

จุฬาลงกรณ์มหาวิทยาลัย

ชื่อ _____

คณะวิศวกรรมศาสตร์

เลขประจำตัว _____

ภาควิชาวิศวกรรมคอมพิวเตอร์

หมายเลขเครื่อง _____

2110-263 DIGITAL COMPUTER LOGIC LAB I

วันที่ _____

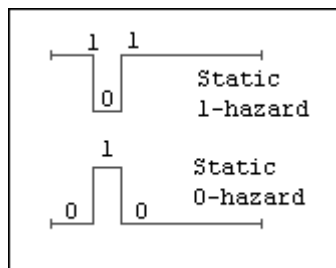
5. การแก้ปัญหา Hazard

วัตถุประสงค์

1. เพื่อให้นิสิตได้ศึกษาปัญหาของ Hazard ที่เกิดขึ้นในวงจรเชิงตรรกะ
2. เพื่อให้นิสิตได้สามารถแก้ปัญหาของ Hazard ที่เกิดขึ้นในวงจรตรรกะโดยใช้วิธีการทางตรรกะ
3. เพื่อให้นิสิตสามารถสร้าง Test vector สำหรับทดลองวงจรตรรกะ

บทนำ

Hazard คือสภาพที่วงจรตรรกะอาจเกิดความผิดพลาดของ output ในช่วงเวลาสั้น (glitch) ซึ่งแบ่งเป็น Static hazard และ Dynamic hazard



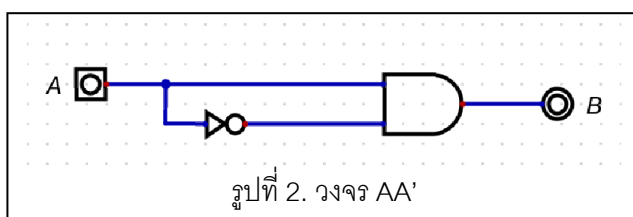
รูปที่ 1 static hazard

ในการทดลองนี้จะสนใจเฉพาะ static hazard เท่านั้น ซึ่งมี 2 ลักษณะ คือ

static 1-hazard คือ เอาต์พุตควรจะเป็น 1 คงที่ แต่เกิด glitch เป็น 0 ชั่วขณะแล้วกลับเป็น 1 ดังในรูปที่ 1

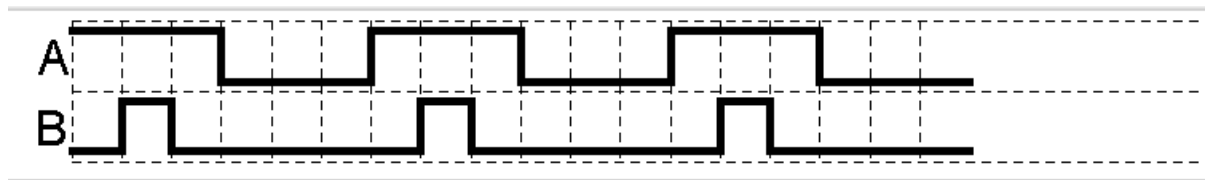
static 0-hazard คือ เอาต์พุตควรจะเป็น 0 คงที่ แต่เกิด glitch เป็น 1 ชั่วขณะแล้วกลับเป็น 0 ดังในรูปที่ 1

Hazard เกิดจากการที่ delay ของสัญญาณในวงจรไม่เท่ากัน ตัวอย่างเช่น จากวงจรในรูปที่ 2 ถ้า A



รูปที่ 2. วงจร AA'

เป็น 0 จะทำให้ output ของ NOT gate เป็น 1 และจะทำให้ output ของ AND gate เป็น 0 และเมื่อ A เปลี่ยนจาก 0 เป็น 1 output ของ NOT gate จะยังคงเป็น 1 อยู่อีกชั่วขณะ ก่อนจึงกลับเป็น 0 (เป็นผลมาจาก gate delay ของ NOT gate) ทำให้ในช่วงเวลานั้น input ของ AND gate เป็น 1 ทั้งคู่ ซึ่งทำให้ output ได้เป็น 1 เมื่อหมดเวลาช่วงนี้แล้ว output จึงกลับเป็น 0 เหตุการณ์นี้เราเรียกว่า เกิด glitch ขึ้น และกล่าวได้ว่าวงจรนี้มี static 0-hazard



รูปที่ 3 waveform แสดงเวลาของวงจร $B=AA'$

เมื่อดูจาก Waveform แล้วจะสังเกตเห็นว่า glitch นั้นไม่ได้เกิดขึ้นทันทีในเวลา que input A เปลี่ยนจาก 0 เป็น 1 สืบเนื่องมาจาก delay ของ AND gate นั้นเอง

พิจารณา สมการบูลีน $B=AA'$ จะเห็นได้ว่า ผลที่ได้ออกมาจะต้องเป็น 0 เสมอ และถ้าไม่ดูจาก waveform เราจะไม่เห็น glitch ที่เกิดขึ้น (ทำนองเดียวกันกับ $A+A'$ ที่ทำให้เกิด static 1-hazard) โดยทั่วไปจะสนใจ hazard ที่เกิดขึ้นจากการเปลี่ยนแปลงของ input ที่มีการเปลี่ยนแปลงทีละ 1 บิต เช่น 00010 เป็น 10010

การแก้ปัญหาของ hazard เริ่มจากการตรวจหา hazard ว่าเป็นแบบใด และเกิดขึ้นตอนไหน โดยเฉพาะการหาว่า “เกิดขึ้นตอนไหน” นี้สำคัญมาก เพราะเราอาจทดสอบวงจรด้วยการป้อน input ทุกค่า แต่อาจไม่พบ glitch เลย ทั้ง ๆ ที่ในวงจรมี hazard อยู่ (glitch เป็นผลของ hazard) ทั้งนี้เพราะ hazard ไม่ได้เกิดขึ้นที่ input ค่าใดค่าหนึ่ง แต่เกิดจากการเปลี่ยนแปลงของ input จากค่าหนึ่งเป็นอีกค่าหนึ่ง เช่น ในตัวอย่าง AA' เราจะได้ glitch เมื่อ A เปลี่ยนจาก 0 เป็น 1 แต่จะไม่เกิดขึ้นเมื่อ A เปลี่ยนจาก 1 เป็น 0

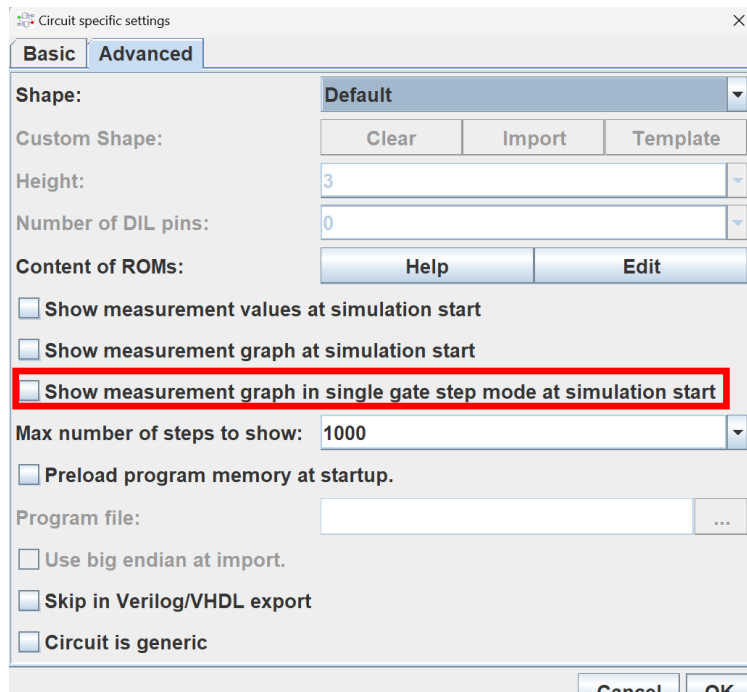
การแก้ไข hazard ทำได้หลายวิธี เช่น พยายามให้สัญญาณมี delay เท่ากันหมด หรือใช้วงจร R-C กรองสัญญาณออกเป็นต้น แต่วิธีที่จะกล่าวต่อไปนี้เป็นวิธีทางตรรกะ โดยจะอาศัย K-Map ช่วยในการวิเคราะห์ แก้ไข hazard โดยจะสนใจเฉพาะวงจร 2 ระดับ (2 level) เท่านั้น

ในการสร้างสมการบูลีน 2 ระดับ ของฟังก์ชันโดยใช้ K-Map จะกำหนดให้แต่ละ cube ที่เลือกแทนด้วยเกท 1 เกท (AND gate ถ้าเป็น sum-of-products หรือ OR gate ถ้าเป็น product-of-sums) เพื่อให้สะดวกต่อการอธิบายจะกล่าวถึงกรณีของ sum-of-products ซึ่งแต่ละ cube จะเป็น AND gate และฟังก์ชันจะเป็นการ OR กันของ AND gate เหล่านี้ ถ้าแต่ละ cube ไม่มี minterm ใดร่วมกันเลย แสดงว่าขณะที่ค่าของฟังก์ชันเป็น 1 AND gate ของ cube หนึ่งจะมีเอาต์พุตเป็น 1 และ AND gate ของ cube ที่เหลือจะมีเอาต์พุตเป็น 0 ในขณะที่อินพุตมีการเปลี่ยนแปลงจาก minterm ใน cube หนึ่งไปเป็น minterm ในอีก cube หนึ่ง ซึ่งทำให้ค่า AND gate อันหนึ่งเปลี่ยนจาก 1 เป็น 0 และทำให้ AND gate อีกอันหนึ่งเปลี่ยนจาก 0 เป็น 1 แล้วจะทำให้เกิด glitch ขึ้นได้ ในขณะที่ ค่าของ AND gate อันแรกเปลี่ยนเป็น 0 ก่อนที่ค่าของ AND gate อันหลังเปลี่ยนเป็น 1 โดยทั่วไปเราจะสนใจการเปลี่ยนแปลงของอินพุตที่มีการเปลี่ยนแปลงเพียง 1 บิต หรือจาก minterm ที่อยู่ข้างเคียงกันใน K-Map

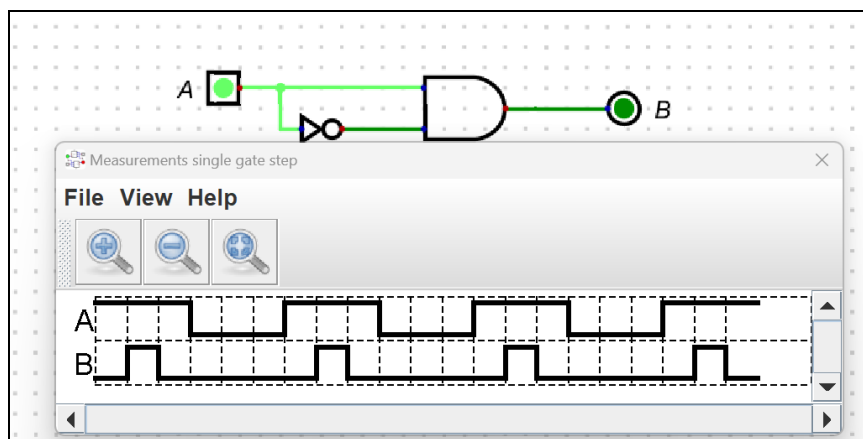
การแก้ไข static 1-hazard ทำโดยเพิ่ม cube ให้คลุม minterm ที่อยู่ติดกันแต่อยู่ต่าง cube กัน

เนื่องจากการเกิด glitch จะเกิดการเปลี่ยนแปลงของ input บางค่า การตรวจสอบโดยใช้ simulator จำเป็นต้องสามารถกำหนดค่า input ที่ต้องการลงไปได้ ในลักษณะของสัญญาณหรือ waveform ซึ่งเรียกว่าเป็นการกำหนด Test vector

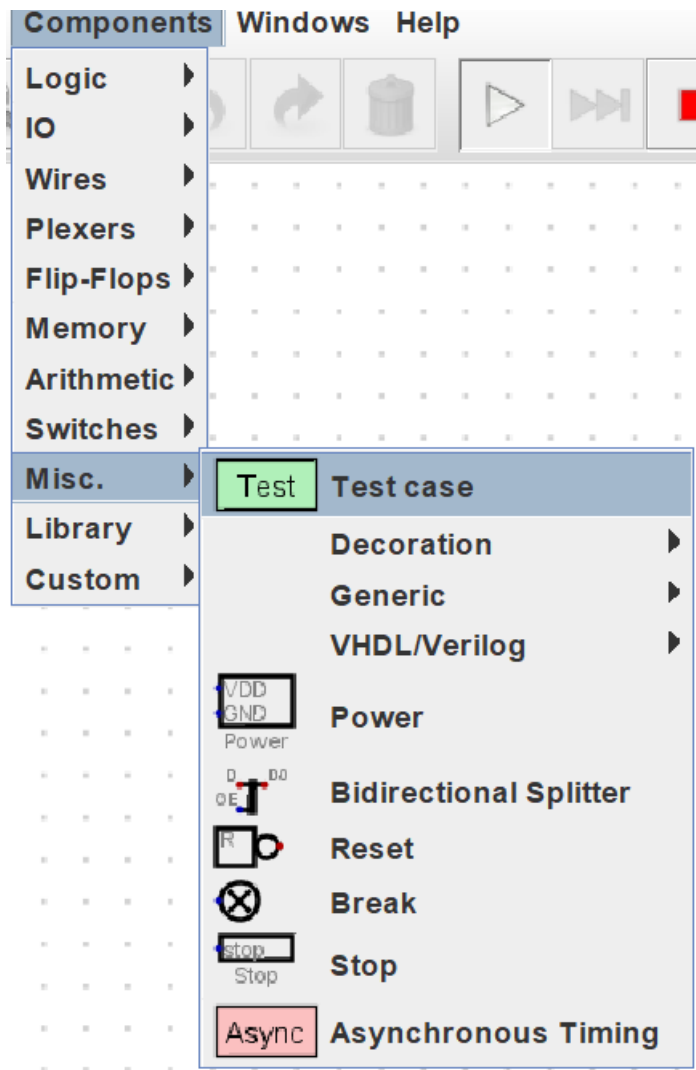
เราสามารถเปิด Timing diagram ให้แสดง glitch โดยคลิกที่เมนู Edit แล้วเลือก circuit specific settings แล้วกดที่แท็บ Advanced แล้วติ๊กที่ Show measurement graph in single gate step mode at simulation start แล้วกด OK



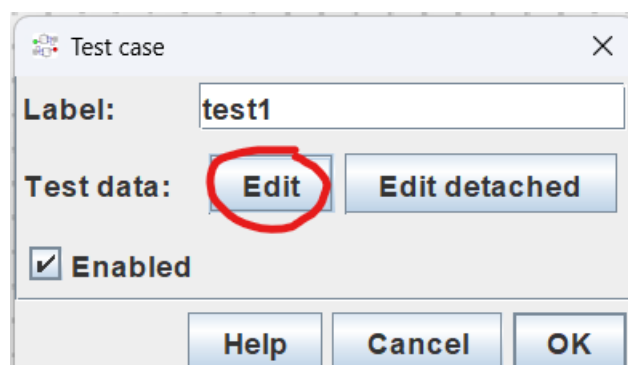
ตัวอย่างการใช้งาน



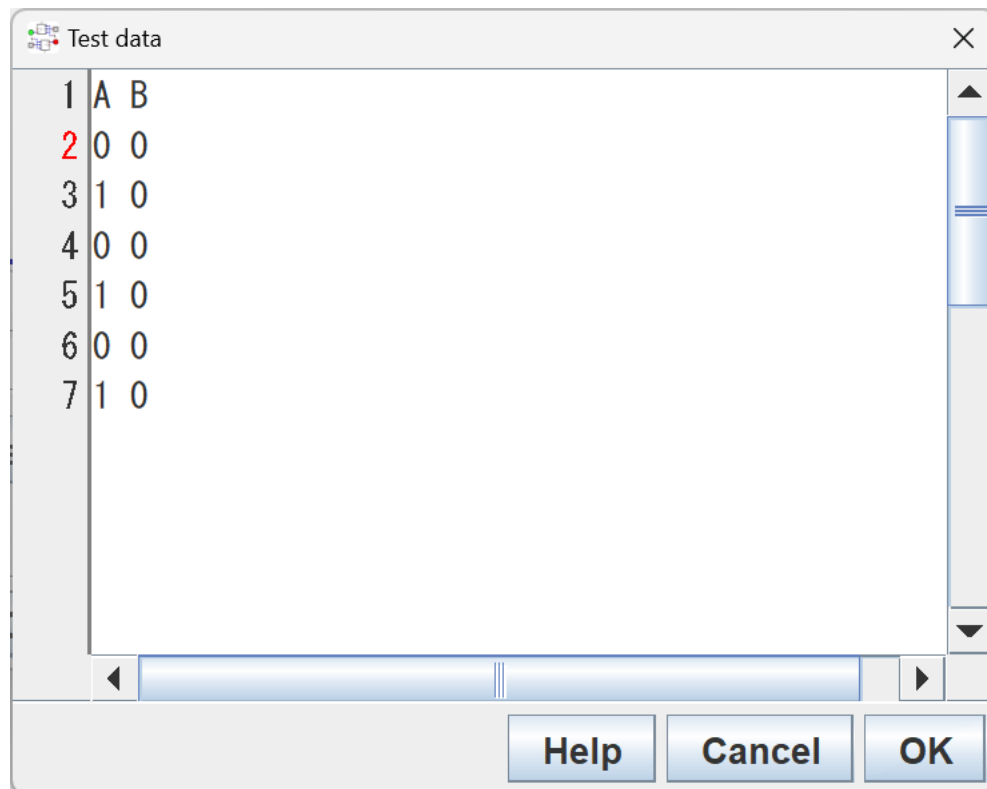
โปรแกรม Digital สามารถสร้าง Test Vector ได้จากโปรแกรมโดยตรง โดยใช้ component ที่ชื่อว่า Test case จาก Misc. -> Test case



เมื่อทำการเลือก test case วางลงใน project ให้ทำการ คลิกขวาที่ test case ที่หัวข้อ Test data กด คำว่า Edit



เมื่อกดแล้วจะมีหน้าต่างเขียนสำหรับเขียน test data จะแสดงขึ้นมา



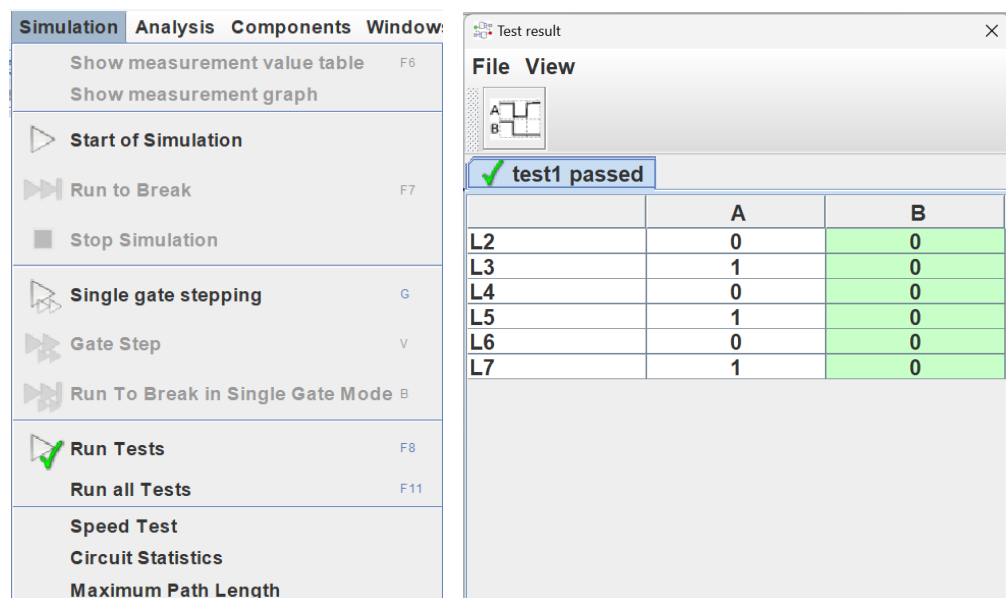
ให้เราใส่ค่าโดย

บรรทัดแรก จะเป็น ชื่อของ input, output

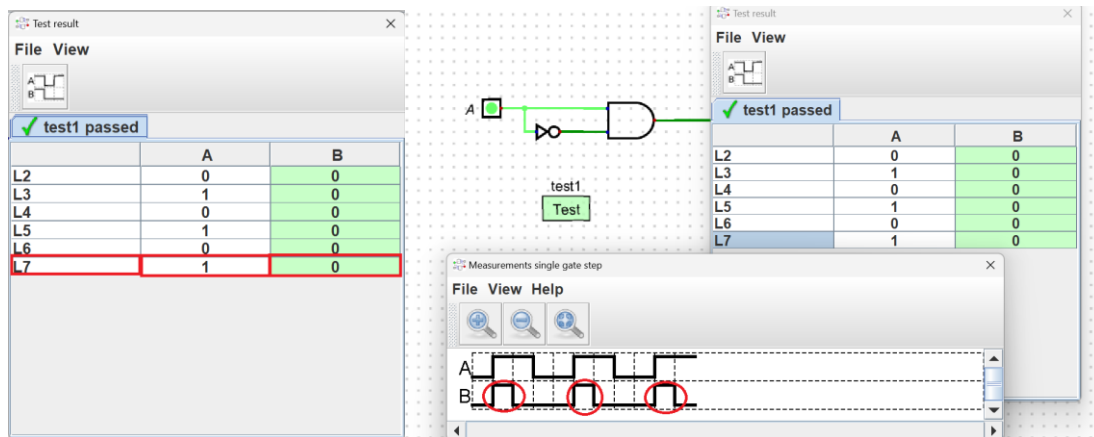
บรรทัดต่อมา จะเป็นการกำหนดค่าของ input, output เพื่อตรวจว่าถ้า Input, output ของวงจรที่เราสร้างตรงกับค่าที่เราตั้งไว้ case นั้นก็จะผ่าน

นอกจาก 0 และ 1 เรายังสามารถใส่ X (not determined) และ Z (high-impedance) ได้อีกด้วย

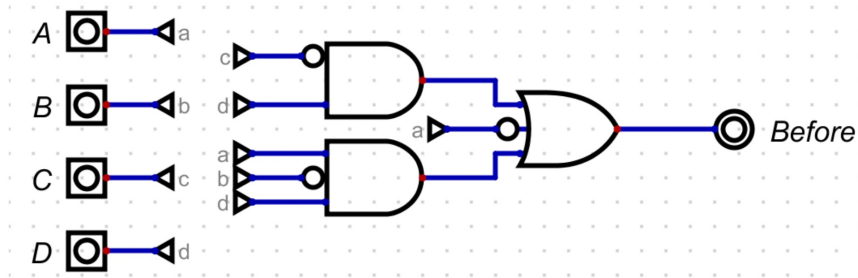
เราสามารถใช้งาน test case ได้จาก simulation -> run Tests



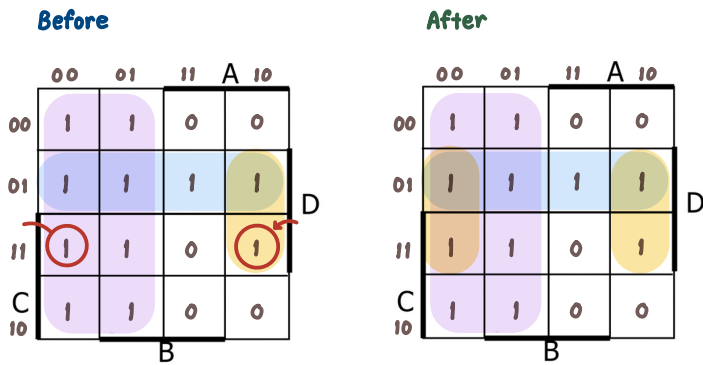
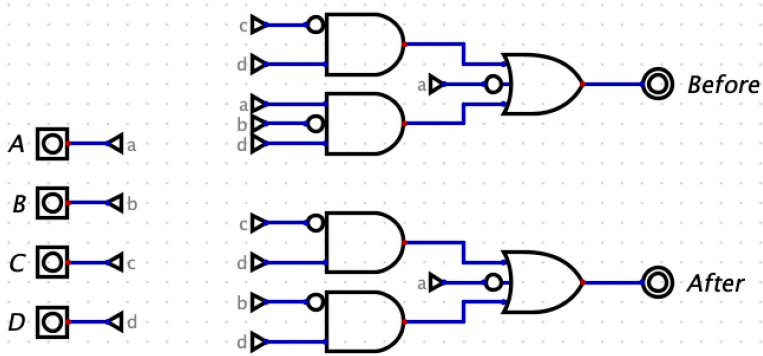
ในการดู glitch ใน test case ให้เรากดดับเบิลคลิกที่ช่องไหนก็ได้ของแถวสุดท้ายแล้วจะได้ Timing diagram ที่แสดง glitch จาก test case



จากสมการบูลีน $F(A,B,C,D) = A' + C'D + AB'D$ จงสร้างวงจรก่อนและหลังกำจัด Hazard แล้วเขียน test vector ที่แสดง Glitch เปรียบเทียบให้เห็นชัดเจนว่าวงจรก่อนกำจัด hazard มี glitch ส่วนวงจรหลังกำจัดไม่มี ให้แสดง K-Map ให้ผู้ตรวจดูด้วย ในวงจรประกอบด้วย Input คือ A,B,C,D ขนาด 1 Bit และ Output คือ Before,After ขนาด 1 Bit โดยที่ Before คือ Output ของวงจรก่อนการแก้ไข Glitch (สร้างวงจรตามสมการบูลีนที่กำหนดให้) และ After คือ Output ของวงจรหลังการแก้ไข Glitch แล้ว



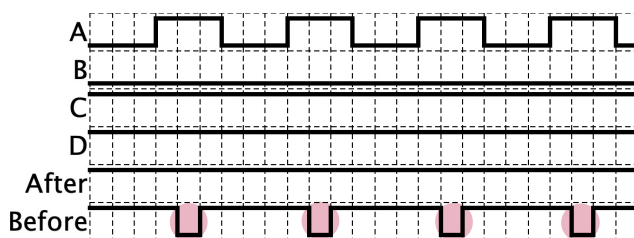
T1
Test



$$A' + C'D + AB'D$$

0011 → 1011 เกิด glitch

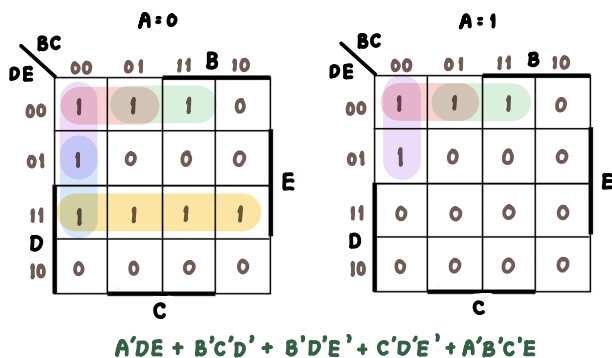
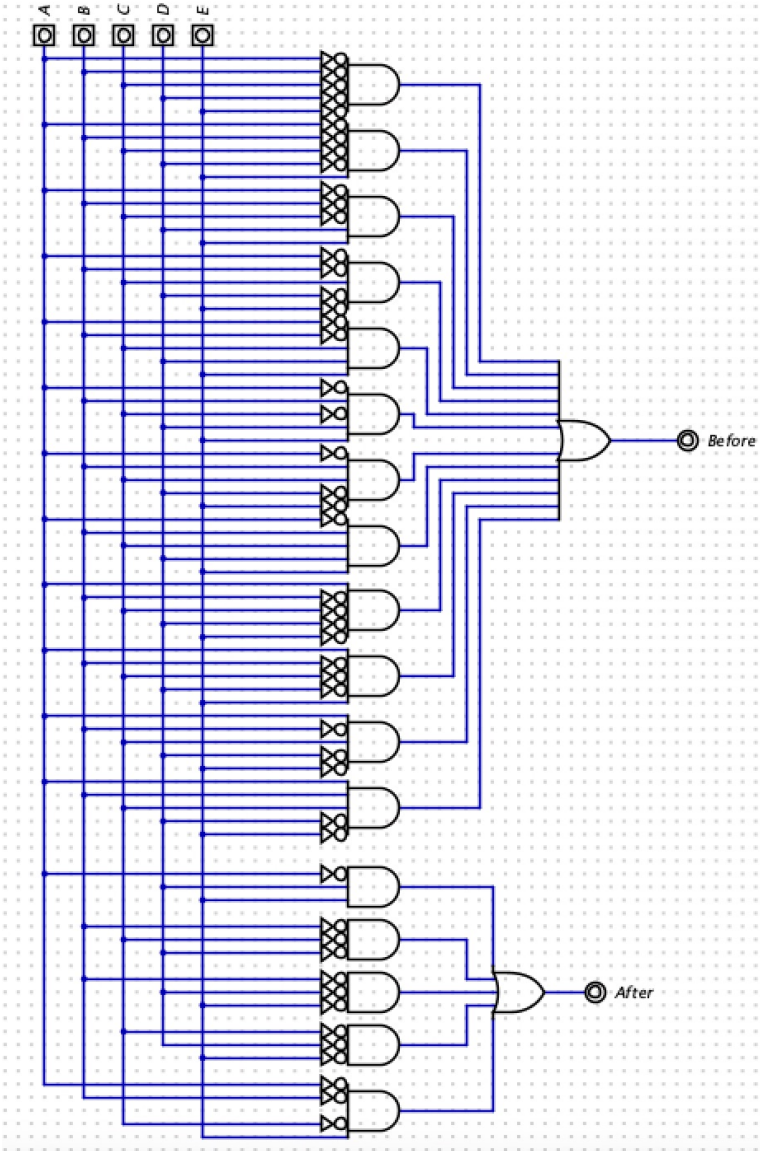
$$A' + C'D + B'D$$



2

Glitch_II

จากสมการบูลีน $F(A,B,C,D,E) = \sum m(0,1,3,4,7,11,12,15,16,17,20,28)$ จงสร้างวงจรก่อนและหลังกำจัด Hazard แล้วเขียน test vector ที่แสดง Glitch เปรียบเทียบให้เห็นชัดเจนว่วงจรก่อนกำจัด hazard มี glitch ส่วนวงจรหลังกำจัดไม่มี ให้แสดง K-Map ให้ผู้ตรวจด้วย ในวงจรประกอบด้วย Input คือ A,B,C,D,E ขนาด 1 Bit และ Output คือ Before, After ขนาด 1 Bit โดยที่ Before คือ Output ของวงจรก่อนการแก้ไข Glitch (สร้างวงจรตามสมการบูลีนที่กำหนดให้) และ After คือ Output ของวงจรหลังการแก้ไข Glitch แล้ว



Note:

