บทที่ 2
แบบจำลองวงจรดิจิตอล จุดเสีย ค่าตรรก และกระบวนการให้กำเนิดรูปแบบสัญญาณทดสอบ

การประยุกต์คอมพิวเตอร์เพื่อให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรดิจิตอลนั้นสามารถทำได้โดยสร้างแบบจำลองวงจรดิจิตอล แบบจำลองจุดเสียรวมทั้งแบบจำลองค่าตรรกเพื่อเลียนแบบพฤติกรรมหรือจำลองการทำงานของวงจรดิจิตอลทั้งที่มีจุดเสียและไม่มีจุดเสีย

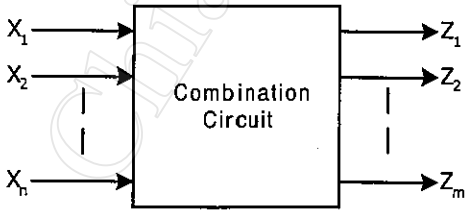
2.1 แบบจำลองวงจรดิจิตอล จุดเสีย ค่าตรรก

2.1.1 วงจรดิจิตอลแบบจำลองวงจรดิจิตอล

ก) วงจรดิจิตอล

วงจรดิจิตอลประกอบขึ้นจากการนำกลุ่มของเกตต่าง ๆ มาต่อกันเป็นวงจรเพื่อให้ทำงานตามฟังก์ชันที่ได้ออกแบบไว้ สามารถแบ่งออกเป็น 2 ชนิด คือ วงจรชนิดประกอบ (Combination Circuit) และวงจรชนิดลำดับ (Sequential Circuit)

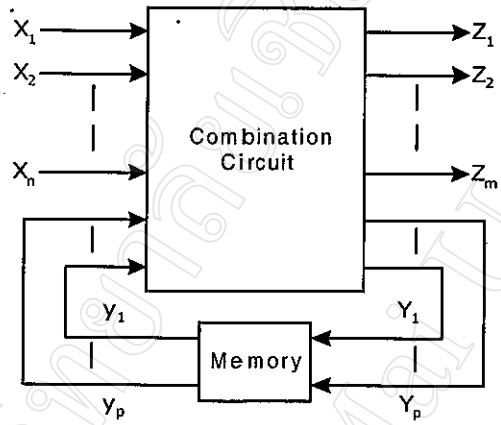
วงจรชนิดประกอบเกิดจากการนำเกตพื้นฐานต่าง ๆ เช่น เกต AND NAND OR NOR หรือ NOT มาต่อเข้าด้วยกันเป็นวงจรเพื่อให้ทำงานใดงานหนึ่งตามที่ได้ออกแบบไว้ ดังภาพที่ 2.1 แสดงบล็อกไดอะแกรมของวงจรชนิดประกอบ ซึ่งมี X_1 , X_2 , ... , X_n เป็นขั้วอินพุตหลัก และ Z_1 , Z_2 , ... , Z_m เป็นขั้วเอาต์พุตหลักของวงจร ค่าตรรกหรือสถานะเอาต์พุตขึ้นอยู่กับฟังก์ชันของวงจรและสถานะของอินพุตที่ป้อนให้กับขั้วอินพุตหลัก ณ เวลาปัจจุบัน



ภาพที่ 2.1 บล็อกไดอะแกรมของวงจรดิจิตอลชนิดประกอบ [11]

วงจรชนิดลำดับประกอบขึ้นจากเกตต่าง ๆ เช่นเดียวกับวงจรชนิดประกอบ แต่มีหน่วยความจำหรือส่วนป้อนกลับเพิ่มขึ้นมา ดังภาพที่ 2.2 แสดงบล็อกไดอะแกรมของวงจรชนิด

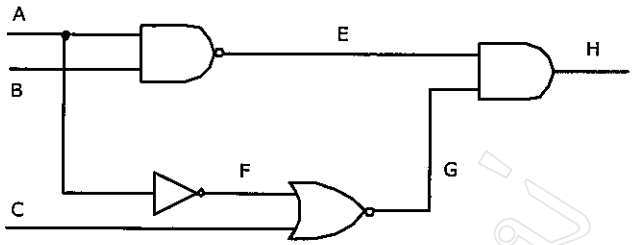
ลำดับ ซึ่งมี $X_1, X_2, \ldots, X_n$ เป็นขั้วอินพุตหลัก $Z_1, Z_2, \ldots, Z_m$ เป็นขั้วเอาต์พุตหลัก $Y_1, Y_2, \ldots, Y_p$ เป็นขั้วอินพุตของหน่วยความจำต่อมาจากส่วนเอาต์พุตของเกตในส่วนวงจรชนิดประกอบ และ $y_1, y_2, \ldots, y_p$ เป็นขั้วเอาต์พุตของหน่วยความจำที่ต่อไปเข้าอินพุตของเกตในส่วนวงจรชนิดประกอบ หน่วยความจำหรือส่วนป้อนกลับได้แก่เกตประเภทฟลิปฟลอป (Flip-Flop) นั่นเอง ส่วนนี้มีหน้าที่เก็บค่าตรรกเอาต์พุตของเกตแล้วนำไปป้อนกลับให้แก่อินพุตเกตใดเกตหนึ่งในช่วงเวลาถัดไป ดังนั้นค่าเอาต์พุตของวงจรชนิดลำดับไม่เพียงแต่ขึ้นอยู่กับฟังก์ชันของวงจรและสถานะของอินพุตที่ป้อนให้กับขั้วอินพุตหลัก ณ เวลาปัจจุบันเท่านั้น แต่ยังขึ้นอยู่กับค่าตรรกที่เก็บไว้ในหน่วยความจำซึ่งเกิดจากการทำงานของวงจรในช่วงเวลาก่อนหน้าอีกด้วย



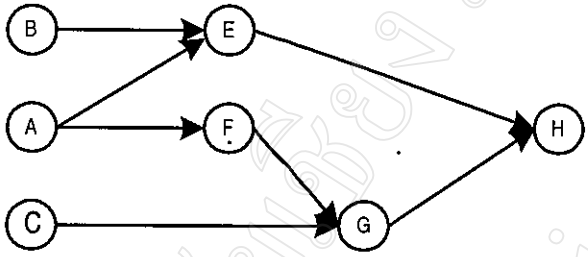
ภาพที่ 2.2 บล็อกไดอะแกรมของวงจรดิจิตอลชนิดลำดับ [11]

ข) แบบจำลองวงจรดิจิตอล (Digital Circuit Model)

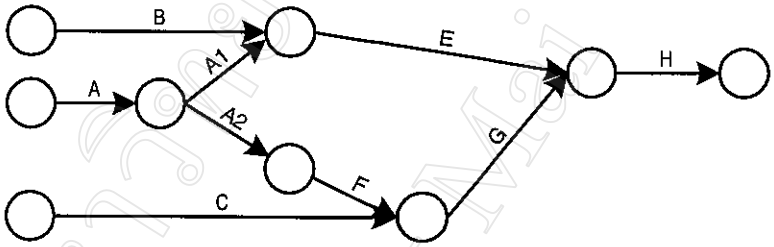
เนื่องจากลักษณะของวงจรดิจิตอลประกอบด้วยเกตหรือโหนดต่าง ๆ และเส้นเชื่อมที่ต่อระหว่างเกตหรือระหว่างโหนด หากแปลงเป็นแบบจำลองทางคณิตศาสตร์หรือโครงสร้างข้อมูลทางคอมพิวเตอร์แล้วแบบจำลองที่เหมาะสมที่สุดได้แก่กราฟ กราฟประกอบด้วยกลุ่มของโหนดและกลุ่มของเส้นเชื่อมระหว่างโหนด และโหนดทุกโหนดต้องมีเส้นเชื่อมต่ออยู่อย่างน้อย 1 เส้นเชื่อม [20] ขั้วอินพุตและเอาต์พุตของเกตต่าง ๆ สามารถแทนด้วยโหนดของกราฟหรือเส้นเชื่อมของกราฟก็ได้ ดังตัวอย่างการแทนวงจรดิจิตอลในภาพที่ 2.3 ก) ด้วยกราฟที่ใช้โหนดของกราฟแทนขั้วอินพุตเอาต์พุตของเกตดังภาพที่ 2.3 ข) และด้วยกราฟที่ใช้เส้นเชื่อมแทนขั้วอินพุตเอาต์พุตของเกตดังภาพที่ 2.3 ค) กราฟที่ใช้จำลองวงจรดิจิตอลต้องเป็นกราฟประเภทมีทิศทาง



ก) วงจรดิจิตอลตัวอย่าง



ข) กราฟที่ใช้โหนดแทนขั้วต่าง ๆ ของเกต



ค) กราฟที่ใช้เส้นเชื่อมแทนขั้วต่าง ๆ ของเกต

ภาพที่ 2.3 การแทนวงจรดิจิตอลด้วยกราฟ

2.1.2 จุดเสียและแบบจำลองจุดเสีย (Fault and Fault Model)

จุดเสียของวงจร VLSI เป็นความบกพร่องทางกายภาพ ซึ่งอาจมีสาเหตุมาจากการออกแบบวงจรผิดพลาด หรือเกิดจากความผิดพลาดในกระบวนการผลิต เช่น อาจเกิดจากความบกพร่องของสารกึ่งตัวนำที่นำมาใช้ หรือเกิดความบกพร่องของจุดต่อหรือเส้นเชื่อมวงจร เป็นต้น [4] จุดเสียแต่ละจุดอาจเกิดจากสาเหตุที่แตกต่างกัน พฤติกรรมของวงจรที่มีจุดเสียก็อาจแตกต่างกันไป ดังนั้นจึงจำเป็นต้องศึกษาพฤติกรรมของจุดเสียเป็นอย่างดีก่อนทำการออกแบบสัญญาณทดสอบ สัญญาณทดสอบสำหรับจุดเสียที่มีพฤติกรรมแบบหนึ่งไม่สามารถใช้ทดสอบกับจุดเสียที่มีพฤติกรรมแบบอื่นได้ จึงต้องเลือกว่าจะทำการทดสอบจุดเสียที่มีพฤติกรรมแบบใด แล้วสร้างแบบ

จำลองจุดเสียที่สามารถเลียนแบบพฤติกรรมทางกายภาพของจุดเสียดังกล่าว ในวงการการทดสอบมีแบบจำลองจุดเสียที่นิยมใช้กันมาก 3 แบบ ได้แก่

ก) แบบจำลองจุดเสียแบบติดค้าง

เป็นแบบจำลองพื้นฐานและนิยมใช้กันมากในการวิเคราะห์และทดสอบวงจรดิจิตอล พฤติกรรมของจุดเสียที่เกิดขึ้นจริงในวงจรมีอยู่ 2 แบบคือ จุดเสียแบบติดค้าง "0" (Stuck-at-0 หรือ s-a-0) และจุดเสียแบบติดค้าง "1" (Stuck-at-1 หรือ s-a-1) คือ โหนดหรือเส้นในวงจรจะมีตรรก "0" หรือ "1" ค้างอยู่ตลอดเวลานั่นเอง สัญลักษณ์ที่ใช้ในการเขียนแทนค่าของจุดเสียนี้ คือ l/s-a-i, l/i หรือ l@i โดย l คือโหนดหรือเส้นในวงจรที่เกิดจุดเสีย i คือค่าตรรก "0" หรือ "1"

ถ้าวงจรที่นำมาทดสอบมีจำนวน m เส้น ภายใต้เงื่อนไขแบบจำลองจุดเสียแบบติดค้างเดี่ยว วงจรนั้นก็จะมีจุดเสียที่สามารถเกิดขึ้นได้เท่ากับ 2m จุด คือ ในแต่ละเส้นของวงจรสามารถเกิดจุดเสียได้ทั้ง s-a-0 และ s-a-1 แต่กรณีที่ในวงจรเกิดจุดเสียได้มากกว่า 1 จุดพร้อมกัน จำนวนจุดเสียที่สามารถเกิดขึ้นได้เท่ากับ 3^m-1 จุด คือ จุดหรือเส้นในวงจรเกิดจุดเสียได้ทั้ง s-a-0, s-a-1 และไม่มีจุดเสีย

ข) แบบจำลองจุดเสียแบบลัดวงจร

จุดเสียแบบนี้เกิดขึ้นจากการลัดวงจรระหว่างจุดหรือเส้นในวงจร จุดเสียแบบนี้อาจไม่มีผลต่อการทำงานของวงจรถ้าจุดเสียที่ลัดวงจรกันนั้นมีค่าตรรกเดียวกัน แต่ถ้าต่างกันก็จะมีผลทำให้การทำงานของวงจรเกิดผิดพลาดขึ้นทันที ซึ่งก็ขึ้นอยู่กับเทคโนโลยีในการสร้างวงจรหรืออุปกรณ์นั้น ๆ เช่น ถ้าเป็น TTL (Transistor-Transistor Logic) จะมีค่าตรรก "0" เป็นตัวข่ม คือ ถ้าเกิดโหนดหรือเส้นที่ลัดวงจรกันมีตรรกเป็น "0" และ "1" ผลลัพธ์ของค่าตรรกที่จุดเสียจะมีค่าเป็น "0" หรือถ้าเป็น ECL (Emitter-Coupling Logic) จะมีค่าตรรก "1"

ค) แบบจำลองจุดเสียแบบทำงานช้า

จุดเสียแบบทำงานช้า จุดเสียแบบนี้จะส่งผลให้การทำงานของวงจรล้าช้าลง วงจรจะไม่สามารถตอบสนองต่อการเปลี่ยนแปลงของสัญญาณได้ตามฟังก์ชันที่ออกแบบไว้

2.1.3 แบบจำลองค่าตรรก (Logic Value Model)

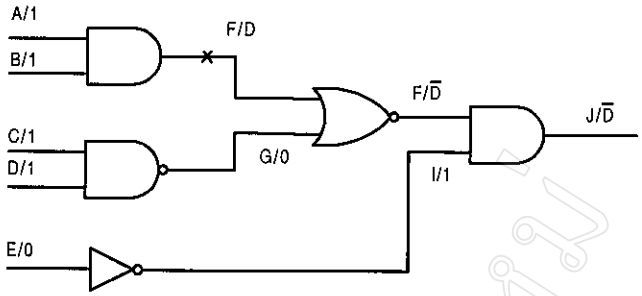
แบบจำลองตรรกที่ใช้ในวงจรดิจิตอลโดยทั่วไปเป็นแบบจำลองตรรก 2 ค่า {1, 0} หรือแบบจำลองตรรก 3 ค่า {1, 0, X} ซึ่งไม่เพียงพอสำหรับการวิเคราะห์วงจรที่มีจุดเสีย ไม่สามารถที่

จะแสดงจุดเสียภายในวงจรได้ แบบจำลองที่นิยมใช้ในการทดสอบวงจรดิจิตอลได้แก่แบบจำลองค่าตรรก 5 ค่า แบบจำลองค่าตรรก 9 ค่า และแบบจำลองค่าตรรก 16 ค่า การเลือกใช้แบบจำลองแบบใดนั้นขึ้นอยู่กับหลักวิธีที่ใช้ในการให้กำเนิดรูปแบบสัญญาณทดสอบและการจำลองการทำงานของจุดเสีย การใช้แบบจำลองตรรกที่เหมาะสมจะช่วยลดจำนวนของทางเลือกในการตัดสินใจและจำนวนการทำงานแบบย้อนรอยของโปรแกรม หลักวิธีที่นำเสนอใช้แบบจำลองค่าตรรก 5 ค่า ซึ่งคิดค้นขึ้นโดย Roth [21] ประกอบด้วยค่าตรรกจำลอง 5 ค่า แสดงทั้งค่าตรรกของวงจรที่ดีและค่าตรรกของวงจรที่เสียไปพร้อม ๆ กัน รายละเอียดแสดงในตารางที่ 2.1 โดย D คือ จุดเสียแบบติดค้าง "0" หากวงจรไม่เกิดจุดเสียจะมีค่าตรรก "1" แต่หากมีจุดเสียเกิดขึ้นจะมีค่าตรรก "0" เช่นเดียวกัน $\overline{D}$ คือ จุดเสียแบบติดค้าง "1" หากไม่เกิดจุดเสียจะมีค่าตรรก "0" หากมีจุดเสียขึ้นมาจะมีค่า "0" ส่วน X เป็นค่าตรรกอะไรก็ได้ แบบจำลองค่าตรรก 5 ค่านี้นิยมใช้กันมากในการทดสอบวงจรวงจรดิจิตอล

ตารางที่ 2.1. แบบจำลองค่าตรรก 5 ค่า

สัญลักษณ์	คำอธิบาย
0	ค่าตรรก "0"
1	ค่าตรรก "1"
X	ค่าตรรกอะไรก็ได้ (Don't care value)
D	ค่าตรรก "1" ในวงจรที่ดีและ "0" ในวงจรที่มีจุดเสีย (s-a-0)
$\overline{D}$	ค่าตรรก "0" ในวงจรที่ดีและ "1" ในวงจรที่มีจุดเสีย (s-a-1)

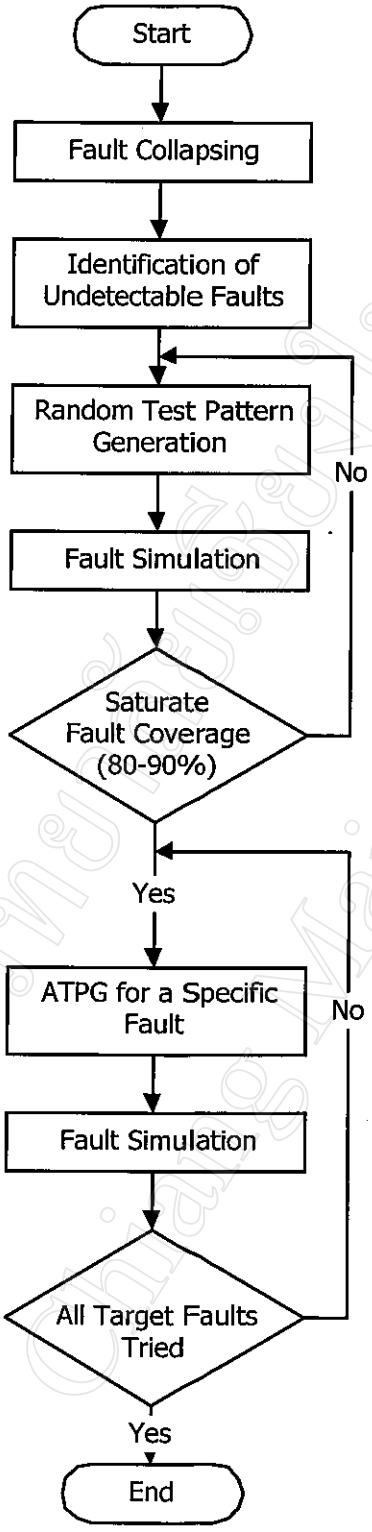
ภาพที่ 2.4 เป็นการจำลองการทำงานของวงจรดิจิตอลซึ่งเกิดจุดเสีย F/s-a-0 (F/D) ขึ้น เมื่อป้อน "11110" .ให้กับขั้วอินพุตหลัก ABCDE ของวงจร จะได้ค่าตรรก $\overline{D}$ ที่ขั้วเอาต์พุตหลัก ซึ่งแสดงว่าหากวงจรไม่มีจุดเสียเกิดขึ้นค่าตรรกเอาต์พุตของวงจรต้องมีค่าเท่ากับ "0" แต่หากมีจุดเสีย F/s-a-0 เกิดขึ้นค่าตรรกเอาต์พุตที่ขั้ว J จะมีค่าเป็น "1" สัญลักษณ์คือ $\overline{D}$ นั่นเอง



ภาพที่ 2.4 การจำลองการทำงานของวงจรดิจิตอลที่มีจุดเสียโดยใช้แบบจำลองค่าตรรก 5 ค่า

2.2 กระบวนการให้กำเนิดรูปแบบสัญญาณทดสอบ

กระบวนการให้กำเนิดรูปแบบสัญญาณทดสอบแบ่งได้เป็น 3 ขั้นตอน ขั้นตอนแรกคือการเตรียมการก่อนการให้กำเนิดรูปแบบสัญญาณทดสอบ (Preprocessing) ทำการแปลงข้อมูลวงจรดิจิตอลให้เป็นแบบจำลองวงจร สร้างและยุบรายการจุดเสีย บางหลักวิธีอาจทำการเรียนรู้พฤติกรรมของวงจรและแยกรายการจุดเสียบางส่วนที่ไม่สามารถตรวจสอบออกจากรายการจุดเสียทั้งหมด ขั้นตอนที่สองเป็นการให้กำเนิดรูปแบบสัญญาณทดสอบ การให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับแต่ละจุดเสียเป็นงานค้นหาคำตอบจากพื้นที่ค้นหาที่กว้างมาก หากต้องค้นหาคำตอบทุกจุดเสียจะทำให้เสียเวลามาก การให้กำเนิดรูปแบบสัญญาณทดสอบจึงแบ่งออกเป็น 2 ช่วง ช่วงแรกเป็นการให้กำเนิดรูปแบบสัญญาณทดสอบแบบสุ่ม (Random TPG) ทำการสุ่มคำตอบขึ้นมาแล้วทำการจำลองการทำงานของจุดเสีย พิจารณาว่าจุดเสียใดบ้างที่สามารถทดสอบด้วยรูปแบบสัญญาณนี้ ทำการให้กำเนิดรูปแบบสัญญาณทดสอบแบบสุ่มจนกว่าจะไม่สามารถครอบคลุมจุดเสียที่เหลืออยู่เพิ่มเติมได้หรือจนเป็นที่พอใจ ช่วงที่สองเป็นการให้กำเนิดรูปแบบสัญญาณสำหรับจุดเสียเฉพาะจุดเป็นจุดเสียที่ยากต่อการหารูปแบบสัญญาณทดสอบ ให้กำเนิดรูปแบบสัญญาณทดสอบจนกว่าจะครอบคลุมจุดเสียได้ทั้งหมดหรือเป็นที่พอใจ ช่วงที่สองนี้ยังเรียกใช้การจำลองการทำงานของจุดเสียเช่นกัน ขั้นตอนสุดท้ายคือการวัดประสิทธิภาพของรูปแบบสัญญาณทดสอบที่ได้ (Fault Evaluate) โดยวัดจากค่าครอบคลุมจุดเสีย (Fault Coverage) ที่เป็นอัตราส่วนระหว่างจำนวนจุดเสียที่สามารถทดสอบได้ต่อจำนวนจุดเสียทั้งหมด ค่าครอบคลุมจุดเสียควรมีค่า 100% หรือใกล้เคียงที่สุด [25] โดยใช้การจำลองการทำงานของจุดเสียเป็นเครื่องมือในการวัดค่าครอบคลุมจุดเสีย นอกจากนี้ยังวัดจากจำนวนรูปแบบสัญญาณทดสอบและเวลาที่ใช้ในการให้กำเนิด กระบวนการให้กำเนิดรูปแบบสัญญาณทดสอบแสดงดังภาพที่ 2.5

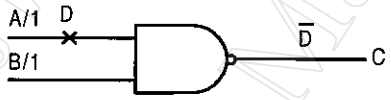


ภาพที่ 2.5 กระบวนการให้กำเนิดรูปแบบสัญญาณทดสอบ [21]

2.2.1 การสร้างรายการจุดเสียและการยุบรวมจุดเสีย (Fault List and Fault Collapsing)

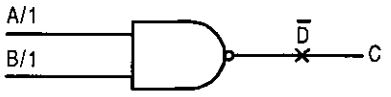
การสร้างรายการจุดเสียเป็นขั้นตอนต่อจากการแปลงวงจรดิจิตอลให้อยู่ในแบบจำลองวงจรทางคอมพิวเตอร์ กรณีที่ใช้แบบจำลองจุดเสียแบบติดค้างเดี่ยวหากวงจรที่นำมาทดสอบมีโหนดหรือเส้นเชื่อมวงจรอยู่ n เส้นจะมีจุดเสียทั้งหมด 2n จุด นั่นคือแต่ละโหนดมีโอกาสเกิดจุดเสียได้ทั้ง s-a-0 และ s-a-1 จุดเสียทั้งหมดในวงจรจะมีบางจุดเสียที่ใช้รูปแบบสัญญาณทดสอบรูปแบบเดียวกัน เช่น เกต NAND ในภาพที่ 2.6 มี 2 ขั้วอินพุต A กับ B มีขั้วเอาต์พุต C จะมีรูปแบบสัญญาณทดสอบสำหรับ A/s-a-0 B/s-a-0 และ C/s-a-1 เหมือนกัน คือ "11" และได้ค่าตรรกเอาต์พุต $\overline{D}$ เหมือนกันทั้ง 3 จุดเสีย

ดังนั้นแทนที่จะทำการให้กำเนิดรูปแบบสัญญาณทดสอบทั้ง 3 จุดก็ทำเพียงจุดเดียว โดยการยุบรวมจุดเสียทั้ง 3 จุดให้เป็นจุดเสียเดียว จะช่วยลดเวลาในการประมวลผลและจำนวนรูปแบบสัญญาณทดสอบลง การยุบรวมจุดเสีย คือ การจัดกลุ่มจุดเสียที่สามารถใช้รูปแบบสัญญาณทดสอบรูปแบบเดียวกัน แล้วเลือกจุดเสียเพื่อเป็นตัวแทนจุดเสียทั้งหมดในกลุ่มเพียงจุดเดียว โดยทั่วไปการยุบรวมจุดเสียแบ่งเป็น 2 วิธี คือ การยุบรวมจุดเสียที่สมมูลกัน (Equivalence Fault Collapsing) และการยุบรวมจุดเสียที่สมนัยกัน (Imply Fault Collapsing)



ก) รูปแบบสัญญาณทดสอบสำหรับเกต NAND ซึ่งมีจุดเสีย A/s-a-0

ข) รูปแบบสัญญาณทดสอบสำหรับเกต NAND ซึ่งมีจุดเสีย B/s-a-0



ค) รูปแบบสัญญาณทดสอบสำหรับเกต NAND ซึ่งมีจุดเสีย C/s-a-1

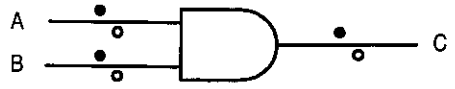
ภาพที่ 2.6 รูปแบบสัญญาณทดสอบสำหรับเกต NAND ซึ่งมีจุดเสีย

ก) การยุบรวมจุดเสียที่สมมูลกัน

จะทำการยุบรวมจุดเสียสำหรับเกต OR NOR AND หรือ NAND หลักการคือทำการยุบรวมจุดเสีย s-a-c ของทุกขั้วอินพุตเข้ากับจุดเสีย s-a-($c \oplus i$) ของขั้วเอาต์พุต แล้วใช้จุดเสีย s-a-($c \oplus i$) ของขั้วเอาต์พุตเป็นตัวแทน โดย c คือ ค่าควบคุม (Controlling Value) และ i คือค่าตรงข้าม (Inversion Value) เช่น เกต AND มี 2 ขั้วอินพุต A กับ B และขั้วเอาต์พุต C จะมีจุดเสียทั้งหมด 6 จุด แสดงดังภาพที่ 3.3 ก) จุดสีดำด้านบนของเส้นเชื่อมแทนจุดเสีย s-a-1 และจุดสีขาวด้านล่างของเส้นเชื่อมแทนจุดเสีย s-a-0 เมื่อทำการยุบจุดเสีย s-a-0 ทุกขั้วอินพุตเข้ากับจุดเสีย s-a-1 ของขั้วเอาต์พุต จะเหลือจุดเสียเพียง 4 จุดดังภาพ 2.7 ข) นอกจากเกตทั้ง 4 เกตข้างต้นแล้วยังสามารถยุบในเกต NOT ได้ โดยยุบ s-a-0 (และ s-a-1) ที่ขั้วอินพุตเข้ากับ s-a-1 (และ s-a-0) ที่ขั้วเอาต์พุต ดังนั้นเกต NOT จะเหลือตัวแทนจุดเสียเพียง 2 จุด

ข) การยุบรวมจุดเสียที่สมนัยกัน

นอกเหนือจากการยุบรวมจุดเสียที่สมมูลกันแล้ว ยังมีจุดเสียที่ขั้วเอาต์พุตของเกตอีกจุดหนึ่งที่สามารถใช้รูปแบบสัญญาณทดสอบจุดเสียที่ขั้วอินพุตบางจุดทดสอบได้ ดังเกต NAND ในภาพที่ 2.8 ก) มีจุดเสีย A/s-a-1 ใช้รูปแบบสัญญาณทดสอบ "01" สำหรับการตรวจสอบ ซึ่ง "01" นี้สามารถใช้ตรวจสอบจุดเสีย C/s-a-0 ได้เช่นกัน ดังภาพ 2.8 ข) ดังนั้นจึงสามารถยุบจุดเสียทั้ง 2 จุดนี้ให้เหลือตัวแทนเพียงจุดเดียว เรียกว่าการยุบจุดเสียที่สมนัยกัน จะทำการยุบรวมจุดเสียสำหรับเกต OR NOR AND หรือ NAND หลักการ คือ ทำการยุบจุดเสีย s-a-($\bar{c} \oplus$ i) ของขั้วเอาต์พุตเข้ากับจุดเสีย s-a-$\bar{c}$ ของบางขั้วอินพุต แล้วใช้จุดเสีย s-a-$\bar{c}$ ของขั้วอินพุตเป็นตัวแทน เช่น เกต AND ในภาพที่ 2.7 ข) ทำการยุบรวมจุดเสียที่สมนัยกัน โดยยุบรวมจุดเสีย C/s-a-1 เข้ากับจุดเสีย A/s-a-1 จะเหลือจุดเสียเพียง 3 จุด ดังภาพที่ 2.7 ค)



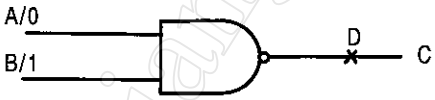
ก) รายการจุดเสียทั้งหมด

ข) จุดเสียที่ผ่านการยุบรวมจุดเสียที่สมมูลกัน

ค) จุดเสียที่ผ่านการยุบรวมจุดเสียที่สมดุลและสมนัยกัน

ภาพที่ 2.7 รายการจุดเสียของเกต AND 2 ขั้วอินพุต

ก) รูปแบบสัญญาณทดสอบสำหรับเกต NAND ซึ่งมีจุดเสีย A/s-a-1



ข) รูปแบบสัญญาณทดสอบสำหรับเกต NAND ซึ่งมีจุดเสีย C/s-a-0

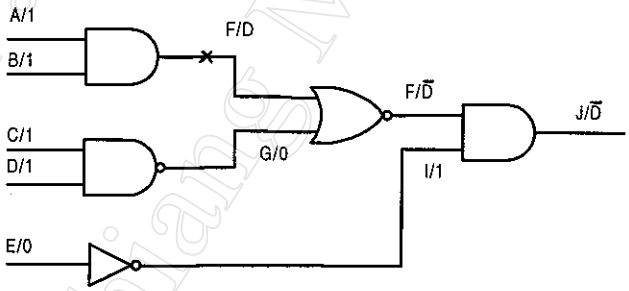
ภาพที่ 2.8 รูปแบบสัญญาณทดสอบสำหรับเกต NAND

มหาวิทยาลัยเชียงใหม่ Chiang Mai University

2.2.2 การให้กำเนิดรูปแบบสัญญาณทดสอบ

ก) การให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรชนิดประกอบ

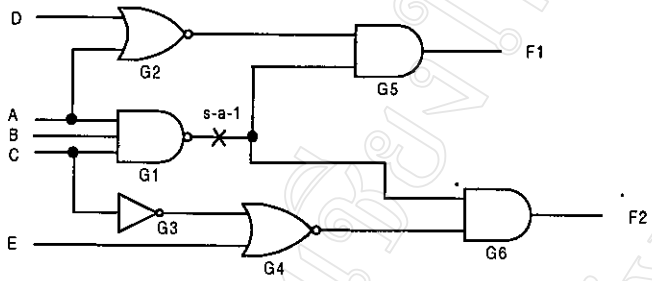
วิธีการให้กำเนิดรูปแบบสัญญาณทดสอบแบ่งเป็น 2 ขั้นตอนใหญ่ ๆ คือ ทำการกระตุ้นจุดเสียให้มีค่าตรรกตรงข้ามกับค่าของจุดเสีย (Fault Activation หรือ Fault Exciting) และทำการแพร่ผลของจุดเสียหรือค่าความผิดปรกติให้ไปปรากฏที่ขั้วเอาต์พุตหลักของวงจร (Fault Effect Propagation) ซึ่งทั้ง 2 ขั้นตอนมีงานที่เหมือนกัน คือ ต้องทำการปรับค่าตรรกแก่โหนดหรือเส้นเชื่อมต่าง ๆ ภายในวงจร (Line Justification) ให้ถูกต้อง ดังตัวอย่างการให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรดิจิตอลในภาพที่ 2.9 ซึ่งมีจุดเสีย F/s-a-0 วิธีการให้กำเนิดรูปแบบสัญญาณทดสอบเริ่มจากการกระตุ้นจุดเสียให้มีค่าตรรกตรงข้าม โดยต้องกำหนด A/1 และ B/1 หากวงจรเกิดจุดเสีย F/s-a-0 ขึ้นค่าตรรกของโหนด F จะมีค่าเป็น "0" แต่หากไม่มีจุดเสียเกิดขึ้นค่าตรรกจะเป็น "1" ในแบบจำลองค่าตรรกแบบ 5 ค่าจะใช้ค่า D จากนั้นทำการแพร่ค่าผลของจุดเสียให้ไปปรากฏที่ขั้วเอาต์พุตหลักของวงจร โดยทำการแพร่ค่าผ่านโหนด H และ J ตามลำดับ จึงจำเป็นต้องกำหนด G/0 และ I/1 งานที่ต้องทำต่อคือการปรับค่าให้แก่โหนดต่าง ๆ เพื่อให้ได้ G/0 และ I/1 ซึ่งสามารถทำได้โดยการกำหนด C/1 D/1 และ E/0 ดังนั้นรูปแบบสัญญาณทดสอบที่ใช้สำหรับตรวจสอบจุดเสีย F/s-a-1 คือ "11110" ค่าตรรกที่ขั้วเอาต์พุตหลักมีค่าเท่ากับ $\overline{D}$



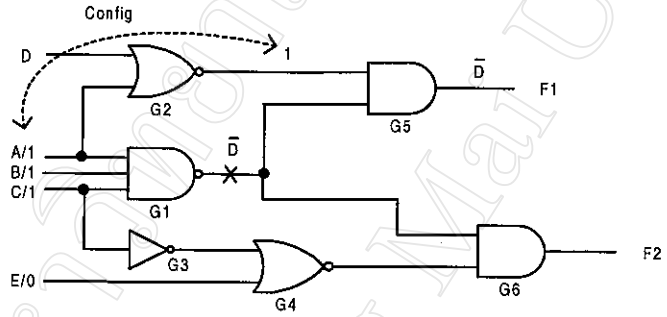
ภาพที่ 2.9 การให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรที่ไม่มีแฟนเอาต์ [5]

วงจรตัวอย่างในภาพที่ 2.9 จะไม่มีปัญหาในการให้กำเนิดรูปแบบสัญญาณทดสอบเนื่องจากเป็นวงจรที่ไม่มีโหนดแฟนเอาต์ แต่หากวงจรมีโหนดแฟนเอาต์และโหนดรีคอนเวอร์เจนต์-แฟนเอาต์ ซึ่งเป็นโหนดรวมสัญญาณทั้งหมดที่แยกมาจากโหนดแฟนเอาต์เดียวกัน จะทำให้การปรับค่าตรรกให้แก่โหนดต่าง ๆ ไม่เป็นไปอย่างอิสระ ดังตัวอย่างการให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรในภาพที่ 2.10 ก) ซึ่งมีจุดเสีย G1/s-a-1 วิธีการคือทำการกระตุ้นจุดเสียให้มีค่าตรรกตรงข้ามกับค่าที่เกิดจากจุดเสียและทำการแพร่ผลของจุดเสียเช่นเดิม จากวงจรจะเห็นว่าใน

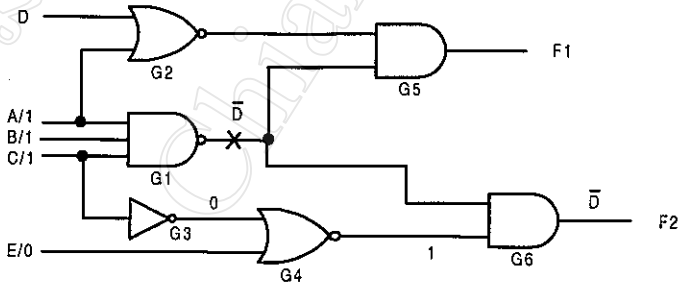
การแพร่ผลที่เกิดจากจุดเสีย G1/s-a-1 นั้นมี 2 ทางเลือกที่สามารถแพร่ไปได้ คือ เลือกผ่าน G5 หรือเลือกผ่าน G6 สมมุติว่าเลือกที่จะแพร่ค่าผ่าน G5 ต้องกำหนด G2/0 การกระตุ้นให้ G1 มีค่าเท่ากับ $\overline{D}$ นั้นจำเป็นต้องกำหนดอินพุต A/1 B/1 และ C/1 แต่การกำหนด G2/0 นั้นจำเป็นต้องกำหนด A/0 และ D/0 จะเห็นว่าโหนด A มีความขัดแย้ง (Conflict) เกิดขึ้นมา ดังภาพที่ 2.10 ข) ไม่สามารถที่จะ



ก) วงจรดิจิตอลที่มีจุดเสีย G1/s-a-1



ข) เลือกแพร่ค่าผลของจุดเสียผ่าน G5 ทำให้เกิดความขัดแย้ง



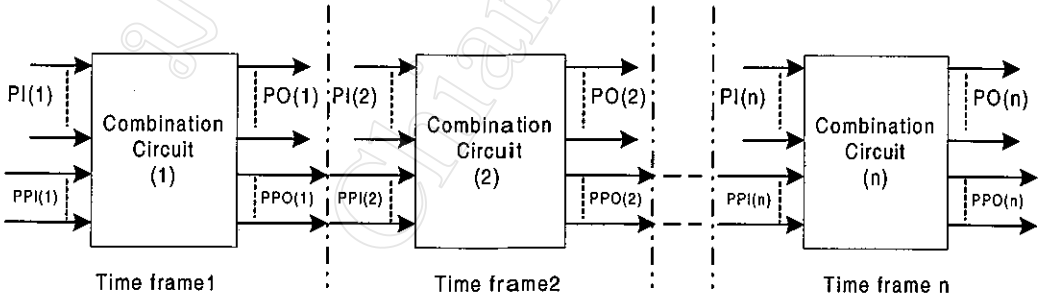
ค) เลือกแพร่ค่าผลของจุดเสียผ่าน G6 สามารถให้กำเนิดรูปแบบสัญญาณทดสอบได้

ภาพที่ 2.10 การให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรที่มีแฟนเอาต์

แพร่ค่าผลของจุดเสียผ่าน G5 ได้ จำเป็นต้องเปลี่ยนเป็นผ่าน G6 แทน ต้องกำหนด G4/1 สามารถทำได้โดยกำหนด E/0 ได้ F2/$\overline{D}$ ดังนั้นรูปแบบสัญญาณทดสอบสำหรับจุดเสีย G1/1 คือ "111X0" ดังภาพที่ 2.10 ค)

ข) การให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรชนิดลำดับ

เนื่องจากวงจรดิจิตอลชนิดลำดับนั้นมีส่วนของหน่วยความจำหรือฟลิบฟลอบเข้ามาเพิ่มเข้ามา ค่าตรรกที่เอาต์พุตไม่ได้ขึ้นอยู่กับฟังก์ชันและค่าตรรกอินพุตเท่านั้นแต่ยังขึ้นอยู่กับค่าตรรกก่อนหน้าที่ป้อนกลับเข้ามาอีกด้วย ดังนั้นจุดเสียแต่ละจุดอาจจะใช้รูปแบบสัญญาณทดสอบมากกว่า 1 รูปแบบสัญญาณ หลักวิธีในการให้กำเนิดรูปแบบสัญญาณทดสอบสำหรับวงจรชนิดลำดับที่นิยมทำโดยทั่ว ๆ ไปคือทำการแปลงวงจรชนิดลำดับให้เป็นวงจรชนิดประกอบโดยการตัดส่วนป้อนกลับหรือฟลิปฟลอปทิ้งไป ทำการสำเนาวงจรชนิดประกอบที่ได้เป็นหลาย ๆ วงจร (Iterative Combination Circuit) เรียกแต่ละวงจรว่ากรอบเวลา หรือ Time Frame เมื่อตัดฟลิปฟลอปออกขั้วเอาต์พุตของฟลิปฟลอบจะกลายเป็นขั้วอินพุตเทียมหรือ PPI (Pseudo Primary Input) และขั้วอินพุตของฟลิปฟลอปกลายเป็นขั้วเอาต์พุตเทียมหรือ PPO (Pseudo Primary Output) นำ PPI ต่อเข้ากับ PPO ของกรอบเวลาก่อนหน้า [16] ทำเหมือนกันทุกกรอบเวลาดังภาพที่ 2.11 การให้กำเนิดรูปแบบสัญญาณทดสอบทำเช่นเดียวกับวงจรชนิดประกอบ โดยเริ่มทำในกรอบเวลาแรกก่อน หากไม่สามารถแพร่ค่าผลของจุดเสียให้ปรากฏที่ขั้วเอาต์พุตหลักของวงจรได้ ยังไม่ถือว่าจุดเสียนี้ไม่สามารถตรวจสอบได้ ให้เลื่อนไปทำในกรอบเวลาถัดไปพร้อมทั้งแพร่ค่าตรรกขั้ว PPO ไปยังขั้ว PPI ของกรอบเวลาถัดไปด้วย [11] [18]



ภาพที่ 2.11 แบบจำลองวงจรชนิดประกอบหลายกรอบเวลาต่อกัน

2.2.3 การจำลองการทำงานของจุดเสีย (Fault Simulation)

ในการให้กำเนิดรูปแบบสัญญาณทดสอบการจำลองการทำงานของจุดเสียนับว่าเป็นงานสำคัญมาก หน้าที่หลักของการจำลองการทำงานจุดเสียคือการหาจุดเสียที่สามารถตรวจสอบได้ด้วยรูปแบบสัญญาณทดสอบที่ให้กำเนิดขึ้นมา และยังเป็นเครื่องมือในการคำนวณค่าครอบคลุมจุดเสีย

การจำลองการทำงานจุดเสียแบ่งออกเป็น 4 แบบ ได้แก่

- การจำลองการทำงานของจุดเสียแบบอนุกรม (Serial Fault Simulation)
- การจำลองการทำงานของจุดเสียแบบขนาน (Parallel Fault Simulation)
- การจำลองการทำงานของจุดเสียแบบคอนดักตีฟ (Conductive Fault Simulation)
- การจำลองการทำงานของจุดเสียแบบคอนเคอร์เรนท์ (Concurrent Fault Simulation)

ก) การจำลองการทำงานของจุดเสียแบบอนุกรม

เป็นการจำลองการทำงานของจุดเสียที่ง่ายที่สุด โดยทำการเปลี่ยนการจำลองการทำงานของวงจรที่ดีให้เป็นการทำงานของวงจรที่มีจุดเสีย นั่นคือใส่ค่าจุดเสียที่สนใจเข้าไปในวงจร จำลองการทำงานของวงจรแล้วสังเกตดูว่าค่าผลที่เกิดจากจุดเสีย ซึ่งได้แก่ค่า D หรือ $\overline{D}$ ปรากฏที่ขั้วเอาต์พุตหลักหรือไม่ ถ้ามีปรากฏที่ขั้วเอาต์พุตหลักแสดงว่าสามารถตรวจสอบจุดเสียจุดนั้น ๆ ได้ ข้อดีของการจำลองการทำงานของจุดเสียแบบอนุกรมคือไม่ต้องอาศัยหลักวิธีที่ยุ่งยากใช้หลักวิธีเดียวกับการจำลองการทำงานของวงจรที่ดี แต่ข้อเสียคือใช้เวลาในการประมวลผลนาน เพราะในแต่ละจุดเสียต้องทำการจำลองการทำงาน 1 ครั้งเสมอ [11]

ข) การจำลองการทำงานของจุดเสียแบบขนาน

เนื่องจากคอมพิวเตอร์มีคำสั่งกระทำระดับบิต โดยเฉพาะคำสั่งทางตรรก เช่น AND OR XOR หรือ NOT ซึ่งคำสั่งกระทำระดับบิตจะมีการประมวลผลแต่ละบิตอย่างเป็นอิสระต่อกัน ดังนั้นหากคอมพิวเตอร์มี 32 บิตก็จะสามารถแก้ปัญหาได้ 32 ปัญหาพร้อมกัน ดังนั้นหากแทนค่าจุดเสีย 1 จุดด้วยข้อมูลขนาด 1 บิต จะสามารถจำลองการทำงานจุดเสียพร้อมกันหลาย ๆ จุดเสีย กล่าวคือถ้าคอมพิวเตอร์มีขนาด n บิตจะสามารถจำลองการทำงานจุดเสียได้ n-1 จุดพร้อม ๆ กัน ส่วนอีกบิตหนึ่งสำหรับจำลองการทำงานของวงจรที่ดี ในการจำลองนี้จะใช้หน้ากาก (Mask) เป็นตัวแยกจัดการระดับบิต หน้ากากที่ใช้มี 2 ตัว คือ mask(S) และ fvalue(S) โดยที่

$mask(S)_i = 1$ ถ้ามีจุดเสียเกิดขึ้นบนโหนด S

$mask(S)_i = 0$ ถ้าไม่มีจุดเสียเกิดขึ้นบนโหนด S

$fvalue(S)_i = 1$ ถ้าจุดเสียที่เกิดบนโหนด S เป็นจุดเสียแบบติดค้าง 1

$fvalue(S)_i = 0$ ถ้าจุดเสียที่เกิดบนโหนด S เป็นจุดเสียแบบติดค้าง 0

i คือบิตใด ๆ

การจำลองการทำงานของจุดเสียระดับบิตสามารถทำโดยการใช้หน้ากากมากรองเพื่อคำนวณค่าแต่ละบิตแต่ละจุดเสียอย่างเป็นอิสระต่อกัน ดังภาพที่ 2.12 หน้ากากที่นำมากรองสร้างขึ้นจากสมการที่ 2.1 โดยที่สัญลักษณ์ + . และ – แทนตัวกระทำ OR AND และ NOT ตามลำดับ

$$S' = S.\overline{mask(S)} + mask(S).fvalue(S) \qquad (2.1)$$

ตัวอย่างการจำลองการทำงานของจุดเสียเกต NOR ที่มีขั้วอินพุต A B ขั้วเอาต์พุต C มีจุดเสีย A/s-a-1 B/s-a-0 C/s-a-1 และ C/s-a-0 ด้วยรูปแบบสัญญาณทดสอบ "01" ขนาดของข้อมูล 5 บิต กำหนดบิตสำหรับแทนจุดเสียดังตารางที่ 2.2 และตารางที่ 2.3 แสดงค่าหน้ากาก mask และ fvalue

ตารางที่ 2.2 การกำหนดบิตสำหรับแทนค่าจุดเสีย

บิต	แทนจุดเสีย
0	ไม่มีจุดเสีย
1	A/s-a-1
2	B/s-a-0
3	C/s-a-1
4	C/s-a-0

ตารางที่ 2.3 ค่า mask และ fvalue ของโหนดต่าง ๆ

โหนด	Mask	Fvalue
A	01000	01000
B	00100	00000
C	00011	00010

เมื่อทำการจำลองการทำงานจะได้

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

$$A' = [00000]\ .\ [10111] + [0100O].[01000]\ =\ [00000] + [01000]\ =\ [01000]$$

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

$$B' = [11111]\ .\ [11011] + [0010O].[00000]\ =\ [11011] + [00000]\ =\ [11011]$$

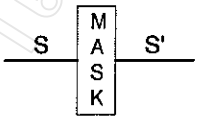
$$C = \overline{A'+B'}$$

$$C = \overline{[01000] + [11011]}\ =\ \overline{[11011]}\ =\ [00100]$$

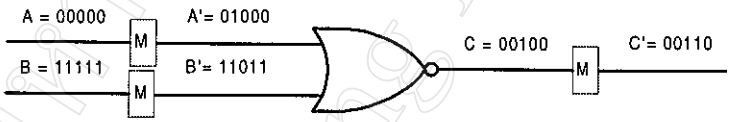
$$C' = C.\overline{mask\ (C)} + mask\ (C).fvalue\ (C)$$

$$C' = [00100]\ .\ [11100] + [00011].[00010]\ =\ [00100] + [00010]\ =\ [00110]$$

ค่าตรรกของโหนดต่าง ๆ แสดงดังภาพที่ 2.12ข)



ก) การใช้หน้ากากจัดการแยกแต่ละบิต



ข) การจำลองการทำงานของจุดเสียแบบขนานของเกต NOR 2 อินพุต

ภาพที่ 2.12 การจำลองการทำงานของจุดเสียแบบขนาน [11]

พิจารณาค่า C' มีค่าเท่ากับ [00110] บิต 0 คือค่าเอาท์พุทของวงจรที่ไม่มีจุดเสียมีค่าเท่ากับ "0" ดังนั้นจุดเสียที่สามารถทดสอบได้ต้องทำให้วงจรมีค่าเอาต์พุทเท่ากับ "1" ซึ่งมี 2 จุด คือ บิตที่ 2 และ 3 แทนจุดเสีย B/s-a-0 และ C/s-a-1 ตามลำดับ จึงสามารถตรวจสอบจุดเสียทั้งสองได้

ค) การจำลองการทำงานของจุดเสียแบบดีดักตีฟ

แบบจำลองจุดเสียแบบดีดักตีฟถูกพัฒนาขึ้นโดย Armstrong ในปี ค.ศ. 1972 [11] มีขั้นตอนในการหาจุดเสียที่สามารถทดสอบได้ดังนี้

ขั้นตอนที่ 1) นำรูปแบบสัญญาณทดสอบที่ได้ไปป้อนให้กับขั้วอินพุตหลักและทำการจำลองการทำงานของวงจรที่ดี ดังนั้นแต่ละโหนดจะได้ค่าตรรก 1 ค่า

ขั้นตอนที่ 2) ใช้พฤติกรรมของวงจรที่ดีตัดจุดเสียในแต่ละโหนดที่ไม่สามารถตรวจสอบได้ออกไป โดยทำการเปรียบเทียบผลของจุดเสียกับค่าตรรกที่ได้จากการจำลองการทำงานของวงจรในขั้นตอนที่ 1) หากผลของจุดเสียมีค่าเดียวกับค่าตรรกของวงจรดีแสดงว่าไม่สามารถทำการตรวจสอบจุดเสียนี้ได้ให้ทำการตัดจุดเสียนี้ออกไป ดังนั้นแต่ละโหนดจะเหลือรายการจุดเสียที่สามารถตรวจสอบได้ L_s โดย s คือโหนดหรือเส้นเชื่อมใด ๆ ตัวอย่างเช่น เกต AND 3 ขั้วอินพุต A B และ C มี Y เป็นขั้วเอาต์พุต มี "110" เป็นรูปแบบสัญญาณทดสอบ เมื่อทำการตัดจุดเสียที่ไม่สามารถตรวจสอบได้ จะได้ L_s ดังนี้

$L_A = \{ A/0 \}$ หรือ $\{A/s\text{-}a\text{-}0\}$

$L_B = \{ B/0 \}$

$L_C = \{ C/1 \}$

$L_Y = \{ Y/1 \}$

ขั้นตอนที่ 3) ทำการแพร่รายการจุดเสียของแต่ละโหนดจากขั้วอินพุตหลักไปจนถึงขั้วเอาต์พุตหลัก โดยใช้ตัวกระทำยูเนียน (Union) อินเตอร์เซกชัน (Intersection) และการเติมเต็ม (Implement) ของทฤษฎีกลุ่ม (Set Theory) สำหรับคำนวณและแพร่รายการจุดเสีย ดังนั้นเมื่อทำการแพร่รายการจุดเสียเกต AND ดังกล่าวจะได้

$L_A = L_A = \{ A/0 \}$

$L_B = L_B = \{ B/0 \}$

$L_C = L_C = \{ C/1 \}$

$L_Y = (L'_A \cap L'_B \cap L_C) \cup L_Y = \{ C/1 , Y/1 \}$

ขั้นตอนที่ 4) รายการจุดเสียที่ขั้วเอาต์พุตหลักของวงจรจะเป็นรายการจุดเสียที่สามารถตรวจสอบได้ด้วยรูปแบบสัญญาณทดสอบที่ป้อนให้วงจร ดังนั้นรูปแบบสัญญาณทดสอบ "110" สามารถตรวจสอบจุดเสียได้ 2 จุดคือ C/s-a-1 และ Y/s-a-1

ง) การจำลองการทำงานของจุดเสียแบบคอนเคอร์เรนท์

โดย Ulrich และ Baker ร่วมกันพัฒนาขึ้นในปี ค.ศ. 1973 [11] แบบจำลองจุดเสียแบบคอนเคอร์เรนท์นี้มีวิธีการจำลองคล้ายกับแบบดีดักตีฟเป็นอย่างมาก โดยทำการจำลองการทำงานของวงจรที่ดีก่อนเช่นเดียวกัน จากนั้นจำลองการทำงานของจุดเสียทุกกรณีที่เกิดขึ้นในแต่ละโหนดด้วยค่าผลของจุดเสีย หากผลของจุดเสียทำให้ตรรกที่ขั้วเอาต์พุตของแต่ละเกตมีตรรกตรงข้ามกับวงจรที่ดี แสดงว่าจุดเสียนั้นสามารถตรวจสอบได้ ตัวอย่างเช่น เกต AND 3 ขั้วอินพุต A B และ C มี Y เป็นขั้วเอาต์พุต มี "110" เป็นรูปแบบสัญญาณทดสอบ เช่นเดิม ทำการจำลองการทำงานของจุดเสียจะได้

วงจรดี

: ABC = "110" ; Y = 0

วงจรเสีย

จุดเสีย { A/0 } : ABC = "**0**10" ; Y = 0

จุดเสีย { B/0 } : ABC = "1**0**0" ; Y = 0

จุดเสีย { C/1 } : ABC = "11**1**" ; Y = 1

จุดเสีย { Y/1 } : ABC = "110" ; Y = **1**

ดังนั้นจุดเสียที่สามารถตรวจสอบได้ ได้แก่ C/s-a-1 และ Y/s-a-1 เพราะค่าตรรกที่ขั้วเอาต์พุตมีค่าตรงข้ามกับวงจรดี