

# วิธีการดำเนินการสอบครั้งที่ 1 วิชา 2110252 Digital Computer Logic ปีการศึกษา 2566 เทอมต้น

- นิสิตสามารถนำกระดาษ A4 เข้าได้หนึ่งแผ่น เขียนหรือพิมพ์ก็ได้ เป็นเอกสารที่เตรียมด้วยตัวเองหรือสำเนาจากผู้อื่นก็ได้ หรือไม่เขียนอะไรเป็นกระดาษเปล่าๆก็ได้
  - ไม่อนุญาตให้เปิด application ใดๆ นอกจาก digital, espresso, basic text editor, VS Code และ browser หนึ่งตัว
    - โดย browser จะมีได้เพียงแค่ tab ที่เป็นการใช้เกรดเดอร์ของวิชานี้และโจทย์ pdf ของวิชานี้เท่านั้น
    - Basic text editor เช่น notepad สำหรับการสร้างไฟล์ .pla ที่ใช้ใน espresso เท่านั้น
    - VS Code สำหรับการสร้างไฟล์ .pla ที่ใช้ใน espresso เท่านั้น
  - **ไม่สามารถเปิดวงจรเก่าที่เคยเขียนไว้**
  - อนุญาตให้ใช้ espresso เพื่อช่วยลดรูปวงจร
  - อนุญาตให้ใช้ gate หรือ IC หรือเทคนิคอื่นใดที่อยู่ในโปรแกรม digital ได้ (รวมทั้งการสร้างวงจรจาก Truth Table)
  - ไม่อนุญาตให้ปรึกษาหรือดูผู้อื่น หรือค้นจากอินเทอร์เน็ต/ข้อมูลในเครื่อง เพิ่มเติม ให้ค้นหาได้เฉพาะจาก A4 ที่นำเข้ามาเท่านั้น
  - บนโต๊ะที่ใช้ทำสอบจะมีได้แค่โน้ตบุ๊ก, กระดาษ A4 ที่นำเข้ามา, ระเบียบการสอบที่ผู้คุมสอบแจกให้ และอุปกรณ์เครื่องเขียนเท่านั้น ที่เหลือให้วางไว้ใต้โต๊ะ
  - ต้องอัด video บันทึกหน้าจอตลอดการสอบ หากไม่มี video ส่งหลังการสอบ จะถือว่าได้คะแนนเป็น 0
  - หากพบว่าทุจริต ไม่ว่าจะเป็นผู้กระทำผิดหรือมีส่วนร่วมในการกระทำผิด นิสิตจะได้ U ทันที และอาจจะ
    - ให้พ้นสภาพนิสิต
    - ให้ถอนรายวิชาอื่นทั้งหมดที่นิสิตผู้นั้นลงทะเบียนไว้ในภาคการศึกษานั้น
    - อ้างอิงจากหมวด 8 ข้อ 60 ของ [ข้อบังคับจุฬาลงกรณ์มหาวิทยาลัยว่าด้วยการศึกษาชั้นปริญญาตรี](#)
  - หลังสอบเสร็จจะมี assignment ใน Mycourseville ซึ่งมีลิงค์ไปยังฟอร์มที่ใช้ในการส่งไฟล์วิดีโอ
  - สอบวันที่ 25 กันยายน 2566 เวลา 09:30 - 11:00 โดยให้มาถึงห้องสอบเพื่อเตรียมพร้อมในเวลา 09:00 หากเข้าสายเกิน 09:15 จะไม่อนุญาตให้เข้าห้องสอบ โดยห้องสอบและเลขที่นั่งสอบ จะแจ้งให้ทราบอีกครั้ง ทั้งนี้เมื่อสอบเสร็จและส่งไฟล์วิดีโอแล้ว ให้นิสิตปิดคอมพิวเตอร์ และรอในห้องสอบจนกว่าจะหมดเวลาสอบ
-

1. จงสร้างวงจรที่รับข้อมูลจาก Input (**in**) 4 Bit และ แสดงผลลัพธ์ด้วย Output (**out**) 1 Bit ดังตารางต่อไปนี้

in	out
0 0 0 0	0
0 0 0 1	1
0 0 1 0	0
0 0 1 1	1
0 1 0 0	1
0 1 0 1	1
0 1 1 0	0
0 1 1 1	0
1 0 0 0	0
1 0 0 1	1
1 0 1 0	1
1 0 1 1	0
1 1 0 0	0
1 1 0 1	1
1 1 1 0	1
1 1 1 1	0

กรุณาเริ่มจาก template\_01.dig: ใน Template จะมี input ชื่อ in เป็นเลข 4 bit, และ output ชื่อ out เป็นเลข 1 bit. ในตัวอย่างหาก in มีค่าเป็น 1110 => out = 1

#### คะแนน

คะแนนเต็ม 100 คะแนน โดยมีจาก Grader 90 คะแนน และ ถ้าถูกต้องทุก Case ภายใน 1 ชม. จะได้อีก 10 คะแนน

2. กำหนดให้มีสัญญาณ Input 3 บิต 2 ตัว (**in0**, **in1**) และสัญญาณ **addsubtract** 1 บิตโดยให้แสดงผลด้วย Output (**out**) 3 บิต และกำหนดให้ numone เป็น function มีค่าเท่ากับจำนวน bit ที่มีค่าเป็น 1 ใน input, ตัวอย่างเช่น numone(101) = 2, numone(010) = 1, etc.

จงสร้างวงจรที่ทำให้ output **out** มีค่าดังต่อไปนี้

เมื่อ **addsubtract** เป็น 0, **out** มีค่าเท่ากับ numone (**in0**) + numone (**in1**)

เมื่อ **addsubtract** เป็น 1, **out** มีค่าเท่ากับ numone(**in0**) – numone(**in1**)

**\*\*ในกรณีที่ **addsubtract** = 1, numone (**in0**) มากกว่าหรือเท่ากับ numone (**in1**) เสมอ\*\***

ตัวอย่างเช่น

**addsubtract** = 0, **in0** = 110, **in1** = 001 => numone(110)+numone(001) = 2 + 1 = 3 => **out** = 011

**addsubtract** = 1, **in0** = 011, **in1** = 010 => numone(011)-numone(010) = 2 - 1 = 1 => **out** = 001

**addsubtract** = 0, **in0** = 111, **in1** = 001 => numone(111)+numone(001) = 3 + 1 => **out** = 100

กรุณาเริ่มจาก template\_02.dig: ใน Template จะมี input ชื่อ **in0** กับ **in1** เป็นเลข 3 bit, **addsubtract** เป็นเลข 1 bit และ output ชื่อ **out** เป็นเลข 1 bit

#### คะแนน

คะแนนเต็ม 100 คะแนน โดยมีจาก Grader 90 คะแนน และ ถ้าถูกต้องทุก Case ภายใน 1 ชม. จะได้ไปอีก 10 คะแนน