

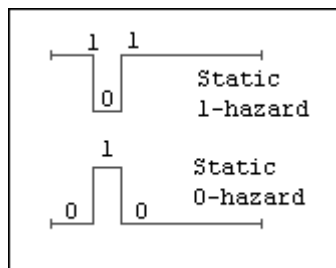
5. การแก้ปัญหา Hazard

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



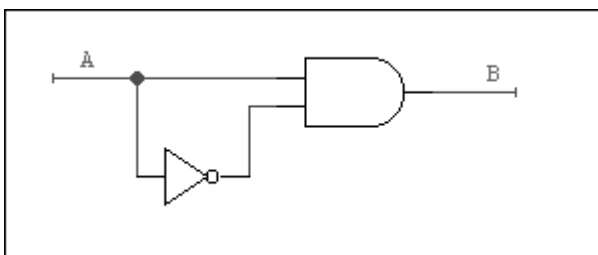
รูปที่ 1 static hazard

ในการทดลองนี้จะสนใจเฉพาะ static hazard เท่านั้น ซึ่งมี 2 ลักษณะ คือ

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

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

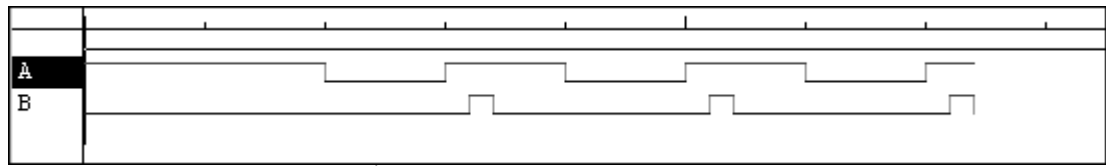
Hazard เกิดจากการที่ delay ของสัญญาณในวงจรไม่เท่ากัน ตัวอย่างเช่น จากวงจรในรูปที่ 2 ถ้า A



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 นั้นไม่ได้เกิดขึ้นทันทีในเวลาที่ 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 ทั้งๆที่ในวงจรมี 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 ช่วยในการวิเคราะห์แก้ไข hazard โดยจะสนใจเฉพาะวงจร 2 ระดับ (2 level) เท่านั้น

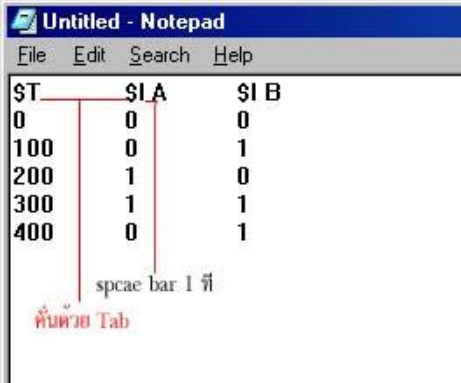
ในการสร้างสมการบูลีน 2 ระดับ ของฟังก์ชันโดยใช้ K-Map จะกำหนดให้แต่ละ cube ที่เลือกแทนด้วยเกท 1 เกท (AND gate ถ้าเป็น sum-of-products หรือ OR gate ถ้าเป็น product-of-sums) เพื่อให้สะดวกต่อการอธิบายจะกล่าวถึงกรณีของ sum-of-products ซึ่งแต่ละ cube จะเป็น AND gate และฟังก์ชันจะเป็นการ OR กันของ AND gate 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

โปรแกรม Logicworks™ ไม่สามารถสร้าง Test Vector ได้จากโปรแกรมโดยตรง แต่สามารถใช้โปรแกรม Text Editor ภายนอกสร้างเป็น Text file แล้วจึง import เข้าไปได้ โดยมีขั้นตอนดังนี้

- ใช้โปรแกรม Text Editor เช่น notepad , Editplus เป็นต้น สร้างไฟล์ที่มี filetype เป็น “.TIM” โดยมีรูปแบบดังตัวอย่างต่อไปนี้



| \$T | \$I A | \$I B |
|-----|-------|-------|
| 0 | 0 | 0 |
| 100 | 0 | 1 |
| 200 | 1 | 0 |
| 300 | 1 | 1 |
| 400 | 0 | 1 |

- บรรทัดแรกเป็น ชื่อของเวลา (\$T) และสัญญาณ input (\$I) เช่น สัญญาณ input A และ input B ก็จะเป็น (\$I A) และ (\$I B) ตามลำดับ โดยที่แต่ละส่วน จะถูกคั่นด้วยการกด Tab 1 ครั้ง จะมีสัญญาณ input ก็อันก็ได้ แต่ชื่อต้องสอดคล้องกับชื่อของ input ในวงจรที่ออกแบบไว้

-บรรทัดต่อมา เป็นค่าที่ต้องการกำหนดตามความต้องการตามเวลา และการเปลี่ยนแปลงค่า input ต่างๆ จากตัวอย่าง ในคอลัมน์แรกเป็นเวลา ในคอลัมน์ต่อไปเป็นค่าของ input A และ B เสร็จแล้วให้ save

-กลับไปโปรแกรม Logicworks ให้หา menu Simulation -> import timing เลือก ไฟล์ test vector ที่สร้างขึ้น แล้วจะได้ตารางเวลาของ input ไปแสดง

- Run simulation แล้วจะได้สัญญาณ output แสดงออกมาตามต้องการ

การทดลอง

ข้อ 1-2 ให้สร้างวงจรมีก่อนและหลังกำจัด hazard ใน circuit sheet เดียวกัน แล้วเขียน test vector ที่แสดง glitch เปรียบเทียบให้เห็นชัดเจนว่าวงจรมีก่อนกำจัด hazard มี glitch ส่วนวงจรมีก่อนกำจัดไม่มี ให้แสดง K-Map ให้ผู้ตรวจดูด้วย

1. $F(A,B,C,D) = A' + C'D + AB'D$

2. $F(A,B,C,D,E) = \sum m(0,1,3,4,7,11,12,15,16,17,20,28)$

(ข้อ 3-4 ไม่เกี่ยวกับ Hazard เป็นการ Preview Lab 6 ต้องส่งในชั่วโมงทุกข้อเช่นกัน)

3. ในระบบเลข 2's complement 8-bit ให้สร้างวงจรมีกอินพุต 8 bit จาก Hex keyboard wo/STB 2 อันและ เอาท์พุทมี 8 บิต ต่อกับ Hex display โดยเอาท์พุทเป็น 2's complement ของอินพุท

(อนุญาตให้ใช้อุปกรณ์ ADDER ได้)

4. สร้างวงจรมีกอินพุทเป็นเลขในระบบ sign and magnitude 4 บิต (ใช้ Hex keyboard wo/STB) แล้ว เปลี่ยนเป็นเลขในระบบ 2's complement 4 บิต (แสดงโดย Hex display)

| sign and magnitude | 2's complement | ค่าของตัวเลข |
|--------------------|----------------|--------------|
| 0001 | 0001 | 1 |
| 1001 | 1111 | -1 |