

จุฬาลงกรณ์มหาวิทยาลัย

ชื่อ _____

คณะวิศวกรรมศาสตร์

เลขประจำตัว _____

ภาควิชาวิศวกรรมคอมพิวเตอร์

หมายเลขเครื่อง _____

2110-263 DIGITAL COMPUTER LOGIC LAB I

วันที่ _____

3. การออกแบบวงจรตรรกะด้วยการลดขนาดนิพจน์บูลีน

วัตถุประสงค์

1. เพื่อให้นิสิตสามารถใช้โปรแกรม Espresso ช่วยในการลดนิพจน์บูลีน
2. เพื่อให้นิสิตออกแบบและจำลองการทดสอบวงจรตรรกะ
3. เพื่อให้นิสิตรู้จักอุปกรณ์ของโปรแกรมจำลองวงจรเพิ่ม

บทนำ

การพยายามลดนิพจน์บูลีนลงก่อน จะทำให้การสร้างวงจรทำได้ง่ายขึ้นและมีขนาดเล็กลง วิธีการลดนิพจน์บูลีนมีหลายวิธี เช่น Karnaugh-Map (K-Map) Quine-McCluskey Method ซึ่งเป็นวิธีที่ใช้กันมานาน ในปัจจุบันมีการใช้คอมพิวเตอร์ช่วยในขั้นตอนนี้ และมีหลายโปรแกรมที่สามารถทำงานนี้ได้ โปรแกรมที่จะใช้ในการปฏิบัติการนี้ คือ Espresso ของ University of California at Berkeley Source code ของ Espresso ได้ถูกนำมา Recompile เพื่อให้ใช้ได้ภายใต้ DOS

การใช้โปรแกรม Espresso เบื้องต้น

โปรแกรม Espresso เป็นโปรแกรมที่ทำงานภายใต้ DOS ที่ทำงานได้ทั้ง Interpreter mode และ Compile mode

- การใช้งานแบบ Interpreter ให้เรียกโปรแกรม Espresso โดยเรียก Command Prompt ขึ้นมาก่อน โดยไปที่ Start -> Programs -> Accessories -> Command Prompt แล้วพิมพ์คำว่า espresso ในโฟลเดอร์ที่มี ไฟล์ espresso.exe อยู่ จากนั้นให้ป้อนคำสั่งต่างๆ และ ข้อมูลลงไป ตาม Berkeley PLA Format เมื่อสิ้นสุดโปรแกรม (โดยคำสั่ง ".e") โปรแกรมก็จะสร้าง ผลของนิพจน์ที่ทำการลดขนาดให้เรียบร้อย การใช้งานแบบนี้จะไม่มีการ save ดังนั้นเมื่อทำผิดก็เริ่มใหม่

ตัวอย่าง $Y = f(A,B,C,D) = \sum m(0,3,5,12,13) + \sum d(1,2,15)$

สามารถสร้าง Berkeley PLA Format ได้ดังนี้

```
.i 4
.o 1
.ilb A B C D
.ob Y
.p 8
0000 1
0011 1
0101 1
1100 1
1101 1
0001 -
0010 -
1111 -
.e
```

- การทำงานแบบ Compiler วิธีนี้จะต้องใช้ Text Editor เช่น Notepad สร้างโปรแกรมและข้อมูล ลงไฟล์ ก่อน ซึ่ง file type ควรเป็น .pla เช่น in.pla เสร็จแล้วใช้คำสั่ง

```
espresso in.pla
```

ถ้าต้องการให้ผลลัพธ์ไปเก็บลงไฟล์ เช่น out.pla ก็ใช้คำสั่ง

```
espresso in.pla > out.pla
```

นอกจากนี้ยังมี Option อื่นๆอีกมากมาย เช่น

- ต้องการให้เอาท์พุทที่ออกมาเป็นสมการบูลีน

```
espresso -o eqntott in.pla
```

- ต้องการได้ เอาท์พุทเป็น Inverse logic

```
espresso -epos in.pla หรือ espresso -epos -o eqntott in.pla
```

เมื่อเอาท์พุทที่ได้มาทำ Inverse อีกครั้ง ก็จะได้ฟังก์ชันออกมาในรูปแบบของ POS (product-of-sums)

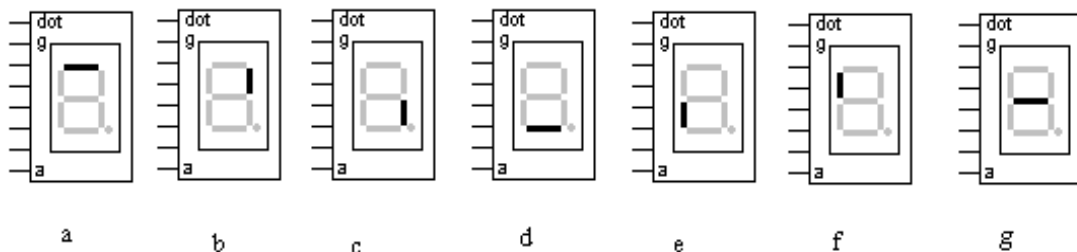
เช่นตามตัวอย่างจะได้ $f(A,B,C,D) = (A \& !B) | (B \& C) | (!A \& B \& !D)$

เพราะฉะนั้น $f(A,B,C,D) = (!A + B) (!B + !C) (A + !B + D)$

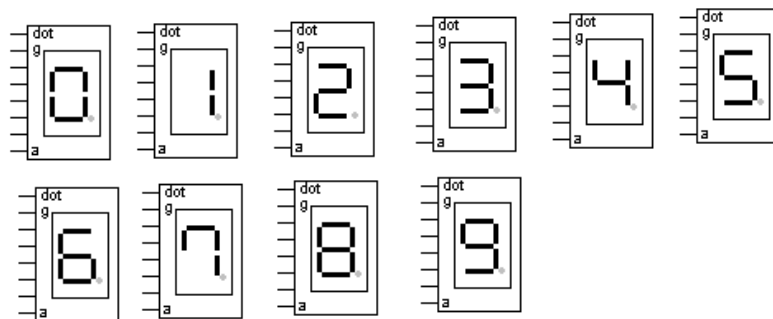
- ต้องการกำหนดให้มีเอาท์พุทมากกว่าหนึ่งค่า ทำได้โดยกำหนดจำนวนเอาท์พุทที่ต้องการที่คำสั่ง “.o “ และตัวแปรที่ต้องการที่คำสั่ง “.ob “ และใส่ค่าของเอาท์พุทเพิ่มตามจำนวนที่กำหนด เช่น จากตัวอย่างข้างต้น ถ้าต้องการ 2 เอาท์พุทก็ใส่เป็น 0000 11 เป็นต้น

อุปกรณ์ใหม่ ที่จะใช้ในการปฏิบัติการคือ 7-segment display , Hex Keyboard wo/STB และ ASCII display

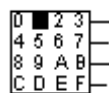
- 7-segment display เป็นอุปกรณ์ที่ใช้แสดงผลที่เห็นได้ทั่วไป ประกอบด้วย ซีด (segment) 7 ซีด และ จุด (dot) 1 จุด ซีดและจุดเหล่านี้ คือ LED (Light Emitting Diode) ซึ่งเมื่อมีกระแสไฟฟ้าวิ่งผ่านจะเปล่งแสงออกมา 7-segment display จะมี input 8 อันสำหรับ ซีดแต่ละซีด และจุด โดยแต่ละซีดจะมีชื่อว่า a b c d e f g ตามลำดับ ตำแหน่งของซีดแสดงในรูปข้างล่าง



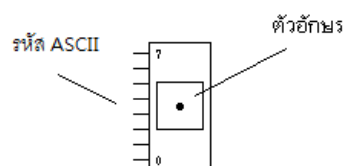
การสว่างและมีดของแต่ละซีดจะทำให้แสดงตัวเลข 0-9 ได้ดังรูป



- Hex Keyboard wo/STB เป็นอุปกรณ์ input ที่ให้ค่า Binary 4 บิต จาก 0000 -1111 โดยจะแสดงค่าที่ถูกกดออกมาที่ขาทั้ง 4 และค่านี้นี้จะคงที่จนกว่าจะถูกกดค่าใหม่



- ASCII display เป็นอุปกรณ์ output ที่จะแสดงตัวอักษร ตามรหัส ASCII 8 bit ที่ได้รับ



การทดลอง

1. จงออกแบบสร้างวงจร BCD to 7-segment Decoder หมายถึงวงจรที่มีอินพุต 4 บิต มีค่า 0-9 ส่วน A – F จะไม่ใช้ (ให้ใช้ Hex Keyboard wo/STB รับค่า) และเอาท์พุท ไปต่อกับขั้วต่างๆของ 7-segment display โดยเมื่อกดแป้นแต่ละแป้นแล้ว ให้เอาท์พุทไปแสดงที่ 7-segment display ตามแป้นที่กด โดยให้ใช้โปรแกรม Espresso ในการลดขนาดนิพจน์บูลีน
2. **ให้นิสิตศึกษารหัส ASCII** แล้วออกแบบและสร้างวงจรที่มีอินพุตเป็น binary switch 3 อัน(ตั้งชื่อ A,B,C โดย A เป็น most significant bit) และเอาท์พุทเป็น ASCII display ถ้าอินพุตเป็น 0-5 จะแสดงนามสกุลของนิสิตเป็นอักษรภาษาอังกฤษ 6 ตัวแรกออกมา เช่น ถ้านามสกุลนิสิตคือ Sridonmeng 6 อักษรแรกจะเป็น Sridon ดังนั้นถ้าอินพุตของวงจรเป็น 0 เอาท์พุทจะแสดง S (อักษรตัวแรกเป็นตัวใหญ่) ถ้าอินพุตของวงจรเป็น 1 เอาท์พุทจะแสดง r เช่นนี้ไปเรื่อยๆ สำหรับอินพุตที่เป็น 6 และ 7 ถือว่าเป็น don't care ใช้ Espresso ช่วยในการออกแบบ ถ้านามสกุลไม่ถึง 6 ตัวอักษร ใช้ abcd... ต่อท้ายจนครบ เช่นถ้านามสกุลเป็น Yong ต้องแสดง Yongab