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 ลงไฟล์ก่อน ซึ่งfile type ควรเป็น .pla เช่น in.pla เสร็จแล้วใช้คำสั่ง

  espresso in.pla

  ถ้าต้องการให้ผลลัพธ์ไปเก็บลงไฟล์ เช่น out.pla ก็ใช้คำสั่ง

  espresso in.pla > out.pla

  นอกจากนี้ยังมี Option อื่นๆอีกมากมาย เช่น

  - ต้องการให้เอาต์พุตที่ออกมาเป็นสมการบูลีน

    espresso –o eqntott in.pla

  - ต้องการได้ เอาต์พุตเป็น Inverse logic

    espresso –epos in.pla หรือ **espresso –epos –o eqntott in.pla**

    เมื่อเอาต์พุตที่ได้มาทำ Inverse อีกครั้ง ก็จะได้ฟังก์ชันออกมาในรูปของPOS(product-of-sums)

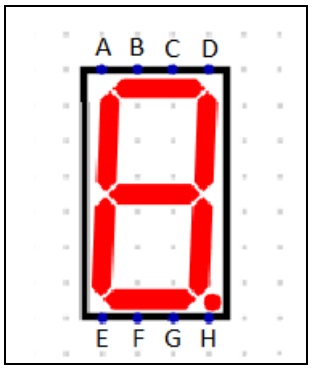
    เช่นตามตัวอย่างจะได้ $\overline{f}$ (A,B,C,D) = (A & !B) | (B & C) | (!A & B & !D)

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

  - ต้องการกำหนดให้มีเอาต์พุตมากกว่าหนึ่งค่า ทำได้โดยกำหนดจำนวนเอาต์พุตที่ต้องการที่คำสั่ง ".o " และตัวแปรที่ต้องการที่คำสั่ง ".ob "และใส่ค่าของเอาต์พุตเพิ่มตามจำนวนที่กำหนด เช่น จากตัวอย่างข้างต้น ถ้าต้องการ 2 เอาต์พุตก็ใส่เป็น 0000 **11** เป็นต้น

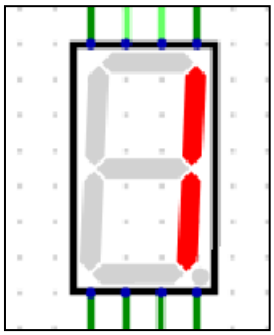
อุปกรณ์ใหม่ ที่จะใช้ในการปฏิบัติการคือ 7-segment display และ Terminal

- 7-segment display เป็นอุปกรณ์ที่ใช้แสดงผลที่เห็นได้ทั่วไป ประกอบด้วย ขีด (segment) 7 ขีด และ จุด (dot) 1 จุด ขีดและจุดเหล่านี้ คือ LED (Light Emitting Diode) ซึ่งเมื่อรับไฟฟ้าหรือสัญญาณ 1 จะเปล่งแสงออกมา 7-segment display จะมี ขา input 8 ขาสำหรับกำกับขีดแต่ละขีดและจุด โดยจะมีทั้งหมด 8 ขา A – H ดังรูป

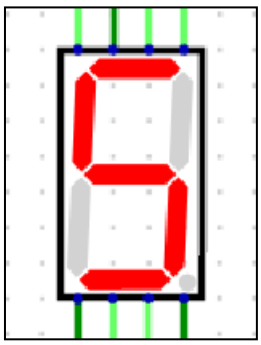


<u>ตัวอย่างการใช้งาน</u>

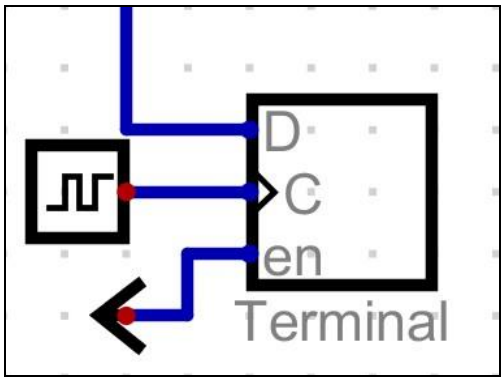
- แสดงเลข 1 โดยใส่กระแสไฟฟ้าในขา B,C



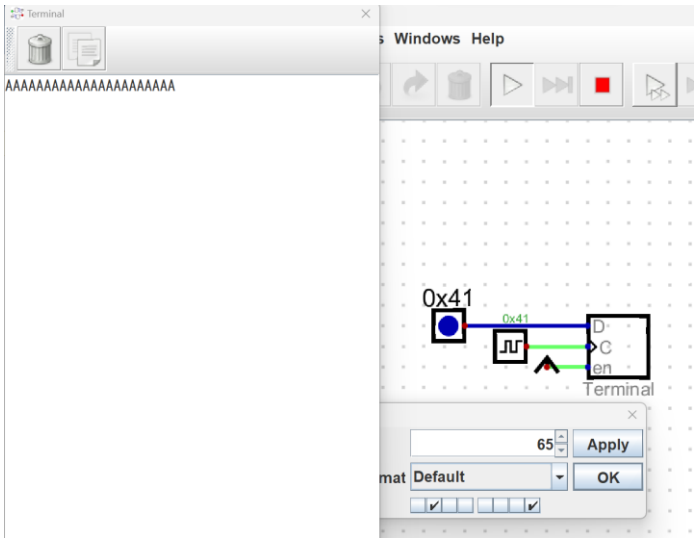
- แสดงเลข 5 โดยใส่กระแสไฟฟ้าในขา A, C, D, F, G



- Terminal เป็นอุปกรณ์ที่ใช้แสดงผลรหัสตัวอักษรตามค่า ASCII ที่รับเข้ามา โดยมีขา input คือ D รับสัญญาณรหัส ASCII 8 Databits ส่วนขา C คือ Clock ให้ต่อกับ Clock และขา en คือ enable ใช้เพื่อเปิด/ปิดอุปกรณ์ โดยอุปกรณ์จะเปิดเมื่อได้ขา en ได้รับสัญญาณ 1



ตัวอย่างการใช้งาน



ก่อนเริ่มการ simulate อย่าลืมเปิด real time clock เมื่อเริ่มการ simulate จะยังไม่มีอะไรเกิดขึ้น จนกว่าเราจะตั้งค่าข้อมูลที่ส่งมาที่ขา D จากตัวอย่างเราให้ input เป็น 65 ซึ่งในรหัส ASCII คือตัวอักษร A แล้วจะมีหน้าต่าง Terminal เปิดขึ้นแสดงตัวอักษร A ซึ่งเป็นรหัส ASCII ที่เราส่งเข้าไป