

บทที่ 4

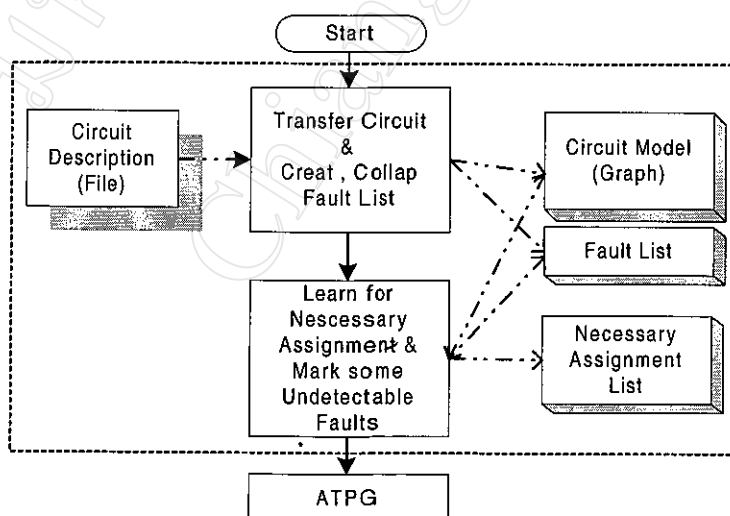
การพัฒนาโปรแกรม เอ.ที.พี.จี. โดยกำหนค่าที่จำเป็น

กระบวนการในการพัฒนาโปรแกรม เอ.ที.พี.จี. โดยกำหนค่าที่จำเป็นแบ่งออกเป็น 3 ขั้นตอนใหญ่ ๆ คือ

- ขั้นตอนการเตรียมการก่อนการให้กำเนิดรูปแบบสัญญาณทดสอบ
- ขั้นตอนการให้กำเนิดรูปแบบสัญญาณทดสอบ
- ขั้นตอนการประเมินผลประสิทธิภาพของรูปแบบสัญญาณทดสอบและ โปรแกรม

4.1 การเตรียมการก่อนการให้กำเนิดรูปแบบสัญญาณทดสอบ

ก่อนการให้กำเนิดรูปแบบสัญญาณทดสอบโปรแกรม เอ.ที.พี.จี. ต้องเตรียมโครงสร้างข้อมูลและข้อมูลที่เป็นสำหรับการให้กำเนิดรูปแบบสัญญาณทดสอบ งานหลักที่ต้องทำในขั้นตอนนี้ได้แก่ การสร้างแบบจำลองวงจรดิจิทัลที่จะทำการทดสอบ การสร้างและยุบรวมรายการจุดเสีย และอีกงานหนึ่งที่ได้เสนอขึ้นมา คือ การเรียนรู้เพื่อเตรียมข้อมูลสำหรับการกำหนค่าที่จำเป็นและสำหรับการระบุจุดเสียที่ไม่สามารถตรวจสอบได้บางส่วนก่อนการให้กำเนิดรูปแบบสัญญาณทดสอบ ระบบงานในขั้นตอนการเตรียมการก่อนการให้กำเนิดรูปแบบสัญญาณทดสอบแสดงได้ดังภาพที่ 4.1 สิ่งที่ได้จากงานในขั้นตอนนี้คือแบบจำลองวงจร รายการจุดเสียและรายการสำหรับการกำหนค่าที่จำเป็น ซึ่งจะถูกใช้งานตลอดทั้ง 3 ขั้นตอน



ภาพที่ 4.1 งานในขั้นตอนก่อนการให้กำเนิดรูปแบบสัญญาณทดสอบ

4.1.1 การสร้างแบบจำลองวงจรดิจิทัล

การให้กำเนิดรูปแบบสัญญาณทดสอบของโปรแกรมเอทีพีจีเริ่มจากการอ่านเพิ่มข้อมูลบรรยายวงจรดิจิทัลที่ต้องการทดสอบแล้วทำการสร้างแบบจำลองวงจรทางคอมพิวเตอร์ที่สามารถเลียนแบบพฤติกรรมของวงจรดิจิทัลดังกล่าวได้

ก) เพิ่มข้อมูลบรรยายวงจรดิจิทัล

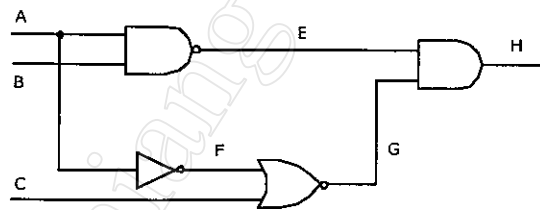
เนื้อหาของเพิ่มข้อมูลบรรยายวงจรบอกรายละเอียดของวงจรดิจิทัลว่ามีขาอินพุตหลัก เอาต์พุตหลัก และเกตอะไรบ้าง ต่อกันอย่างไร งานที่พัฒนาขึ้นนี้ใช้เพิ่มบรรยายวงจรมาตรฐาน ISCAS Benchmark ดังตัวอย่างเนื้อหาของเพิ่มบรรยายวงจรในภาพที่ 4.2 ที่บรรยายถึงวงจรดิจิทัลในภาพที่ 4.3

```
#Example Circuit Description

INPUT(A)
INPUT(B)
INPUT(C)
OUTPUT(H)

E=NAND(A, B)
F=NOT(A)
G=NOR(F,C)
H=AND(E, G)
```

ภาพที่ 4.2 เนื้อหาของเพิ่มข้อมูลบรรยายวงจรดิจิทัล



ภาพที่ 4.3 วงจรดิจิทัลตัวอย่าง

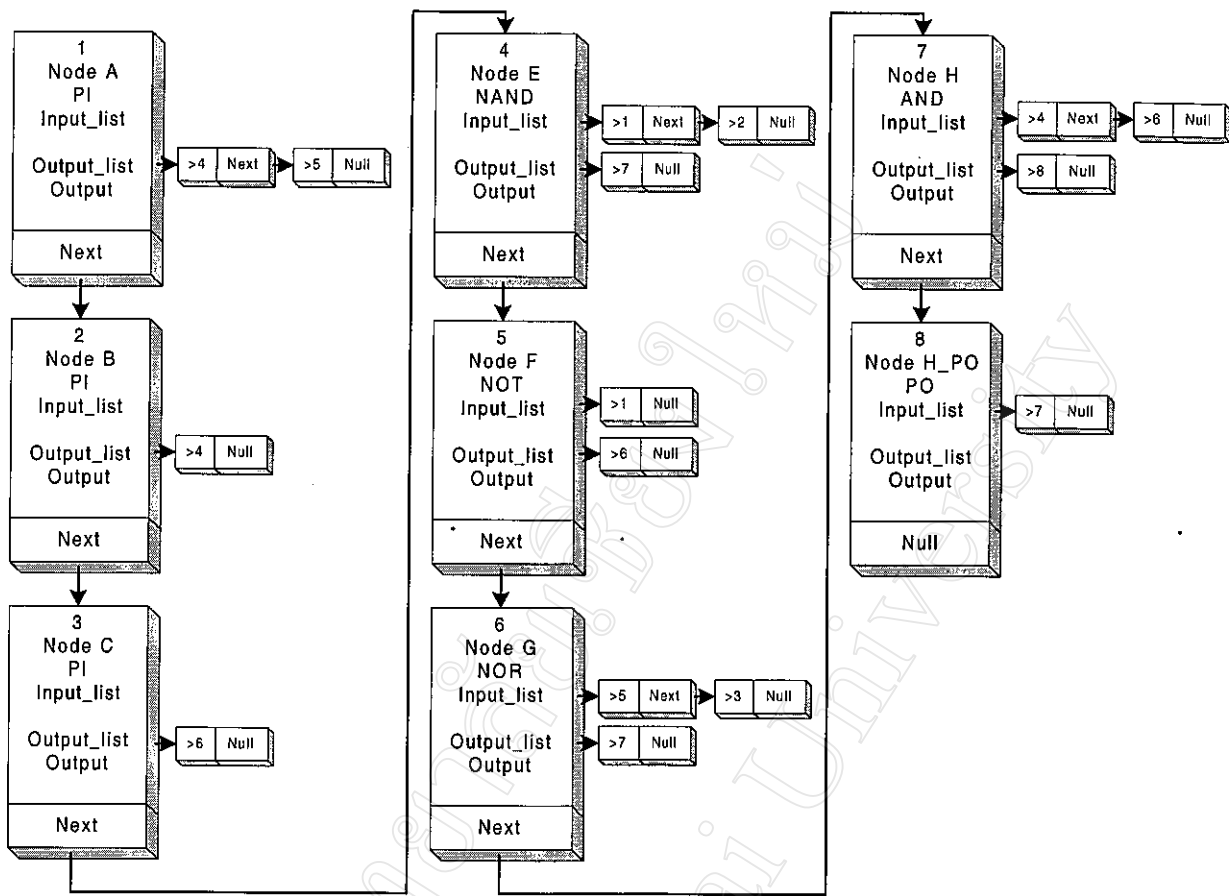
ข) แบบจำลองวงจรดิจิทัล

ใช้โครงสร้างข้อมูลชนิดกราฟเป็นแบบจำลองวงจรดิจิทัล โหนดของกราฟแทนโหนดของวงจร ในแต่ละโหนดจะเก็บข้อมูลต่าง ๆ ของโหนด เช่น ชื่อ ชนิดหรือฟังก์ชัน จำนวนและรายการขาอินพุตและขาเอาต์พุต ค่าตรรก และอื่น ๆ ดังรายการเขตข้อมูล (Field) ต่าง ๆ ในตารางที่ 4.1 ตัวอย่างวงจรดิจิทัลในภาพที่ 4.3 เมื่อสร้างแบบจำลองจะได้โครงสร้างข้อมูลชนิดกราฟดังภาพที่ 4.4

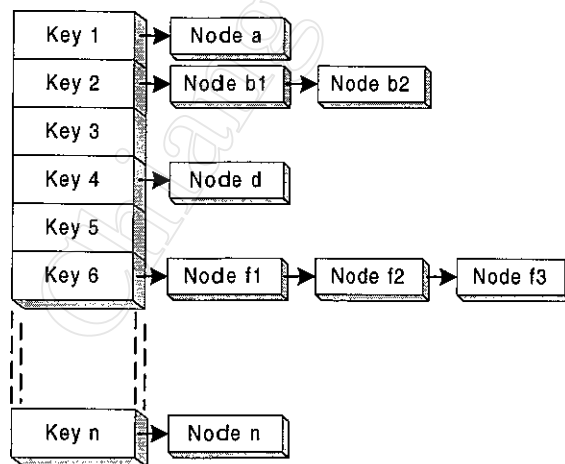
ตารางที่ 4.1 เขตข้อมูลสำคัญ ๆ ของกราฟจำลองวงจรดิจิทัล

เขตข้อมูล	คำอธิบาย
Node_Id	หมายเลขประจำโหนด
Node	ชื่อโหนด
Function	ชนิดของโหนด
N_input	จำนวนอินพุตของโหนด
Input List	รายการขั้วอินพุตของโหนด
N_output	จำนวนเอาต์พุต(อาจมากกว่า 1 กรณีแฟนเอาต์)
Output List	รายการขั้วเอาต์พุตของโหนด
Output Value	ค่าตรรกเอาต์พุตของโหนด
Ptr_Fault	รายการจุดเสียของโหนด
NecAssLst	รายการข้อมูลสำหรับการกำหนดค่าที่จำเป็นของโหนด
Next	โหนดถัดไป

แต่เนื่องจาก วงจร VLSI เป็นวงจรขนาดใหญ่มากประกอบด้วยโหนดซับซ้อนมากมาย ขั้วเอาต์พุตของเกตหนึ่ง ๆ อาจต่อเข้าเกตอื่นมากกว่า 1 เกต เนื้อหาของแฟ้มบรรยายวงจรดิจิทัลย่อมมีโหนดใดโหนดหนึ่งปรากฏซ้ำ ๆ หลายครั้ง ดังนั้นก่อนนำโหนดของวงจรมาสร้างเป็นโหนดของกราฟนั้นต้องตรวจสอบทุกครั้งว่าโหนดนั้น ๆ มีอยู่ในกราฟหรือยัง หากมีอยู่แล้วไม่ต้องสร้างโหนดขึ้นใหม่เพียงแต่ต้องแก้ไขข้อมูลที่เกี่ยวข้องบางส่วนเท่านั้น จึงนำตารางแฮชชิง (Hashing Table) ซึ่งมีประสิทธิภาพในการค้นหาข้อมูลสูงมากมาใช้งานร่วมกับกราฟ ดังภาพที่ 4.5 แสดงโครงสร้างของตารางแฮชชิงที่ใช้ร่วมกับกราฟจำลองวงจร การเก็บหรือการค้นหาข้อมูลของตารางแฮชชิงทำโดยการนำชื่อของโหนดแต่ละโหนดไปเข้าฟังก์ชันแฮชแล้วจะได้ค่าเป็นตำแหน่งที่อยู่ของแต่ละโหนดในตารางแฮชชิง ทำให้สามารถค้นหาข้อมูลได้อย่างรวดเร็ว [24]



ภาพที่ 4.4 กราฟจำลองวงจรดิจิทัล



ภาพที่ 4.5 โครงสร้างของตารางแฮชซึ่งใช้ร่วมกับกราฟจำลองวงจร

4.1.2 การยุบรวมจุดเสีย

ทุกโหนดของวงจรมีจุดเสีย 2 จุดคือ $s-a-0$ และ $s-a-1$ แต่เมื่อทำการยุบรวมจุดเสียแล้ว รายการจุดเสียจะเก็บเฉพาะจุดเสียที่เป็นตัวแทนของกลุ่มจุดเสียเท่านั้น งานที่พัฒนาขึ้นนี้ใช้การยุบรวมแบบสมมูลเพียงอย่างเดียว โดยมีคำสั่งเทียมในการสร้างรายการจุดเสียดังภาพที่ 4.6

PROCEDURE Create_Faults_List()

Begin

For all gate G

Begin

{ Input Faults }

for all Input of the G

Begin

If the Input of the G has >1 fan-outs Then

Begin

If G is OR, NOR, AND, or NAND gate Then

{ c is Controlling Value of gate G }

Add $s-a-\bar{c}$ Fault to the input of G

Else

Add both $s-a-c$ and $s-a-\bar{c}$ Fault to the input of G

End

End

{ Output Faults }

If the G has 1 Output and (a successor of G is PO or has >1 input) Then

Begin

If the successor of G is OR, NOR, AND, or NAND gate Then

Add $s-a-\bar{c}$ Fault to the output of the G

Else

Add both $s-a-c$ and $s-a-\bar{c}$ Fault to the output of the G

End

{ Fan-out Faults }

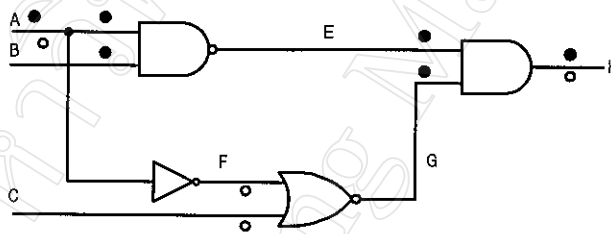
Else If G is Fan-out Then

Add both $s - a - c$ and $s - a - \bar{c}$ Fault to output of G

End.

ภาพที่ 4.6 คำสั่งเทียมการสร้างรายการจุดเสียทำการยุบรวมจุดเสียแบบสมมูล

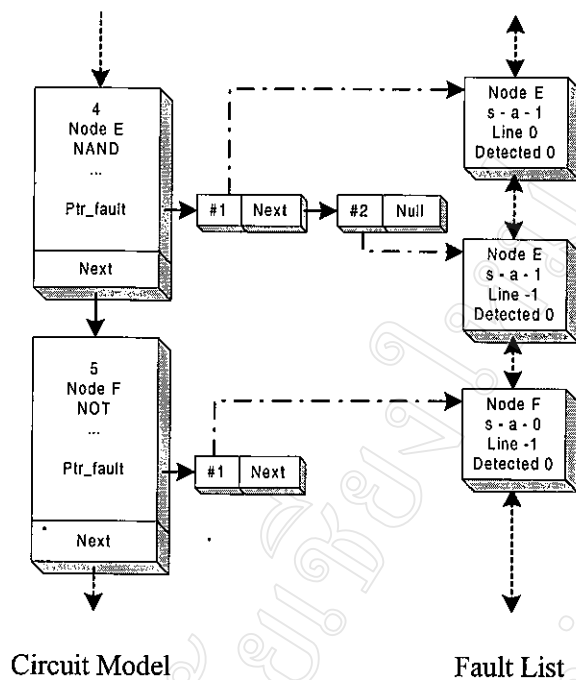
เมื่อผ่านการยุบรวมจุดเสียแล้วจะเหลือจุดเสียเฉพาะที่เป็นตัวแทนของกลุ่มเท่านั้น เช่น จากวงจรในภาพที่ 4.3 เมื่อผ่านการยุบรวมแล้วเหลือจุดเสียที่เป็นตัวแทนดังภาพที่ 4.7 แต่ละโหนดมีจุดเสียแตกต่างกันไป เช่น โหนด E มีจุดเสีย 2 จุด คือที่ขั้วอินพุตเส้นที่ 0 และที่ขั้วเอาต์พุต เป็นจุดเสีย s-a-1 ทั้งคู่ โปรแกรมจะรวบรวมจุดเสียตัวแทนเหล่านี้เข้าด้วยกันเป็นรายการจุดเสีย รายการจุดเสียใช้โครงสร้างข้อมูลแบบลิงค์ลิสต์ 2 ทิศทาง โดยมีเขตข้อมูลสำคัญ ๆ ดังตารางที่ 4.2 และในแบบจำลองวงจรแต่ละโหนดจะมีเขตข้อมูล Ptr_Fault ซึ่งไปยังรายการจุดเสียของโหนดนั้น ๆ ดังภาพที่ 4.8 แสดงรายการจุดเสียส่วนที่เป็นจุดเสียของโหนด E และโหนด F



ภาพที่ 4.7 จุดเสียตัวแทนที่ผ่านการยุบรวมจุดเสียแบบสมมูล

ตารางที่ 4.2 เขตข้อมูลสำคัญของรายการจุดเสีย

เขตข้อมูล	คำอธิบาย
Node	โหนดที่เกิดจุดเสีย
Type	ชนิดจุดเสีย
Line	หมายเลขขั้วที่เกิดจุดเสีย
Detected	ตรวจสอบได้หรือยัง
Previous	ชี้ไปยังรายการจุดเสียจุดก่อนหน้า
Next	ชี้ไปยังรายการจุดเสียจุดถัดไป



ภาพที่ 4.8 รายการจุดเสียของโหนด E และ โหนด F

4.1.3 การเรียนรู้เพื่อเตรียมข้อมูลสำหรับการกำหนดที่จำเป็น

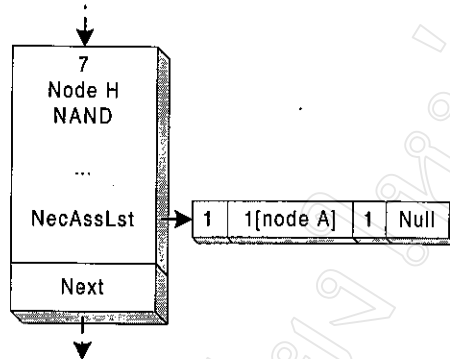
โหนดที่ต้องมีการกำหนดค่าที่จำเป็นคือโหนดที่มีความสัมพันธ์ทางตรรกกับโหนดอื่น ดังนั้นข้อมูลสำหรับการกำหนดค่าที่จำเป็นก็คือข้อมูลความสัมพันธ์ทางตรรก ซึ่งเป็นข้อมูลที่บอกว่า หากต้องการกำหนดให้โหนดใดโหนดหนึ่งมีค่าตรรกเป็น “0” หรือ “1” ต้องกำหนดโหนดอื่นที่สัมพันธ์กันให้มีค่าตรรกเป็นอย่างไร จึงทำการเรียนรู้เพื่อสร้างรายการข้อมูลสำหรับการกำหนดค่าที่จำเป็นสำหรับแต่ละโหนด โดยมีเขตข้อมูลดังตารางที่ 4.3

ตารางที่ 4.3 เขตข้อมูลสำคัญของรายการข้อมูลสำหรับการกำหนดที่จำเป็น

เขตข้อมูล	คำอธิบาย
Source Value	ค่าตรรกของโหนดปัจจุบัน
Target Node	หมายเลขโหนดที่ต้องกำหนดค่าที่จำเป็น
Target Value	ค่าตรรกที่ต้องกำหนดค่าที่จำเป็น
Next	รายการข้อมูลสำหรับการกำหนดที่จำเป็นถัดไป

เช่น ข้อมูลสำหรับการกำหนดที่จำเป็นของ H/1 คือ A/1 ดังนั้นที่โหนด H ต้องมีรายการข้อมูลสำหรับการกำหนดที่จำเป็นประกอบด้วยค่าตรรกที่ต้องการกำหนดให้กับโหนด H คือ “1”

โหนด A และค่าตรรกะที่ต้องกำหนดให้กับโหนด A คือ “1” แสดงได้ดังรูปที่ 4.9 ส่วนในภาพที่ 4.10 แสดงคำสั่งเทียมสำหรับการเรียนรู้เพื่อเตรียมข้อมูลสำหรับการกำหนดค่าที่จำเป็น



ภาพที่ 4.9 รายการข้อมูลสำหรับการกำหนดค่าที่จำเป็นของโหนด H

Procedure Learning_for_Necessary_Assignment()

Begin

For all Fan-out node

Begin

For all branch node

Begin

If the branch node is OR, NOR, AND, or NAND gate then

Begin

Forward_Implication(N : Fan-out, cv : Controlling value)

For all node M which has value w or $\bar{w}$ $\{cv, w \in \{0,1\}\}$

Begin

{Using contra-positive make Necessary Assignment Set of node M }

If Output Value of $M = w$ Then

Begin

$\{[M / \bar{w}] \Rightarrow [N / cv]\}$

Add $[N / cv]$ to $nec_ass([M / \bar{w}])$

{Check Conflict Assignment Relation}

If $[N / cv]$ belongs to $nec_ass([M / \bar{w}])$

Add $[M / \bar{w}]$ to Uncontrollable Set


```

End
Else
Begin
    { $[M/w] \Rightarrow [N/\overline{cv}]$ }
    Add  $[N/\overline{cv}]$  to  $nec\_ass([M/w])$ 
    {Check Conflict Assignment Relation}
    If  $[N/cv]$  belongs to  $nec\_ass([M/w])$ 
        Add  $[M/w]$  to Uncontrollable Set
    End
End
End
End
End.

```

ภาพที่ 4.10 คำสั่งเทียมสำหรับการสร้างรายการข้อมูลสำหรับการกำหนดค่าที่จำเป็น

4.1.4 การระบุจุดเสียที่ไม่สามารถตรวจสอบได้

จากการเรียนรู้เพื่อสร้างรายการข้อมูลสำหรับการกำหนดค่าที่จำเป็นในบางวงจรอาจพบว่ามีโหนดบางโหนดที่ไม่สามารถกำหนดให้มีค่าตรรกค่าใดค่าหนึ่งได้ โหนดเหล่านี้จะถูกเก็บไว้ในกลุ่มไม่สามารถควบคุมค่าตรรกได้ (Uncontrollable Set) ซึ่งโครงสร้างข้อมูลของกลุ่มนี้จะประกอบด้วยเขตข้อมูลต่าง ๆ ดังตารางที่ 4.4

ดังที่กล่าวไว้ในบทที่ 3 กลุ่มจุดเสียที่ไม่สามารถตรวจสอบได้เนื่องจากไม่สามารถกำหนดให้โหนด (N) บางโหนดมีค่าตรรก (v) บางค่าได้ (N/v) แบ่งออกเป็น 2 กลุ่มคือ กลุ่มจุดเสียที่ต้องการ N/v สำหรับการกระตุ้นจุดเสียให้มีค่าตรรกตรงข้ามกับค่าที่เกิดจากจุดเสีย กับกลุ่มจุดเสียที่ต้องการ N/v สำหรับการแพร่ค่าให้ไปปรากฏที่ขั้วเอาต์พุต ในกลุ่มแรกสามารถทำได้ด้วยการแพร่ค่า N/v ไปข้างหน้าแล้วส่วนกลุ่มหลังนั้นสามารถใช้วิธีการค้นหาข้อมูลในกราฟ ซึ่งสามารถใช้ได้ทั้งค้นหาแนวลึก (Depth - First Search) หรือแนวกว้าง (Breadth - First Search) คำสั่งเทียมสำหรับการระบุจุดเสียที่ไม่สามารถตรวจสอบได้แสดงดังภาพที่ 4.11

ตารางที่ 4.4 เขตข้อมูลสำคัญของโหนดที่ไม่สามารถกำหนดค่าตรรกได้

เขตข้อมูล	คำอธิบาย
Node	โหนดที่ไม่สามารถกำหนดค่าได้
Value	ค่าตรรกที่ไม่สามารถกำหนดได้
Next	ชี้ไปรายการถัดไป

Procedure Mark_Undetectable_Fault()

Begin

For all element M/w in Uncontrollable Set

Begin

Remove Faults that require M/w as a Necessary Condition for Fault Activation

{Using Implication}

Remove Faults that require M/w as a Necessary Condition for Fault Propagation

{Using Searching in Graph}

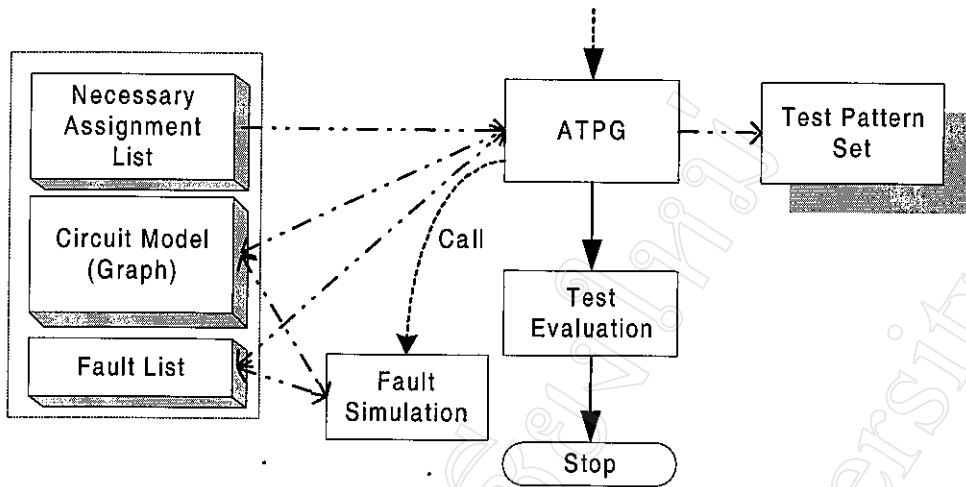
End

End.

ภาพที่ 4.11 คำสั่งเทียมสำหรับระบุจุดเสียที่ไม่สามารถตรวจสอบได้

4.2 การให้กำเนิดรูปแบบสัญญาณทดสอบ

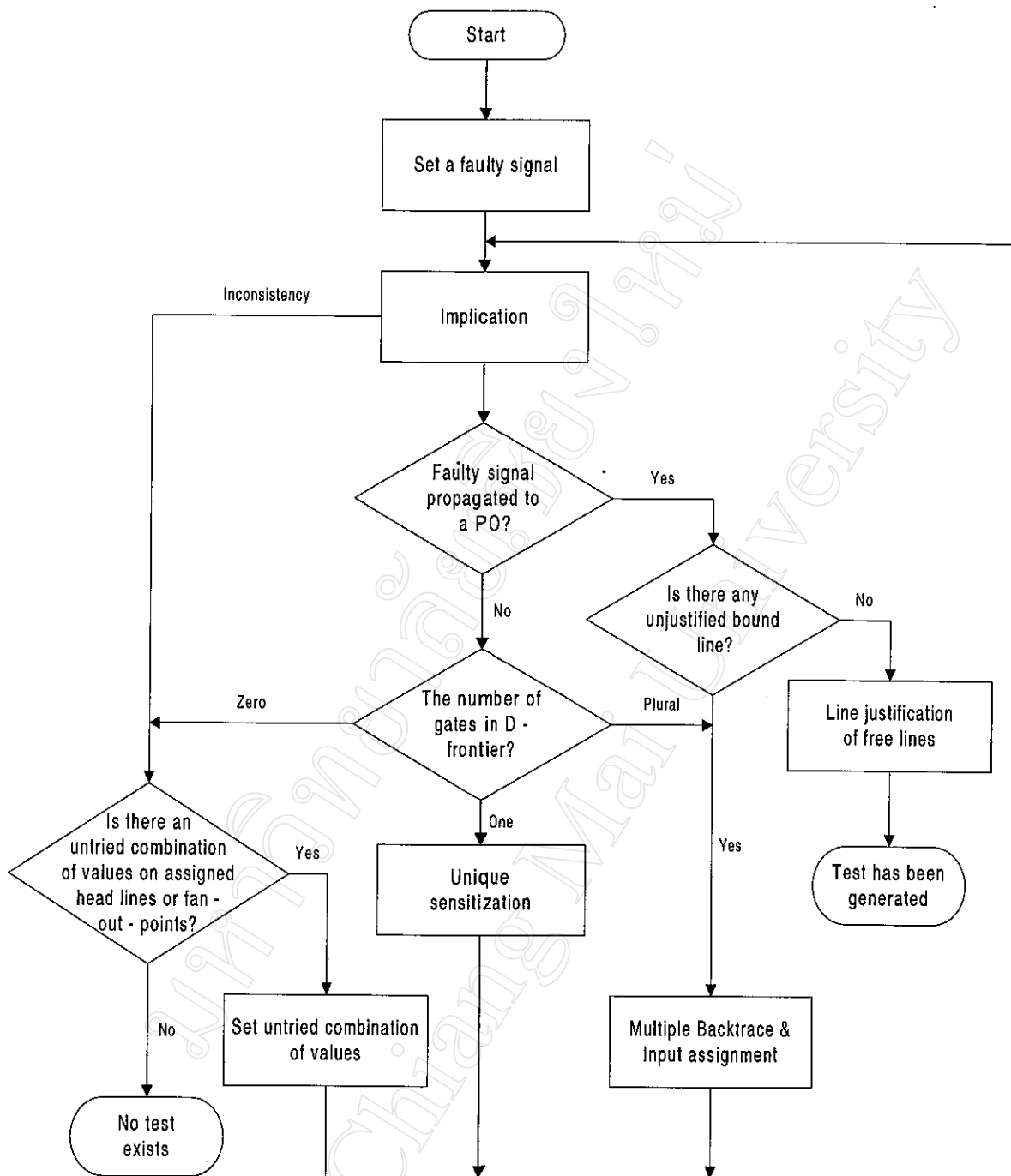
กระบวนการให้กำเนิดรูปแบบสัญญาณทดสอบทั้งแบบสุ่มและแบบระบุจุดเสียนั้นหลังจากที่ได้รูปแบบสัญญาณทดสอบแล้วจะเรียกใช้กระบวนการจำลองการทำงานของจุดเสียทุกครั้งเพื่อหาจุดเสียที่สามารถใช้รูปแบบสัญญาณทดสอบนี้ตรวจสอบได้ หากจุดเสียใดสามารถตรวจสอบได้จะทำการระบุไว้ในรายการจุดเสียที่ตรวจสอบได้แล้ว รูปแบบสัญญาณทดสอบที่ได้จะถูกเก็บไว้ในแฟ้มข้อมูลสัญญาณทดสอบ เมื่อพยายามให้กำเนิดรูปแบบสัญญาณทดสอบครบทุกจุดแล้วทำการประเมินผลประสิทธิภาพของรูปแบบสัญญาณทดสอบว่าครอบคลุมจุดเสียมากน้อยเพียงใด โดยคำนวณจากอัตราส่วนระหว่างจำนวนจุดเสียที่สามารถตรวจสอบได้ต่อจำนวนจุดเสียทั้งหมด ที่เรียกว่าค่าครอบคลุมจุดเสีย ภาพที่ 4.12 แสดงระบบงานการให้กำเนิดรูปแบบสัญญาณทดสอบ



ภาพที่ 4.12 ระบบงานการให้กำเนิดรูปแบบสัญญาณทดสอบ

4.2.1 การให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรชนิดประกอบ

การให้กำเนิดรูปแบบสัญญาณทดสอบโดยการกำหนดค่าที่จำเป็นพัฒนาขึ้นบนพื้นฐานของหลักวิธี FAN ซึ่งมีผังงานดังภาพที่ 4.13 ทำการแก้ไขเพิ่มเติมในส่วนของการแพร่ค่าตรรกะไปข้างหลัง โดยใช้การกำหนดค่าที่จำเป็น หากพบว่าโหนดที่ต้องการกำหนดค่ามีข้อมูลสำหรับการกำหนดค่าที่จำเป็นให้กำหนดค่าที่จำเป็นเหล่านั้นให้เรียบร้อยแล้วค่อยแพร่ค่าตรรกะด้วยวิธีทั่วไป ซึ่งมีคำสั่งเทียมดังภาพที่ 4.14

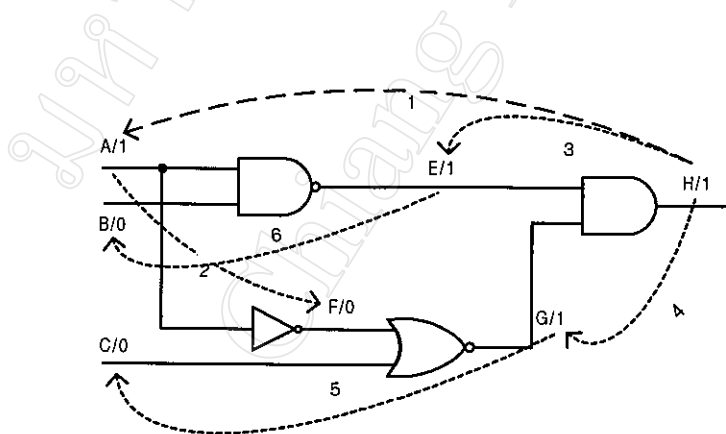


ภาพที่ 4.13 ฟังก์ชันของหลักวิธี FAN [10]

Procedure Backward_Implication(Node)
Begin
If Node has the Necessary Assignment Then
Begin
Implication using Necessary Assignment
Forward Implication
End
Tradition Implication
End.

ภาพที่ 4.14 คำสั่งเทียมในส่วนของการแพร่ค่าตรรกะไปข้างหลัง

ตัวอย่างเช่นวงจรในภาพที่ 4.3 หากต้องการกำหนด H/1 ซึ่งมีข้อมูลสำหรับค่ากำหนดที่จำเป็นคือ A/1 เมื่อแพร่ค่าตรรกะไปข้างหลังจะต้องกำหนด A/1 ก่อน ดังลูกศรหมายเลข 1 ในภาพที่ 4.15 เป็นการแพร่ค่าตรรกะไปข้างหลังด้วยการกำหนดค่าที่จำเป็น จากนั้นทำการแพร่ค่าตรรกะด้วยวิธีทั่วไปคือการคำนวณตามฟังก์ชันของเกตและค่าตรรกะระหว่างข้ออินพุตกับข้อเอาต์พุตซึ่งจะได้ F/0 E/1 G/1 B/0 และ C/0 ตามลำดับ



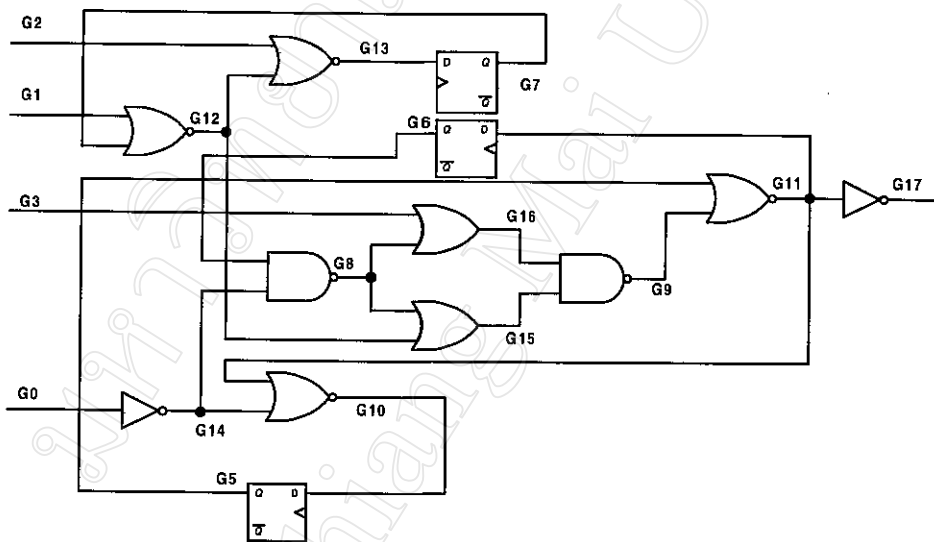
ภาพที่ 4.15 การปรับค่าให้โหนด H มีค่าตรรกะเท่ากับ “1” โดยใช้การกำหนดค่าที่จำเป็น

ข) การให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรชนิดลำดับ

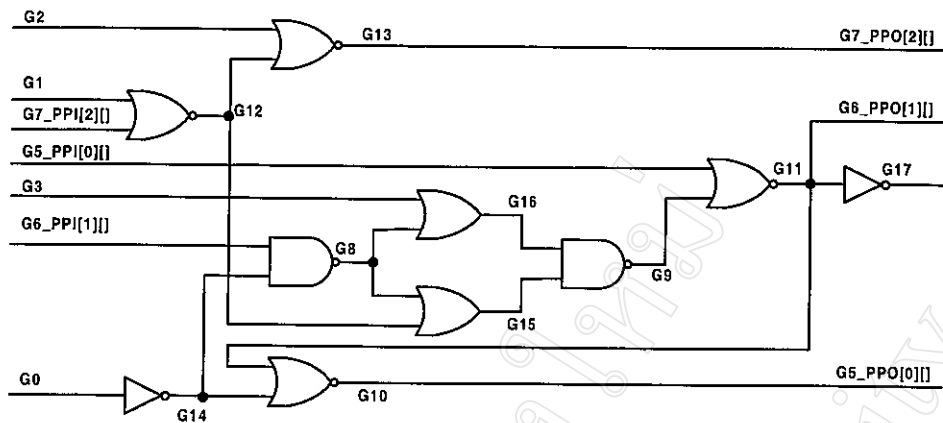
วิธีการให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรชนิดลำดับโดยทั่วไปมีหลักการสำคัญ คือ ทำการแปลงวงจรชนิดลำดับให้เป็นวงจรชนิดประกอบ และทำการให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรชนิดประกอบที่ได้

ขั้นตอนการให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรชนิดลำดับ

ขั้นตอนที่ 1) ทำการแปลงวงจรชนิดลำดับให้อยู่ในรูปแบบของวงจรชนิดประกอบ โดยตัดส่วนป้อนกลับหรือฟลิปฟล็อปออก ขั้วเอาต์พุตและอินพุตของฟลิปฟล็อปเปลี่ยนเป็นขั้วอินพุตและเอาต์พุตเทียมของวงจร เช่น วงจร S27.BENCH ดังภาพที่ 4.16 เมื่อทำการแปลงเป็นวงจรชนิดประกอบ ฟลิปฟล็อป G5 G6 และ G7 จะถูกตัดออกไป ขั้วเอาต์พุตของฟลิปฟล็อป (Q) กลายเป็นขั้วอินพุตเทียมของวงจร ให้ชื่อ G5_PPI[0][], G6_PPI[1][] และ G7_PPI[2][] ตามลำดับ วงเล็บหลังคือกรอบเวลา ส่วนขั้วอินพุตของฟลิปฟล็อป (D) กลายเป็นขั้วเอาต์พุตเทียมของวงจรเช่นกัน ได้แก่ G5_PPO[0][], G6_PPO[1][] และ G7_PPO[2][] ดังภาพที่ 4.17



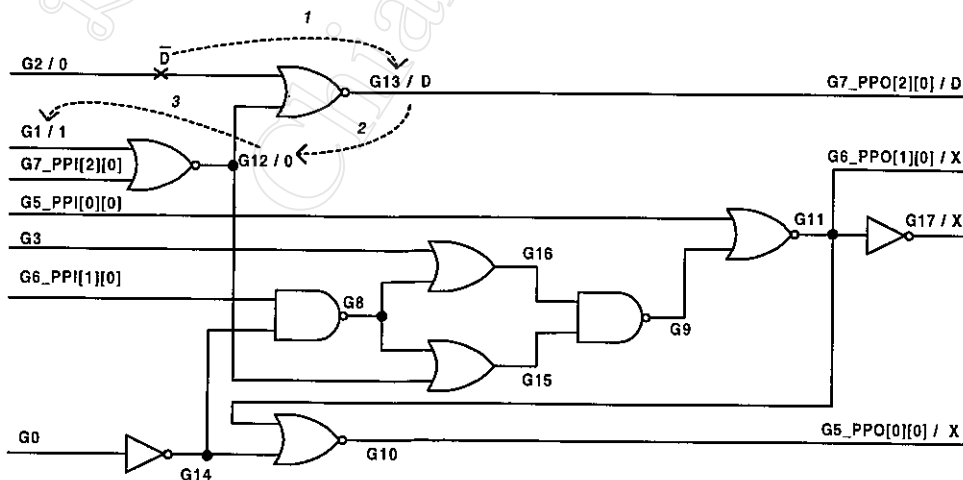
ภาพที่ 4.16 วงจร S27.BENCH



ภาพที่ 4.17 วงจรประกอบที่ได้จากการแปลงวงจร S27.BENCH

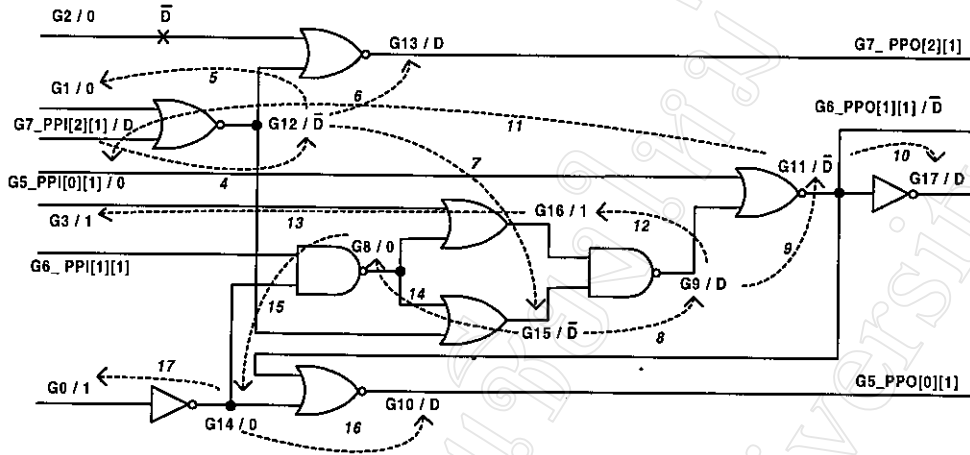
ขั้นตอนที่ 2) ทำให้กำหนดรูปแบบสัญญาณทดสอบสำหรับวงจรประกอบที่ได้ โดยเสมือนว่าขั้วอินพุตเทียมเป็นขั้วอินพุตหลักขั้วหนึ่ง หากไม่สามารถแพร่ค่าผลของจุดเสียให้ไปปรากฏที่ขั้วเอาต์พุตหลักได้ ให้เพิ่มกรอบเวลาขึ้นอีก 1 กรอบเวลา นำขั้วอินพุตเทียมของกรอบเวลาใหม่ต่อเข้ากับขั้วเอาต์พุตเทียมของกรอบเวลาก่อนหน้า แพร่ค่าตรรกต่าง ๆ จากขั้วเอาต์พุตเทียมในกรอบเวลาก่อนหน้าไปยังขั้วอินพุตเทียมของกรอบเวลาใหม่ พยายามแพร่ผลของจุดเสียให้ไปปรากฏที่ขั้วเอาต์พุตหลักของวงจรต่อไป หากยังไม่ปรากฏให้เพิ่มกรอบเวลาขึ้นอีกจนกว่าผลของจุดเสียจะไปปรากฏที่ขั้วเอาต์พุตหลักหรือเกินจำนวนกรอบเวลาที่จำกัดไว้

ตัวอย่างการให้กำหนดรูปแบบสัญญาณทดสอบสำหรับจุดเสีย G2/s-a-1 ในวงจร S27.BENCH เริ่มในกรอบเวลาแรก พบว่าไม่สามารถทำให้ผลของจุดเสียปรากฏที่ขั้วเอาต์พุตหลักได้ แต่อย่างไรก็ตามค่าผลของจุดเสียได้แพร่ถึง G7_PPO[2][0] ดังภาพที่ 4.18 จึงเพิ่มกรอบเวลาที่สองขึ้นมาแล้วพยายามแพร่ผลจุดเสียต่อไป



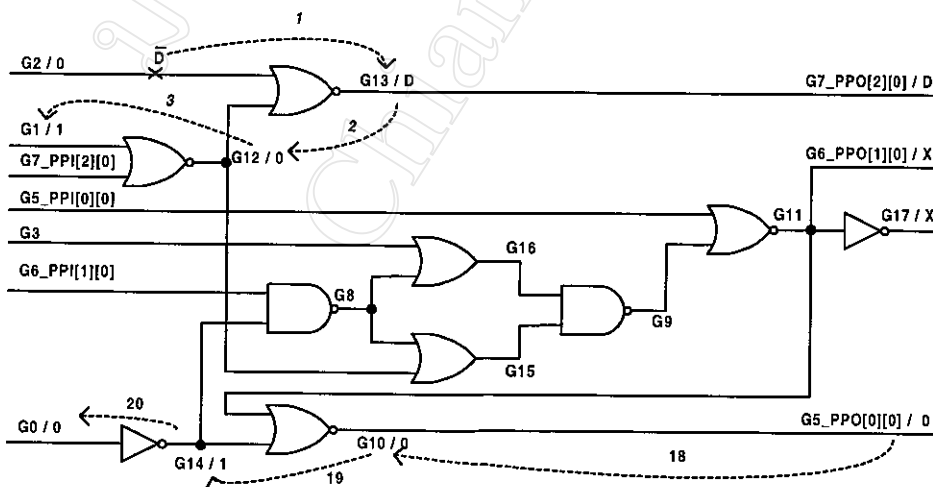
ภาพที่ 4.18 การให้กำหนดรูปแบบสัญญาณทดสอบในกรอบเวลาที่ 1

ในกรอบเวลาที่สองนอกจากจะมีจุดเสีย G2/s-a-1 แล้วยังมีผลของจุดเสีย (D) แพร่มาถึง G7_PPI[2][1] ด้วย ในกรอบเวลานี้สามารถแพร่ผลของจุดเสียปรากฏที่หัวเอาต์พุตหลักของวงจรได้ ดังภาพที่ 4.19



ภาพที่ 4.19 การให้กำเนิดรูปแบบสัญญาณทดสอบในกรอบเวลาที่ 2

แต่เนื่องจากหัวอินพุตเทียมไม่ใช่หัวอินพุตหลักของวงจรค่าตรรกะของหัวอินพุตเทียมไม่สามารถกำหนดได้ตามใจชอบ จะได้รับค่ามาจากหัวเอาต์พุตเทียมของกรอบเวลาก่อนหน้า หากต้องการค่าตรรกะเป็นอย่างไรต้องปรับค่าตรรกะของโหนดต่าง ๆ ของกรอบเวลาก่อนหน้าเพื่อให้หัวเอาต์พุตเทียมดังกล่าวมีค่าตรรกะตามที่ต้องการ จากตัวอย่างในกรอบเวลาที่ 2 จะเห็นว่าการ G5_PPI[0][1] ที่มีค่าตรรกะเท่ากับ “0” ดังนั้นจึงต้องย้อนกลับไปกรอบเวลาแรกเพื่อปรับค่าตรรกะของโหนดต่าง ๆ จน G5_PPO[0][0] มีค่าตรรกะเท่ากับ “0” ดังภาพที่ 4.20 ดังนั้นรูปแบบสัญญาณทดสอบสำหรับ G2/s-a-1 มี 2 ชุดคือ “010X” และ “1001”



ภาพที่ 4.20 การปรับค่าตรรกะเพื่อให้โหนด G5_PPO[0][0] มีค่าตรรกะเท่ากับ “0”

สำหรับในกรอบเวลาแรกหาค่าตรรกที่ชั่วอินพุตเทียมมีค่าไม่เท่ากับ X จำเป็นต้องเพิ่ม
กรอบเวลาก่อนหน้าขึ้นอีกพยายามปรับค่าตรรกให้แก่ชั่วอินพุตเทียมจนมีค่าตรรกเท่ากับ X ทุกชั่ว
จึงจะสามารถให้กำเนิดรูปแบบสัญญาณทดสอบได้

มหาวิทยาลัยเชียงใหม่
Chiang Mai University