บทที่ 3

หลักวิธีการกำเนิดรายการลดทอนและรูปแบบสัญญาณทดสอบ

การหารูปแบบสัญญาณทดสอบสำหรับวงจรดิจิตอลจะมีค่าตรรกให้เลือกหลายค่า เพื่อเลือกให้แก่ขั้วขาเข้าของวงจรหรือของเกต รายการลดทอน คือ รายการที่เก็บค่าตรรกที่เป็นไปไม่ได้ที่จะเลือกให้กับทุกขั้วขาเข้าของวงจรหรือของเกต ก็จะช่วยลดทอนค่าตรรกที่จะป้อนให้แก่ขั้วขาเข้าของวงจรหรือของเกตได้

3.1 วิธีการกำเนิดรายการลดทอน

การสร้างรายการลดทอน มีอยู่ 2 ขั้นตอน คือ ขั้นตอนการทำรายการลดทอนไปข้างหน้า (Forward Reduction List) [17] และ Contrapositive Law [22]

3.1.1 รายการลดทอนไปข้างหน้าหรือเซตฟังก์ชัน (Set Functions)

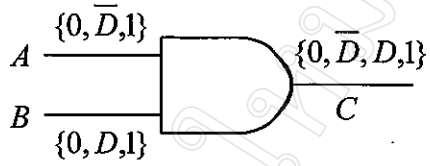
นิยามที่ 1 ถ้า f เป็นฟังก์ชันจาก A ไปยัง B จะเขียนแทนด้วย $f: A \to B$ ถ้า $a \in A$ เรียกสมาชิกใน B ที่จับคู่กับ a โดยฟังก์ชัน f ว่าภาพ (Images) ของ a ภายใต้ f ซึ่งเขียนแทนด้วย $f(a)$ เรียกเซตของค่าเหล่านี้ว่า ภาพ ของ f ซึ่งจะเขียนแทนด้วย $f(A)$ และถ้า $f(a) \in B$ เรียกสมาชิกใน A ที่จับคู่กับ b โดยฟังก์ชัน f ว่าส่วนกลับภาพ (Inverse Images) ของ b ภายใต้ f ซึ่งเขียนแทนด้วย $f^{-1}(B)$ [1]

$$f(A) = \{f(a) \mid a \in A\}$$
$$f^{-1}(B) = \{a \mid f(a) \in B\}$$

ถ้าให้ f เป็นฟังก์ชันจาก X x Y ไปยัง Z $f: XxY \to Z$ เป็นฟังก์ชัน 2 ตัวแปร และให้ $A \subseteq X, B \subseteq Y, C \subseteq Z$ ภาพฟังก์ชัน (Images Function) $f(A,B)$ ของ AxB ภายใต้ฟังก์ชัน f ก็คือ เซตของทุกภาพ $f(x,y)$ โดยที่ $x \in A$ และ $y \in B$ เขียนได้ดังนี้ [15]

$$f(A,B) = \{f(x,y) \mid x \in A \text{ และ } y \in B\}$$
$$f^{-X}_{|AxB}(C) = \{(x \in A \mid f(x,y) \in C \text{ for some } y \in B) \cap (AxB)\}$$
$$f^{-Y}_{|AxB}(C) = \{(y \in B \mid f(x,y) \in C \text{ for some } x \in A\} \cap (AxB)\}$$

ตัวอย่างการให้กำเนิดรายการลดทอน พิจารณา AND เกต 2 ขั้วขาเข้า แสดงในรูปที่ 3.1



รูปที่ 3.1 ภาพฟังก์ชันสำหรับ AND เกต 2 ขั้วขาเข้า [16]

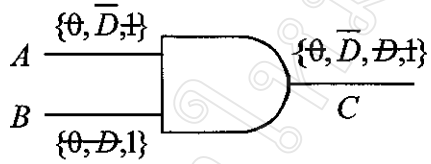
กำหนดให้ เซต $A = \{0, \overline{D},1\}, B = \{0, D,1\}$ ดังนั้นเซตของค่าตรรกที่ขั้วขาออกของ AND เกตได้มาจากการหาภาพของฟังก์ชันหรือการแพร่ค่าตรรกไปข้างหน้าได้ดังนี้

$$\begin{aligned} AND(\{0,\overline{D},1\},\{0,D,1\}) = \{&AND(0,0), AND(0,D),\\ &AND(0,1), AND(\overline{D},0),\\ &AND(\overline{D},D), AND(\overline{D},1),\\ &AND(1,0), AND(1,D),\\ &AND(1,1)\}\\ =\{&0,\overline{D},D,1\} \end{aligned}$$

$$\begin{aligned} f^{-X}_{|AxB}(\overline{D}) &= \{((1,\overline{D}),(\overline{D},1),(\overline{D},\overline{D}))\cap(AxB)\}\\ AND^{-1}_{|AxB}(\overline{D}) &= \{((1,\overline{D}),(\overline{D},1),(\overline{D},\overline{D}))\cap(AxB)\}\\ &= \{(\overline{D},1)\}\\ AND^{-X}_{|AxB} &= \{\overline{D}\}\\ AND^{-Y}_{|AxB} &= \{1\} \end{aligned}$$

การใช้แบบจำลองตรรก 16 ค่า และ รายการลดทอน มีแนวโน้มที่คาดว่าจะลดทอนค่าตรรกที่เป็นไปไม่ได้เสียก่อน ที่จะพบความขัดแย้งในการแพร่ตรรกไปข้างหน้าจนถึงขั้วขาออกหลัก จากตัวอย่างในรูปที่ 3.1 ถ้าสามารถตรวจพบค่าที่เกิดจุดเสีย $(\overline{D}, D)$ ที่ขั้วขาออกของ AND เกตได้ ทำให้สามารถตัดค่าตรรก "0" ที่ขั้วขาออกและทุกขั้วขาเข้าของเกตออกได้ โดยใช้ส่วนกลับภาพฟังก์ชัน (Inverse Images Function) [16] ทำให้ค่าที่เลือกให้แก่ขั้วขาเข้าของวงจรหรือ

ของเกตลดน้อยลง เพราะว่าสามารถที่จะตัดค่าที่เป็นไปไม่ได้ออกจากจุดหรือเส้นในวงจรดังแสดงในรูปที่ 3.2 (ถ้าเลือกจุดเสีย $\overline{D}$ ที่ขั้วขาออกของเกต)



รูปที่ 3.2 ส่วนกลับภาพฟังก์ชันสำหรับ AND เกต 2 ขั้วขาเข้า [16]

ในจุดหรือเส้นหนึ่งๆ ในวงจรจะมีค่า $\{\overline{D}, D, 0, 1\}$ ปรากฏอยู่ ถ้าสมมุติให้จุด X มีค่าตรรกเป็น "0" ทำให้สามารถตัดค่า $\{\overline{D}, D, 1\}$ ออกจากจุด X ได้ รายการลดทอนในจุด X เขียนใหม่ได้ดังนี้

$$X = 0 \quad \Rightarrow \quad \{X \neq \overline{D},\ X \neq D,\ X \neq 1\}$$
$$X = 1 \quad \Rightarrow \quad \{X \neq \overline{D},\ X \neq D,\ X \neq 0\}$$
$$X = D \quad \Rightarrow \quad \{X \neq \overline{D},\ Z \neq 1,\ X \neq 0\}$$
$$X = \overline{D} \quad \Rightarrow \quad \{X \neq D,\ X \neq 1,\ X \neq 0\}$$

ถ้ากำหนดให้ X, Y เป็นขั้วขาเข้าของ AND เกต และ Z เป็นขั้วขาออก รายการลดทอนไปข้างหน้า เมื่อ Z = 1 จะได้ $\{Z \neq \overline{D},\ Z \neq D,\ Z \neq 0,\ X \neq \overline{D},\ X \neq D,\ X \neq 0,\ Y \neq \overline{D},\ Y \neq D,\ Y \neq 0\}$

3.1.2 ความสัมพันธ์การแพร่ค่าตรรกไปข้างหน้าของเกตพื้นฐาน

กำหนดให้ X, Y เป็นขั้วขาเข้า และ Z เป็นขั้วขาออกของเกต '&' แทนตรรกและ '|' แทนตรรกหรือ

1.) AND เกต

if Z=0 then (Z=0 & (X=0 | Y=0 | (X=D & Y=$\overline{D}$) | (X=$\overline{D}$ & Y=D)))

if Z=1 then (Z=1 & X=1 & Y=1)

if Z=D then (Z=D & ((X=1 & Y=D) | (X=D&Y=1) | (X=D & Y=D)))

if Z=$\overline{D}$ then (Z=$\overline{D}$ & ((X=1 & Y=$\overline{D}$) | (X=$\overline{D}$&Y=1) | (X=$\overline{D}$ & Y=$\overline{D}$)))

2.) OR เกต

if Z=0 then (Z=0 & X=0 & Y=0)

if Z=1 then (Z=1 & (X=1 | Y=1 | (X=D & Y=$\overline{D}$) | (X=$\overline{D}$ & Y=D)))

if Z=D then (Z=D & ((X=0 & Y=D) | (X=D&Y=0) | (X=D & Y=D)))

if Z=$\overline{D}$ then (Z=$\overline{D}$ & ((X=0 & Y=$\overline{D}$) | (X=$\overline{D}$ & Y=0) | (X=$\overline{D}$ & Y=$\overline{D}$)))

3.) XOR เกต

if Z=0 then (Z=0 & ((X=0 & Y=0) | (X=1 & Y=1) | (X=D&Y=D) |
(X=$\overline{D}$ & Y=$\overline{D}$)))

if Z=1 then (Z=1 & ((X=0 & Y=1) | (X=1 & Y=0) | (X=D&Y=$\overline{D}$) |
(X=$\overline{D}$ & Y=D)))

if Z=D then (Z=D & ((X=0 & Y=D) | (X=1 & Y=$\overline{D}$) | (X=D&Y=0) |
(X=$\overline{D}$ & Y=1)))

if Z=$\overline{D}$ then (Z=$\overline{D}$ & ((X=0 & Y=$\overline{D}$) | (X=1 & Y=D) | (X=D&Y=1) |
(X=$\overline{D}$ & Y=0)))

3.1.3 Contrapositive Law

คุณสมบัติทางตรรกที่เรียกว่า Contrapositive Law มีนิพจน์ดังต่อไปนี้

$$(P \Rightarrow Q) \Leftrightarrow (\neg Q \Rightarrow \neg P) \quad (1)$$

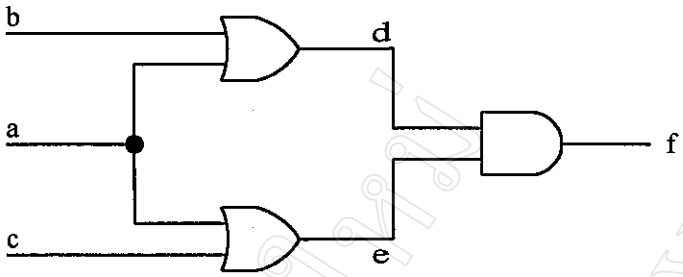
$$(P \Rightarrow (\neg Q)) \Leftrightarrow (\neg(\neg Q) \Rightarrow \neg P) \quad (2)$$

$$(P \Rightarrow \neg Q) \Leftrightarrow (Q \Rightarrow \neg P) \quad (3)$$

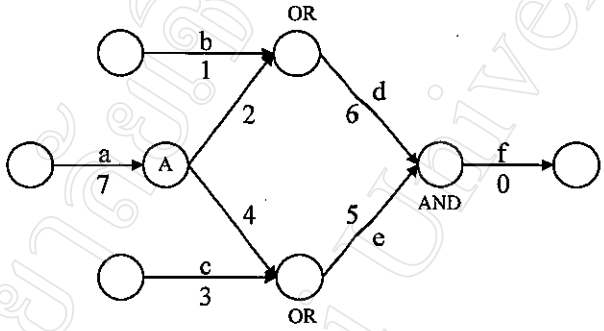
ถ้าแทน P ด้วย $(a = 1)$ และแทน Q ด้วย $(b \neq 1)$ จากสมการที่ (3) สามารถนำมาเขียนใหม่ได้ดังนี้

$$((a = 1) \Rightarrow (b \neq 1)) \Leftrightarrow ((b = 1) \Rightarrow (a \neq 1)) \quad (4)$$

จากสมการที่ 4 ถ้า a = 1 แล้ว b $\neq$ 1 ก็ต่อเมื่อ ถ้า b = 1 แล้ว a $\neq$ 1



รูปที่ 3.3 วงจรตัวอย่าง ในการทำรายการลดทอน [22]



รูปที่ 3.4 แทนวงจรด้วยกราฟแบบมีทิศทาง จากรูปที่ 3.3 [22]

จากรูปที่ 3.4 การสร้างรายการลดทอน จากการทำรายการลดทอนไปข้างหน้า และ Contrapositive Law ถ้าเส้นที่ 0 มีค่าตรรกเป็น "0" จะมีรายการลดทอนเป็น

$$0 = 0 : \{7 \neq \overline{D}, 7 \neq D, 7 \neq 1, 0 \neq D, 0 \neq \overline{D}, 0 \neq 1\}$$

ในทำนองเดียวกัน ถ้าเส้นที่ 7 มีค่าตรรกเป็น $\{\overline{D}, D, 1\}$ ในรายการลดทอนของเส้นที่ 7 ก็จะมีเส้นที่ $0 \neq 0$ ปรากฏอยู่ด้วย เช่นถ้าเส้นที่ 7 มีค่าเท่ากับ "$\overline{D}$" เส้นที่ 7 ก็จะมีรายการลดทอนดังนี้

$$7 = \overline{D} : \{7 \neq D, 7 \neq 1, 7 \neq 0, \underline{0 \neq 0}, 0 \neq D, 2 \neq D, 2 \neq 1, 2 \neq 0, 4 \neq D, 4 \neq 1, 4 \neq 0,$$
$$5 \neq D, 5 \neq 0, 6 \neq D, 6 \neq 0\}$$

และถ้าเส้นที่ 0 มีค่าเป็น $\{\overline{D}, D, 1\}$ ก็จะมีเส้นที่ $0 \neq 0$ อยู่ในรายการลดทอนนั้นด้วยตามลำดับ โดยใช้คุณสมบัติของ Contrapositive Law

ตัวอย่างเช่น แทนค่า P = $(0 = 0)$ และ Q = $(0 \neq \overline{D})$ ลงสมการที่ 3, 4 จะได้

$$((0 = 0) \Rightarrow (0 \neq \overline{D})) \Leftrightarrow ((O = \overline{D}) \Rightarrow (0 \neq 0))$$

จะเห็นว่าถ้าเส้นที่ $0 = 0$ จะมี $0 \neq \overline{D}$ อยู่ในรายการลดทอนของ $0 = 0$ ในทางกลับกัน ถ้าเส้นที่ $0 = \overline{D}$ จะมี $0 \neq 0$ อยู่ในรายการลดทอนของ $0 = \overline{D}$ นั้นด้วย

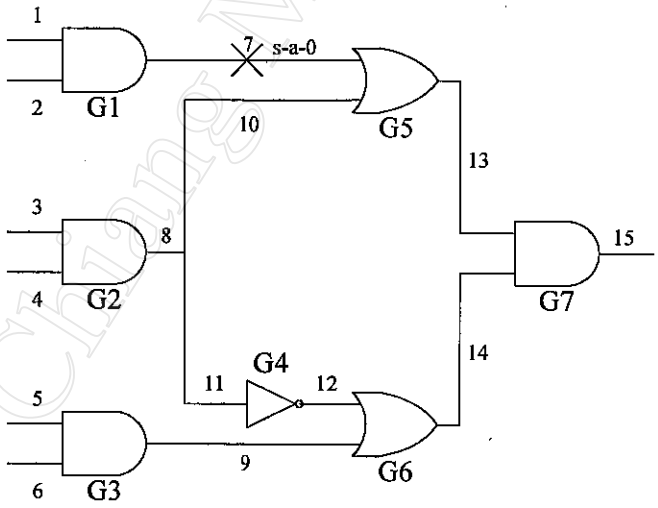
3.2 วิธีการให้กำเนิดรูปแบบสัญญาณทดสอบ

3.2.1 วิธีเส้นทางการแพร่ค่าตรรก (Sensitized Path)

วิธีการหาเส้นทางการแพร่ค่าตรรกจากจุดเสียไปจนถึงขั้วขาออกหลักของวงจร สามารถนำไปใช้ในการหารูปแบบสัญญาณทดสอบจุดเสียที่เกิดขึ้นในวงจรได้ มีอยู่ 2 ส่วน คือ

1.) สร้างเส้นทางจากจุดเสียไปจนถึงขั้วขาออกหลักของวงจร

2.) กำหนดค่าตรรกให้กับขั้วขาเข้าของแต่ละเกตตามเส้นทางเดินที่ได้จากข้อที่ 1 และทำการย้อนรอยกลับไปเลือกค่าตรรกให้กับขั้วขาเข้าของเกตที่เหลือในวงจรจนถึงขั้วขาเข้าหลักของวงจร



รูปที่ 3.5 วงจรตัวอย่างการสร้างเส้นทางการแพร่ตรรก [21]

ตัวอย่างการสร้างเส้นทางการแพร่ตรรก จากรูปที่ 3.5 เกิดจุดเสีย s-a-0 เส้นที่ 7 แสดงในตารางที่ 3.1 ในส่วนที่ 1 เป็นการสร้างเส้นทางการแพร่ค่าตรรกจากจุดเสียไปจนถึงขั้วขาออกหลักของวงจร อยู่ในขั้นตอนที่ 1 - 3

ขั้นตอนที่ 1 ตรวจสอบจุดเสียเส้นที่ 7 ขั้วขาออกของ AND เกต G1 ต้องกำหนดค่าตรรกในเส้นที่ 1 และ 2 มีค่าเท่ากับ "1" เพื่อตรวจสอบจุดเสีย s-a-0 เส้นที่ 7

ขั้นตอนที่ 2 แพร่ค่าตรรกผ่านเกต G5 ต้องกำหนดให้เส้นที่ 10 มีค่าตรรกเท่ากับ "0"

ขั้นตอนที่ 3 แพร่ค่าตรรกผ่านเกต G7 ต้องกำหนดให้เส้นที่ 14 มีค่าตรรกเท่ากับ "1"

ในส่วนที่ 2 ทำการย้อนรอยกลับไปเลือกค่าตรรกให้กับขั้วขาเข้าของเกตที่เหลือในวงจร เพราะว่าได้ค่าตรรกที่ขั้วขาออกของเกตแล้วในส่วนที่ 1 จนถึงขั้วขาเข้าหลักของวงจร อยู่ในขั้นตอนที่ 4 - 7

ตารางที่ 3.1 ขั้นตอนการสร้างเส้นทางการแพร่ค่าตรรก [21]

ขั้นตอนที่	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1					1								
2										0					
3														1	
	1	1					1			0			1	1	0
4								0			0	1			
	1	1					1	0		0	0	1	1	1	0
5									X						
	1	1					1	0	X	0	0	1	1	1	0
6					X	X									
	1	1			X	X	1	0	X	0	0	1	1	1	0
7(a)			0	X											
7(b)			X	0											

ขั้นตอนที่ 4 เส้นที่ 10 และ 14 ถูกกำหนดค่าตรรกไว้แล้ว ซึ่งเป็นขั้วขาออกของเกต G2 และ G6 ตามลำดับ ในเส้นที่ 10 สามารถกำหนดค่าตรรกเส้นที่ 8, 11 และ 12 ได้ในขั้นตอนนี้

ขั้นตอนที่ 5 เมื่อเส้นที่ 14 ถูกกำหนดให้มีค่าตรรกเท่ากับ "1" (ขั้วขาออกของเกต G6) และในขั้นตอนที่ 4 กำหนดค่าตรรกเส้นที่ 12 มีค่าเท่ากับ "1" อยู่แล้ว จะมีผลทำให้เส้นที่ 14 มีค่าตรรกเท่ากับ "1" ตามไปด้วย ตรงตามเงื่อนไขที่ต้องการ ส่วนที่ในเส้นที่ 9 กำหนดค่าตรรกอะไรก็ได้ (X)

ขั้นตอนที่ 6 เส้นที่ 5 และ 6 (ขั้วขาเข้าของเกต G3) กำหนดค่าตรรกอะไรก็ได้ (X) เพราะว่าเส้นที่ 9 (ขั้วขาออกของเกต G3) เป็นค่าตรรกอะไรก็ได้

ขั้นตอนที่ 7 การกำหนดค่าตรรกให้กับขั้วขาเข้าของเกต G2 ในเส้นที่ 3 และ 4 นั้นทำอย่างไรก็ได้ที่มีผลทำให้ขั้วขาออกของเกต G2 มีค่าเท่ากับ "0" (ในเส้นที่ 8 ถูกกำหนดมาแล้วในขั้นตอนที่ 4) กำหนดให้เส้นที่ 3 มีค่าตรรกเท่ากับ "0" และ เส้นที่ 4 มีค่าตรรกเท่ากับ "X" หรือกำหนดให้เส้นที่ 3 มีค่าตรรกเท่ากับ "X" และ เส้นที่ 4 มีค่าตรรกเท่ากับ "0" ทั้ง 2 กรณีมีผลทำให้เส้นที่ 8 มีค่าตรรกเท่ากับ "0" เมื่อค่าตรรกถูกกำหนดให้กับทุกขั้วขาเข้าหลักของวงจรเรียบร้อยแล้ว จะได้รูปแบบสัญญาณทดสอบจุดเสีย s-a-0 เส้นที่ 7 คือ {(110XXX), (11X0XX)}

การสร้างเส้นทางการแพร่ค่าตรรก ก็เป็นวิธีการหนึ่งในการให้กำเนิดรูปแบบสัญญาณทดสอบจุดเสีย ปัญหาที่เกิดขึ้นในการสร้างเส้นทางการแพร่ค่าตรรก คือ มีเส้นทางให้เลือกหลายเส้นทางเพราะว่าในวงจรบางเกตมีขั้วขาออกมีจุดต่อแยกหลายจุดหรือมีขั้วขาออกหลักในวงจรหลายขั้ว ทำให้ต้องเลือกเส้นทางใดเส้นทางหนึ่งมีโอกาสทำให้เลือกผิด และเมื่อมีจุดต่อแยกหลายจุดจากขั้วขาออกของเกต ก็จะมีอยู่จุดหนึ่งในวงจรเป็นที่รวมสัญญาณจากจุดต่อแยกหลายจุด (Reconvergent Fan-out Path) ก็จะทำให้การกำหนดค่าตรรกให้กับขั้วขาเข้าของเกตผิด มีผลทำให้การแพร่ค่าตรรกจุดเสียไม่สามารถไปถึงขั้วขาออกหลักของวงจรได้ การสร้างเส้นทางการแพร่ค่าตรรก จะทำได้เพียงเส้นทางเดียว (Single Path) แต่มีหลักวิธี D-Algorithm ที่สามารถทำได้หลายเส้นทาง (Multiple Path)

3.2.2 วิธี D-Algorithm

D-Algorithm เป็นหลักวิธีที่เป็นพื้นฐานในการทำการให้กำเนิดรูปแบบสัญญาณทดสอบโดยอัตโนมัติสำหรับวงจรประกอบ [22] โดยกำหนดค่าตรรก 2 ค่า ที่จะป้อนให้กับวงจรชนิดแรกก็คือค่าตรรกที่ขั้วขาเข้าที่ทำให้เกิดผลที่ขั้วขาออกที่ตรงกันข้ามกับค่าตรรกที่เกิดจุดเสียอยู่ในวงจรเรียกว่า Primitive D-cube of a fault (PDcf) [22] ชนิดที่สอง คือ ค่าตรรกที่ขั้วขาเข้าที่สามารถนำค่าตรรกที่จุดเสียที่เกิดขึ้นในวงจรไปปรากฏที่เอาต์พุตของวงจร เรียกว่า Propagation D-cube (PDc) [22] และยังมีค่าตรรกที่เป็นจุดเสีย $(D,\overline{D})$ ที่ปรากฏที่ขั้วขาออก เรียกว่า D-frontier [22] ค่าตรรกที่ครอบคุมการทำงานทั้งหมดของเกตตามตารางความจริง เรียกว่า Singular Cover [21] และ D-Intersection [21] คือ การจำลองการทำงาน การแพร่ค่าตรรกผ่านเกตต่างๆ ในวงจร โดยใช้กฎในตารางที่ 3.8

ตารางที่ 3.2 PDc สำหรับ AND เกต 2 ขั้วขาเข้า [22]

A	B	C
{1}	$\{D\}$	$\{D\}$
$\{D\}$	{1}	$\{D\}$
{1}	$\{\overline{D}\}$	$\{\overline{D}\}$
$\{\overline{D}\}$	{1}	$\{\overline{D}\}$

ตารางที่ 3.3 PDc สำหรับ NAND เกต 2 ขั้วขาเข้า [21]

A	B	C
{1}	$\{D\}$	$\{\overline{D}\}$
$\{D\}$	{1}	$\{\overline{D}\}$
{1}	$\{\overline{D}\}$	$\{D\}$
$\{\overline{D}\}$	{1}	$\{D\}$

ตารางที่ 3.4 PDc สำหรับ OR เกต 2 ขั้วขาเข้า [21]

A	B	C
{0}	$\{D\}$	$\{D\}$
$\{D\}$	{0}	$\{D\}$
{0}	$\{\overline{D}\}$	$\{\overline{D}\}$
$\{\overline{D}\}$	{0}	$\{\overline{D}\}$

ตารางที่ 3.5 PDc สำหรับ NOR เกต 2 ขั้วขาเข้า [21]

A	B	C
{0}	$\{D\}$	$\{\overline{D}\}$
$\{D\}$	{0}	$\{\overline{D}\}$
{0}	$\{\overline{D}\}$	$\{D\}$
$\{\overline{D}\}$	{0}	$\{D\}$

ตารางที่ 3.6 PDcf สำหรับ AND เกต 2 ขั้วขาเข้า [22]

A	B	C
{1}	{1}	$\{D\}$
{0}	{1}	$\{\overline{D}\}$
{1}	{0}	$\{\overline{D}\}$

ตารางที่ 3.7 Singular Cover สำหรับ AND เกต 2 ขั้วขาเข้า [21]

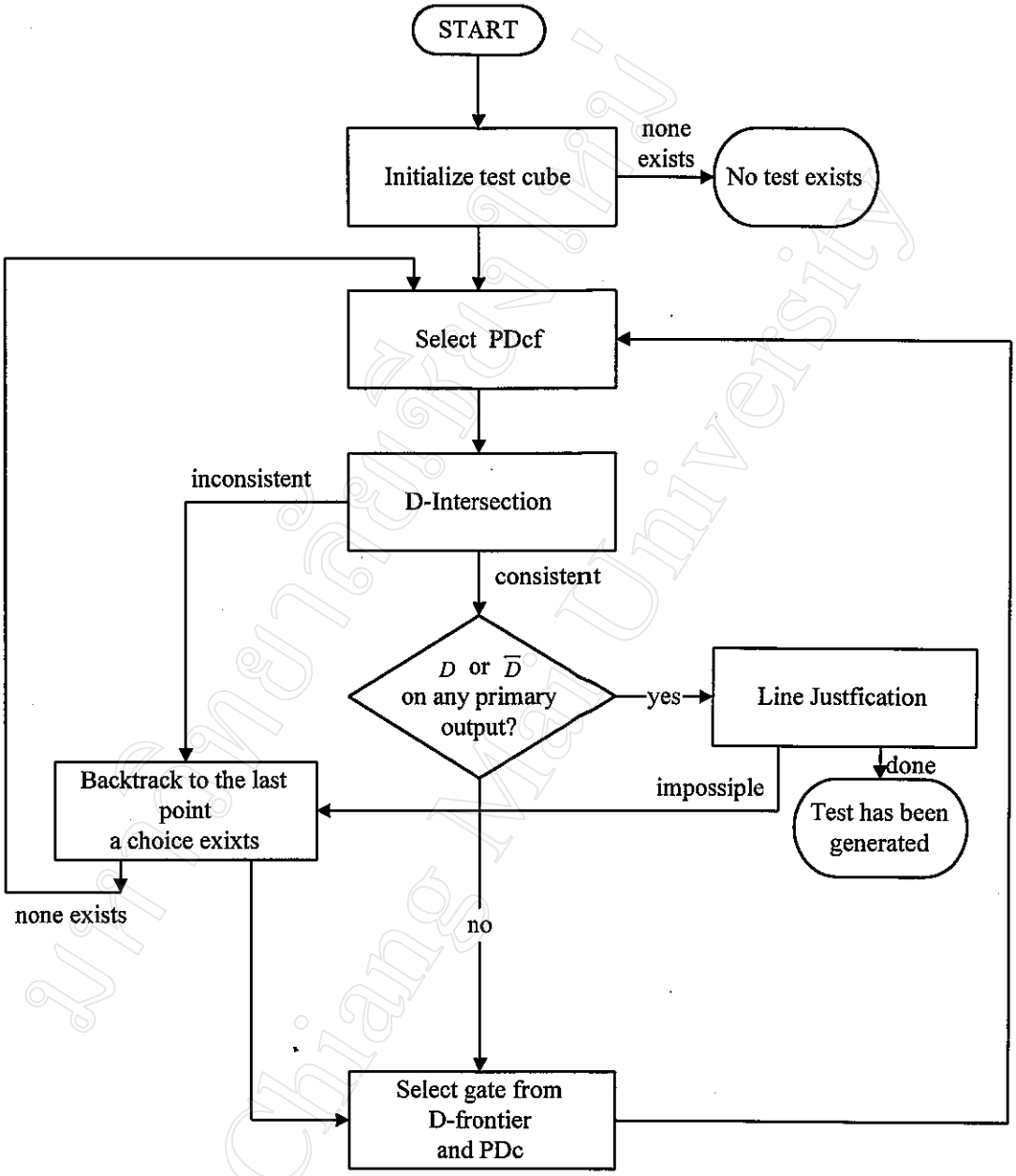
A	B	C
{0}	{X}	{0}
{X}	{0}	{0}
{1}	{1}	{1}

ตารางที่ 3.8 กฎการทำ D-Intersection [21]

$\cap$	{0}	{1}	{X}	$\{D\}$	$\{\overline{D}\}$
{0}	{0}	$\{\phi\}$	{0}	$\{\varphi\}$	$\{\varphi\}$
{1}	$\{\phi\}$	{1}	{1}	$\{\varphi\}$	$\{\varphi\}$
{X}	{0}	{1}	{X}	$\{D\}$	$\{\overline{D}\}$
$\{D\}$	$\{\varphi\}$	$\{\varphi\}$	$\{D\}$	$\{D\}$	$\{\varphi\}$
$\{\overline{D}\}$	$\{\varphi\}$	$\{\varphi\}$	$\{\overline{D}\}$	$\{\varphi\}$	$\{\overline{D}\}$

หมายเหตุ

- ϕ คือ เซตว่าง หลังจากการทำ D-Intersection
- φ คือ เซตที่ไม่ต้องกำหนดค่า



รูปที่ 3.6 ผังงานหลักวิธี D-Algorithm [8]

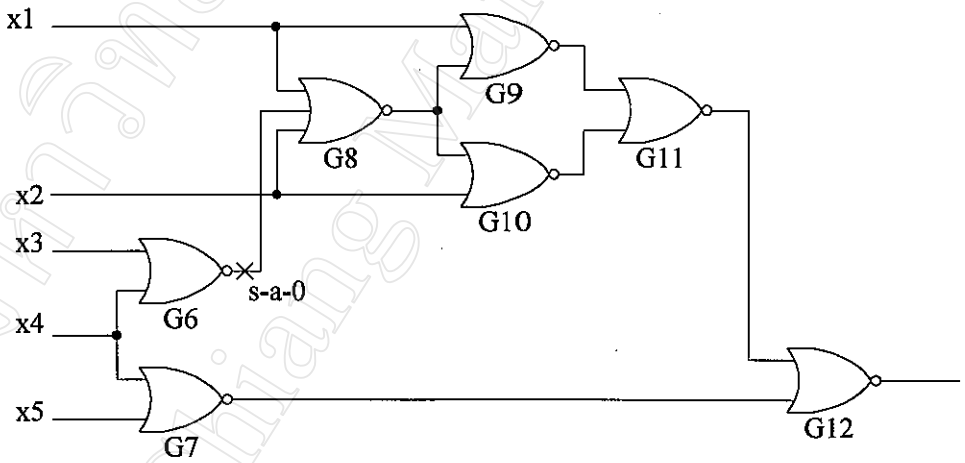
ขั้นตอนการทำงานของหลักวิธี D-Algorithm

1.) เลือกจุดเสีย จากรายการจุดเสีย

2.) ทำ PDcf สำหรับเกตที่เกิดจุดเสียในวงจร

3.) เลือกเส้นทางการแพร่ค่าตรรกมา 1 เส้นทาง จากที่เกิดจุดเสียในวงจร จากนั้นหา D-frontier ทุกเกตที่ต่ออยู่กับขั้วขาออกของเกตที่เกิดจุดเสีย โดยทำ D-Intersection ระหว่าง PDcf ของเกตที่เกิดจุดเสีย กับ PDc ของเกตตัวแรกที่ต่ออยู่ขั้วขาออกของเกตที่เกิดจุดเสีย โดยใช้กฎการทำ D-Intersection (ตารางที่ 3.8) ผลลัพธ์ที่ได้จะเก็บไว้ เพื่อนำไป D-Intersection กับ PDc ของเกตอื่นๆ ที่เหลือตามเส้นทางการแพร่ค่าตรรก ทำจนกระทั่งค่าตรรกที่เป็นจุดเสีย (D,$\overline{D}$) ปรากฏที่ขั้วขาออกหลักของวงจร

4.) เช็คความถูกต้อง (Consistency Check) เมื่อมีการกำหนดค่าตรรกให้กับขั้วขาเข้าของเกตหรือของวงจรเรียบร้อยแล้ว (ตามข้อที่ 1, 2) ต้องตรวจสอบความถูกต้องตามตารางความจริง ตาราง PDcf และตาราง PDc ของแต่ละเกต ถ้าผิดต้องกลับไปเลือกค่าตรรกให้กับขั้วขาเข้าของเกตหรือของวงจรใหม่ และถ้าถูกต้องจะได้รูปแบบสัญญาณทดสอบจุดเสียที่เกิดขึ้นในวงจร



รูปที่ 3.7 วงจรตัวอย่างการใช้หลักวิธี D-Algorithm [21]

ตัวอย่างการใช้หลักวิธี D-Algorithm จากรูปที่ 3.7 เกิดจุดเสีย s-a-0 ที่ขั้วขาออกของเกต G6 แสดงขั้นตอนการทำงานในตารางที่ 3.9

ขั้นตอนที่ 1 กำหนดค่า PDcf สำหรับเกต 6 โดยกำหนดให้ x4 และ x3 มีค่าตรรกเท่ากับ “0” เพราะว่าต้องการค่าตรรกที่ขั้วขาออกของเกต 6 เท่ากับ “1” ค่าที่เป็นปกติ (ตรงกันข้ามกับค่าตรรกจุดเสีย D)

ขั้นตอนที่ 2 แพร่ค่าตรรก (PDc) จุดเสียผ่านเกต 8 สามารถกำหนดค่าตรรก "0" ให้กับ x1 และ x2 ได้เลย (ตามตาราง PDc ของ NOR เกต) ดังนั้นขั้วขาออกของเกต 8 มีค่าตรรกเท่ากับ "$\overline{D}$"

ตารางที่ 3.9 แสดงขั้นตอนการทำงานของหลักวิธี D-Algorithm [21]

ขั้นตอนที่	X1	X2	X3	X4	X5	G6	G7	G8	G9	G10	G11	G12
1			0	0		D						
2	0	0				D		$\overline{D}$				
3	0	0	0	0		D		$\overline{D}$				
4	0							$\overline{D}$	D			
5	0	0	0	0		D		$\overline{D}$	D			
6		0						$\overline{D}$		D		
7	0	0	0	0		D		$\overline{D}$	D	D		
8									D	D	D	
9	0	0	0	0		D		$\overline{D}$	D	D	D	
10							0				D	$\overline{D}$
11	0	0	0	0		D	0	$\overline{D}$	D	D	D	$\overline{D}$
12				X	1		0					
13	0	0	0	0	1	D	0	$\overline{D}$	D	D	D	$\overline{D}$

ขั้นตอนที่ 3 ผลลัพธ์ที่ทำ D-Intersection ระหว่าง PDcf ของเกต 6 (ขั้นตอนที่ 1) กับ PDc ของเกต 8 (ขั้นตอนที่ 2) และต้องตรวจสอบค่าตรรกที่ขั้วขาเข้าและที่ขั้วขาออกของเกตเกิดการเปลี่ยนแปลงหรือไม่ หลังจากการทำ D-intersection แล้ว ถ้ามีการเปลี่ยนแปลงจะต้องกลับย้อนไปเลือกค่าตรรกให้กับขั้วขาเข้าของเกตใหม่

ขั้นตอนที่ 4 ถึง 7 D-Algorithm จะทำการแพร่ค่าตรรกผ่านทุกเส้นทางที่เป็นไปได้ โดยผ่านเกต 8 เกต 9 ไปจนถึงขั้วขาเข้าของเกต 11

ขั้นตอนที่ 4 แพร่ค่าตรรก (PDc) ผ่านเกต 9

ขั้นตอนที่ 5 ผลลัพธ์ที่ทำ D-Intersection ระหว่าง PDc เกต 9 กับค่าตรรกในขั้นตอนที่ 3

ขั้นตอนที่ 6 แพร่ค่าตรรก (PDc) ผ่านเกต 10

ขั้นตอนที่ 7 ผลลัพธ์ที่ทำ D-Intersection ระหว่าง PDc เกต 10 กับค่าตรรกในขั้นตอนที่ 5

ขั้นตอนที่ 8 แพร่ค่าตรรก (PDc) ผ่านเกต 11

ขั้นตอนที่ 9 ผลลัพธ์ที่ทำ D-Intersection ระหว่าง PDc เกต 11 กับค่าตรรกในขั้นตอนที่ 7

ขั้นตอนที่ 10 แพร่ค่าตรรก (PDc) ผ่านเกต 12

ขั้นตอนที่ 11 ผลลัพธ์ที่ทำ D-Intersection ระหว่าง PDc เกต 12 กับค่าตรรกในขั้นตอนที่ 9

เมื่อแพร่ค่าตรรกมาตามเส้นทางจนถึงขั้นตอนที่ 11 ตรวจพบว่าค่าตรรกจุดเสีย $(D,\overline{D})$ ปรากฏที่ขั้วขาออกหลักของวงจร ขั้นตอนต่อไปจะเป็นขบวนการ การตรวจสอบความถูกต้อง โดยการกำหนดค่าตรรก "0" ที่ขั้วขาออกของเกต 7

ขั้นตอนที่ 12 กำหนดค่าตรรกให้กับขั้วขาเข้าสำหรับเกต 7 เมื่อต้องการค่าตรรก "0" ที่ขั้วขาออก จากตัวอย่างกำหนดให้ X5 มีค่าตรรกเท่ากับ "1" และ X4 เป็นค่าตรรกอะไรก็ได้ (X) ขั้นตอนนี้มีการเปลี่ยนแปลงค่าตรรกที่กำหนดให้กับ X4 ซึ่งค่าตรรกของ X4 ถูกกำหนดมาแล้วในขั้นตอนที่ 1 ดังนั้นค่าตรรกที่กำหนดให้กับ X4 มีค่าเท่า "0" ตามขั้นตอนที่ 1

ขั้นตอนที่ 13 ผลลัพธ์ที่ทำ D-Intersection ระหว่าง PDc เกต 7 กับค่าตรรกในขั้นตอนที่ 11 ขั้นตอนนี้เป็นขั้นตอนสุดท้าย ทุกขั้วขาเข้าของเกตหรือของวงจรถูกกำหนดค่าตรรกเรียบร้อยแล้ว สำหรับทดสอบจุดเสีย s-a-0 ที่ขั้วขาออกของเกต 6 มีรูปแบบสัญญาณทดสอบดังนี้ X1=X2=X3=x4= "0" และ X5="1" ถ้ามีค่าตรรก "0" ที่ขั้วขาออกหลักของวงจรเป็นค่าปกติ และถ้ามีค่าตรรก "1" แสดงว่าวงจรนั้นเสีย

3.2.3 หลักวิธี PODEM

D-Algorithm จะทำการตรวจสอบโครงสร้างทั้งหมดในโครงสร้างข้อมูลรายละเอียดแต่ละโหนดของวงจรในหน่วยความจำ (Search Space) ซึ่งในระหว่างทำการตรวจสอบ จะต้องทำการย้อนรอย ทำให้มีเวลาที่สูญเสียไปจากการทำมากขึ้น ถ้าเป็นวงจรที่มีขนาดใหญ่จะต้องใช้เวลานานมาก

เพื่อที่จะเพิ่มประสิทธิภาพของกระบวนการ การให้กำเนิดรูปแบบของสัญญาณทดสอบ ในส่วนของเวลาที่ใช้ในการทำงาน ได้มีการเสนอ PODEM Algorithm (Path-Oriented Decision Making Algorithm) [6] PODEM สามารถที่จะลดจำนวนของการย้อนรอยลงในเอทีพีจี โดยใช้ "branch-and-bound" Algorithm [18]

หลักวิธีการนี้ขั้นแรกพยายามที่จะตรวจสอบค่าของตรรก (D หรือ $\overline{D}$) ที่ตำแหน่งที่เกิดจุดเสีย โดยทำการป้อนค่าตรรกที่ขั้วขาเข้าที่มีผลต่อจุดเสียที่เกิดขึ้นกับโหนดโดยตรง ทันทีที่ป้อนค่าเข้าสู่ขั้วขาเข้า จะทำการตรวจสอบว่าค่าตรรกที่ป้อนเข้าไปสอดคล้อง และถูกต้องหรือไม่ ถ้าพบว่าไม่ถูกต้อง จะพยายามป้อนค่าตรรกใหม่เสริมเข้าไปในจุดที่เกิดความขัดแย้ง ถ้าค่าตรรกที่ป้อนเสริมเข้าไปยังคงไม่ถูกต้อง ก็จะต้องทำการย้อนรอยเลือกค่าตรรกใหม่อีกครั้งหนึ่ง

หลังจากตำแหน่งของจุดเสียได้มีการตรวจสอบเรียบร้อยแล้ว ขั้นตอนต่อไปคือการเตรียมทางเดิน (Path) เพื่อไปสู่ขั้วขาออกของวงจรของ D-frontier อย่างไรก็ตามมันไม่ใช่เรื่องง่ายที่จะทำการตรวจสอบทางเดินให้เป็นไปอย่างรวดเร็ว

ในหลักวิธีการของ PODEM หลักการค้นหาจะอยู่ในรูปโครงสร้างข้อมูลแบบ Binary Tree จุดประสงค์ที่สำคัญที่สุดคือ ป้อนข้อมูลที่กำหนดไว้เข้าไปที่ขั้วขาเข้า หลักวิธีการนี้ใช้หลักการตัดเอาแนวทางค้นหาที่ไม่ใช่ออก อย่างไรก็ตามหลักวิธี PODEM ยังคงมีสิ่งที่ต้องปรับปรุงแก้ไขการย้อนรอยเมื่อพบค่าตรรกที่เกิดความขัดแย้งในการกำหนดค่าให้กับโหนดในวงจร

3.2.4 หลักวิธี FAN

Fan-out-oriented test generation Algorithm (FAN) เป็นหลักวิธี ที่ใช้เพื่อพยายามลดจำนวนของการย้อนรอย และทำให้ช่วงเวลาประมวลผลระหว่างการหาเส้นทางการย้อนรอยน้อยลง [8][9][10] เป็นวิธีการพยายามที่หาคำตอบที่มีอยู่เท่าที่จะทำได้ โดยใช้หลักวิธี "branch-and-bound " ซึ่งจะคล้ายกับหลักวิธี PODEM เพื่อที่จะหลีกเลี่ยงการค้นหาที่ไม่จำเป็นในภายหลังทำการย้อนอรยทันทีที่ไม่มีคำตอบในโหนดปัจจุบัน ในโครงสร้างข้อมูลแบบต้นไม้เพื่อการตัดสินใจ (Decision Tree) ถูกใช้เพื่อทำการลดจำนวนของตัวเลือกอิสระโดยการวิเคราะห์การเชื่อมต่อของวงจร (Circuit Connectivity) ในขั้นตอนของการประมวลผลหลักวิธี FAN จะทำการบอกถึงโหนดในวงจรที่มีสายสัญญาณขั้วขาออกของเกตมีมากกว่า 1 เส้น ผ่านเกตอื่น ๆ แล้วก็กลับมารวมกัน ณ เกตตัวอีกตัวหนึ่งในวงจร (Reconvergent Fan-out Loops) ซึ่งจัดการได้ดีกว่าของหลักวิธี PODEM โดยคำจำกัดความของเส้นที่ใช้ในวงจรดังนี้

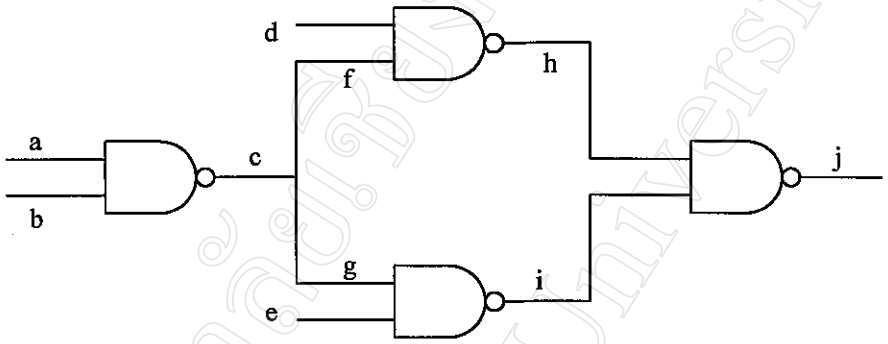
เส้นอิสระ (Freeline) เป็นโหนดของวงจรที่ไม่มีอะไรอยู่ก่อนหน้าเป็นส่วนหนึ่งของ Fan-out Loop ดังนั้นเส้นอิสระสามารถถูกกำหนดค่าได้อิสระ

เส้นขอบเขต (Boundline) เป็นโหนดของวงจรที่ตรงข้ามกับเส้นอิสระมันจะไม่สามารถถูกกำหนดค่าได้อย่างอิสระเนื่องจากการรวมจุดในวงจร เส้นขอบเขตใน Reconvergence Loop ขึ้นอยู่กับการกำหนดค่าระหว่างกัน

เส้นส่วนหัว (Headline) เป็นเส้นอิสระที่ทำให้เกตทำงาน ซึ่งเป็นส่วนหนึ่งของ Reconvergent Fan-out Loop ดังนั้นเส้นส่วนหัวสามารถถูกกำหนดค่าได้อิสระและสามารถถูกทำให้เหมือนกับขั้วขาเข้าของวงจร

ตัวอย่างพิจารณาวงจรที่แสดงในรูปที่ 3.8 เส้น a, b, c, d, e และ j เป็นเส้นอิสระและเส้น f, g, h และ j เป็นเส้นขอบเขต ในเส้นอิสระเหล่านี้ c, d และ e ก็เป็นเส้นส่วนหัวด้วย

หลักวิธี FAN สามารถถือได้ว่าเป็นส่วนขยายของหลักวิธี PODEM ขึ้นอยู่กับคุณสมบัติของเส้นที่กล่าวถึงในวงจร วิธีการนี้จะทำการกำหนดค่าตรรกหลายๆ ค่าเท่าที่เป็นไปได้ สามารถทำได้โดยการหาเส้นทาง (Tracing) ของค่าตรรก การไปข้างหน้า (Forward) และการย้อนกลับ (Backward) ในวงจร



รูปที่ 3.8 วงจรตัวอย่างการใช้หลักวิธี FAN [22]

ผลการทดลองแสดงว่าหลักวิธี PODEM มีความเร็วกว่าหลักวิธี D ในลักษณะทั่ว ๆ ไปของวงจรประกอบ [6] และหลักวิธี FAN มีความเร็วมากกว่าและมีประสิทธิภาพมากกว่าหลักวิธี PODEM ในแง่ของเวลาการประมวลผลจำนวนของการย้อนรอย และค่าครอบคลุมจุดเสีย (Fault Coverage) [9] อย่างไรก็ตามเป็นการยากที่จะเปรียบเทียบประสิทธิภาพของหลักวิธี D, PODEM และหลักวิธี FAN ในวิธีใด ๆ เนื่องจากหลักวิธีการส่วนมากถูกเขียนในภาษาที่แตกต่างกัน ทำงานบนเครื่องต่างกัน และถูกใช้เพียงชุดของวงจรเล็ก ๆ หรือชุดของการนำเสนอเล็ก ๆ เท่านั้น

3.2.5 หลักวิธี SOCRATES Algorithm

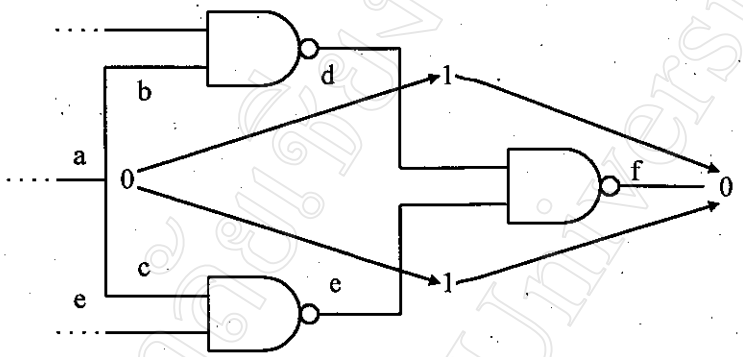
SOCRATES (Structure-Oriented Cost-Reducing Automatic TESt Pattern Generation) [23] มีจุดประสงค์เพื่อพัฒนา และเพิ่มความเร็วของเอทีพีจี

ตัวอย่างพิจารณาตัวอย่างของวงจรที่แสดงในรูปที่ 3.8 วงจรที่เป็นเพียงส่วนเล็กๆ ของวงจรทั้งหมด ความสัมพันธ์ระหว่างโหนดในวงจร สมมุติว่าค่าตรรก “0” ถูกกำหนดไปยังโหนด a ดังนั้นความสัมพันธ์ที่เกิดขึ้นระหว่างโหนดในวงจร คือ โหนด d=1 โหนด e=1 และโหนด f=0 ด้วยตามลำดับ ดังแสดงในรูปที่ 3.9 โดยใช้พีชคณิตบูลีน ที่เรียกว่า Contrapositive มีสมการดังนี้

$$(P \Rightarrow Q) \Leftrightarrow (\neg Q \Rightarrow \neg P) \qquad (5)$$

หลักวิธี SOCRATES ใช้วิธีการแบบนี้ในการแสดงค่าที่เหมาะสมในวงจร สำหรับวงจรในรูป ถ้าตัวแปร P และ Q (สมการ 5) ถูกแทนที่โดยการใส่ค่าตรรกในโหนด a เท่ากับ "0" และค่าตรรกของโหนด f เท่ากับ "0" ด้วยตามลำดับ แทนลงในสมการ (5) สามารถเขียนได้ใหม่ดังนี้

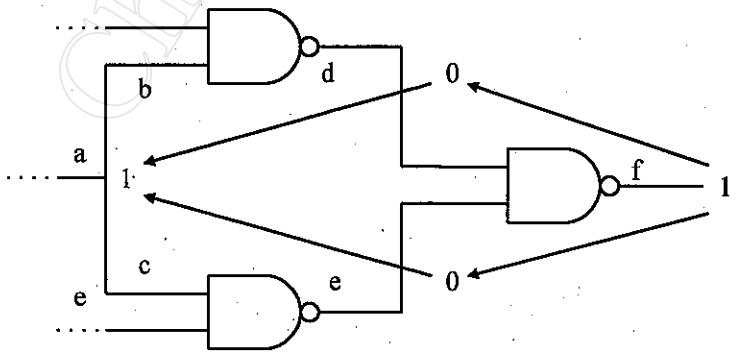
$$((a=0)\Rightarrow(f=0))\Leftrightarrow(\neg(f=0)\Rightarrow\neg(a=0)) \qquad (6)$$



รูปที่ 3.9 แสดงถึงความสัมพันธ์ระหว่างโหนด (Forward Propagation) [22]

ส่วนทางด้านขวามือของสมการ (6) สามารถถูกเขียนในทางกลับกัน (Negations) ได้ดังสมการ (7) แสดงว่าค่าตรรก "1" สามารถกำหนดให้โหนด f ได้ทันที ทำให้ที่โหนด a มีค่าตรรกเท่ากับ "1" ตามไปด้วยสามารถอธิบายได้ดังรูปที่ 3.10

$$((a=0)\Rightarrow(f=0))\Leftrightarrow((f=1)\Rightarrow(a=1)) \qquad (7)$$



รูปที่ 3.10 แสดงความสัมพันธ์ระหว่างโหนด (Backward Propagation) [22]

3.3 วิธีการจำลองการทำงานจุดเสีย

การจำลองการทำงานจุดเสีย เป็นส่วนหนึ่งที่สำคัญในขบวนการให้กำเนิดรูปแบบสัญญาณทดสอบโดยอัตโนมัติ เมื่อได้รูปแบบสัญญาณทดสอบมา 1 รูปแบบ นำรูปแบบสัญญาณที่ได้มาจำลองการทำงานเพื่อทดสอบจุดเสียจุดอื่นๆ ที่เหลืออยู่ในรายการจุดเสีย เมื่อจำลองการทำงานจุดเสียไม่สามารถทดสอบจุดเสียจุดอื่นๆ เพิ่มได้อีกก็จะหยุดการทำงาน และการให้กำเนิดรูปแบบสัญญาณทดสอบโดยอัตโนมัติก็จะทำงานต่อ เพื่อหารูปแบบสัญญาณทดสอบจุดเสียที่เหลืออยู่ การทำงานจะสลับกันไปจนกว่าจะได้ทดสอบจุดเสียครบทั้งหมดที่เหลืออยู่ในรายการจุดเสีย หรือการให้กำเนิดรูปแบบสัญญาณทดสอบโดยอัตโนมัติไม่สามารถหารูปแบบสัญญาณเพื่อมาทดสอบจุดเสียได้อีก การจำลองการทำงานจุดเสียประกอบด้วย 5 ส่วนที่สำคัญ คือ

1.) จำลองการทำงานวงจรที่ดี

2.) เลือกจุดเสียที่จะจำลองการทำงานจากรายการจุดเสีย

3.) ใส่ค่าตรรกจุดเสียลงในวงจร

4.) แพร่ค่าตรรกไปข้างหน้า

5.) เมื่อทดสอบหาจุดได้เพิ่มขึ้น ก็จะทำการตัดจุดเสียจุดนั้นออกจากรายจุดเสีย

3.3.1 การจำลองการทำงานแบบขนาน (Parallel Fault Simulation)

การจำลองการทำงานแบบขนานนี้จะใช้ขนาดความกว้างของคำ (Word Length : N bits) ของเครื่องคอมพิวเตอร์ จะจำลองการทำงานเท่ากับ n=W-1 (n=จำวนจุดเสีย , W=คำ, 1 คือค่าตรรกที่เป็นปกติ) ในการทำงานจะใช้คำสั่ง AND, OR, XOR, NOT ตัวอย่างเช่น

[1010] AND [1100] = [1000]

[1010] OR [1100] = [1110]

NOT [1010] = [0101]

[1010] XOR [1100] = [0110]

กำหนดสัญญาณ ($mask(S)$) และจุดเสีย ($fvalue(S)$) ลงในแหน่งบิต (i) ของคำ

$mask(S)_i$ = 1 (กำหนดจุดเสีย mask bit (i) =1)

$fvalue(S)_i$ = 1 ถ้าจุดเสียเป็น s-a-1, 0 ถ้าจุดเสียเป็น s-a-0

$$S' = S.\overline{mask(s)} + mask(S).fvalue(S)$$

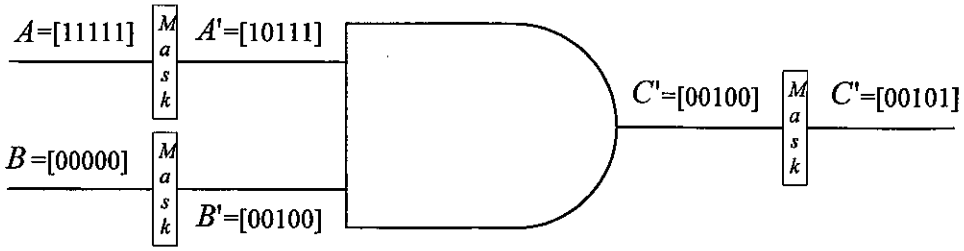
ตัวอย่างการจำลองการทำงานจุดเสียแบบขนานของ AND เกต 2 ขั้วขาเข้า ใช้ความกว้างของคำเท่ากับ 5 บิต

ตารางที่ 3.10 ตำแหน่งบิตที่เกิดจุดเสีย [8]

ตำแหน่งบิต	จุดเสีย
1	ค่าตรรกที่เป็นปกติ
2	A s-a-0
3	B s-a-1
4	C s-a-0
5	C s-a-1

ตารางที่ 3.11 กำหนดค่าในแต่ละบิตที่เกิดจุดเสีย [8]

เส้นที่	Mask Bit	Fvalue
A	[01000]	[00000]
B	[00100]	[00100]
C	[00011]	[00001]



รูปที่ 3.11 จำลองการทำงานจุดเสียแบบขนานของ AND เกต 2 ขั้วขาเข้า [8]

กำหนดให้ A="1", B="0" และ C="0"

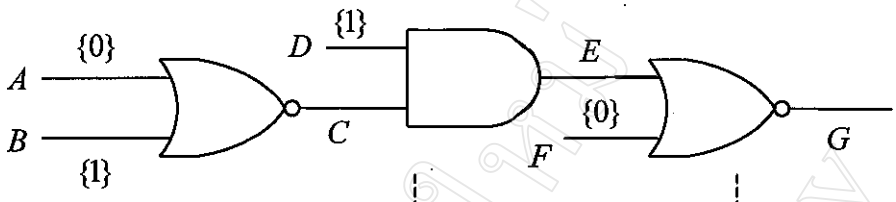
$$A' = A.\overline{mask(A)} + mask(A).fvalue(A) = [11111].[10111] + [01000].[00000]$$
$$= [10111] + [00000] = [10111]$$
$$B' = B.\overline{mask(B)} + mask(B).fvalue(B) = [00000].[11011] + [00100].[00100]$$
$$= [00000] + [00100] = [00100]$$
$$C = A'.B' = [00100]$$
$$C' = C.\overline{mask(C)} + mask(C).fvalue(C) = [00100].[11100] + [00011].[00001]$$
$$= [00100] + [00001] = [00101]$$

ในบิตที่ 3, 5 ผลลัพธ์ของ C' มีค่าตรรกเท่ากับ "1" ซึ่งมีค่าตรงกันข้ามกับบิตที่ 0 ซึ่งเป็นค่าตรรกที่เป็นปกติ ("0") แสดงว่ารูปแบบสัญญาณทดสอบ A="1" และ B="0" สามารถทดสอบจุดเสีย B s-a-1 และ C s-a-0 ได้ ส่วนบิตที่ 2 และ 4 ไม่สามารถทดสอบจุดเสียได้เพราะมีค่าตรรก "0" ซึ่งตรงกับค่าตรรกที่เป็นปกติ

3.3.2 การจำลองการทำงานจุดเสียแบบ Concurrent (Concurrent Fault Simulation)

เป็นเทคนิคที่รวมเอาการจำลองการทำงานจุดเสียแบบขนานและแบบ Deductive ไว้ด้วยกัน การจำลองการทำงานแบบนี้จะตรวจสอบค่าตรรกจุดเสียที่ขั้วขาออกของเกต หรือของวงจรแตกต่างจากค่าตรรกที่เป็นปกติหรือไม่ ดังแสดงในตัวอย่างต่อไปนี้

จากรูปที่ 3.12 กำหนดให้ A=F="0" และ B=D="1" ในระดับที่ 1 มีรายการจุดเสียเป็น A/1, B/0, C/1 แต่ A/1 ไม่สามารถทดสอบได้เพราะว่าค่าตรรกที่ขั้วขาออกของ NOR เกต เท่ากับค่าตรรกที่เป็นปกติ ทำให้สามารถตัดจุดเสีย A/1 ออกจากรายการจุดเสียได้ ในระดับที่ 2 มีรายการจุดเสียเป็น B/0, C/1, D/0, E/1 มีรายการจุดเสีย D/0 ที่ไม่สามารถทดสอบได้ ทำให้ D/0 ถูกตัดออกจากรายการจุดเสีย ในระดับสุดท้าย มีรายการจุดเสียดังนี้ B/0, C/1, E/1/, F1/, G/0 ซึ่งสามารถทดสอบจุดเสียได้ทั้งหมด โดยมีรูปแบบสัญญาณทดสอบ A=F="0" และ B=D="1"



	A	B	C		C	D	E		E	F	G
ค่าตรรกปกติ	***0***	***1***	***0***		***0***	***1***	***0***		***0***	***0***	***1***
รายการจุดเสีย A/1	1	1	0	B/0	1	1	1	B/0	1	0	0
B/0	0	0	1	C/1	1	1	1	C/1	1	0	0
C/1	0	1	1	D/0	0	0	0	E/1	1	0	0
				E/1	0	1	1	F/1	0	1	0
								G/0	0	0	0

รูปที่ 3.12 จำลองการทำงานจุดเสียแบบ Concurrent [21]

มหาวิทยาลัยเชียงใหม่ Chiang Mai University