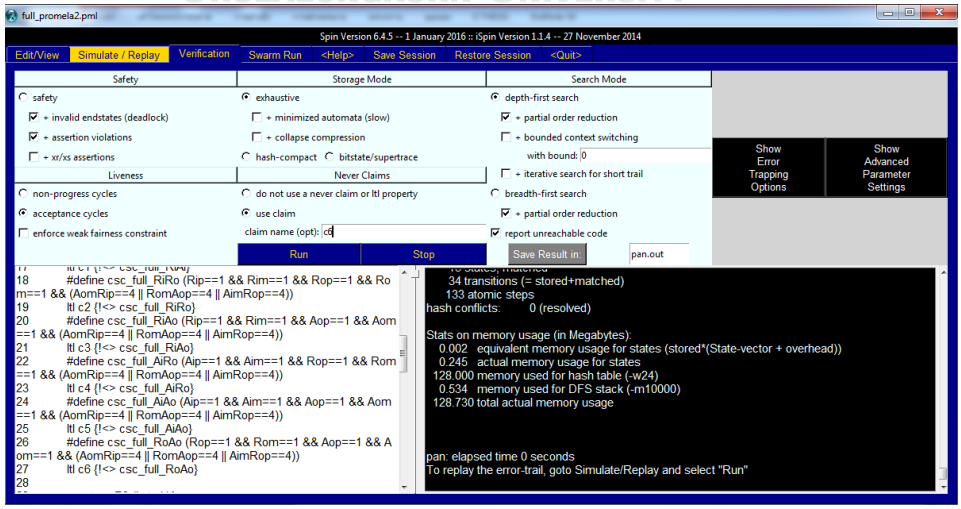



รูปที่ 5-10 ผลการทวนสอบจากเครื่องมือสปินของคู่สัญญาณ Ri, Ao



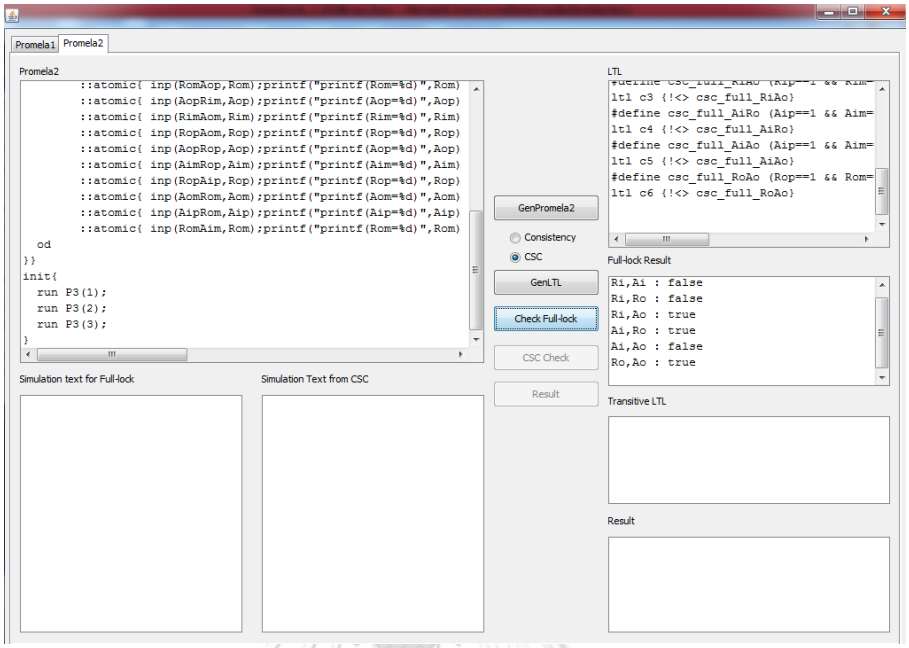

รูปที่ 5-11 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Ro, Ai





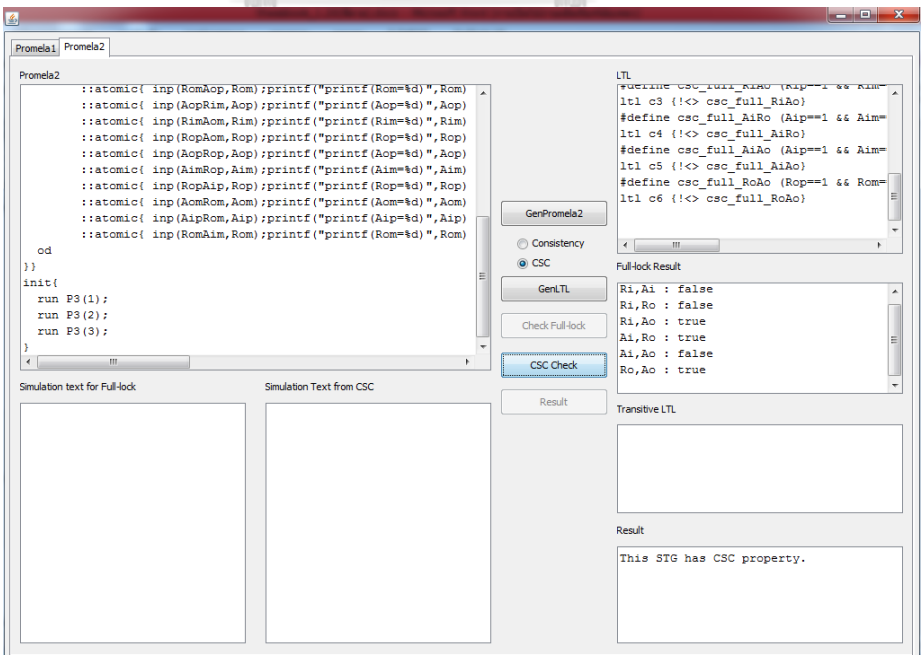

รูปที่ 5-12 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Ro, Ao

เมื่อนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น พบว่าทั้งสามคู่สัญญาณเป็นฟูลล็อค ดังแสดงในรูปที่ 5-13




รูปที่ 5-13 ผลการตรวจสอบฟูลล็อคจากเครื่องมือที่พัฒนาขึ้น

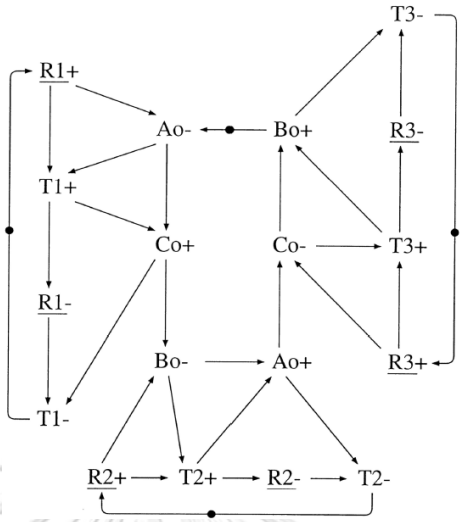
จากนั้นจึงตรวจสอบว่าพบแทรนซิทีฟล็อคจากวัฏจักรที่มีจุดยอดไม่ซ้ำกันที่พบหรือไม่โดยเครื่องมือที่พัฒนาขึ้น ซึ่งเครื่องมือแสดงผลว่า พบแทรนซิทีฟล็อค ดังแสดงในรูปที่ 5-14 จึงสามารถสรุปได้ว่าวงจรอสมวารฟูล มีคุณสมบัติการกำหนดสถานะที่สมบูรณ์




รูปที่ 5-14 ผลการทวนตรวจสอบแทรนซิทีฟล็อคจากเครื่องมือที่พัฒนาขึ้น

### 5.1.2 กรณีศึกษาที่ 2 การทวนสอบตัวอย่างวงจรอสมวารไทรมอสเซนท์

วงจรอสมวารไทรมอสเซนท์ มีซิกแนลแทรนซิชันกราฟดังรูป 5-15




รูปที่ 5-15 ซิกแนลแทรนซิชันกราฟของวงจรอสมวารไทรมอสเซนท์

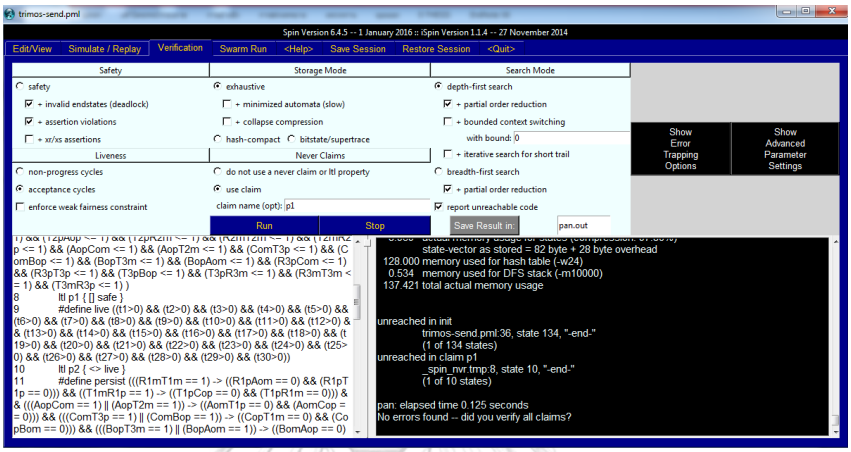
1) การทวนสอบคุณสมบัติความปลอดภัย

คุณสมบัติความปลอดภัยใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยวงจรอสมวารไทรมอสเซนท์ สามารถแสดงเน็ตลิสต์ได้ดังรูปที่ 5-16

| | |
|---|---|
| .model trimos-send | T2+ Ao+ R2- |
| .input R1 R2 R3 | R2- T2- |
| .output T1 T2 T3 Ao Bo Co | T2- R2+ |
| .graph | Ao+ Co- T2- |
| R1+ Ao- T1+ | Co- T3+ Bo+ |
| T1+ Co+ R1- | Bo+ T3- Ao- |
| R1- T1- | R3+ Co- T3+ |
| T1- R1+ | T3+ Bo+ R3- |
| Ao- T1+ Co+ | R3- T3- |
| Co+ T1- Bo- | T3- R3+ |
| Bo- Ao+ T2+ | .marking {<T1-,R1+>,<T2-,R2+>,<T3-,R3+>,<Bo+,Ao->} |
| R2+ T2+ Bo- | .end |

รูปที่ 5-16 เน็ตลิสต์ของวงจรอสมวารไทรมอสเซนท์

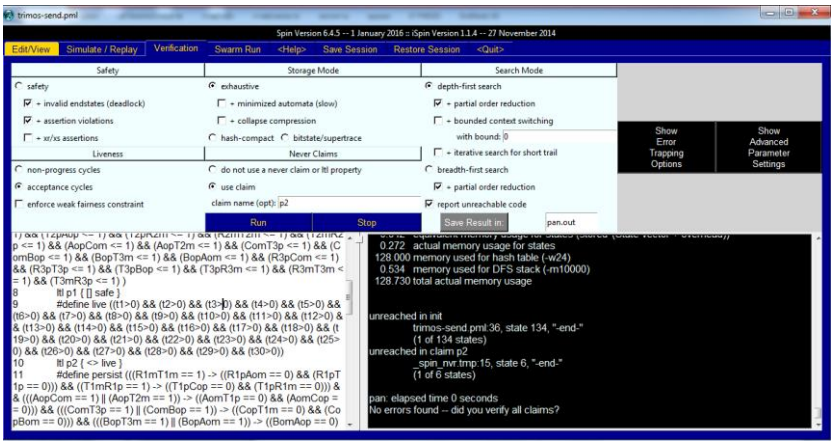
รหัสโพรเมลาแบบ A1 ของวงจรอสมวารไทรมอสเซนท์ แสดงในภาคผนวก ก.2.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความปลอดภัยตามหัวข้อ 4.4.2 ได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัยซึ่งแสดงในภาคผนวก ก.2.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-17 ซึ่งสปินไม่พบความผิดพลาดแสดงว่าวงจรอสมวารไทรมอสเซนท์ มีคุณสมบัติความปลอดภัย




รูปที่ 5-17 ผลการทวนสอบคุณสมบัติความปลอดภัยจากวงจรอสมวารไทรมอสเซนท์

2) การทวนสอบคุณสมบัติไลฟ์เนส

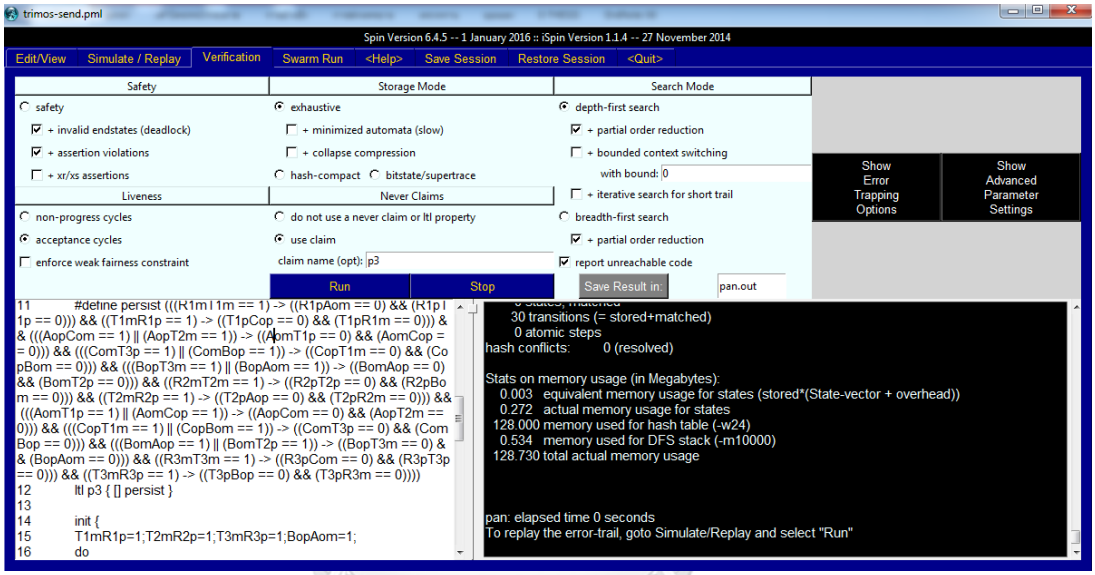
การทวนสอบคุณสมบัติไลฟ์เนสใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบโดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารไทรมอสเซนท์ แสดงในภาคผนวก ก.2.1 เมื่อได้รหัส โพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติไลฟ์เนสตามหัวข้อ 4.4.3 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนสซึ่งแสดงในภาคผนวก ก.2.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-18 ซึ่งสปินไม่พบความผิดพลาด แสดงว่าวงจรอสมวารไทรมอสเซนท์ มีคุณสมบัติไลฟ์เนส




รูปที่ 5-18 ผลการทวนสอบคุณสมบัติความไลฟ์เนสจากวงจรอสมวารไทรมอสเซนท์

3) การทวนสอบคุณสมบัติความทนทาน

การทวนสอบคุณสมบัติความทนทานใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารไทรมอสเซนท์ แสดงในภาคผนวก ก.2.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความทนทานตามหัวข้อ 4.4.4 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทานซึ่งแสดงในภาคผนวก ก.2.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-19 ซึ่งสปินไม่พบความผิดพลาดแสดงว่าวงจรอสมวารไทรมอสเซนท์ มีคุณสมบัติความทนทาน

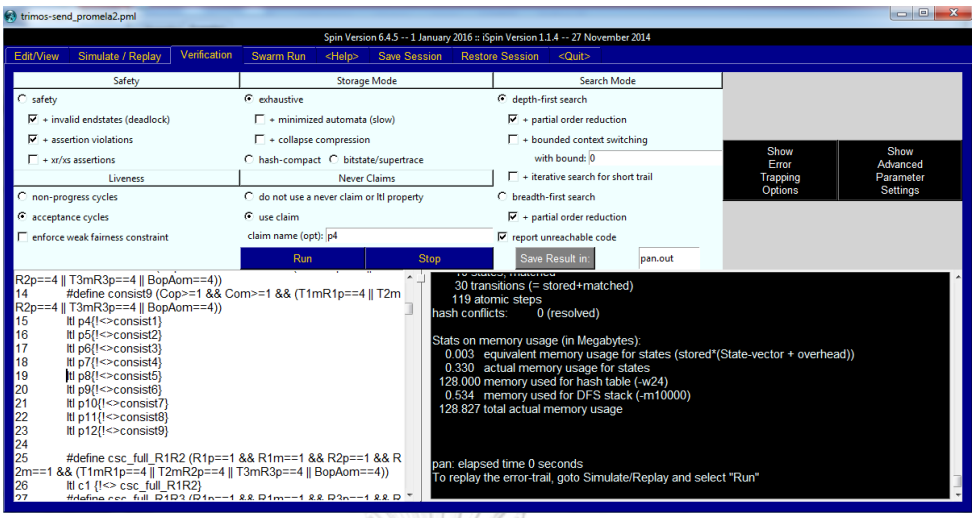



รูปที่ 5-19 ผลการทวนสอบคุณสมบัติความทนทานจากวงจรอสมวารไทรมอสเซนท์

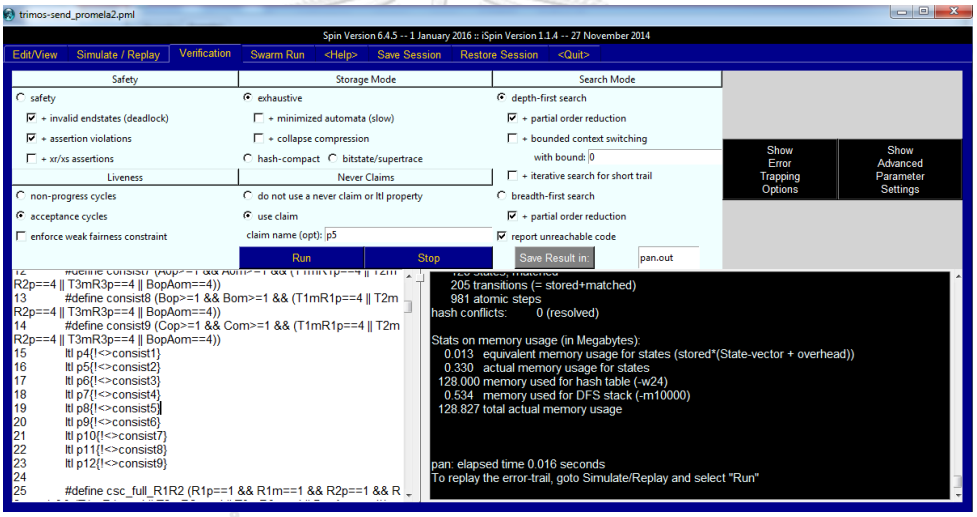


4) การทวนสอบคุณสมบัติความต้องกัน

การทวนสอบคุณสมบัติความต้องกันใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารไทรมอสเซนท์ แสดงในภาคผนวก ก.2.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความต้องกันตามหัวข้อ 4.4.5 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันซึ่งแสดงในภาคผนวก ก.2.4 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-20 ถึงรูปที่ 5-28 ซึ่งสปินพบความผิดพลาดของทุกสัญญาณแสดงว่าวงจรอสมวารไทรมอสเซนท์ มีคุณสมบัติความต้องกัน

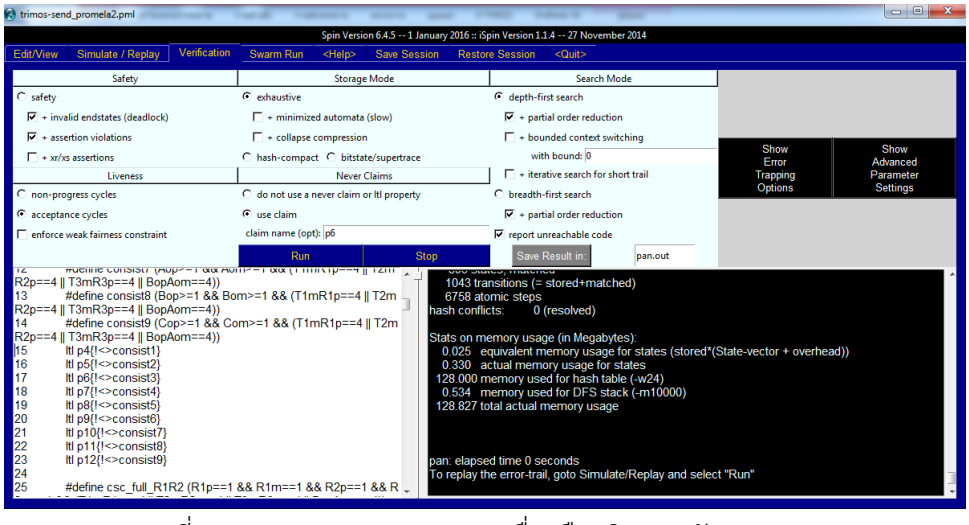



รูปที่ 5-20 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ R1

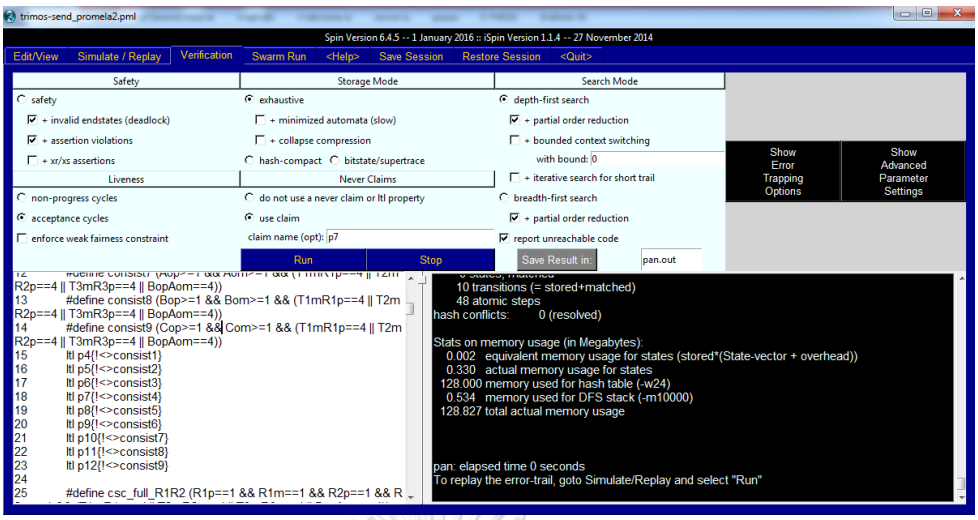



รูปที่ 5-21 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ R2

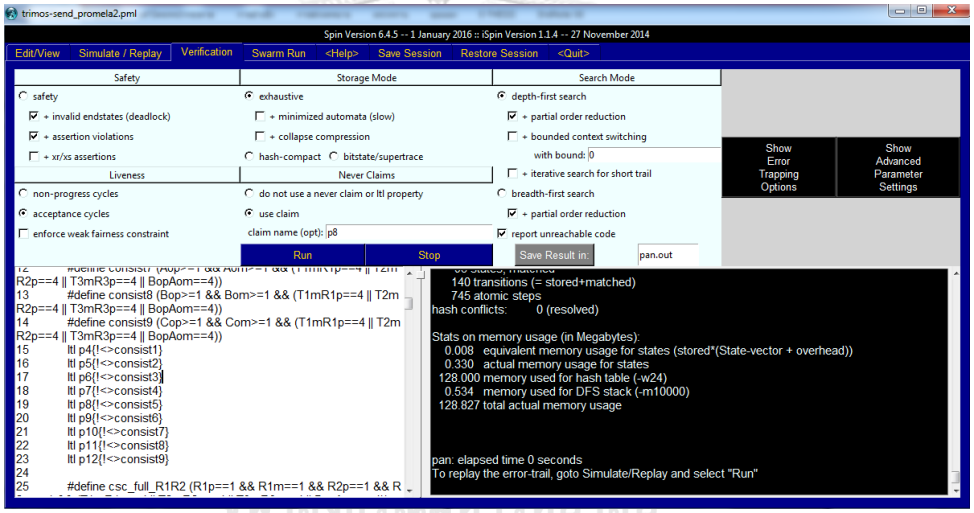





รูปที่ 5-22 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ R3

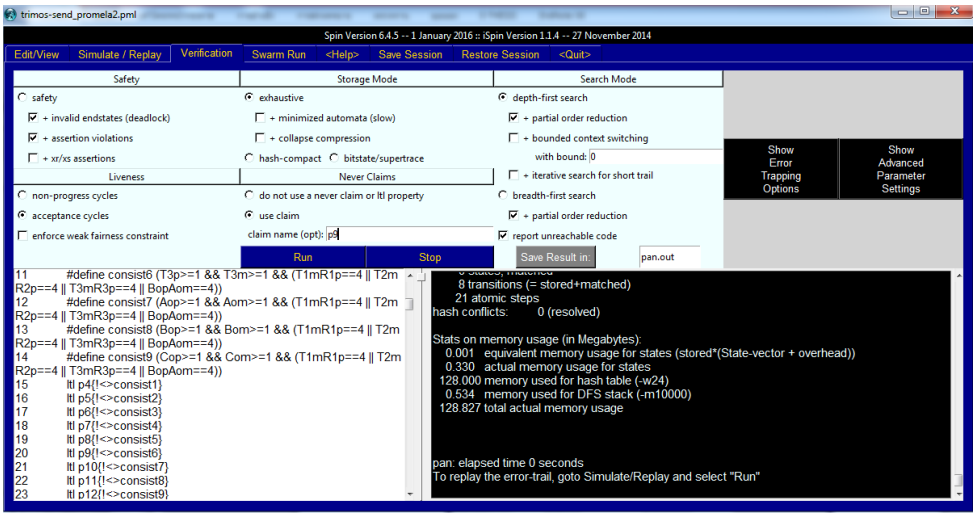



รูปที่ 5-23 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ T2

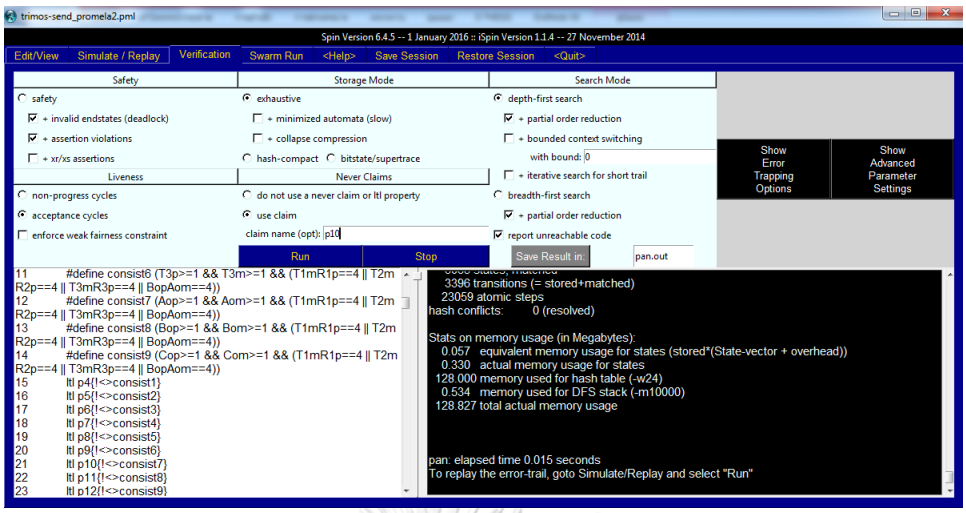



รูปที่ 5-24 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ T1

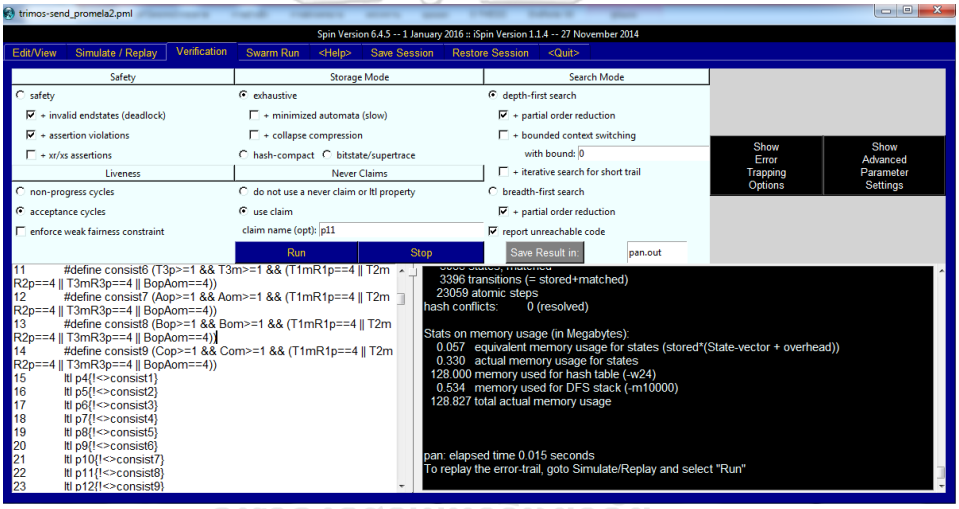





รูปที่ 5-25 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ T3

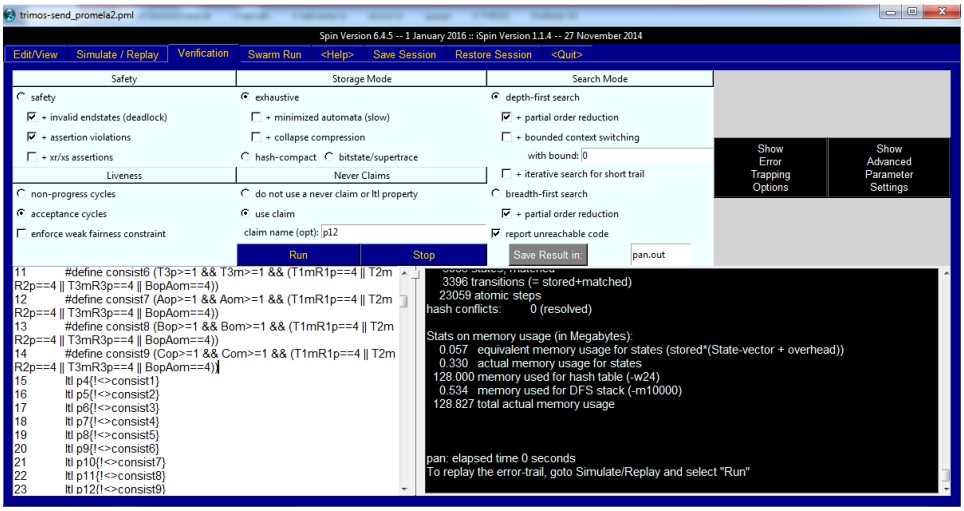



รูปที่ 5-26 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Ao




รูปที่ 5-27 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Bo

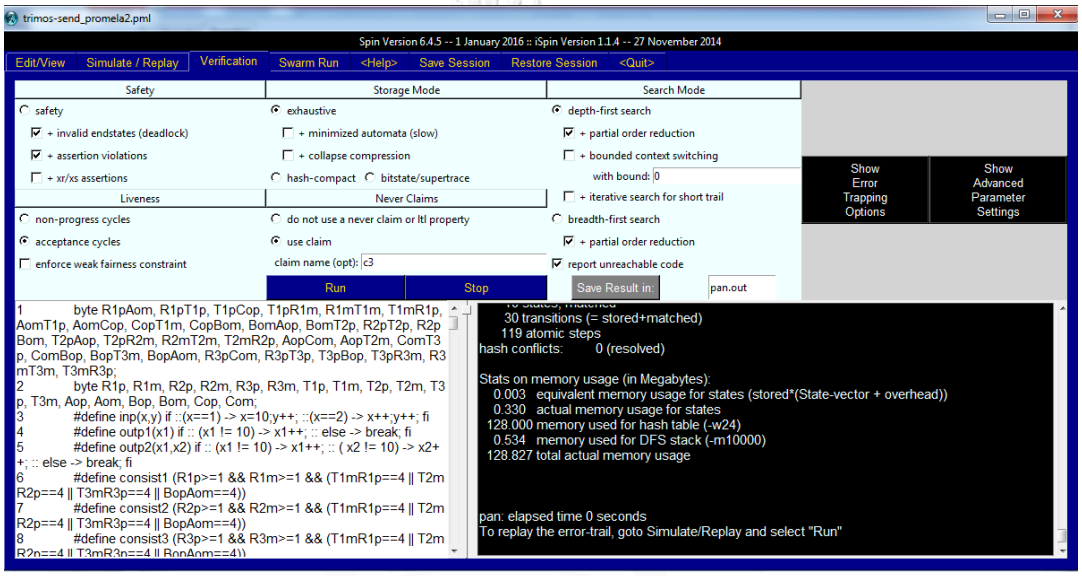




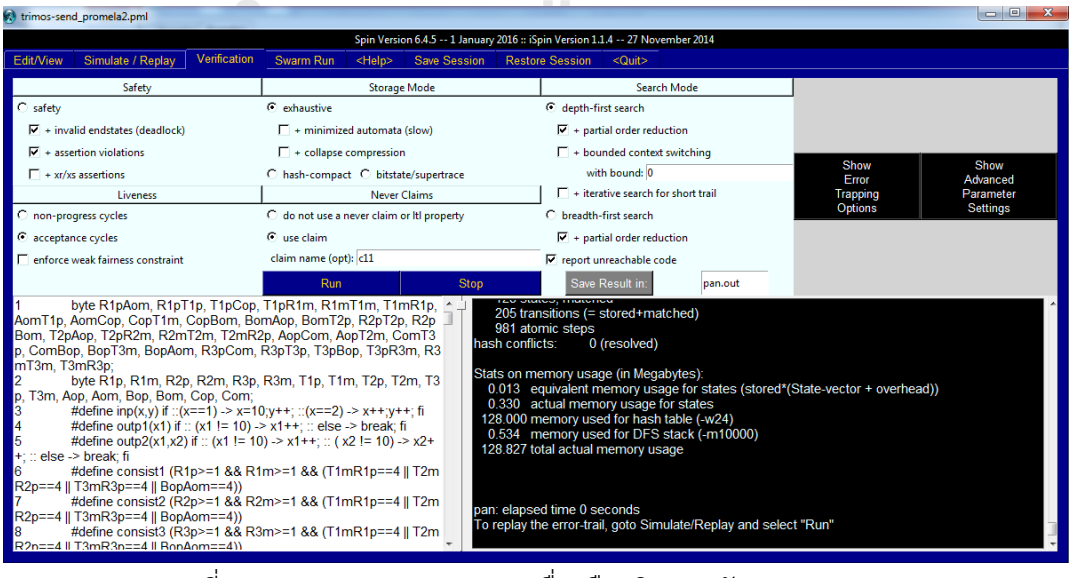

รูปที่ 5-28 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Co

5) การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์

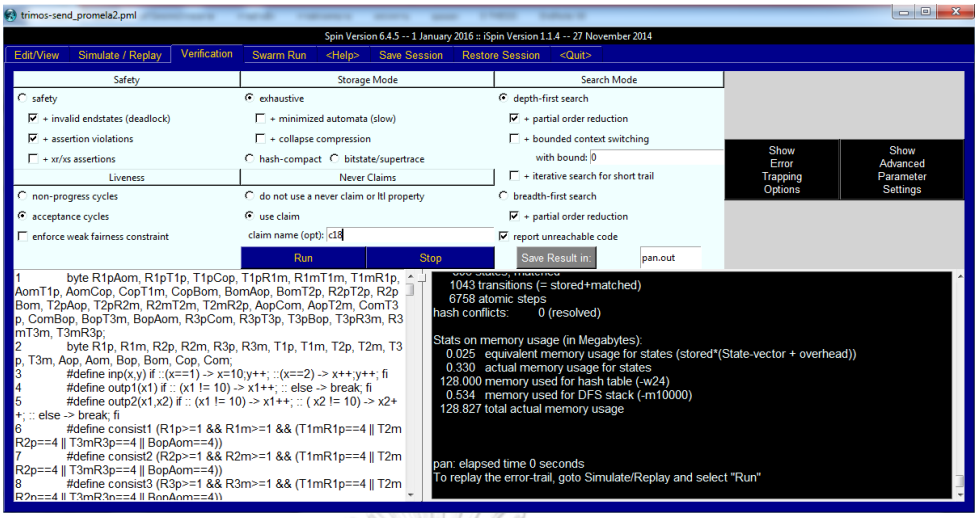
การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารไทรมอสเซนท์ แสดงในภาคผนวก ก.2.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ตามหัวข้อ 4.4.6 จะได้ตรรกะเวลาเชิงเส้นของฟูลล็อคซึ่งแสดงในภาคผนวก ก.2.5 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปิน พบว่าจากคู่สัญญาณทั้งหมดทวนสอบพบความผิดพลาดดังรูปที่ 5-29 ถึงรูปที่ 5-34 ซึ่งเครื่องมือสปินพบความผิดพลาดจึงนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น



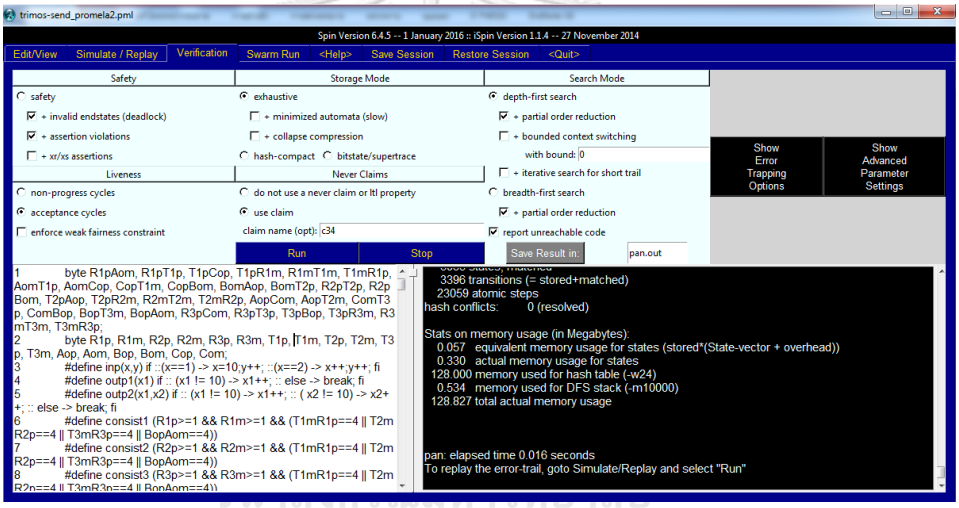

รูปที่ 5-29 ผลการทวนสอบจากเครื่องมือสปินของคู่สัญญาณ R1, T1



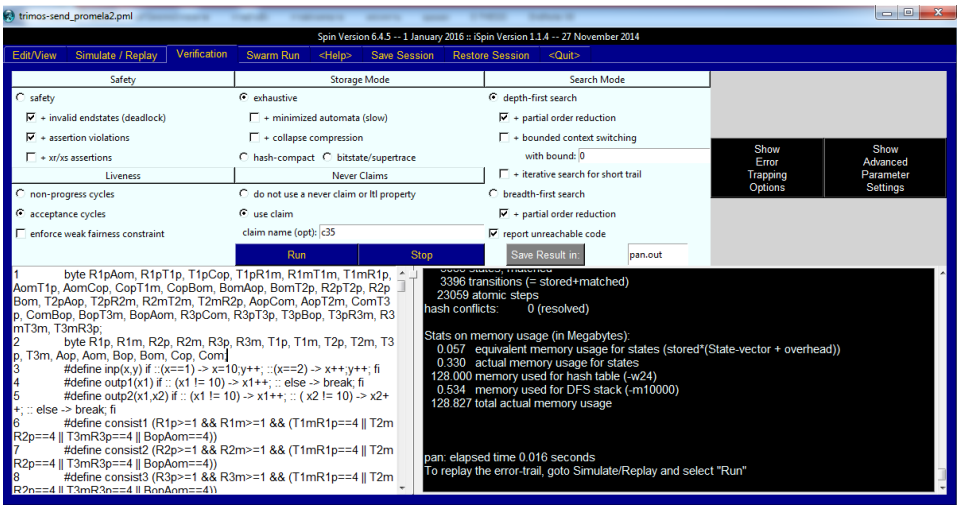

รูปที่ 5-30 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ R2, T2



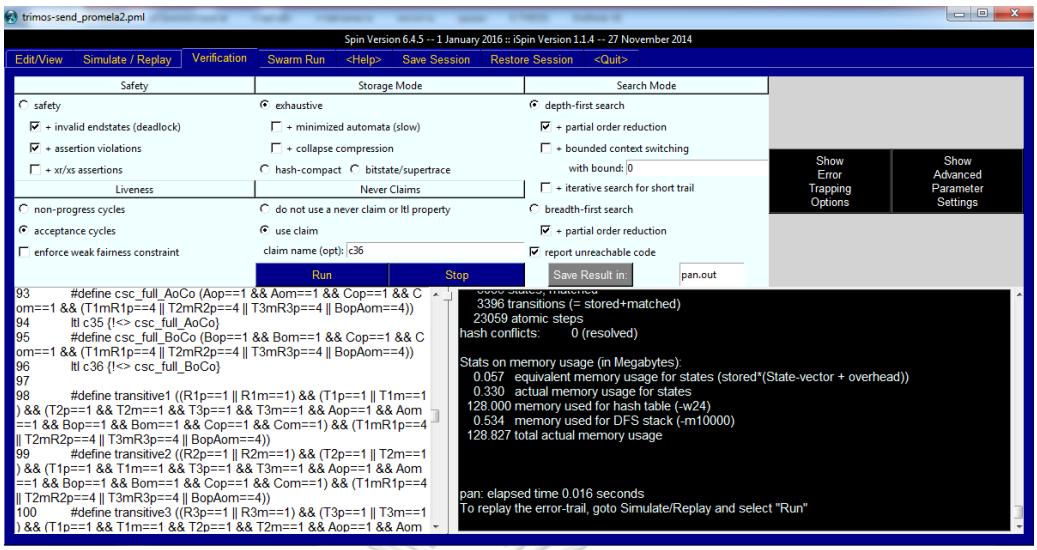

รูปที่ 5-31 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ R3, T3




รูปที่ 5-32 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Ao, Bo

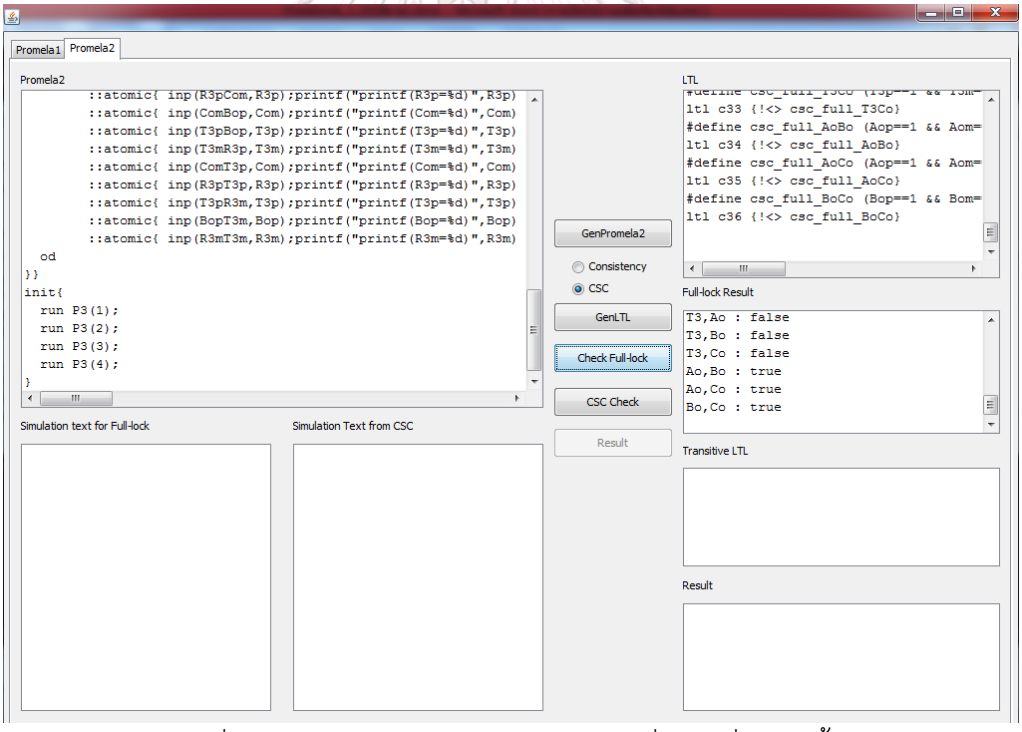



รูปที่ 5-33 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Ao, Co



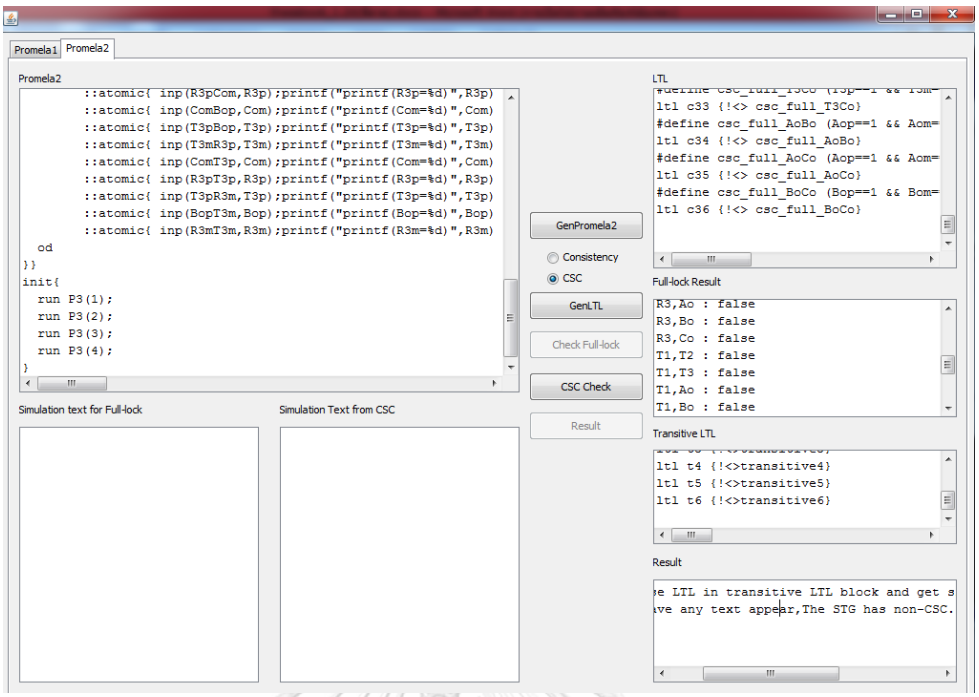

รูปที่ 5-34 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ Bo, Co

เมื่อนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น พบว่า ทั้งหกคู่สัญญาณเป็นฟูลล็อค ดังแสดงในรูปที่ 5-35



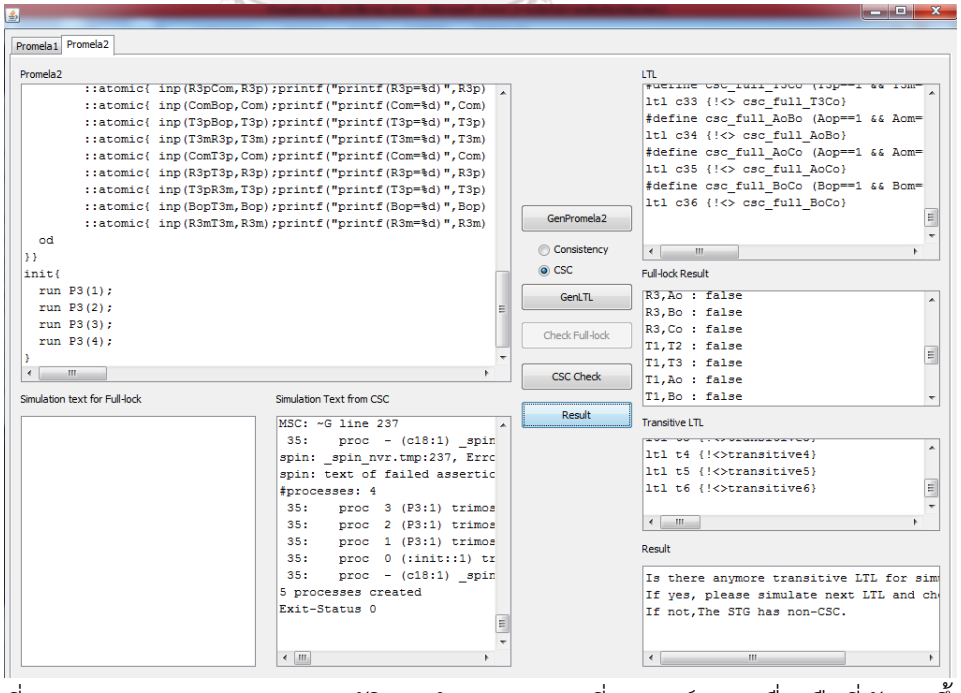

รูปที่ 5-35 ผลการตรวจสอบฟูลล็อคจากเครื่องมือที่พัฒนาขึ้น

จากนั้นนำผลที่ได้สัญญานฟูลล็อคทั้งหกคู่สัญญาณมาสร้างตรรกะเวลาเชิงเส้นเพื่อหาแทรนซิทีฟล็อค ซึ่งได้ตรรกะเวลาเชิงเส้น แสดงดังรูปที่ 5-36




รูปที่ 5-36 ตรรกะเวลาเชิงเส้นเพื่อหาแทรนซิทีฟล็อคจากเครื่องมือที่พัฒนาขึ้น

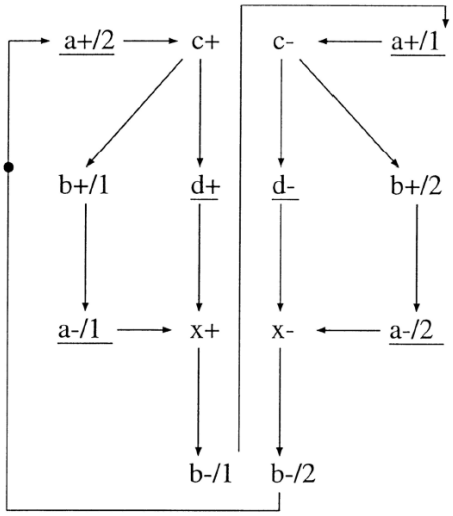
จากนั้นจึงตรวจสอบว่าพบแทรนซิทีฟล็อคจากวัฏจักรที่มีจุดยอดไม่ซ้ำกันที่พบหรือไม่โดยเครื่องมือที่พัฒนาขึ้น ซึ่งเครื่องมือแสดงผลว่า ไม่พบแทรนซิทีฟล็อคจากวัฏจักรที่มีจุดยอดไม่ซ้ำกันที่เจอทั้งหมด ดังแสดงในรูปที่ 5-37 จึงสามารถสรุปได้ว่าวงจรอสมวารไทรมอสเซนท์ ไม่มีคุณสมบัติการกำหนดสถานะที่สมบูรณ์




รูปที่ 5-37 ผลการทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์จากเครื่องมือที่พัฒนาขึ้น

### 5.1.3 กรณีศึกษาที่ 3 การทวนสอบตัวอย่างวงจรอสมวารอีเบอร์เจ็น

วงจรอสมวารอีเบอร์เจ็น มีซิกแนลแทรนซิชันกราฟดังรูป 5-38




รูปที่ 5-38 ซิกแนลแทรนซิชันกราฟของวงจรอสมวารอีเบอร์เจ็น

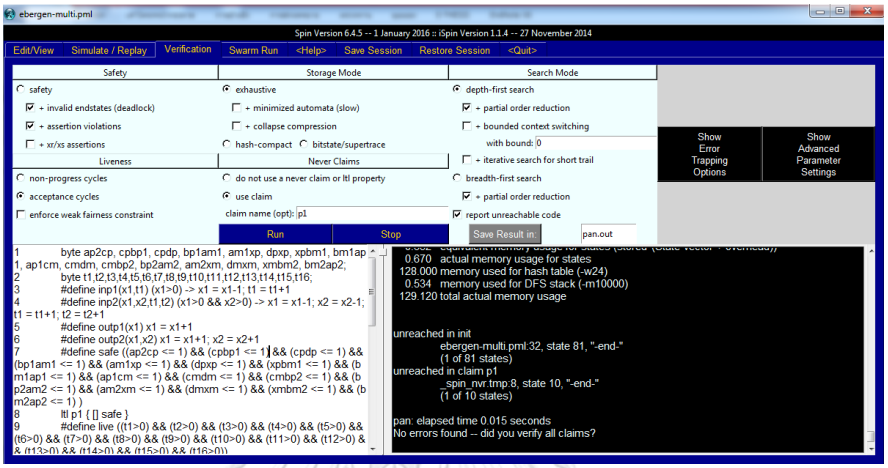
1) การทวนสอบคุณสมบัติความปลอดภัย

คุณสมบัติความปลอดภัยใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยวงจรอสมวารอีเบอร์เจ็น สามารถแสดงเน็ตลิสต์ได้ดังรูปที่ 5-39

| | |
|---|---|
| .model ebergen | b-/1 a+/1 |
| .input a d | a+/1 c- |
| .output b c x | c- d- b+/2 |
| .graph | b+/2 a-/2 |
| a+/2 c+ | a-/2 x- |
| c+ b+/1 d+ | d- x- |
| b+/1 a-/1 | x- b-/2 |
| a-/1 x+ | b-/2 a+/2 |
| d+ x+ | .marking {<b-/2,a+/2>} |
| x+ b-/1 | .end |

รูปที่ 5-39 เน็ตลิสต์ของวงจรอสมวารอีเบอร์เจ็น

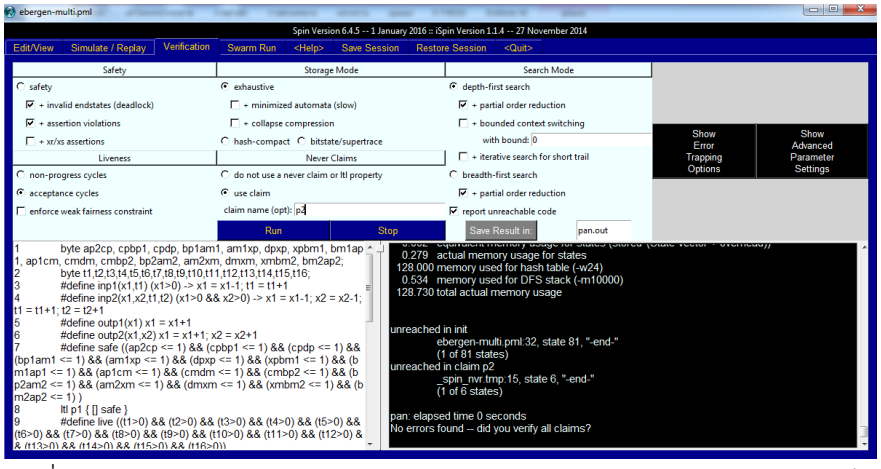
รหัสโพรเมลาแบบ A1 ของวงจรอสมวารอีเบอร์เจ็น แสดงในภาคผนวก ก.3.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความปลอดภัยตามหัวข้อ 4.4.2 ได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัยซึ่งแสดงในภาคผนวก ก.3.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-40 ซึ่งสปินไม่พบความผิดพลาดแสดงว่าวงจรอสมวารอีเบอร์เจ็น มีคุณสมบัติความปลอดภัย




รูปที่ 5-40 ผลการทวนสอบคุณสมบัติความปลอดภัยจากวงจรอสมวารอีเบอร์เจ็น

2) การทวนสอบคุณสมบัติไลฟ์เนส

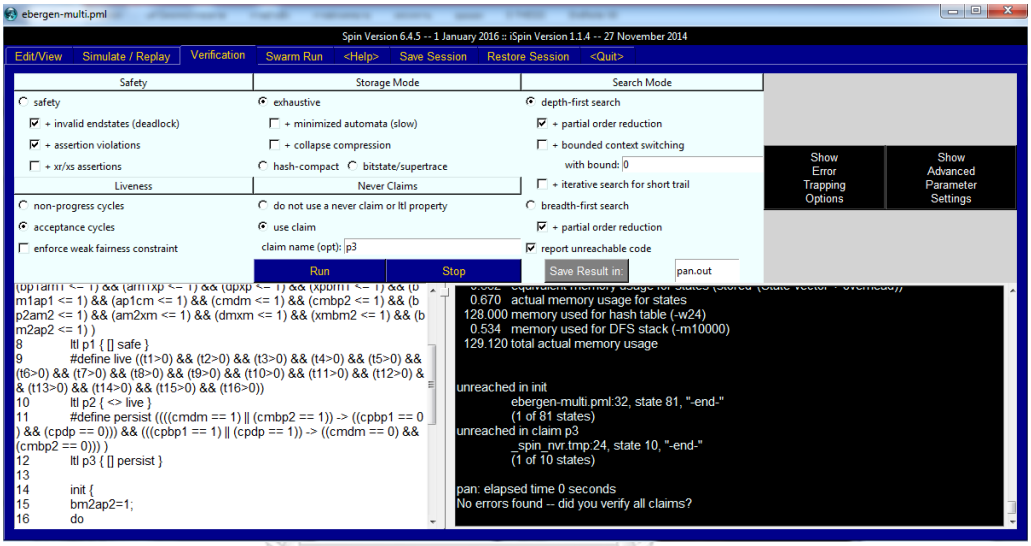
การทวนสอบคุณสมบัติไลฟ์เนสใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบโดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารอีเบอร์เจ็น แสดงในภาคผนวก ก.3.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติไลฟ์เนสตามหัวข้อ 4.4.3 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนสซึ่งแสดงในภาคผนวก ก.3.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-41 ซึ่งสปินไม่พบความผิดพลาดแสดงว่าวงจรอสมวารอีเบอร์เจ็น มีคุณสมบัติไลฟ์เนส




รูปที่ 5-41 ผลการทวนสอบคุณสมบัติความไลฟ์เนสจากวงจรอสมวารอีเบอร์เจ็น

3) การทวนสอบคุณสมบัติความทนทาน

การทวนสอบคุณสมบัติความทนทานใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารอีเบอร์เจ็น แสดงในภาคผนวก ก.3.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความทนทานตามหัวข้อ 4.4.4 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทานซึ่งแสดงในภาคผนวก ก.3.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-42 ซึ่งสปินไม่พบความผิดพลาดแสดงว่าวงจรอสมวารอีเบอร์เจ็น มีคุณสมบัติความทนทาน

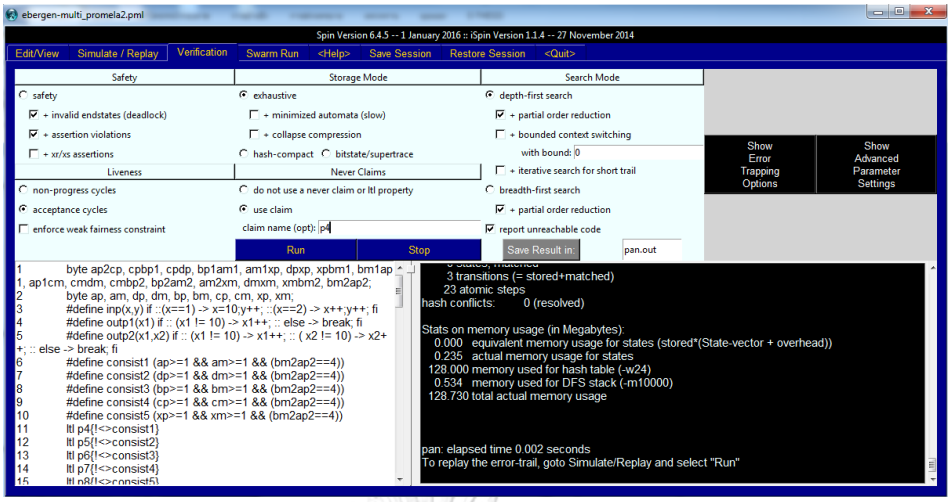



รูปที่ 5-42 ผลการทวนสอบคุณสมบัติความทนทานจากวงจรอสมวารอีเบอร์เจ็น

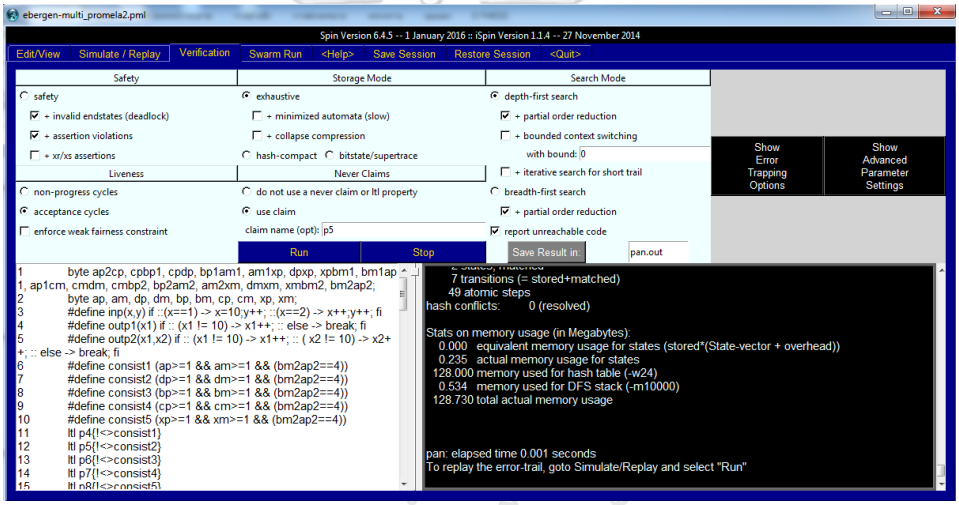


4) การทวนสอบคุณสมบัติความต้องกัน

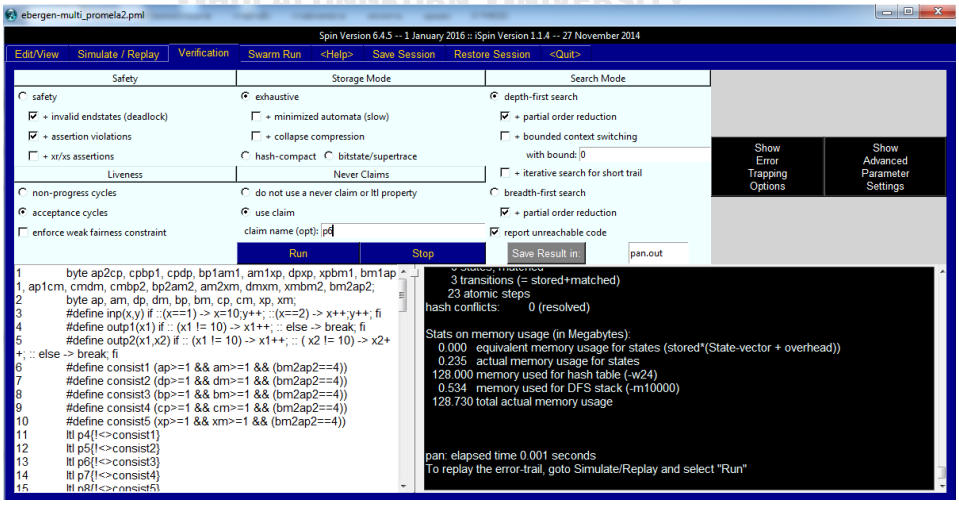
การทวนสอบคุณสมบัติความต้องกันใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารอีเบอร์เจ็น แสดงในภาคผนวก ก.3.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความต้องกันตามหัวข้อ 4.4.5 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันซึ่งแสดงในภาคผนวก ก.3.4 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-43 ถึงรูปที่ 5-47 ซึ่งสปินพบความผิดพลาดของทุกสัญญาณแสดงว่าวงจรอสมวารอีเบอร์เจ็น มีคุณสมบัติความต้องกัน



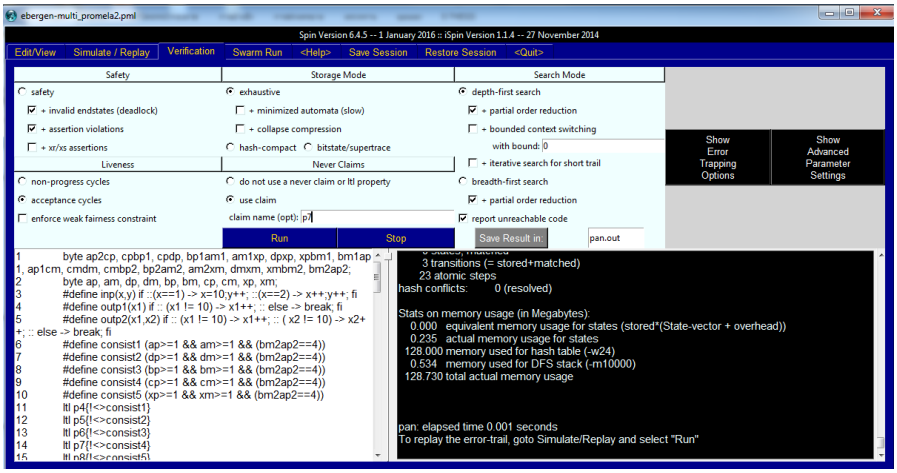

รูปที่ 5-43 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ a



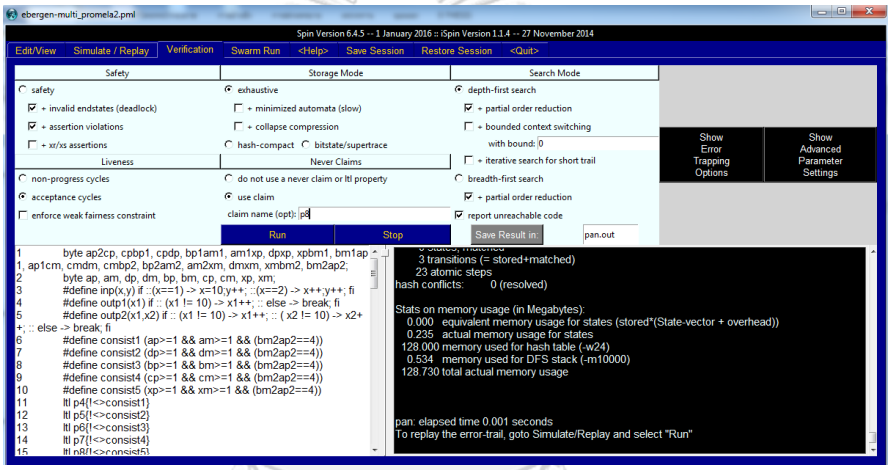

รูปที่ 5-44 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ d




รูปที่ 5-45 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ b




รูปที่ 5-46 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ c

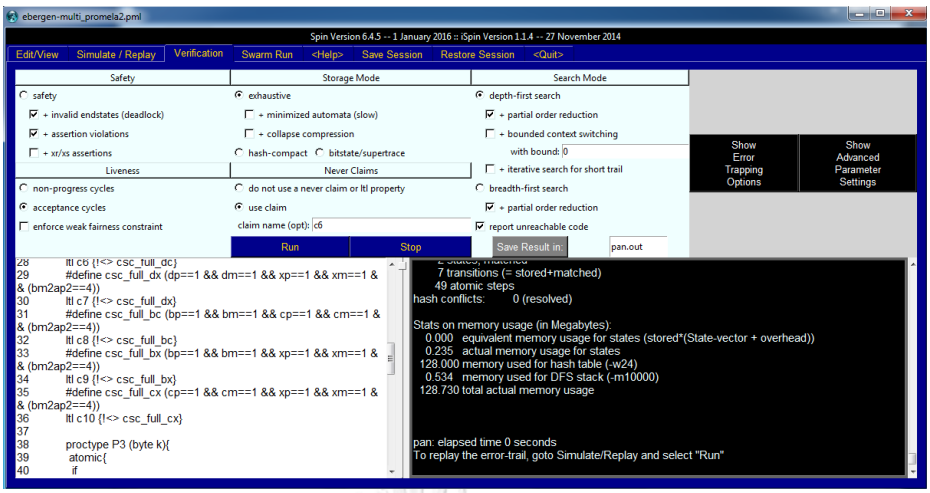



รูปที่ 5-47 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ x

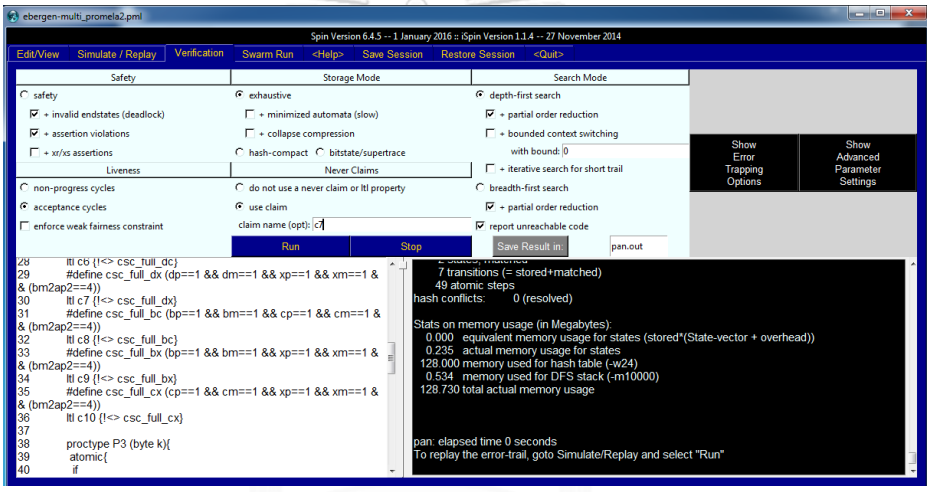


5) การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์

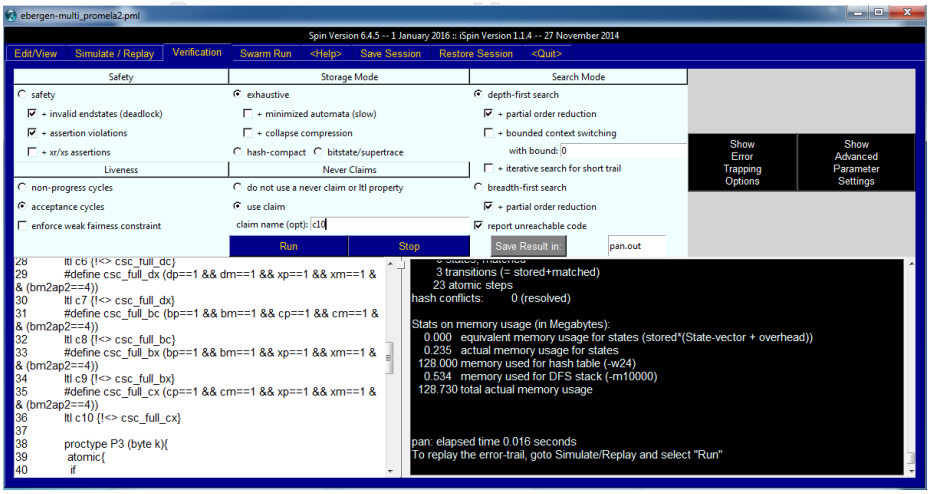
การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารอีเบอร์เจ็น แสดงในภาคผนวก ก.3.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ตามหัวข้อ 4.4.6 จะได้ตรรกะเวลาเชิงเส้นของฟูลล็อคซึ่งแสดงในภาคผนวก ก.3.5 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินพบว่าจากคู่สัญญาณทั้งหมดทวนสอบพบความผิดพลาดดังรูปที่ 5-48 ถึงรูปที่ 5-50 ซึ่งเครื่องมือสปินพบความผิดพลาดจึงนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น




รูปที่ 5-48 ผลการทวนสอบจากเครื่องมือสปินของคู่สัญญาณ d, c

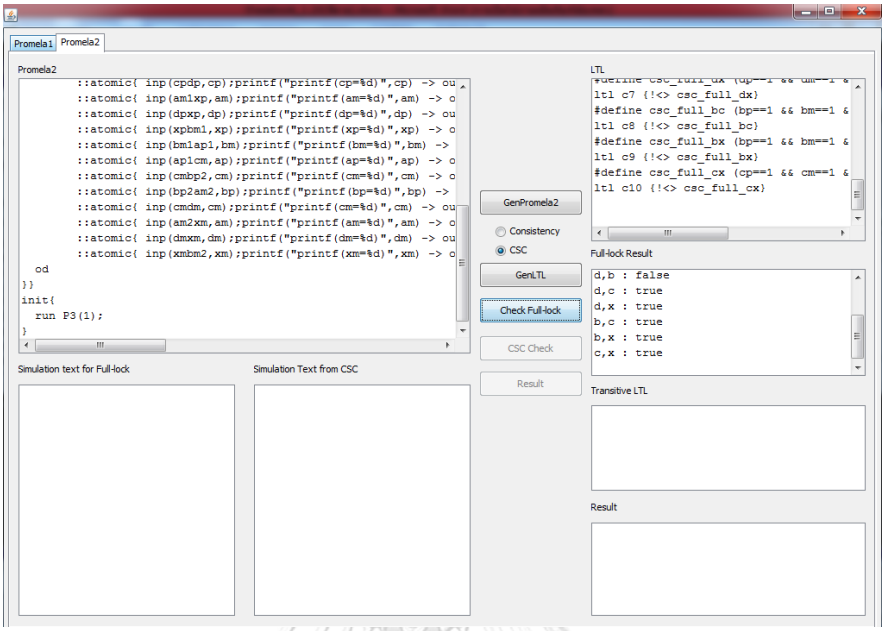



รูปที่ 5-49 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ d, x



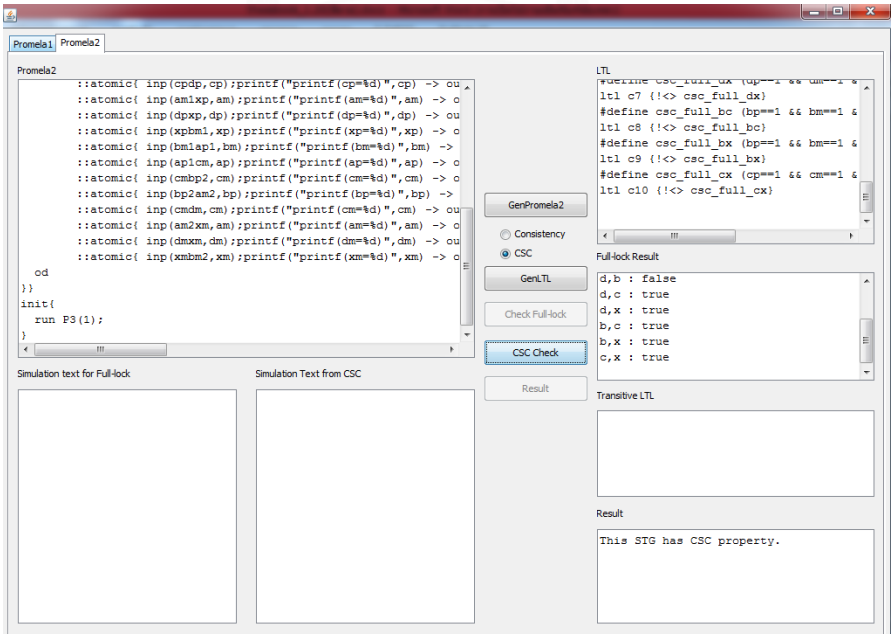

รูปที่ 5-50 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ c, x

เมื่อนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น พบคู่สัญญาณเป็นฟูลล็อค ดังแสดงในรูปที่ 5-51




รูปที่ 5-51 ผลการตรวจสอบฟูลล็อคจากเครื่องมือที่พัฒนาขึ้น

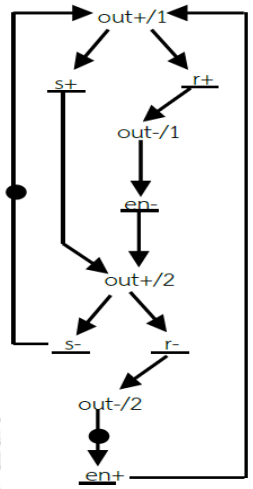
จากนั้นนำคู่สัญญาณเป็นฟูลล็อคตรวจสอบว่าพบแทรนซิทีฟล็อคจากวัฏจักรที่มีจุดยอดไม่ซ้ำกันที่พบหรือไม่โดยเครื่องมือที่พัฒนาขึ้น ซึ่งเครื่องมือแสดงผลว่า พบแทรนซิทีฟล็อค ดังแสดงในรูปที่ 5-52 จึงสามารถสรุปได้ว่าวงจรอสมวารอีเบอร์เจ็น มีคุณสมบัติการกำหนดสถานะที่สมบูรณ์




รูปที่ 5-52 ผลการทวนสอบแทรนซิทีฟล็อคจากเครื่องมือสปิน

### 5.1.4 กรณีศึกษาที่ 4 การทวนสอบตัวอย่างวงจรอสมวารอินพุต

วงจรอสมวารอินพุต มีซิกแนลแทรนซิชันกราฟดังรูป 5-53




รูปที่ 5-53 ซิกแนลแทรนซิชันกราฟของวงจรอสมวารอินพุต

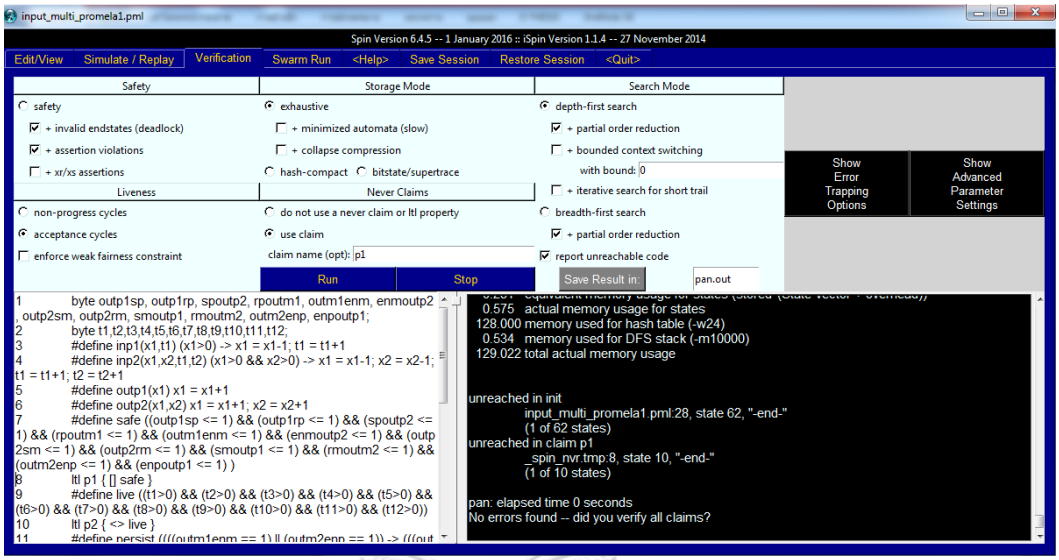
1) การทวนสอบคุณสมบัติความปลอดภัย

การทวนสอบคุณสมบัติความปลอดภัยใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยวงจรอสมวารอินพุต สามารถแสดงเน็ตลิสต์ได้ดังรูปที่ 5-54

| | |
|---|---|
| .model input_multi | en- out+/2 |
| .input s r en | out+/2 s- r- |
| .output out | s- out+/1 |
| .graph | r- out-/2 |
| out+/1 s+ r+ | out-/2 en+ |
| s+ out+/2 | en+ out+/1 |
| r+ out-/1 | .marking {<s-,out+/1>,<out-/2,en+>} |
| out-/1 en- | .end |

รูปที่ 5-54 เน็ตลิสต์ของวงจรอสมวารอินพุต

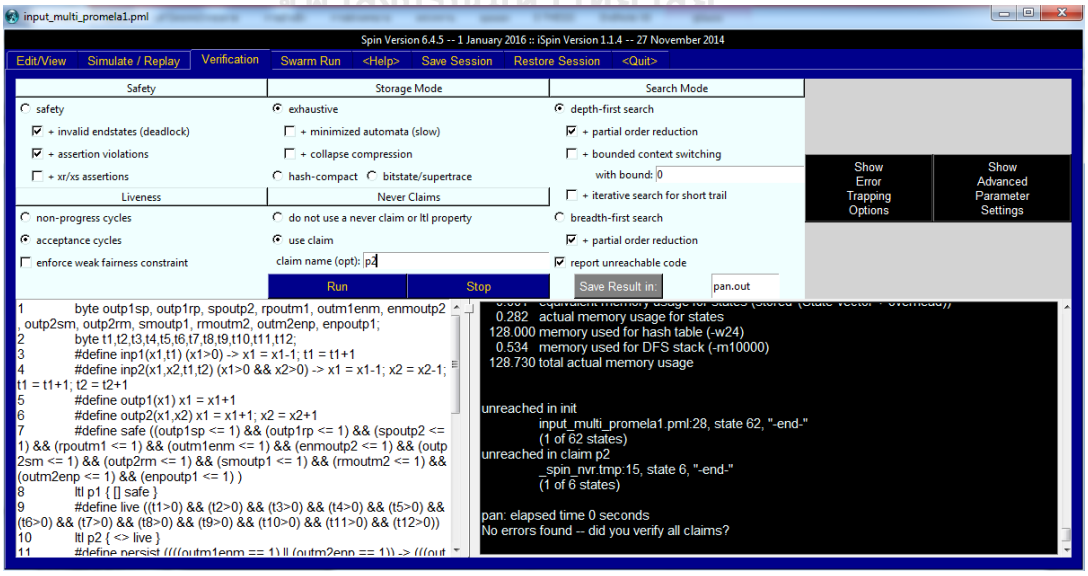
รหัสโพรเมลาแบบ A1 ของวงจรอสมวารอินพุต แสดงในภาคผนวก ก.4.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความปลอดภัยตามหัวข้อ 4.4.2 ได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัยซึ่งแสดงในภาคผนวก ก.4.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-55 ซึ่งสปินไม่พบความผิดพลาดแสดงว่าวงจรอสมวารอินพุต มีคุณสมบัติความปลอดภัย




รูปที่ 5-55 ผลการทวนสอบคุณสมบัติความปลอดภัยจากวงจรอสมวารอินพุต

2) การทวนสอบคุณสมบัติไลฟ์เนส

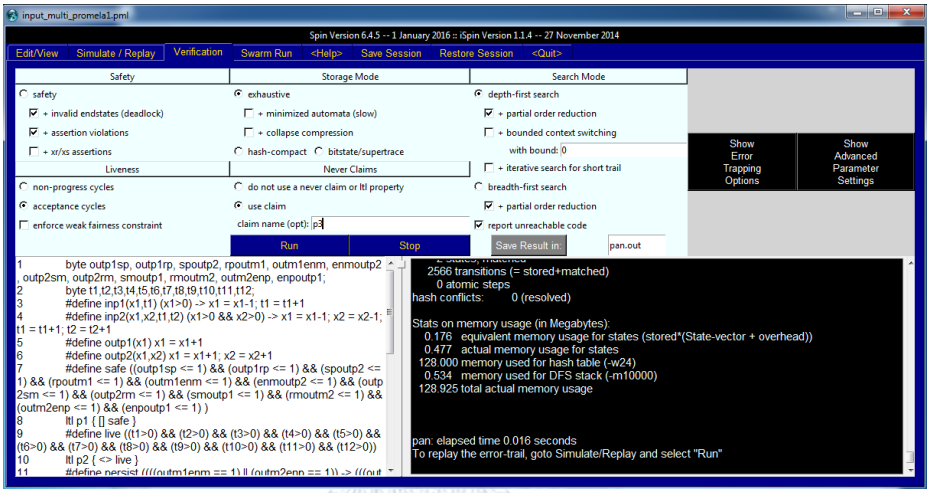
การทวนสอบคุณสมบัติไลฟ์เนสใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบโดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวาร อินพุต แสดงในภาคผนวก ก.4.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติไลฟ์เนสตามหัวข้อ 4.4.3 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนสซึ่งแสดงในภาคผนวก ก.4.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-56 ซึ่งสปินไม่พบความผิดพลาด แสดงว่าวงจรอสมวารอินพุต มีคุณสมบัติไลฟ์เนส




รูปที่ 5-56 ผลการทวนสอบคุณสมบัติความไลฟ์เนสจากวงจรอสมวารอินพุต

3) การทวนสอบคุณสมบัติความทนทาน

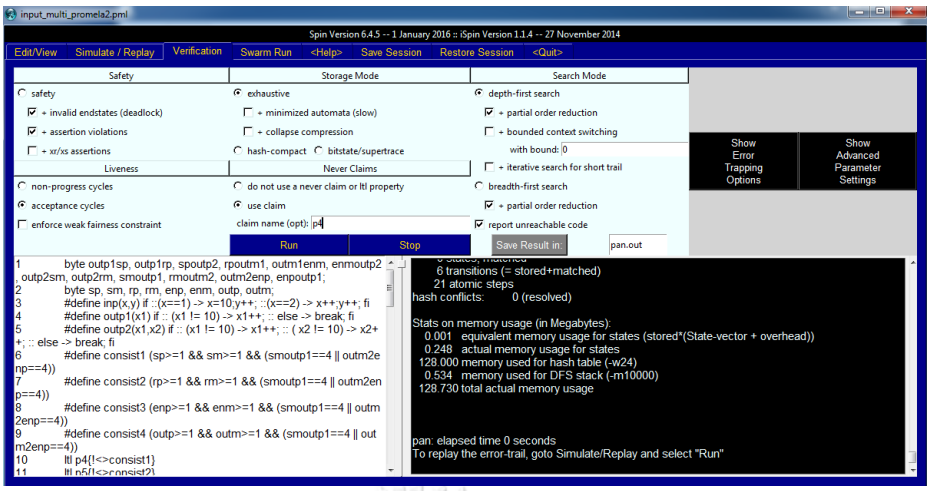
การทวนสอบคุณสมบัติความทนทานใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวาร อินพุต แสดงในภาคผนวก ก.4.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความทนทานตามหัวข้อ 4.4.4 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทานซึ่งแสดงในภาคผนวก ก.4.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-57 ซึ่งสปินพบความผิดพลาดแสดงว่าวงจรอสมวารอินพุต ไม่มีคุณสมบัติความทนทาน



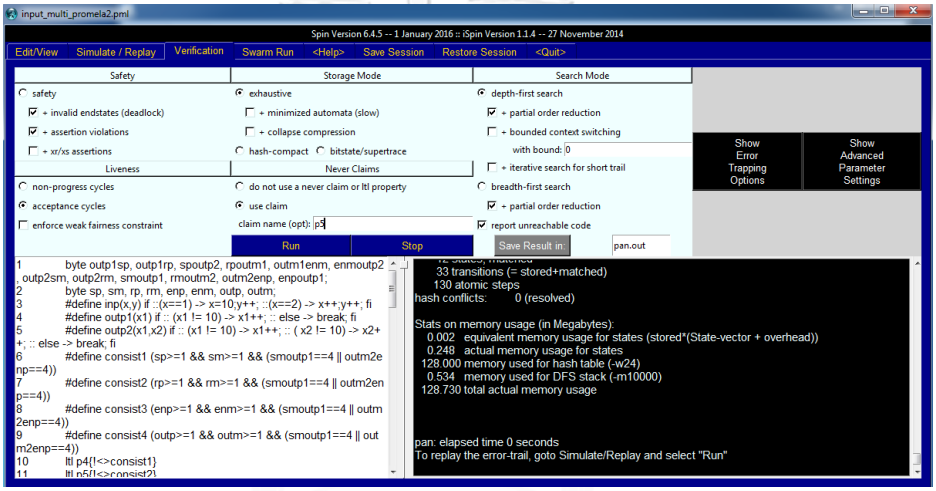

รูปที่ 5-57 ผลการทวนสอบคุณสมบัติความทนทานจากวงจรอสมวารอินพุต

4) การทวนสอบคุณสมบัติความต้องกัน

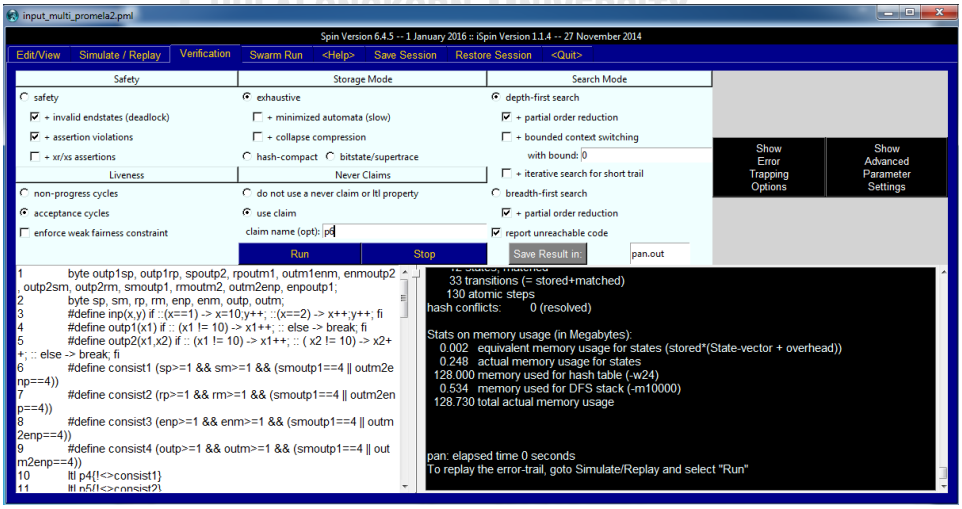
การทวนสอบคุณสมบัติความต้องกันใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารอินพุต แสดงในภาคผนวก ก.4.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความต้องกันตามหัวข้อ 4.4.5 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันซึ่งแสดงในภาคผนวก ก.4.4 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-58 ถึงรูปที่ 5-61 ซึ่งสปินพบความผิดพลาดของทุกสัญญาณแสดงว่าวงจรอสมวารอินพุต มีคุณสมบัติความต้องกัน



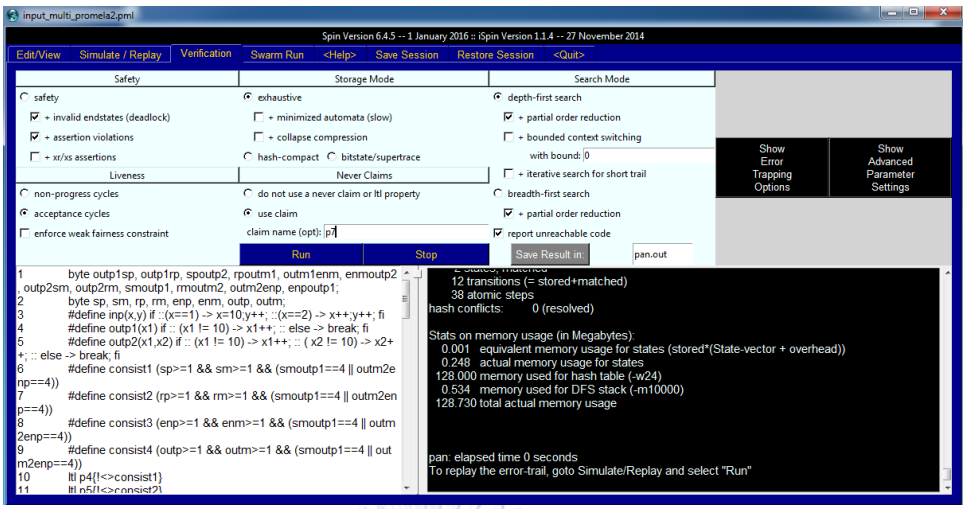

รูปที่ 5-58 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ s




รูปที่ 5-59 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ r




รูปที่ 5-60 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ en

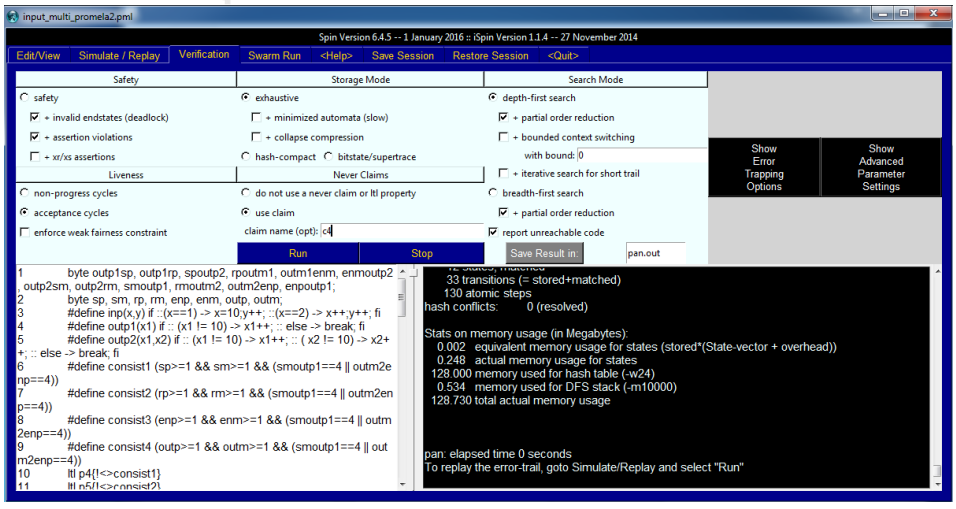



รูปที่ 5-61 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ out

5) การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์

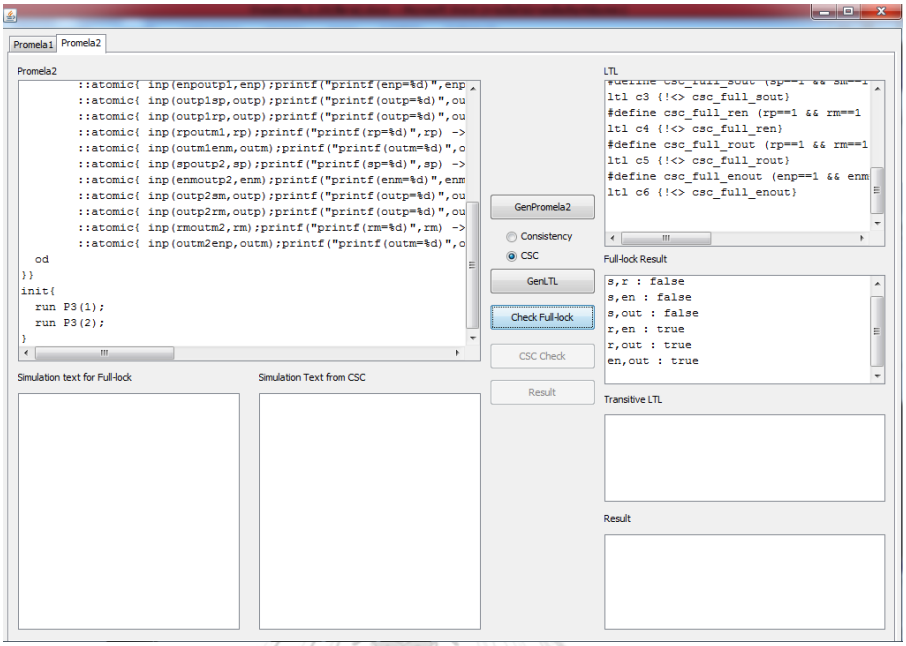
การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารอินพุต แสดงในภาคผนวก ก.4.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ตามหัวข้อ 4.4.6 จะได้ตรรกะเวลาเชิงเส้นของฟูลล็อคซึ่งแสดงในภาคผนวก ก.4.5 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินพบว่าจากคู่สัญญาณทั้งหมดทวนสอบพบความผิดพลาดดังรูปที่ 5-62 ซึ่งเครื่องมือสปินพบความผิดพลาดจึงนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น





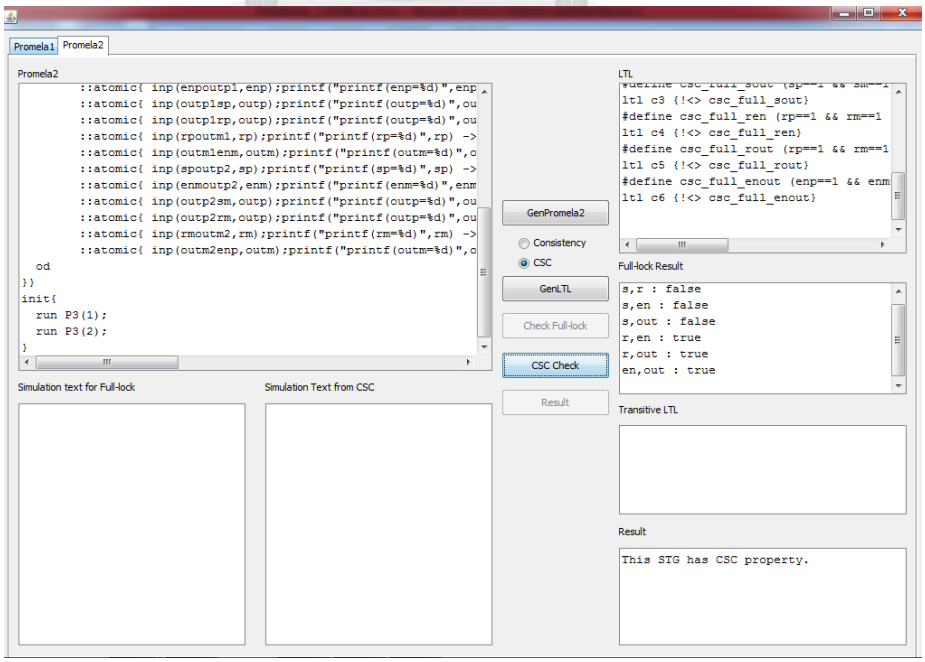

รูปที่ 5-62 ผลการทวนสอบจากเครื่องมือสปินของคู่สัญญาณ r, en

เมื่อนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น พบคู่สัญญาณฟูลล็อค ดังแสดงในรูปที่ 5-63




รูปที่ 5-63 ผลการตรวจสอบฟูลล็อคจากเครื่องมือที่พัฒนาขึ้น

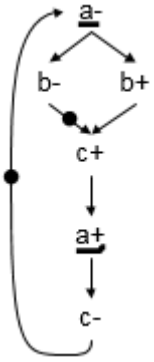
จากนั้นจึงตรวจสอบว่าพบแทรนซิทีฟล็อคจากวัฏจักรที่มีจุดยอดไม่ซ้ำกันที่พบหรือไม่โดยเครื่องมือที่พัฒนาขึ้น ซึ่งเครื่องมือแสดงผลว่า พบแทรนซิทีฟล็อค ดังแสดงในรูปที่ 5-64 จึงสามารถสรุปได้ว่าวงจรอสมวารอินพุต มีคุณสมบัติการกำหนดสถานะที่สมบูรณ์




รูปที่ 5-64 ผลการทวนสอบแทรนซิทีฟล็อคจากเครื่องมือที่พัฒนาขึ้น

### 5.1.5 กรณีศึกษาที่ 5 การทวนสอบตัวอย่างวงจรอสมวารทดลอง 1

วงจรอสมวารทดลอง 1 มีซิกแนลแทรนซิชันกราฟดังรูป 5-65




รูปที่ 5-65 ซิกแนลแทรนซิชันกราฟของวงจรอสมวารทดลอง 1

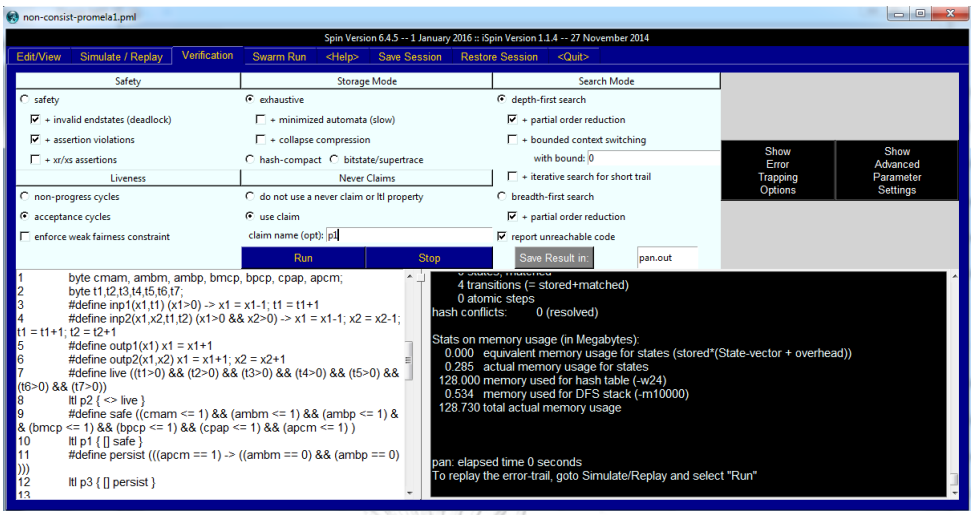
1) การทวนสอบคุณสมบัติความปลอดภัย

คุณสมบัติความปลอดภัยใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยวงจรอสมวารทดลอง 1 สามารถแสดงเน็ตลิสต์ได้ดังรูปที่ 5-66

| | |
|---|---|
| .model non-consist | b+ c+ |
| .input a | c+ a+ |
| .output b c | a+ c- |
| .graph | .marking {<c-,a->,<b-,c+>} |
| c- a- | .end |
| a- b- b+ | |
| b- c+ | |

รูปที่ 5-66 เน็ตลิสต์ของวงจรอสมวารทดลอง 1

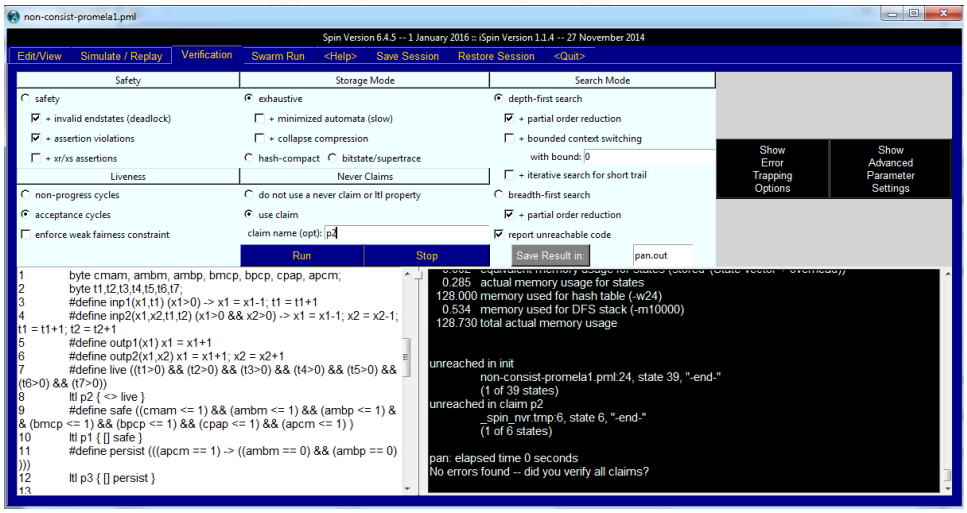
รหัสโพรเมลาแบบ A1 ของวงจรอสมวารทดลอง 1 แสดงในภาคผนวก ก.5.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความปลอดภัยตามหัวข้อ 4.4.2 ได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัยซึ่งแสดงในภาคผนวก ก.5.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-67 ซึ่งสปินพบความผิดพลาด แสดงว่าวงจรอสมวารทดลอง 1 ไม่มีคุณสมบัติความปลอดภัย




รูปที่ 5-67 ผลการทวนสอบคุณสมบัติความปลอดภัยจากวงจรอสมวารทดลอง 1

2) การทวนสอบคุณสมบัติไลฟ์เนส

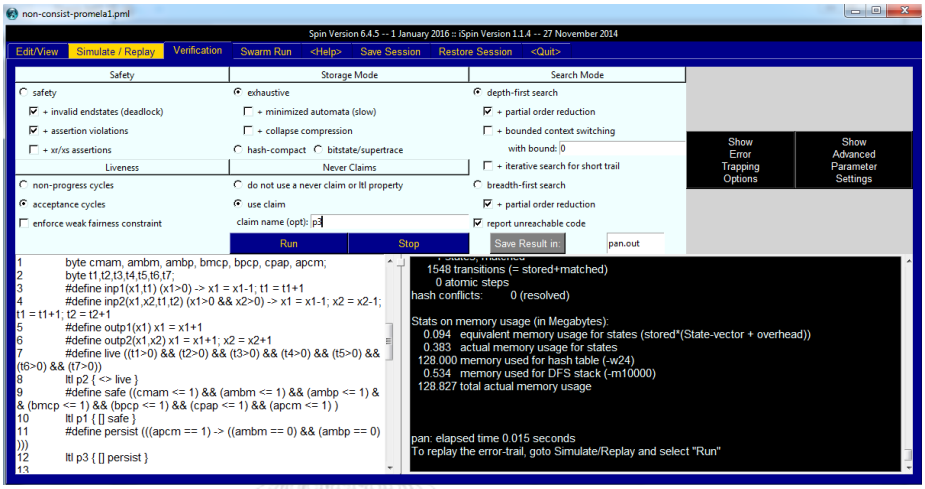
การทวนสอบคุณสมบัติไลฟ์เนสใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบโดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารทดลอง 1 แสดงในภาคผนวก ก.5.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติไลฟ์เนสตามหัวข้อ 4.4.3 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนสซึ่งแสดงในภาคผนวก ก.5.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-68 ซึ่งสปินไม่พบความผิดพลาด แสดงว่าวงจรอสมวารทดลอง 1 มีคุณสมบัติไลฟ์เนส




รูปที่ 5-68 ผลการทวนสอบคุณสมบัติความไลฟ์เนสจากวงจรอสมวารทดลอง 1

3) การทวนสอบคุณสมบัติความทนทาน

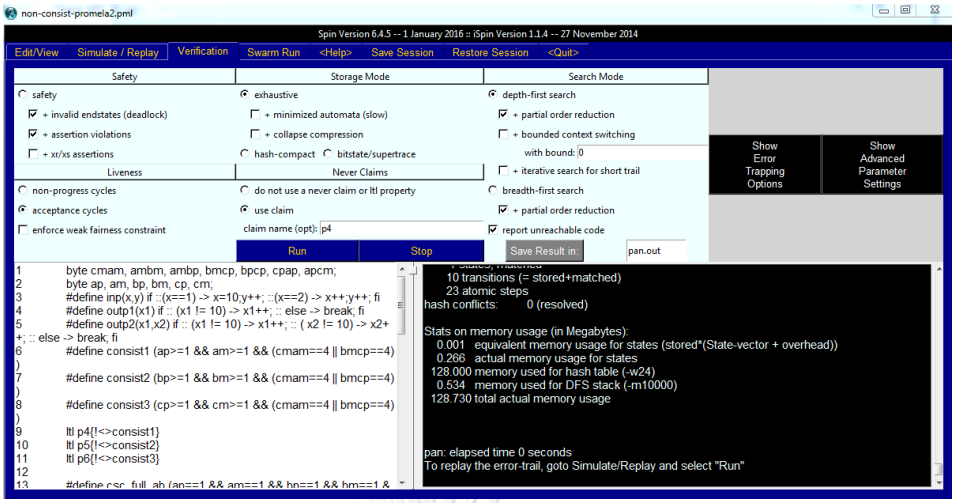
การทวนสอบคุณสมบัติความทนทานใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารทดลอง 1 แสดงในภาคผนวก ก.5.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความทนทานตามหัวข้อ 4.4.4 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทานซึ่งแสดงในภาคผนวก ก.5.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-69 ซึ่งสปินพบความผิดพลาดแสดงว่าวงจรอสมวารทดลอง 1 ไม่มีคุณสมบัติความทนทาน



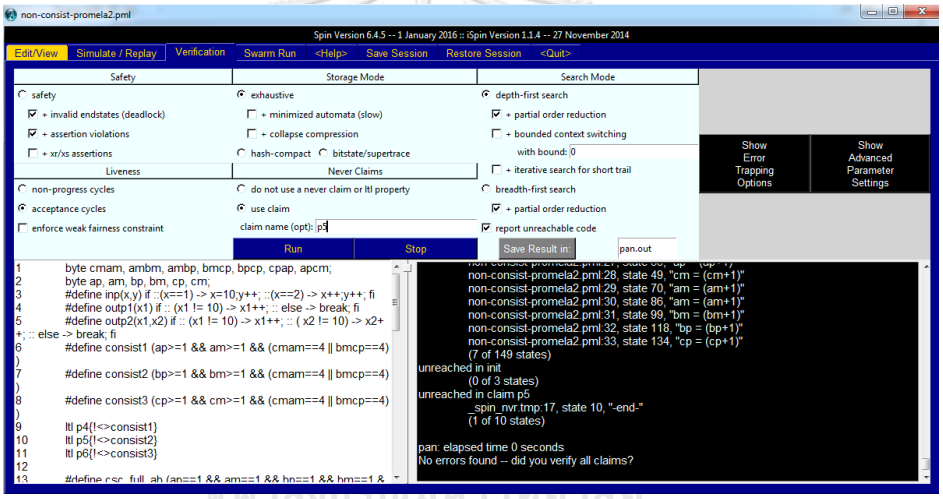

รูปที่ 5-69 ผลการทวนสอบคุณสมบัติความทนทานจากวงจรอสมวารทดลอง 1

4) การทวนสอบคุณสมบัติความต้องกัน

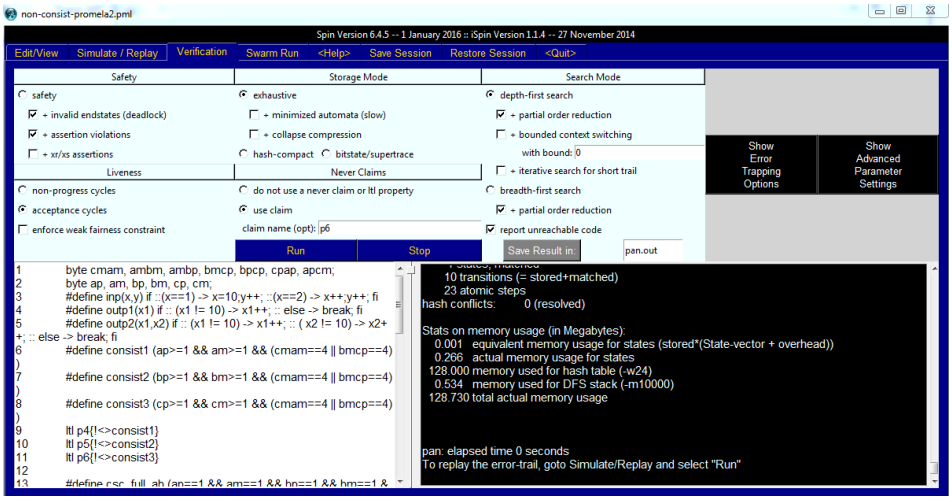
การทวนสอบคุณสมบัติความต้องกันใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารทดลอง 1 แสดงในภาคผนวก ก.5.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความต้องกันตามหัวข้อ 4.4.5 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความต้องกันซึ่งแสดงในภาคผนวก ก.5.4 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-70 ถึงรูปที่ 5-72 ซึ่งสปินไม่พบความผิดพลาดของสัญญาณ b แสดงว่าวงจรอสมวารทดลอง 1 ไม่มีคุณสมบัติความต้องกัน




รูปที่ 5-70 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ a



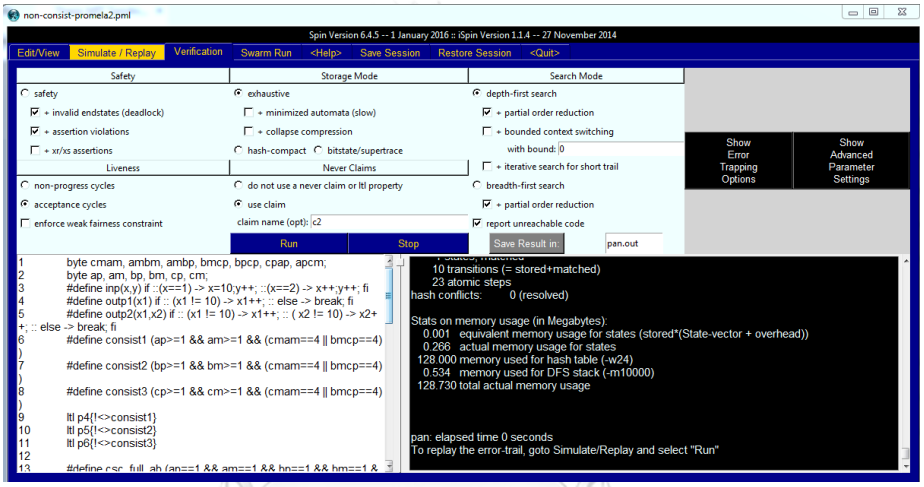

รูปที่ 5-71 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ b




รูปที่ 5-72 ผลการทวนสอบจากเครื่องมือสปินของสัญญาณ c

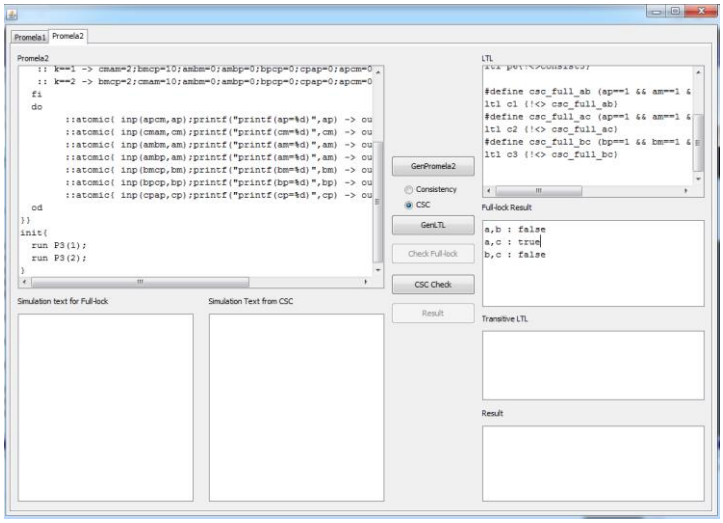
5) การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์

การทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ใช้รหัสโพรเมลาแบบ A2 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A2 ของวงจรอสมวารทดลอง 1 แสดงในภาคผนวก ก.5.3 เมื่อได้รหัสโพรเมลาแบบ A2 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติการกำหนดสถานะที่สมบูรณ์ตามหัวข้อ 4.4.6 จะได้ตรรกะเวลาเชิงเส้นของฟูลล็อคซึ่งแสดงในภาคผนวก ก.5.5 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปิน พบว่าจากคู่สัญญาณทั้งหมดทวนสอบพบความผิดพลาดดังรูปที่ 5-73 ซึ่งเครื่องมือสปินพบความผิดพลาดจึงนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น



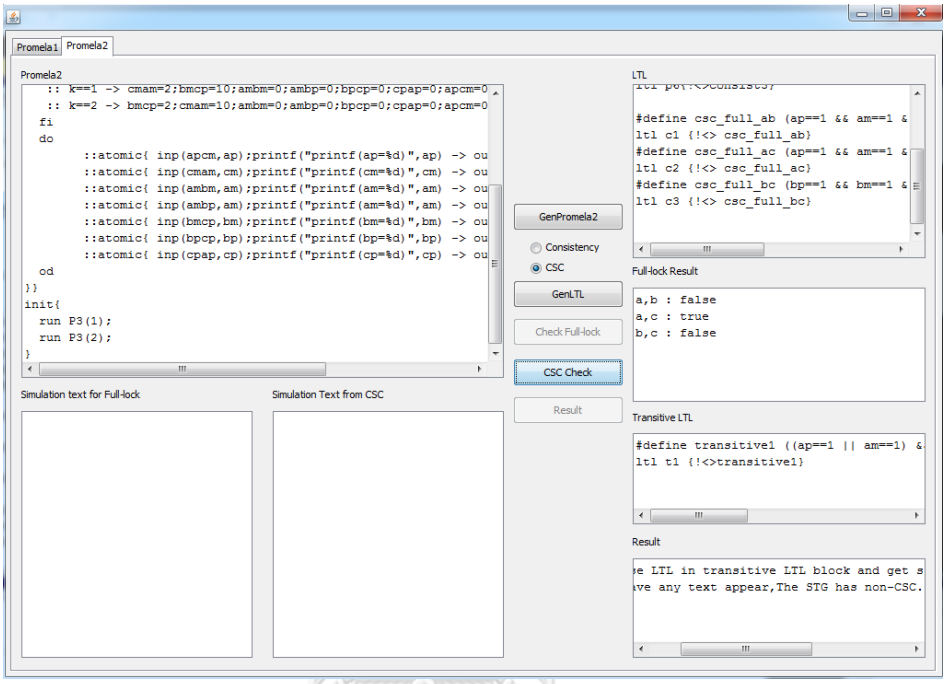

รูปที่ 5-73 ผลการทวนสอบจากเครื่องมือสปินของคู่สัญญาณ a, c

เมื่อนำผลการจำลองมาตรวจสอบฟูลล็อคในเครื่องมือที่พัฒนาขึ้น พบคู่สัญญาณฟูลล็อคดังแสดงในรูปที่ 5-74




รูปที่ 5-74 ผลการตรวจสอบฟูลล็อคจากเครื่องมือที่พัฒนาขึ้น

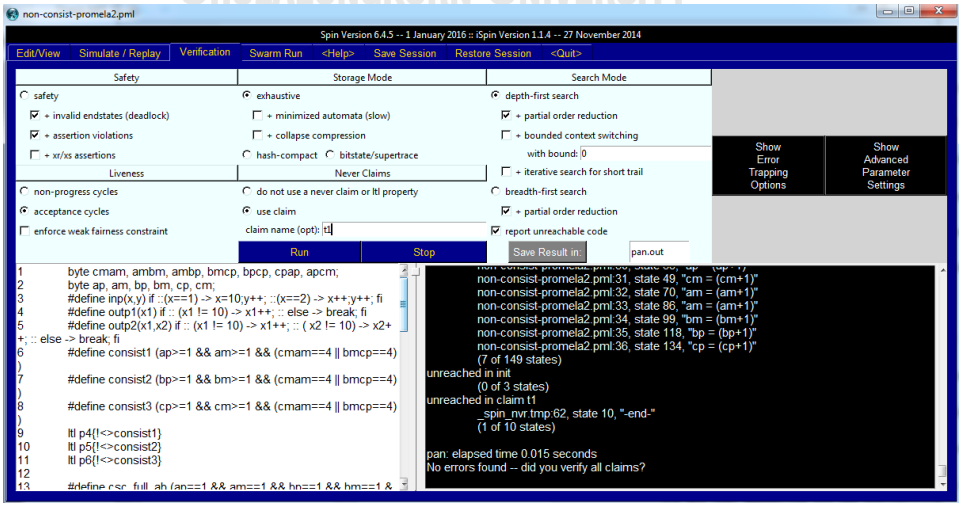
จากนั้นจึงตรวจสอบว่าพบแทรนซิทีฟล็อคจากวัฏจักรที่มีจุดยอดไม่ซ้ำกันที่พบหรือไม่โดยเครื่องมือที่พัฒนาขึ้น ซึ่งเครื่องมือแสดงผลว่า ไม่พบแทรนซิทีฟล็อคเครื่องมือจึงสร้างตรรกะเวลาเชิงเส้นเพื่อการตรวจหาแทรนซิทีฟล็อค ดังแสดงในรูปที่ 5-75




รูปที่ 5-75 ผลการทวนสอบแทรนซิทีฟล็อคจากเครื่องมือที่พัฒนาขึ้น

จากนั้นนำแบบจำลองและตรรกะเวลาเชิงเส้นของแทรนซิทีฟล็อคที่ได้จากเครื่องมือมาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-76 ซึ่งสปินไม่พบความผิดพลาด แสดงว่าวงจรอสมวารทดลอง 1 ไม่มีคุณสมบัติการกำหนดสถานะที่สมบูรณ์

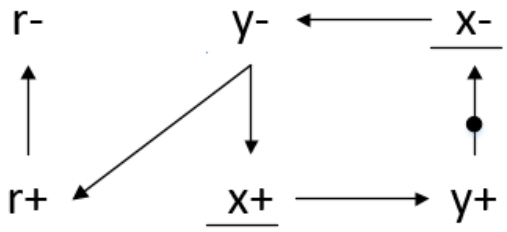





รูปที่ 5-76 ผลการทวนสอบแทรนซิทีฟล็อคจากเครื่องมือสปิน

### 5.1.6 กรณีศึกษาที่ 6 การทวนสอบตัวอย่างวงจรอสมวารทดลอง 2

วงจรอสมวารทดลอง 2 มีซิกแนลแทรนซิชันกราฟดังรูป 5-77




รูปที่ 5-77 ซิกแนลแทรนซิชันกราฟของวงจรอสมวารทดลอง 2

1) การทวนสอบคุณสมบัติความปลอดภัย

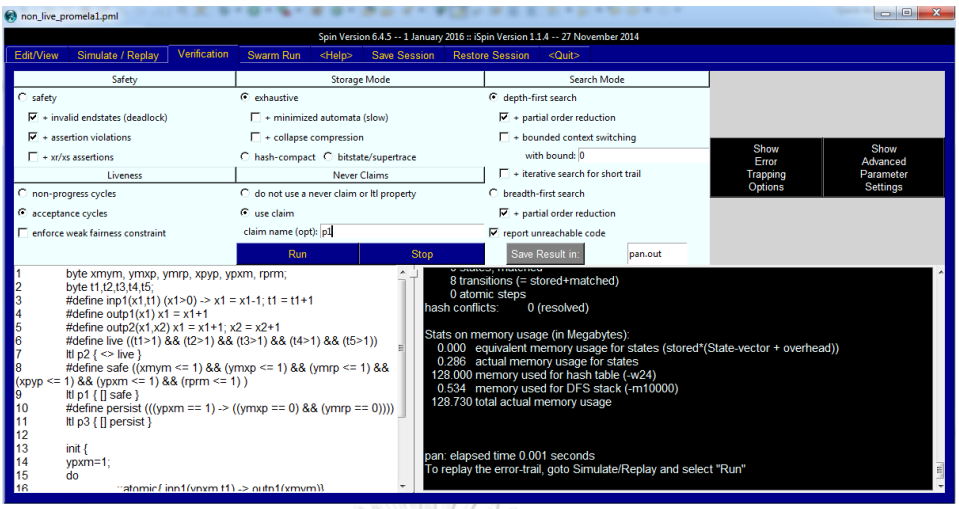
คุณสมบัติความปลอดภัยใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยวงจรอสมวารทดลอง 2 สามารถแสดงเน็ตลิสต์ได้ดังรูปที่ 5-78

```
.model non_live                 x+ y+
.input x                        y+ x-
.output y r                     r+ r-
.graph                          .marking {<y+,x->}
x- y-                           .end
y- x+ r+
```

รูปที่ 5-78 เน็ตลิสต์ของวงจรอสมวารทดลอง 2



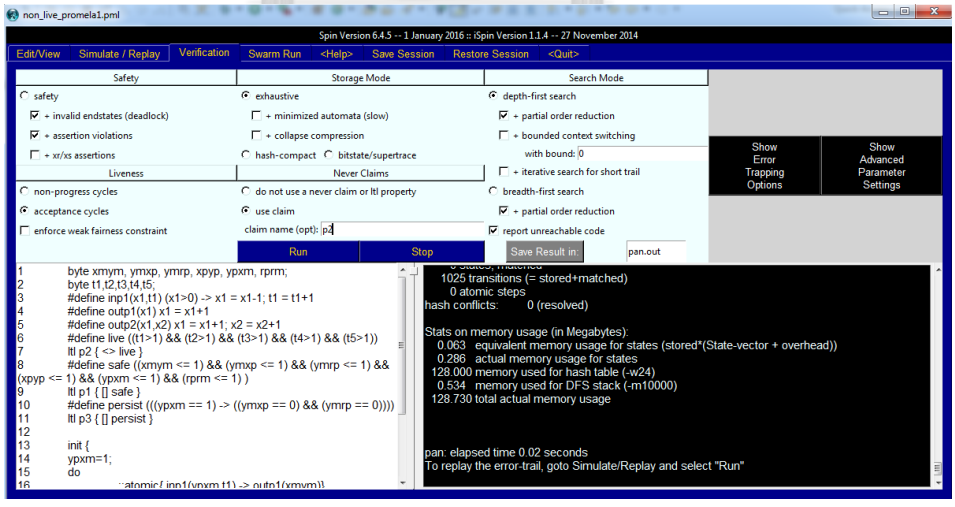
รหัสโพรเมลาแบบ A1 ของวงจรอสมวารทดลอง 2 แสดงในภาคผนวก ก.6.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความปลอดภัยตามหัวข้อ 4.4.2 ได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความปลอดภัยซึ่งแสดงในภาคผนวก ก.6.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-79 ซึ่งสปินพบความผิดพลาดแสดงว่าวงจรอสมวารทดลอง 2 ไม่มีคุณสมบัติความปลอดภัย




รูปที่ 5-79 ผลการทวนสอบคุณสมบัติความปลอดภัยจากวงจรอสมวารทดลอง 2

2) การทวนสอบคุณสมบัติไลฟ์เนส

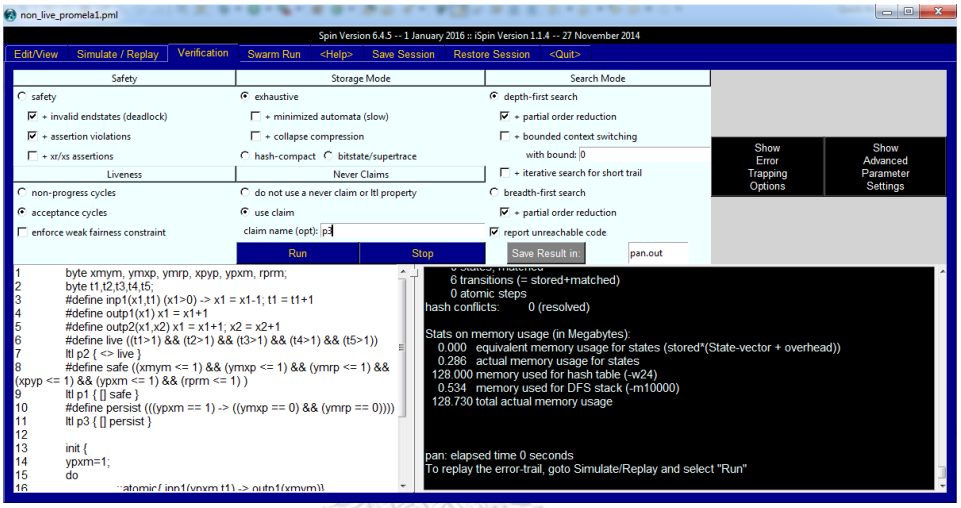
การทวนสอบคุณสมบัติไลฟ์เนสใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบโดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารทดลอง 2 แสดงในภาคผนวก ก.6.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติไลฟ์เนสตามหัวข้อ 4.4.3 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติไลฟ์เนสซึ่งแสดงในภาคผนวก ก.5.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-80 ซึ่งสปินพบความผิดพลาด แสดงว่าวงจรอสมวารทดลอง 2 ไม่มีคุณสมบัติไลฟ์เนส




รูปที่ 5-80 ผลการทวนสอบคุณสมบัติความไลฟ์เนสจากวงจรอสมวารทดลอง 2

3) การทวนสอบคุณสมบัติความทนทาน

การทวนสอบคุณสมบัติความทนทานใช้รหัสโพรเมลาแบบ A1 เป็นแบบจำลองในการทวนสอบ โดยรหัสโพรเมลาแบบ A1 ของวงจรอสมวารทดลอง 2 แสดงในภาคผนวก ก.6.1 เมื่อได้รหัสโพรเมลาแบบ A1 แล้วจึงนำรหัสที่ได้มาสร้างตรรกะเวลาเชิงเส้นสำหรับการทวนสอบคุณสมบัติความทนทานตามหัวข้อ 4.4.4 จะได้ตรรกะเวลาเชิงเส้นของคุณสมบัติความทนทานซึ่งแสดงในภาคผนวก ก.5.2 จากนั้นจึงนำรหัสทั้งสองส่วนที่ได้มาทวนสอบในเครื่องมือสปินได้ผลดังรูปที่ 5-81 ซึ่งสปินพบความผิดพลาดแสดงว่า วงจรอสมวารทดลอง 2 ไม่มีคุณสมบัติความทนทาน




รูปที่ 5-81 ผลการทวนสอบคุณสมบัติความทนทานจากวงจรอสมวารทดลอง 2

## 5.2 ผลการทวนสอบ



จากการทวนสอบคุณสมบัติของซิกแนลแทรนซิชันกราฟของวงจรอสมวารทั้ง 6 กรณี สามารถสรุปได้ว่าวงจรอสมวารทั้ง 6 กรณีมีคุณสมบัติคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์ หรือไท่ ดังแสดงในตารางที่ 5-1 และสามารถสรุปได้ว่าวิธีการทวนสอบนี้สามารถให้ผลที่ถูกต้อง

ตารางที่ 5-1 ตารางสรุปผลการทวนสอบกรณีศึกษาทั้ง 6 กรณี

| **ชื่อวงจร** | **คุณสมบัติความปลอดภัย** | **คุณสมบัติไลฟ์เนส** | **คุณสมบัติความทนทาน** | **คุณสมบัติความต้องกัน** | **คุณสมบัติการกำหนดสถานะที่สมบูรณ์** |
|---|---|---|---|---|---|
| ฟูล | มี | มี | มี | มี | มี |
| ไทรมอสเซนท์ | มี | มี | ไม่มี | มี | ไม่มี |
| อีเบอร์เจ็น | มี | มี | มี | มี | มี |
| อินพุต | มี | มี | ไม่มี | มี | มี |
| ทดลอง 1 | ไม่มี | มี | ไม่มี | ไม่มี | ไม่มี |
| ทดลอง 2 | ไม่มี | ไม่มี | ไม่มี | - | - |

# บทที่ 6
# สรุปผลการวิจัยและข้อเสนอแนะ

จากการวิเคราะห์ ศึกษา วิจัยและพัฒนาวิธีการทวนสอบซิกแนลแทรนซิชันกราฟและคุณสมบัติซึ่งประกอบด้วย คุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์ สามารถสรุปผลการวิจัย ข้อจำกัดของวิธีการและแนวทางในการพัฒนาต่อในอนาคต โดยมีรายละเอียดดังต่อไปนี้

## 6.1 สรุปผลการวิจัย

งานวิจัยนี้เป็นการนำเสนอการสร้างแบบจำลองของซิกแนลแทรนซิชันกราฟด้วยรหัสโพรเมลา และกำหนดคุณสมบัติต่างๆโดยตรรกะเวลาเชิงเส้น และนำทั้งแบบจำลองโพรเมลาและตรรกะเวลาเชิงเส้นไปทวนสอบกับเครื่องมือสปินโดยประกอบด้วย 5 คุณสมบัติ คือ คุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์ ในขั้นแรกซิกแนลแทรนซิชันกราฟจะถูกนำมาแปลงเป็นรหัสโพรเมลาแบบ A1 และแบบ A2 จากนั้นซิกแนลแทรนซิชันกราฟจะถูกนำมาแปลงเป็นตรรกะเวลาเชิงเส้นของทั้ง 5 คุณสมบัติซึ่งประกอบด้วย คุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส คุณสมบัติความทนทาน คุณสมบัติความต้องกัน และคุณสมบัติการกำหนดสถานะที่สมบูรณ์ โดยอ้างอิงจากรหัสโพรเมลาแบบ A1 และ A2 จากนั้นรหัสโพรเมลาและตรรกะเวลาเชิงเส้นจะถูกนำมาทวนสอบในเครื่องมือสปินเพื่อทวนสอบและสรุปผล

การสร้างแบบจำลองด้วยโพรเมลาและตรรกะเวลาเชิงเส้นเพื่อทวนสอบซิกแนลแทรนซิชันกราฟในการสร้างวงจรอสมวารที่ได้นำเสนอในงานวิจัยนี้ เมื่อนำมาทดสอบกับตัวอย่างซิกแนลแทรนซิชันกราฟของวงจรอสมวารต่างๆ ผลลัพธ์ที่ได้มีความถูกต้องและสอดคล้องกัน



## 6.2 ข้อจำกัดของงานวิจัย

1) วิธีการที่เสนอนี้ไม่สามารถใช้กับวัฏจักรแบบตัวเลือกอิสระ (Free choice) ได้
2) วิธีการนี้เสนอนี้ไม่สามารถใช้กับซิกแนลแทรนซิชันกราฟประเภทที่นำเวลามาเกี่ยวข้องได้
3) เครื่องมือที่พัฒนาขึ้นนั้นไม่สามารถนำเข้าข้อมูลในลักษณะรูปภาพได้
4) การทวนสอบคุณสมบัติความต้องกันและคุณสมบัติการกำหนดสถานะที่สมบูรณ์มีขั้นตอนหลายขั้นตอนและต้องทำงานร่วมกับเครื่องมือที่พัฒนาขึ้น

### 6.3 ข้อเสนอแนะและแนวทางการดำเนินงานต่อ

ขั้นตอนวิธีการทวนสอบและเครื่องมือที่พัฒนาขึ้นในงานวิจัยนี้ยังมีส่วนที่ไม่เป็นอัตโนมัติและมีหลายขั้นตอนในการทวนสอบ นอกจากนี้ยังไม่ครอบคลุมซิกแนลแทรนซิชันกราฟแบบตัวเลือกอิสระและซิกแนลแทรนซิชันกราฟที่มีเวลามาเกี่ยวข้อง ควรมีการพัฒนาเพื่อให้เครื่องมือเป็นอัตโนมัติสำหรับทุกคุณสมบัติและขยายสู่ซิกแนลแทรนซิชันกราฟแบบตัวเลือกอิสระและประเภทที่มีเวลามาเกี่ยวข้อง นอกจากนี้ประสิทธิภาพในการสร้างแบบจำลองด้วยรหัสโพรเมลานั้นควรมีการวิเคราะห์ปรับปรุงให้ดีขึ้น ดังตัวอย่างเช่น จำนวนบรรทัดของรหัสโพรเมลา เป็นต้น

# รายการอ้างอิง


1. Berkel, C.H.V., et al., *Applications of asynchronous circuits.* Proceedings of the IEEE, 1999. 87(2): p. 69-93.
2. Harrison, J., *Formal verification at Intel.* Proceedings of the18th Annual IEEE Symposium of Logic in Computer Science, 2003: p. 45.
3. Chu, T.-A., *Synthesis of self-timed VLSI circuits from graph-theoretic specifications*, in *Electrical Engineering and Computer Science*. June 1987, Massachusetts Institute of Technology.
4. Park, S.B., *Synthesis of Asynchronous VLSI Circuits from Signal Transition Graph Specifications*, in *Department of Engineering-Computer Science*. 1996, Tokyo Institute of Technology.
5. Baier, C. and J.-P. Katoen, *Principles Of Model Checking*. Vol. 950. 2008: MIT Press.
6. Ben-Ari, M., *Principles of the SPIN Model Checker*. 2008.
7. Lawsunnee, W., A. Thongtak, and W. Vatanawood, *Signal persistence checking of asynchronous system implementation using SPIN.* Lecture Notes in Engineering and Computer Science, 2015. 2: p. 604-609.
8. Sbaï, Z. and M. Escheikh, *Model Checking Techniques for Verification of an Encryption Scheme for Wireless Sensor Networks.* Proceeding of the International Conference on Information Processing and Wireless Systems (IPWiS), 2012.
9. Holzmann, G.J. and D. Bošnački, *Multi-core model checking with Spin.* Proceedings - 21st International Parallel and Distributed Processing Symposium, IPDPS 2007; Abstracts and CD-ROM, 2007.
10. Lin, K.-J. and C.-s. Lin, *Automatic Synthesis of Asynchronous Circuits STG and Lock Relation Verification and Rectification of a Feasible STG.* 1991: p. 296-301.
11. Vanbekbergen, P., et al., *Optimized Synthesis of Asynchronous Control Circuits from Graph-Theoretic Specifications.* 1992. 11(1): p. 1426-1438.

12. Seiitovicti, E.M., et al., *Sequential Circuit Design Using Synthesis and Optimization.* 1992.

13. Esparza, J., P. Janˇ, and A. Miller, *On the Complexity of Consistency and Complete State Coding for Signal Transition Graphs.* 2006(1).

14. Esparza, J., *A Polynomial-Time Algorithm for Checking Consistency of Free-Choice Signal Transition Graphs.* 2003.

15. Sudeng, S. and A. Thongtak, *Signal Transition Graph Based Logic Synthesis for Asynchronous Control Circuits Using Template Based Method.* 2007.

16. Cortadella, E.P.J., *Polynomial Algorithms for the Synthesis of Hazard-free Circuits from Signal Transition Graphs.* 1993.

17. Lavagnoj, L., et al., *Complete state encoding based on the theory of regions.* 1996: p. 36-47.

18. Lin, K.-J. and C.-S. Lin, *On the verification of state-coding in STGs.* Computer-Aided Design, 1992. ICCAD-92. Digest of Technical Papers., 1992 IEEE/ACM International Conference on, 1992: p. 118-122.

# ภาคผนวก

# ภาคผนวก ก

## ก.1 รหัสโพรเมลาและตรรกะเวลาเชิงเส้นจากวงจรอสมวารฟูล

### ก.1.1 รหัสโพรเมลาแบบ A1 ของวงจรอสมวารฟูล

```
byte RipAop, AopRop, AopRim, RimAom, AomRip, AomRom, RopAip, RopAom, AipRom, RomAim,
RomAop, AimRop;
byte t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12;
#define inp1(x1,t1)         (x1>0) -> x1 = x1-1; t1 = t1+1
#define inp2(x1,x2,t1,t2)  (x1>0 && x2>0) -> x1 = x1-1; x2 = x2-1; t1 = t1+1; t2 = t2+1
#define outp1(x1)           x1 = x1+1
#define outp2(x1,x2)        x1 = x1+1; x2 = x2+1
init {
AomRip=1;RomAop=1;AimRop=1;
do
::atomic{ inp1(AomRip,t1) -> outp1(RipAop)}
::atomic{ inp2(RipAop,RomAop,t2,t3) -> outp2(AopRop,AopRim)}
::atomic{ inp1(AopRim,t4) -> outp1(RimAom)}
::atomic{ inp2(RimAom,RopAom,t5,t6) -> outp2(AomRip,AomRom)}
::atomic{ inp2(AopRop,AimRop,t7,t8) -> outp2(RopAip,RopAom)}
::atomic{ inp1(RopAip,t9) -> outp1(AipRom)}
::atomic{ inp2(AomRom,AipRom,t10,t11) -> outp2(RomAim,RomAop)}
::atomic{ inp1(RomAim,t12) -> outp1(AimRop)}
od
}
```

รูปที่ ก-1 รหัสโพรเมลาแบบ A1 วงจรอสมวารฟูล จากข้อที่ 5.1.1

### ก.1.2 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส และคุณสมบัติความทนทานวงจรอสมวารฟูล

```
#define safe ((RipAop <= 1) && (AopRop <= 1) && (AopRim <= 1) && (RimAom <= 1) && (AomRip <= 1) && (AomRom <= 1) && (RopAip <= 1) && (RopAom <= 1) && (AipRom <= 1) && (RomAim <= 1) && (RomAop <= 1) && (AimRop <= 1) )
ltl p1 { [] safe }
#define live ((t1>1) && (t2>1) && (t3>1) && (t4>1) && (t5>1) && (t6>1) && (t7>1) && (t8>1) && (t9>1) && (t10>1) && (t11>1) && (t12>1))
ltl p2 { <> live }
#define persist ((((AomRip == 1) || (AomRom == 1)) -> ((AopRop == 0) && (AopRim == 0))) && (((AopRop == 1) || (AopRim == 1)) -> ((AomRip == 0) && (AomRom == 0))) && (((RomAim == 1) || (RomAop == 1)) -> ((RopAip == 0) && (RopAom == 0))) && (((RopAip == 1) || (RopAom == 1)) -> ((RomAim == 0) && (RomAop == 0))))
ltl p3 { [] persist }
```

รูปที่ ก-2 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานจากข้อที่ 5.1.1

### ก.1.3 รหัสโพรเมลาแบบ A2 ของวงจรอสมวารฟูล

```
byte RipAop, AopRop, AopRim, RimAom, AomRip, AomRom, RopAip, RopAom, AipRom, RomAim, RomAop, AimRop;
byte Rip, Rim, Aip, Aim, Rop, Rom, Aop, Aom;
#define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++;y++; fi
#define outp1(x1)          if :: (x1 != 10) -> x1++; :: else -> break; fi
#define outp2(x1,x2)       if :: (x1 != 10) -> x1++; :: ( x2 != 10) -> x2++; :: else -> break; fi
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> AomRip=2;RomAop=10;AimRop=10;RipAop=0;AopRop=0;AopRim=0;RimAom=0;
AomRom=0;RopAip=0;RopAom=0;AipRom=0;RomAim=0;Rip=0;Rim=0;Aip=0;Aim=0;Rop=0;Rom=0;
Aop=0;Aom=0;
   :: k==2 -> RomAop=2;AomRip=10;AimRop=10;RipAop=0;AopRop=0;AopRim=0;RimAom=0;
AomRom=0;RopAip=0;RopAom=0;AipRom=0;RomAim=0;Rip=0;Rim=0;Aip=0;Aim=0;Rop=0;Rom=0;
Aop=0;Aom=0;
   :: k==3 -> AimRop=2;AomRip=10;RomAop=10;RipAop=0;AopRop=0;AopRim=0;RimAom=0;
AomRom=0;RopAip=0;RopAom=0;AipRom=0;RomAim=0;Rip=0;Rim=0;Aip=0;Aim=0;Rop=0;Rom=0;
```

```
Aop=0;Aom=0;
  fi
  do
          ::atomic{ inp(AomRip,Aom);printf("printf(Aom=%d)",Aom) -> outp1(RipAop)}
          ::atomic{ inp(RipAop,Rip);printf("printf(Rip=%d)",Rip) -> outp2(AopRop,AopRim)}
          ::atomic{ inp(RomAop,Rom);printf("printf(Rom=%d)",Rom) -> outp2(AopRop,AopRim)}
          ::atomic{ inp(AopRim,Aop);printf("printf(Aop=%d)",Aop) -> outp1(RimAom)}
          ::atomic{ inp(RimAom,Rim);printf("printf(Rim=%d)",Rim) -> outp2(AomRip,AomRom)}
          ::atomic{ inp(RopAom,Rop);printf("printf(Rop=%d)",Rop) -> outp2(AomRip,AomRom)}
          ::atomic{ inp(AopRop,Aop);printf("printf(Aop=%d)",Aop) -> outp2(RopAip,RopAom)}
          ::atomic{ inp(AimRop,Aim);printf("printf(Aim=%d)",Aim) -> outp2(RopAip,RopAom)}
          ::atomic{ inp(RopAip,Rop);printf("printf(Rop=%d)",Rop) -> outp1(AipRom)}
          ::atomic{ inp(AomRom,Aom);printf("printf(Aom=%d)",Aom) -> outp2(RomAim,RomAop)}
          ::atomic{ inp(AipRom,Aip);printf("printf(Aip=%d)",Aip) -> outp2(RomAim,RomAop)}
          ::atomic{ inp(RomAim,Rom);printf("printf(Rom=%d)",Rom) -> outp1(AimRop)}
  od
}}
init{
  run P3(1);
  run P3(2);
  run P3(3);
}
```

รูปที่ ก-3 รหัสโพรเมลาแบบ A2 วงจรอสมวารฟูล จากข้อที่ 5.1.1



### ก.1.4 ตรรกะเวลาเชิงเส้นคุณสมบัติความต้องกันของวงจรอสมวารฟูล

```
#define consist1 (Rip>=1 && Rim>=1 && (AomRip==4 || RomAop==4 || AimRop==4))
#define consist2 (Aip>=1 && Aim>=1 && (AomRip==4 || RomAop==4 || AimRop==4))
#define consist3 (Rop>=1 && Rom>=1 && (AomRip==4 || RomAop==4 || AimRop==4))
#define consist4 (Aop>=1 && Aom>=1 && (AomRip==4 || RomAop==4 || AimRop==4))
ltl p4{!<>consist1}
ltl p5{!<>consist2}
ltl p6{!<>consist3}
ltl p7{!<>consist4}
```

รูปที่ ก-4 ตรรกะเวลาเชิงเส้นคุณสมบัติต้องกันจากข้อที่ 5.1.1

### ก.1.5 ตรรกะเวลาเชิงเส้นคุณสมบัติการกำหนดสถานะที่สมบูรณ์ของวงจรอสมวารฟูล

```
#define csc_full_RiAi (Rip==1 && Rim==1 && Aip==1 && Aim==1 && (AomRip==4 || RomAop==4 || AimRop==4))
ltl c1 {!<> csc_full_RiAi}
#define csc_full_RiRo (Rip==1 && Rim==1 && Rop==1 && Rom==1 && (AomRip==4 || RomAop==4 || AimRop==4))
ltl c2 {!<> csc_full_RiRo}
#define csc_full_RiAo (Rip==1 && Rim==1 && Aop==1 && Aom==1 && (AomRip==4 || RomAop==4 || AimRop==4))
ltl c3 {!<> csc_full_RiAo}
#define csc_full_AiRo (Aip==1 && Aim==1 && Rop==1 && Rom==1 && (AomRip==4 || RomAop==4 || AimRop==4))
ltl c4 {!<> csc_full_AiRo}
#define csc_full_AiAo (Aip==1 && Aim==1 && Aop==1 && Aom==1 && (AomRip==4 || RomAop==4 || AimRop==4))
ltl c5 {!<> csc_full_AiAo}
#define csc_full_RoAo (Rop==1 && Rom==1 && Aop==1 && Aom==1 && (AomRip==4 || RomAop==4 || AimRop==4))
ltl c6 {!<> csc_full_RoAo}
```

รูปที่ ก-5 ตรรกะเวลาเชิงเส้นการกำหนดสถานะที่สมบูรณ์จากข้อที่ 5.1.1

## ก.2 รหัสโพรเมลาและตรรกะเวลาเชิงเส้นจากวงจรอสมวารไทรมอสเซนท์

### ก.2.1 รหัสโพรเมลาแบบ A1 ของวงจรอสมวารไทรมอสเซนท์

```
byte R1pAom,R1pT1p,T1pCop,T1pR1m, R1mT1m, T1mR1p, AomT1p, AomCop, CopT1m, CopBom,
BomAop,BomT2p, R2pT2p, R2pBom, T2pAop, T2pR2m, R2mT2m, T2mR2p, AopCom, AopT2m,
ComT3p,ComBop, BopT3m, BopAom, R3pCom, R3pT3p, T3pBop, T3pR3m, R3mT3m, T3mR3p;
byte t1, t2, t3, t4, t5, t6, t7, t8, t9, t10, t11, t12, t13, t14, t15, t16, t17, t18, t19, t20, t21, t22, t23,
t24, t25, t26, t27, t28, t29, t30;
#define inp1(x1,t1)         (x1>0) -> x1 = x1-1; t1 = t1+1
#define inp2(x1,x2,t1,t2)  (x1>0 && x2>0) -> x1 = x1-1; x2 = x2-1; t1 = t1+1; t2 = t2+1
#define outp1(x1)           x1 = x1+1
#define outp2(x1,x2)        x1 = x1+1; x2 = x2+1
init {
T1mR1p=1;T2mR2p=1;T3mR3p=1;BopAom=1;
do
        ::atomic{ inp1(T1mR1p,t1) -> outp2(R1pAom,R1pT1p)}
        ::atomic{ inp2(R1pT1p,AomT1p,t2,t3) -> outp2(T1pCop,T1pR1m)}
        ::atomic{ inp1(T1pR1m,t4) -> outp1(R1mT1m)}
        ::atomic{ inp2(R1mT1m,CopT1m,t5,t6) -> outp1(T1mR1p)}
        ::atomic{ inp2(R1pAom,BopAom,t7,t8) -> outp2(AomT1p,AomCop)}
        ::atomic{ inp2(T1pCop,AomCop,t9,t10) -> outp2(CopT1m,CopBom)}
        ::atomic{ inp2(CopBom,R2pBom,t11,t12) -> outp2(BomAop,BomT2p)}
        ::atomic{ inp1(T2mR2p,t13) -> outp2(R2pT2p,R2pBom)}
        ::atomic{ inp2(BomT2p,R2pT2p,t14,t15) -> outp2(T2pAop,T2pR2m)}
        ::atomic{ inp1(T2pR2m,t16) -> outp1(R2mT2m)}
        ::atomic{ inp2(R2mT2m,AopT2m,t17,t18) -> outp1(T2mR2p)}
        ::atomic{ inp2(BomAop,T2pAop,t19,t20) -> outp2(AopCom,AopT2m)}
        ::atomic{ inp2(AopCom,R3pCom,t21,t22) -> outp2(ComT3p,ComBop)}
        ::atomic{ inp2(ComBop,T3pBop,t23,t24) -> outp2(BopT3m,BopAom)}
        ::atomic{ inp1(T3mR3p,t25) -> outp2(R3pCom,R3pT3p)}
        ::atomic{ inp2(ComT3p,R3pT3p,t26,t27) -> outp2(T3pBop,T3pR3m)}
        ::atomic{ inp1(T3pR3m,t28) -> outp1(R3mT3m)}
        ::atomic{ inp2(BopT3m,R3mT3m,t29,t30) -> outp1(T3mR3p)}
od}
```

รูปที่ ก-6 รหัสโพรเมลาแบบ A1 วงจรอสมวารไทรมอสเซนท์ จากข้อที่ 5.1.2

### ก.2.2 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย คุณสมบัติไลฟ์เนส และคุณสมบัติความทนทานวงจรอสมไทรมอสเซนท์

```
#define safe ((R1pAom <= 1) && (R1pT1p <= 1) && (T1pCop <= 1) && (T1pR1m <= 1) &&
(R1mT1m <= 1) && (T1mR1p <= 1) && (AomT1p <= 1) && (AomCop <= 1) && (CopT1m <= 1) &&
(CopBom <= 1) && (BomAop <= 1) && (BomT2p <= 1) && (R2pT2p <= 1) && (R2pBom <= 1) &&
(T2pAop <= 1) && (T2pR2m <= 1) && (R2mT2m <= 1) && (T2mR2p <= 1) && (AopCom <= 1) &&
(AopT2m <= 1) && (ComT3p <= 1) && (ComBop <= 1) && (BopT3m <= 1) && (BopAom <= 1) &&
(R3pCom <= 1) && (R3pT3p <= 1) && (T3pBop <= 1) && (T3pR3m <= 1) && (R3mT3m <= 1) &&
(T3mR3p <= 1) )
ltl p1 { [] safe }
#define live ((t1>1) && (t2>1) && (t3>1) && (t4>1) && (t5>1) && (t6>1) && (t7>1) && (t8>1) &&
(t9>1) && (t10>1) && (t11>1) && (t12>1) && (t13>1) && (t14>1) && (t15>1) && (t16>1) && (t17>1)
&& (t18>1) && (t19>1) && (t20>1) && (t21>1) && (t22>1) && (t23>1) && (t24>1) && (t25>1) &&
(t26>1) && (t27>1) && (t28>1) && (t29>1) && (t30>1))
ltl p2 { <> live }
#define persist (((R1mT1m == 1) -> ((R1pAom == 0) && (R1pT1p == 0))) && ((T1mR1p == 1) ->
((T1pCop == 0) && (T1pR1m == 0))) && (((AopCom == 1) || (AopT2m == 1)) -> ((AomT1p == 0) &&
(AomCop == 0))) && (((ComT3p == 1) || (ComBop == 1)) -> ((CopT1m == 0) && (CopBom == 0)))
&& (((BopT3m == 1) || (BopAom == 1)) -> ((BomAop == 0) && (BomT2p == 0))) && ((R2mT2m == 1)
-> ((R2pT2p == 0) && (R2pBom == 0))) && ((T2mR2p == 1) -> ((T2pAop == 0) && (T2pR2m == 0)))
&& (((AomT1p == 1) || (AomCop == 1)) -> ((AopCom == 0) && (AopT2m == 0))) && (((CopT1m ==
1) || (CopBom == 1)) -> ((ComT3p == 0) && (ComBop == 0))) && (((BomAop == 1) || (BomT2p ==
1)) -> ((BopT3m == 0) && (BopAom == 0))) && ((R3mT3m == 1) -> ((R3pCom == 0) && (R3pT3p ==
0))) && ((T3mR3p == 1) -> ((T3pBop == 0) && (T3pR3m == 0))))
ltl p3 { [] persist }
```

รูปที่ ก-7 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานจากข้อที่ 5.1.2

**ก.2.3 รหัสโพรเมลาแบบ A2 ของวงจรอสมวารไทรมอสเซนท์**

```
byte R1pAom, R1pT1p, T1pCop, T1pR1m, R1mT1m, T1mR1p, AomT1p, AomCop, CopT1m,
CopBom, BomAop, BomT2p, R2pT2p, R2pBom, T2pAop, T2pR2m, R2mT2m, T2mR2p, AopCom,
AopT2m, ComT3p, ComBop, BopT3m, BopAom, R3pCom, R3pT3p, T3pBop, T3pR3m, R3mT3m,
T3mR3p;
byte R1p, R1m, R2p, R2m, R3p, R3m, T1p, T1m, T2p, T2m, T3p, T3m, Aop, Aom, Bop, Bom, Cop,
Com;
#define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++;y++; fi
#define outp1(x1)          if :: (x1 != 10) -> x1++; :: else -> break; fi
#define outp2(x1,x2)       if :: (x1 != 10) -> x1++; :: ( x2 != 10) -> x2++; :: else -> break; fi
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> T1mR1p=2;T2mR2p=10;T3mR3p=10;BopAom=10;R1pAom=0;R1pT1p=0;T1pCop=0;
T1pR1m=0;R1mT1m=0;AomT1p=0;AomCop=0;CopT1m=0;CopBom=0;BomAop=0;BomT2p=0;
R2pT2p=0;R2pBom=0;T2pAop=0;T2pR2m=0;R2mT2m=0;AopCom=0;AopT2m=0;ComT3p=0;
ComBop=0;BopT3m=0;R3pCom=0;R3pT3p=0;T3pBop=0;T3pR3m=0;R3mT3m=0;R1p=0;R1m=0;
R2p=0;R2m=0;R3p=0;R3m=0;T1p=0;T1m=0;T2p=0;T2m=0;T3p=0;T3m=0;Aop=0;Aom=0;Bop=0;
Bom=0;Cop=0;Com=0;
   :: k==2 -> T2mR2p=2;T1mR1p=10;T3mR3p=10;BopAom=10;R1pAom=0;R1pT1p=0;T1pCop=0;
T1pR1m=0;R1mT1m=0;AomT1p=0;AomCop=0;CopT1m=0;CopBom=0;BomAop=0;BomT2p=0;
R2pT2p=0;R2pBom=0;T2pAop=0;T2pR2m=0;R2mT2m=0;AopCom=0;AopT2m=0;ComT3p=0;
ComBop=0;BopT3m=0;R3pCom=0;R3pT3p=0;T3pBop=0;T3pR3m=0;R3mT3m=0;R1p=0;R1m=0;
R2p=0;R2m=0;R3p=0;R3m=0;T1p=0;T1m=0;T2p=0;T2m=0;T3p=0;T3m=0;Aop=0;Aom=0;Bop=0;
Bom=0;Cop=0;Com=0;
   :: k==3 ->
T3mR3p=2;T1mR1p=10;T2mR2p=10;BopAom=10;R1pAom=0;R1pT1p=0;T1pCop=0;T1pR1m=0;
R1mT1m=0;AomT1p=0;AomCop=0;CopT1m=0;CopBom=0;BomAop=0;BomT2p=0;R2pT2p=0;
R2pBom=0;T2pAop=0;T2pR2m=0;R2mT2m=0;AopCom=0;AopT2m=0;ComT3p=0;ComBop=0;
BopT3m=0;R3pCom=0;R3pT3p=0;T3pBop=0;T3pR3m=0;R3mT3m=0;R1p=0;R1m=0;R2p=0;R2m=0;
R3p=0;R3m=0;T1p=0;T1m=0;T2p=0;T2m=0;T3p=0;T3m=0;Aop=0;Aom=0;Bop=0;Bom=0;Cop=0;
Com=0;
   :: k==4 -> BopAom=2;T1mR1p=10;T2mR2p=10;T3mR3p=10;R1pAom=0;R1pT1p=0;T1pCop=0;
T1pR1m=0;R1mT1m=0;AomT1p=0;AomCop=0;CopT1m=0;CopBom=0;BomAop=0;BomT2p=0;
R2pT2p=0;R2pBom=0;T2pAop=0;T2pR2m=0;R2mT2m=0;AopCom=0;AopT2m=0;ComT3p=0;
```

```
ComBop=0;BopT3m=0;R3pCom=0;R3pT3p=0;T3pBop=0;T3pR3m=0;R3mT3m=0;R1p=0;R1m=0;
R2p=0;R2m=0;R3p=0;R3m=0;T1p=0;T1m=0;T2p=0;T2m=0;T3p=0;T3m=0;Aop=0;Aom=0;Bop=0;
Bom=0;Cop=0;Com=0;
 fi
 do
        ::atomic{ inp(T1mR1p,T1m);printf("printf(T1m=%d)",T1m) -> outp2(R1pAom,R1pT1p)}
        ::atomic{ inp(R1pT1p,R1p);printf("printf(R1p=%d)",R1p) -> outp2(T1pCop,T1pR1m)}
        ::atomic{ inp(AomT1p,Aom);printf("printf(Aom=%d)",Aom) -> outp2(T1pCop,T1pR1m)}
        ::atomic{ inp(T1pR1m,T1p);printf("printf(T1p=%d)",T1p) -> outp1(R1mT1m)}
        ::atomic{ inp(R1mT1m,R1m);printf("printf(R1m=%d)",R1m) -> outp1(T1mR1p)}
        ::atomic{ inp(CopT1m,Cop);printf("printf(Cop=%d)",Cop) -> outp1(T1mR1p)}
        ::atomic{ inp(R1pAom,R1p);printf("printf(R1p=%d)",R1p) -> outp2(AomT1p,AomCop)}
        ::atomic{ inp(BopAom,Bop);printf("printf(Bop=%d)",Bop) -> outp2(AomT1p,AomCop)}
        ::atomic{ inp(T1pCop,T1p);printf("printf(T1p=%d)",T1p) -> outp2(CopT1m,CopBom)}
        ::atomic{ inp(AomCop,Aom);printf("printf(Aom=%d)",Aom) -> outp2(CopT1m,CopBom)}
        ::atomic{ inp(CopBom,Cop);printf("printf(Cop=%d)",Cop) -> outp2(BomAop,BomT2p)}
        ::atomic{ inp(R2pBom,R2p);printf("printf(R2p=%d)",R2p) -> outp2(BomAop,BomT2p)}
        ::atomic{ inp(T2mR2p,T2m);printf("printf(T2m=%d)",T2m) -> outp2(R2pT2p,R2pBom)}
        ::atomic{ inp(BomT2p,Bom);printf("printf(Bom=%d)",Bom) -> outp2(T2pAop,T2pR2m)}
        ::atomic{ inp(R2pT2p,R2p);printf("printf(R2p=%d)",R2p) -> outp2(T2pAop,T2pR2m)}
        ::atomic{ inp(T2pR2m,T2p);printf("printf(T2p=%d)",T2p) -> outp1(R2mT2m)}
        ::atomic{ inp(R2mT2m,R2m);printf("printf(R2m=%d)",R2m) -> outp1(T2mR2p)}
        ::atomic{ inp(AopT2m,Aop);printf("printf(Aop=%d)",Aop) -> outp1(T2mR2p)}
        ::atomic{ inp(BomAop,Bom);printf("printf(Bom=%d)",Bom) -> outp2(AopCom,AopT2m)}
        ::atomic{ inp(T2pAop,T2p);printf("printf(T2p=%d)",T2p) -> outp2(AopCom,AopT2m)}
        ::atomic{ inp(AopCom,Aop);printf("printf(Aop=%d)",Aop) -> outp2(ComT3p,ComBop)}
        ::atomic{ inp(R3pCom,R3p);printf("printf(R3p=%d)",R3p) -> outp2(ComT3p,ComBop)}
        ::atomic{ inp(ComBop,Com);printf("printf(Com=%d)",Com) -> outp2(BopT3m,BopAom)}
        ::atomic{ inp(T3pBop,T3p);printf("printf(T3p=%d)",T3p) -> outp2(BopT3m,BopAom)}
        ::atomic{ inp(T3mR3p,T3m);printf("printf(T3m=%d)",T3m) -> outp2(R3pCom,R3pT3p)}
        ::atomic{ inp(ComT3p,Com);printf("printf(Com=%d)",Com) -> outp2(T3pBop,T3pR3m)}
        ::atomic{ inp(R3pT3p,R3p);printf("printf(R3p=%d)",R3p) -> outp2(T3pBop,T3pR3m)}
        ::atomic{ inp(T3pR3m,T3p);printf("printf(T3p=%d)",T3p) -> outp1(R3mT3m)}
        ::atomic{ inp(BopT3m,Bop);printf("printf(Bop=%d)",Bop) -> outp1(T3mR3p)}
        ::atomic{ inp(R3mT3m,R3m);printf("printf(R3m=%d)",R3m) -> outp1(T3mR3p)}
```

```
  od
}}
init{
  run P3(1);
  run P3(2);
  run P3(3);
  run P3(4);
}
```

รูปที่ ก-8 รหัสโพรเมลาแบบ A2 วงจรอสมวารไทรมอสเซนท์ จากข้อที่ 5.1.2

### ก.2.4 ตรรกะเวลาเชิงเส้นคุณสมบัติความต้องกันของวงจรอสมวารไทรมอสเซนท์

```
#define consist1 (R1p>=1 && R1m>=1 && (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 ||
BopAom==4))
#define consist2 (R2p>=1 && R2m>=1 && (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 ||
BopAom==4))
#define consist3 (R3p>=1 && R3m>=1 && (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 ||
BopAom==4))
#define consist4 (T1p>=1 && T1m>=1 && (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 ||
BopAom==4))
#define consist5 (T2p>=1 && T2m>=1 && (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 ||
BopAom==4))
#define consist6 (T3p>=1 && T3m>=1 && (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 ||
BopAom==4))
#define consist7 (Aop>=1 && Aom>=1 && (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 ||
BopAom==4))
#define consist8 (Bop>=1 && Bom>=1 && (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 ||
BopAom==4))
#define consist9 (Cop>=1 && Com>=1 && (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 ||
BopAom==4))
ltl p4{!<>consist1}
ltl p5{!<>consist2}
ltl p6{!<>consist3}
ltl p7{!<>consist4}
ltl p8{!<>consist5}
```

```
ltl p9{!<>consist6}
ltl p10{!<>consist7}
ltl p11{!<>consist8}
ltl p12{!<>consist9}
```

รูปที่ ก-9 ตรรกะเวลาเชิงเส้นคุณสมบัติต้องกันจากข้อที่ 5.1.2

### ก.2.5 ตรรกะเวลาเชิงเส้นคุณสมบัติการกำหนดสถานะที่สมบูรณ์ของวงจรอสมวารไทรมอสเซนท์

```
#define csc_full_R1R2 (R1p==1 && R1m==1 && R2p==1 && R2m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c1 {!<> csc_full_R1R2}
#define csc_full_R1R3 (R1p==1 && R1m==1 && R3p==1 && R3m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c2 {!<> csc_full_R1R3}
#define csc_full_R1T1 (R1p==1 && R1m==1 && T1p==1 && T1m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c3 {!<> csc_full_R1T1}
#define csc_full_R1T2 (R1p==1 && R1m==1 && T2p==1 && T2m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c4 {!<> csc_full_R1T2}
#define csc_full_R1T3 (R1p==1 && R1m==1 && T3p==1 && T3m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c5 {!<> csc_full_R1T3}
#define csc_full_R1Ao (R1p==1 && R1m==1 && Aop==1 && Aom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c6 {!<> csc_full_R1Ao}
#define csc_full_R1Bo (R1p==1 && R1m==1 && Bop==1 && Bom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c7 {!<> csc_full_R1Bo}
#define csc_full_R1Co (R1p==1 && R1m==1 && Cop==1 && Com==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c8 {!<> csc_full_R1Co}
#define csc_full_R2R3 (R2p==1 && R2m==1 && R3p==1 && R3m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c9 {!<> csc_full_R2R3}
#define csc_full_R2T1 (R2p==1 && R2m==1 && T1p==1 && T1m==1 && (T1mR1p==4 ||
```

```
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c10 {!<> csc_full_R2T1}
#define csc_full_R2T2 (R2p==1 && R2m==1 && T2p==1 && T2m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c11 {!<> csc_full_R2T2}
#define csc_full_R2T3 (R2p==1 && R2m==1 && T3p==1 && T3m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c12 {!<> csc_full_R2T3}
#define csc_full_R2Ao (R2p==1 && R2m==1 && Aop==1 && Aom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c13 {!<> csc_full_R2Ao}
#define csc_full_R2Bo (R2p==1 && R2m==1 && Bop==1 && Bom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c14 {!<> csc_full_R2Bo}
#define csc_full_R2Co (R2p==1 && R2m==1 && Cop==1 && Com==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c15 {!<> csc_full_R2Co}
#define csc_full_R3T1 (R3p==1 && R3m==1 && T1p==1 && T1m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c16 {!<> csc_full_R3T1}
#define csc_full_R3T2 (R3p==1 && R3m==1 && T2p==1 && T2m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c17 {!<> csc_full_R3T2}
#define csc_full_R3T3 (R3p==1 && R3m==1 && T3p==1 && T3m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c18 {!<> csc_full_R3T3}
#define csc_full_R3Ao (R3p==1 && R3m==1 && Aop==1 && Aom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c19 {!<> csc_full_R3Ao}
#define csc_full_R3Bo (R3p==1 && R3m==1 && Bop==1 && Bom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c20 {!<> csc_full_R3Bo}
#define csc_full_R3Co (R3p==1 && R3m==1 && Cop==1 && Com==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c21 {!<> csc_full_R3Co}
```

```
#define csc_full_T1T2 (T1p==1 && T1m==1 && T2p==1 && T2m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c22 {!<> csc_full_T1T2}
#define csc_full_T1T3 (T1p==1 && T1m==1 && T3p==1 && T3m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c23 {!<> csc_full_T1T3}
#define csc_full_T1Ao (T1p==1 && T1m==1 && Aop==1 && Aom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c24 {!<> csc_full_T1Ao}
#define csc_full_T1Bo (T1p==1 && T1m==1 && Bop==1 && Bom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c25 {!<> csc_full_T1Bo}
#define csc_full_T1Co (T1p==1 && T1m==1 && Cop==1 && Com==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c26 {!<> csc_full_T1Co}
#define csc_full_T2T3 (T2p==1 && T2m==1 && T3p==1 && T3m==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c27 {!<> csc_full_T2T3}
#define csc_full_T2Ao (T2p==1 && T2m==1 && Aop==1 && Aom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c28 {!<> csc_full_T2Ao}
#define csc_full_T2Bo (T2p==1 && T2m==1 && Bop==1 && Bom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c29 {!<> csc_full_T2Bo}
#define csc_full_T2Co (T2p==1 && T2m==1 && Cop==1 && Com==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c30 {!<> csc_full_T2Co}
#define csc_full_T3Ao (T3p==1 && T3m==1 && Aop==1 && Aom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c31 {!<> csc_full_T3Ao}
#define csc_full_T3Bo (T3p==1 && T3m==1 && Bop==1 && Bom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c32 {!<> csc_full_T3Bo}
#define csc_full_T3Co (T3p==1 && T3m==1 && Cop==1 && Com==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
```

```
ltl c33 {!<> csc_full_T3Co}
#define csc_full_AoBo (Aop==1 && Aom==1 && Bop==1 && Bom==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c34 {!<> csc_full_AoBo}
#define csc_full_AoCo (Aop==1 && Aom==1 && Cop==1 && Com==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c35 {!<> csc_full_AoCo}
#define csc_full_BoCo (Bop==1 && Bom==1 && Cop==1 && Com==1 && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl c36 {!<> csc_full_BoCo}
#define transitive1 ((R1p==1 || R1m==1) && (T1p==1 || T1m==1) && (T2p==1 && T2m==1 &&
T3p==1 && T3m==1 && Aop==1 && Aom==1 && Bop==1 && Bom==1 && Cop==1 && Com==1)
&& (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 || BopAom==4))
#define transitive2 ((R2p==1 || R2m==1) && (T2p==1 || T2m==1) && (T1p==1 && T1m==1 &&
T3p==1 && T3m==1 && Aop==1 && Aom==1 && Bop==1 && Bom==1 && Cop==1 && Com==1)
&& (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 || BopAom==4))
#define transitive3 ((R3p==1 || R3m==1) && (T3p==1 || T3m==1) && (T1p==1 && T1m==1 &&
T2p==1 && T2m==1 && Aop==1 && Aom==1 && Bop==1 && Bom==1 && Cop==1 && Com==1)
&& (T1mR1p==4 || T2mR2p==4 || T3mR3p==4 || BopAom==4))
#define transitive4 ((Aop==1 || Aom==1) && (Bop==1 || Bom==1) && (T1p==1 && T1m==1 &&
T2p==1 && T2m==1 && T3p==1 && T3m==1 && Cop==1 && Com==1) && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
#define transitive5 ((Aop==1 || Aom==1) && (Cop==1 || Com==1) && (T1p==1 && T1m==1 &&
T2p==1 && T2m==1 && T3p==1 && T3m==1 && Bop==1 && Bom==1) && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
#define transitive6 ((Bop==1 || Bom==1) && (Cop==1 || Com==1) && (T1p==1 && T1m==1 &&
T2p==1 && T2m==1 && T3p==1 && T3m==1 && Aop==1 && Aom==1) && (T1mR1p==4 ||
T2mR2p==4 || T3mR3p==4 || BopAom==4))
ltl t1 {!<>transitive1}
ltl t2 {!<>transitive2}
ltl t3 {!<>transitive3}
ltl t4 {!<>transitive4}
ltl t5 {!<>transitive5}
ltl t6 {!<>transitive6}
```

รูปที่ ก-10 ตรรกะเวลาเชิงเส้นการกำหนดสถานะที่สมบูรณ์จากข้อที่ 5.1.2

## ก.3 รหัสโพรเมลาและตรรกะเวลาเชิงเส้นจากวงจรอสมวารอีเบอร์เจ็น

### ก.3.1 รหัสโพรเมลาแบบ A1 ของวงจรอสมวารอีเบอร์เจ็น

```
byte ap2cp, cpbp1, cpdp, bp1am1, am1xp, dpxp, xpbm1, bm1ap1, ap1cm, cmdm, cmbp2,
bp2am2, am2xm, dmxm, xmbm2, bm2ap2;
byte t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15,t16;
#define inp1(x1,t1)         (x1>0) -> x1 = x1-1; t1 = t1+1
#define inp2(x1,x2,t1,t2)  (x1>0 && x2>0) -> x1 = x1-1; x2 = x2-1; t1 = t1+1; t2 = t2+1
#define outp1(x1)           x1 = x1+1
#define outp2(x1,x2)        x1 = x1+1; x2 = x2+1
init {
bm2ap2=1;
do
        ::atomic{ inp1(bm2ap2,t1) -> outp1(ap2cp)}
        ::atomic{ inp1(ap2cp,t2) -> outp2(cpbp1,cpdp)}
        ::atomic{ inp1(cpbp1,t3) -> outp1(bp1am1)}
        ::atomic{ inp1(bp1am1,t4) -> outp1(am1xp)}
        ::atomic{ inp1(cpdp,t5) -> outp1(dpxp)}
        ::atomic{ inp2(am1xp,dpxp,t6,t7) -> outp1(xpbm1)}
        ::atomic{ inp1(xpbm1,t8) -> outp1(bm1ap1)}
        ::atomic{ inp1(bm1ap1,t9) -> outp1(ap1cm)}
        ::atomic{ inp1(ap1cm,t10) -> outp2(cmdm,cmbp2)}
        ::atomic{ inp1(cmbp2,t11) -> outp1(bp2am2)}
        ::atomic{ inp1(bp2am2,t12) -> outp1(am2xm)}
        ::atomic{ inp1(cmdm,t13) -> outp1(dmxm)}
        ::atomic{ inp2(am2xm,dmxm,t14,t15) -> outp1(xmbm2)}
        ::atomic{ inp1(xmbm2,t16) -> outp1(bm2ap2)}
od
}
```

รูปที่ ก-11 รหัสโพรเมลาแบบ A1 วงจรอสมวารอีเบอร์เจ็น จากข้อที่ 5.1.3

### ก.3.2 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานวงจรอสมวารอีเบอร์เจ็น

```
#define safe ((ap2cp <= 1) && (cpbp1 <= 1) && (cpdp <= 1) && (bp1am1 <= 1) && (am1xp <= 1)
&& (dpxp <= 1) && (xpbm1 <= 1) && (bm1ap1 <= 1) && (ap1cm <= 1) && (cmdm <= 1) &&
(cmbp2 <= 1) && (bp2am2 <= 1) && (am2xm <= 1) && (dmxm <= 1) && (xmbm2 <= 1) &&
(bm2ap2 <= 1) )
ltl p1 { [] safe }
#define live ((t1>1) && (t2>1) && (t3>1) && (t4>1) && (t5>1) && (t6>1) && (t7>1) && (t8>1) &&
(t9>1) && (t10>1) && (t11>1) && (t12>1) && (t13>1) && (t14>1) && (t15>1) && (t16>1))
ltl p2 { <> live }
#define persist ((((cmdm == 1) || (cmbp2 == 1)) -> ((cpbp1 == 0) && (cpdp == 0))) && (((cpbp1 ==
1) || (cpdp == 1)) -> ((cmdm == 0) && (cmbp2 == 0))) )
ltl p3 { [] persist }
```

รูปที่ ก-12 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานจากข้อที่ 5.1.3

### ก.3.3 รหัสโพรเมลาแบบ A2 ของวงจรอสมวารอีเบอร์เจ็น

```
byte ap2cp, cpbp1, cpdp, bp1am1, am1xp, dpxp, xpbm1, bm1ap1, ap1cm, cmdm, cmbp2,
bp2am2, am2xm, dmxm, xmbm2, bm2ap2;
byte ap, am, dp, dm, bp, bm, cp, cm, xp, xm;
#define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++;y++; fi
#define outp1(x1)          if :: (x1 != 10) -> x1++; :: else -> break; fi
#define outp2(x1,x2)       if :: (x1 != 10) -> x1++; :: ( x2 != 10) -> x2++; :: else -> break; fi
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> bm2ap2=2;ap2cp=0;cpbp1=0;cpdp=0;bp1am1=0;am1xp=0;dpxp=0;xpbm1=0;
bm1ap1=0;ap1cm=0;cmdm=0;cmbp2=0;bp2am2=0;am2xm=0;dmxm=0;xmbm2=0;ap=0;am=0;
dp=0;dm=0;bp=0;bm=0;cp=0;cm=0;xp=0;xm=0;
  fi
  do
          ::atomic{ inp(bm2ap2,bm);printf("printf(bm=%d)",bm) -> outp1(ap2cp)}
          ::atomic{ inp(ap2cp,ap);printf("printf(ap=%d)",ap) -> outp2(cpbp1,cpdp)}
          ::atomic{ inp(cpbp1,cp);printf("printf(cp=%d)",cp) -> outp1(bp1am1)}
          ::atomic{ inp(bp1am1,bp);printf("printf(bp=%d)",bp) -> outp1(am1xp)}
```

```
            ::atomic{ inp(cpdp,cp);printf("printf(cp=%d)",cp) -> outp1(dpxp)}
            ::atomic{ inp(am1xp,am);printf("printf(am=%d)",am) -> outp1(xpbm1)}
            ::atomic{ inp(dpxp,dp);printf("printf(dp=%d)",dp) -> outp1(xpbm1)}
            ::atomic{ inp(xpbm1,xp);printf("printf(xp=%d)",xp) -> outp1(bm1ap1)}
            ::atomic{ inp(bm1ap1,bm);printf("printf(bm=%d)",bm) -> outp1(ap1cm)}
            ::atomic{ inp(ap1cm,ap);printf("printf(ap=%d)",ap) -> outp2(cmdm,cmbp2)}
            ::atomic{ inp(cmbp2,cm);printf("printf(cm=%d)",cm) -> outp1(bp2am2)}
            ::atomic{ inp(bp2am2,bp);printf("printf(bp=%d)",bp) -> outp1(am2xm)}
            ::atomic{ inp(cmdm,cm);printf("printf(cm=%d)",cm) -> outp1(dmxm)}
            ::atomic{ inp(am2xm,am);printf("printf(am=%d)",am) -> outp1(xmbm2)}
            ::atomic{ inp(dmxm,dm);printf("printf(dm=%d)",dm) -> outp1(xmbm2)}
            ::atomic{ inp(xmbm2,xm);printf("printf(xm=%d)",xm) -> outp1(bm2ap2)}
  od
}}
init{
  run P3(1);
}
```

รูปที่ ก-13 รหัสโพรเมลาแบบ A2 วงจรอสมวารอีเบอร์เจ็น จากข้อที่ 5.1.3

### ก.3.4 ตรรกะเวลาเชิงเส้นคุณสมบัติความต้องกันของวงจรอสมวารอีเบอร์เจ็น

```
#define consist1 (ap>=1 && am>=1 && (bm2ap2==4))
#define consist2 (dp>=1 && dm>=1 && (bm2ap2==4))
#define consist3 (bp>=1 && bm>=1 && (bm2ap2==4))
#define consist4 (cp>=1 && cm>=1 && (bm2ap2==4))
#define consist5 (xp>=1 && xm>=1 && (bm2ap2==4))
ltl p4{!<>consist1}
ltl p5{!<>consist2}
ltl p6{!<>consist3}
ltl p7{!<>consist4}
ltl p8{!<>consist5}
```

รูปที่ ก-14 ตรรกะเวลาเชิงเส้นคุณสมบัติต้องกันจากข้อที่ 5.1.3

### ก.3.5 ตรรกะเวลาเชิงเส้นคุณสมบัติการกำหนดสถานะที่สมบูรณ์ของวงจรอสมวารอีเบอร์เจ็น

```
#define csc_full_ad (ap==1 && am==1 && dp==1 && dm==1 && (bm2ap2==4))
ltl c1 {!<> csc_full_ad}
#define csc_full_ab (ap==1 && am==1 && bp==1 && bm==1 && (bm2ap2==4))
ltl c2 {!<> csc_full_ab}
#define csc_full_ac (ap==1 && am==1 && cp==1 && cm==1 && (bm2ap2==4))
ltl c3 {!<> csc_full_ac}
#define csc_full_ax (ap==1 && am==1 && xp==1 && xm==1 && (bm2ap2==4))
ltl c4 {!<> csc_full_ax}
#define csc_full_db (dp==1 && dm==1 && bp==1 && bm==1 && (bm2ap2==4))
ltl c5 {!<> csc_full_db}
#define csc_full_dc (dp==1 && dm==1 && cp==1 && cm==1 && (bm2ap2==4))
ltl c6 {!<> csc_full_dc}
#define csc_full_dx (dp==1 && dm==1 && xp==1 && xm==1 && (bm2ap2==4))
ltl c7 {!<> csc_full_dx}
#define csc_full_bc (bp==1 && bm==1 && cp==1 && cm==1 && (bm2ap2==4))
ltl c8 {!<> csc_full_bc}
#define csc_full_bx (bp==1 && bm==1 && xp==1 && xm==1 && (bm2ap2==4))
ltl c9 {!<> csc_full_bx}
#define csc_full_cx (cp==1 && cm==1 && xp==1 && xm==1 && (bm2ap2==4))
ltl c10 {!<> csc_full_cx}
```

รูปที่ ก-15 ตรรกะเวลาเชิงเส้นการกำหนดสถานะที่สมบูรณ์จากข้อที่ 5.1.3

## ก.4 รหัสโพรเมลาและตรรกะเวลาเชิงเส้นจากวงจรอสมวารอินพุต

### ก.4.1 รหัสโพรเมลาแบบ A1 ของวงจรอสมวารอินพุต

```
byte outp1sp, outp1rp, spoutp2, rpoutm1, outm1enm, enmoutp2, outp2sm, outp2rm, smoutp1,
rmoutm2, outm2enp, enpoutp1;
byte t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12;
#define inp1(x1,t1)         (x1>0) -> x1 = x1-1; t1 = t1+1
#define inp2(x1,x2,t1,t2)  (x1>0 && x2>0) -> x1 = x1-1; x2 = x2-1; t1 = t1+1; t2 = t2+1
#define outp1(x1)           x1 = x1+1
#define outp2(x1,x2)        x1 = x1+1; x2 = x2+1
init {
smoutp1=1;outm2enp=1;
do
        ::atomic{ inp2(smoutp1,enpoutp1,t1,t2) -> outp2(outp1sp,outp1rp)}
        ::atomic{ inp1(outp1sp,t3) -> outp1(spoutp2)}
        ::atomic{ inp1(outp1rp,t4) -> outp1(rpoutm1)}
        ::atomic{ inp1(rpoutm1,t5) -> outp1(outm1enm)}
        ::atomic{ inp1(outm1enm,t6) -> outp1(enmoutp2)}
        ::atomic{ inp2(spoutp2,enmoutp2,t7,t8) -> outp2(outp2sm,outp2rm)}
        ::atomic{ inp1(outp2sm,t9) -> outp1(smoutp1)}
        ::atomic{ inp1(outp2rm,t10) -> outp1(rmoutm2)}
        ::atomic{ inp1(rmoutm2,t11) -> outp1(outm2enp)}
        ::atomic{ inp1(outm2enp,t12) -> outp1(enpoutp1)}
od}
```



รูปที่ ก-16 รหัสโพรเมลาแบบ A1 วงจรอสมวารอินพุต จากข้อที่ 5.1.4

### ก.4.2 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานวงจรอสมวารอินพุต

```
#define safe ((outp1sp <= 1) && (outp1rp <= 1) && (spoutp2 <= 1) && (rpoutm1 <= 1) &&
(outm1enm <= 1) && (enmoutp2 <= 1) && (outp2sm <= 1) && (outp2rm <= 1) && (smoutp1 <= 1)
&& (rmoutm2 <= 1) && (outm2enp <= 1) && (enpoutp1 <= 1) )
ltl p1 { [] safe }
#define live ((t1>1) && (t2>1) && (t3>1) && (t4>1) && (t5>1) && (t6>1) && (t7>1) && (t8>1) &&
(t9>1) && (t10>1) && (t11>1) && (t12>1))
ltl p2 { <> live }
#define persist ((((outm1enm == 1) || (outm2enp == 1)) -> (((outp1sp == 0) && (outp1rp == 0)) &&
((outp2sm == 0) && (outp2rm == 0)))))
ltl p3 { [] persist }
```

รูปที่ ก-17 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานจากข้อที่ 5.1.4

### ก.4.3 รหัสโพรเมลาแบบ A2 ของวงจรอสมวารอินพุต

```
byte outp1sp, outp1rp, spoutp2, rpoutm1, outm1enm, enmoutp2, outp2sm, outp2rm, smoutp1,
rmoutm2, outm2enp, enpoutp1;
byte sp, sm, rp, rm, enp, enm, outp, outm;
#define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++;y++; fi
#define outp1(x1)          if :: (x1 != 10) -> x1++; :: else -> break; fi
#define outp2(x1,x2)       if :: (x1 != 10) -> x1++; :: ( x2 != 10) -> x2++; :: else -> break; fi
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> smoutp1=2;outm2enp=10;outp1sp=0;outp1rp=0;spoutp2=0;rpoutm1=0;
outm1enm=0;enmoutp2=0;outp2sm=0;outp2rm=0;rmoutm2=0;enpoutp1=0;sp=0;sm=0;
rp=0; rm=0;enp=0;enm=0;outp=0;outm=0;
   :: k==2 -> outm2enp=2;smoutp1=10;outp1sp=0;outp1rp=0;spoutp2=0;rpoutm1=0;
outm1enm=0;enmoutp2=0;outp2sm=0;outp2rm=0;rmoutm2=0;enpoutp1=0;sp=0;sm=0;
rp=0;rm=0;enp=0;enm=0;outp=0;outm=0;
  fi
  do
          ::atomic{ inp(smoutp1,sm);printf("printf(sm=%d)",sm) -> outp2(outp1sp,outp1rp)}
          ::atomic{ inp(enpoutp1,enp);printf("printf(enp=%d)",enp) -> outp2(outp1sp,outp1rp)}
```

```
        ::atomic{ inp(outp1sp,outp);printf("printf(outp=%d)",outp) -> outp1(spoutp2)}
        ::atomic{ inp(outp1rp,outp);printf("printf(outp=%d)",outp) -> outp1(rpoutm1)}
        ::atomic{ inp(rpoutm1,rp);printf("printf(rp=%d)",rp) -> outp1(outm1enm)}
        ::atomic{ inp(outm1enm,outm);printf("printf(outm=%d)",outm) -> outp1(enmoutp2)}
        ::atomic{ inp(spoutp2,sp);printf("printf(sp=%d)",sp) -> outp2(outp2sm,outp2rm)}
        ::atomic{ inp(enmoutp2,enm);printf("printf(enm=%d)",enm) -> outp2(outp2sm,outp2rm)}
        ::atomic{ inp(outp2sm,outp);printf("printf(outp=%d)",outp) -> outp1(smoutp1)}
        ::atomic{ inp(outp2rm,outp);printf("printf(outp=%d)",outp) -> outp1(rmoutm2)}
        ::atomic{ inp(rmoutm2,rm);printf("printf(rm=%d)",rm) -> outp1(outm2enp)}
        ::atomic{ inp(outm2enp,outm);printf("printf(outm=%d)",outm) -> outp1(enpoutp1)}
  od
}}
init{
  run P3(1);
  run P3(2);
}
```

รูปที่ ก-18 รหัสโพรเมลาแบบ A2 วงจรอสมวารอินพุต จากข้อที่ 5.1.4

### ก.4.4 ตรรกะเวลาเชิงเส้นคุณสมบัติความต้องกันของวงจรอสมวารอินพุต

```
#define consist1 (sp>=1 && sm>=1 && (smoutp1==4 || outm2enp==4))
#define consist2 (rp>=1 && rm>=1 && (smoutp1==4 || outm2enp==4))
#define consist3 (enp>=1 && enm>=1 && (smoutp1==4 || outm2enp==4))
#define consist4 (outp>=1 && outm>=1 && (smoutp1==4 || outm2enp==4))
ltl p4{!<>consist1}
ltl p5{!<>consist2}
ltl p6{!<>consist3}
ltl p7{!<>consist4}
```

รูปที่ ก-19 ตรรกะเวลาเชิงเส้นคุณสมบัติต้องกันจากข้อที่ 5.1.4

### ก.4.5 ตรรกะเวลาเชิงเส้นคุณสมบัติการกำหนดสถานะที่สมบูรณ์ของวงจรอสมวารอินพุต

```
#define csc_full_sr (sp==1 && sm==1 && rp==1 && rm==1 && (smoutp1==4 || outm2enp==4))
ltl c1 {!<> csc_full_sr}
#define csc_full_sen (sp==1 && sm==1 && enp==1 && enm==1 && (smoutp1==4 ||
outm2enp==4))
ltl c2 {!<> csc_full_sen}
#define csc_full_sout (sp==1 && sm==1 && outp==1 && outm==1 && (smoutp1==4 ||
outm2enp==4))
ltl c3 {!<> csc_full_sout}
#define csc_full_ren (rp==1 && rm==1 && enp==1 && enm==1 && (smoutp1==4 ||
outm2enp==4))
ltl c4 {!<> csc_full_ren}
#define csc_full_rout (rp==1 && rm==1 && outp==1 && outm==1 && (smoutp1==4 ||
outm2enp==4))
ltl c5 {!<> csc_full_rout}
#define csc_full_enout (enp==1 && enm==1 && outp==1 && outm==1 && (smoutp1==4 ||
outm2enp==4))
ltl c6 {!<> csc_full_enout}
```

รูปที่ ก-20 ตรรกะเวลาเชิงเส้นการกำหนดสถานะที่สมบูรณ์จากข้อที่ 5.1.4

## ก.5 รหัสโพรเมลาและตรรกะเวลาเชิงเส้นจากวงจรอสมวารทดลอง 1

### ก.5.1 รหัสโพรเมลาแบบ A1 ของวงจรอสมวารทดลอง 1

```
byte cmam, ambm, ambp, bmcp, bpcp, cpap, apcm;
byte t1,t2,t3,t4,t5,t6,t7;
#define inp1(x1,t1)          (x1>0) -> x1 = x1-1; t1 = t1+1
#define inp2(x1,x2,t1,t2)  (x1>0 && x2>0) -> x1 = x1-1; x2 = x2-1; t1 = t1+1; t2 = t2+1
#define outp1(x1)            x1 = x1+1
#define outp2(x1,x2)         x1 = x1+1; x2 = x2+1
init {
cmam=1;bmcp=1;
do
        ::atomic{ inp1(apcm,t1) -> outp1(cmam)}
        ::atomic{ inp1(cmam,t2) -> outp2(ambm,ambp)}
        ::atomic{ inp1(ambm,t3) -> outp1(bmcp)}
        ::atomic{ inp1(ambp,t4) -> outp1(bpcp)}
        ::atomic{ inp2(bmcp,bpcp,t5,t6) -> outp1(cpap)}
        ::atomic{ inp1(cpap,t7) -> outp1(apcm)}
od
}
```

รูปที่ ก-21 รหัสโพรเมลาแบบ A1 วงจรอสมวารอสมวารทดลอง 1 จากข้อที่ 5.1.5



### ก.5.2 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานวงจรอสมวารทดลอง 1

```
#define safe ((cmam <= 1) && (ambm <= 1) && (ambp <= 1) && (bmcp <= 1) && (bpcp <= 1) &&
(cpap <= 1) && (apcm <= 1) )
ltl p1 { [] safe }
#define live ((t1>1) && (t2>1) && (t3>1) && (t4>1) && (t5>1) && (t6>1) && (t7>1))
ltl p2 { <> live }
#define persist (((apcm == 1) -> ((ambm == 0) && (ambp == 0))))
ltl p3 { [] persist }
```

รูปที่ ก-22 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานจากข้อที่ 5.1.5

### ก.5.3 รหัสโพรเมลาแบบ A2 ของวงจรอสมวารทดลอง 1

```
byte cmam, ambm, ambp, bmcp, bpcp, cpap, apcm;
byte ap, am, bp, bm, cp, cm;
#define inp(x,y) if ::(x==1) -> x=10;y++; ::(x==2) -> x++;y++; fi
#define outp1(x1)          if :: (x1 != 10) -> x1++; :: else -> break; fi
#define outp2(x1,x2)       if :: (x1 != 10) -> x1++; :: ( x2 != 10) -> x2++; :: else -> break; fi
proctype P3 (byte k){
 atomic{
  if
   :: k==1 -> cmam=2;bmcp=10;ambm=0;ambp=0;bpcp=0;cpap=0;apcm=0;ap=0;am=0;bp=0;
bm=0;cp=0;cm=0;
   :: k==2 -> bmcp=2;cmam=10;ambm=0;ambp=0;bpcp=0;cpap=0;apcm=0;ap=0;am=0;bp=0;
bm=0;cp=0;cm=0;
  fi
  do
          ::atomic{ inp(apcm,ap);printf("printf(ap=%d)",ap) -> outp1(cmam)}
          ::atomic{ inp(cmam,cm);printf("printf(cm=%d)",cm) -> outp2(ambm,ambp)}
          ::atomic{ inp(ambm,am);printf("printf(am=%d)",am) -> outp1(bmcp)}
          ::atomic{ inp(ambp,am);printf("printf(am=%d)",am) -> outp1(bpcp)}
          ::atomic{ inp(bmcp,bm);printf("printf(bm=%d)",bm) -> outp1(cpap)}
          ::atomic{ inp(bpcp,bp);printf("printf(bp=%d)",bp) -> outp1(cpap)}
          ::atomic{ inp(cpap,cp);printf("printf(cp=%d)",cp) -> outp1(apcm)}
  od
}}
init{
  run P3(1);
  run P3(2);
}
```

รูปที่ ก-23 รหัสโพรเมลาแบบ A2 วงจรอสมวารทดลอง 1จากข้อที่ 5.1.5

### ก.5.4 ตรรกะเวลาเชิงเส้นคุณสมบัติความต้องกันของวงจรอสมวารทดลอง 1

```
#define consist1 (ap>=1 && am>=1 && (cmam==4 || bmcp==4))
#define consist2 (bp>=1 && bm>=1 && (cmam==4 || bmcp==4))
#define consist3 (cp>=1 && cm>=1 && (cmam==4 || bmcp==4))
ltl p4{!<>consist1}
ltl p5{!<>consist2}
ltl p6{!<>consist3}
```

รูปที่ ก-24 ตรรกะเวลาเชิงเส้นคุณสมบัติต้องกันจากข้อที่ 5.1.5

### ก.5.5 ตรรกะเวลาเชิงเส้นคุณสมบัติการกำหนดสถานะที่สมบูรณ์ของวงจรอสมวารทดลอง 1

```
#define csc_full_ab (ap==1 && am==1 && bp==1 && bm==1 && (cmam==4 || bmcp==4))
ltl c1 {!<> csc_full_ab}
#define csc_full_ac (ap==1 && am==1 && cp==1 && cm==1 && (cmam==4 || bmcp==4))
ltl c2 {!<> csc_full_ac}
#define csc_full_bc (bp==1 && bm==1 && cp==1 && cm==1 && (cmam==4 || bmcp==4))
ltl c3 {!<> csc_full_bc}
#define transitive1 ((ap==1 || am==1) && (cp==1 || cm==1) && (bp==1 && bm==1) && (cmam==4
|| bmcp==4))
ltl t1 {!<>transitive1}
```

รูปที่ ก-25 ตรรกะเวลาเชิงเส้นการกำหนดสถานะที่สมบูรณ์จากข้อที่ 5.1.5

## ก.6 รหัสโพรเมลาและตรรกะเวลาเชิงเส้นจากวงจรอสมวารทดลอง 2

### ก.6.1 รหัสโพรเมลาแบบ A1 ของวงจรอสมวารทดลอง 2

```
byte xmym, ymxp, ymrp, xpyp, ypxm, rprm;
byte t1,t2,t3,t4,t5;
#define inp1(x1,t1)     (x1>0) -> x1 = x1-1; t1 = t1+1
#define outp1(x1)       x1 = x1+1
#define outp2(x1,x2)    x1 = x1+1; x2 = x2+1
init {
ypxm=1;
do
        ::atomic{ inp1(ypxm,t1) -> outp1(xmym)}
        ::atomic{ inp1(xmym,t2) -> outp2(ymxp,ymrp)}
        ::atomic{ inp1(ymxp,t3) -> outp1(xpyp)}
        ::atomic{ inp1(xpyp,t4) -> outp1(ypxm)}
        ::atomic{ inp1(ymrp,t5) -> outp1(rprm)}
od
}
```

รูปที่ ก-26 รหัสโพรเมลาแบบ A1 วงจรอสมวารอสมวารทดลอง 2 จากข้อที่ 5.1.6



### ก.6.2 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานวงจรอสมวารทดลอง 2

```
#define safe ((xmym <= 1) && (ymxp <= 1) && (ymrp <= 1) && (xpyp <= 1) && (ypxm
<= 1) && (rprm <= 1) )
ltl p1 { [] safe }
#define live ((t1>1) && (t2>1) && (t3>1) && (t4>1) && (t5>1) )
ltl p2 { <> live }
#define persist (((ypxm == 1) -> ((ymxp == 0) && (ymrp == 0))))
ltl p3 { [] persist }
```

รูปที่ ก-27 ตรรกะเวลาเชิงเส้นคุณสมบัติความปลอดภัย ไลฟ์เนส และความทนทานจากข้อที่ 5.1.6

# ประวัติผู้เขียนวิทยานิพนธ์

นายคณุตม์ บุญเรืองขาว เกิดเมื่อวันที่ 3 กรกฎาคม พ.ศ. 2535 ภูมิลำเนาเป็นคนตำบลบางพึ่ง อำเภอพระประแดง จ.สมุทรปราการ สำเร็จการศึกษาระดับปริญญาตรี หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเกษตรศาสตร์ ปีการศึกษา 2556 และเข้าศึกษาต่อระดับปริญญามหาบัณฑิต ปีการศึกษา 2558 หลักสูตรวิศวกรรมศาสตรมหาบัณฑิต (วศ.ม) สาขาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย