

จุฬาลงกรณ์มหาวิทยาลัย

คณะวิศวกรรมศาสตร์

ภาควิชาวิศวกรรมคอมพิวเตอร์

2110-263 DIGITAL COMPUTER LOGIC LAB I

ชื่อ นรวิทย์ นามวงศ์

เลขประจำตัว 643208221

หมายเลขเครื่อง

วันที่ 18 / 2 / 2565

1. การใช้โปรแกรมจำลองการทำงานของวงจรตรรกะ (Simulator)

วัตถุประสงค์

1. เพื่อให้นิสิตสามารถใช้โปรแกรมเพื่อสร้างวงจรตรรกะ
2. เพื่อให้นิสิตสามารถใช้โปรแกรมเพื่อจำลองการทำงานของวงจรตรรกะ

บทนำ

การทดลองและศึกษาการออกแบบและการทำงานของวงจรตรรกะอาจทำได้โดยสร้างวงจรขึ้นมาโดยใช้อุปกรณ์ (IC) แล้วใช้เครื่องมือเช่น Oscilloscope หรือ logic probe วัดผล

อีกวิธีหนึ่งที่ได้คือใช้โปรแกรมจำลองการทำงาน หรือ Simulator ซึ่ง เป็น โปรแกรมที่สามารถแสดงผลการทำงานของวงจรจาก “รายละเอียดวงจร” (Design description) การกำหนดลักษณะของวงจรทำได้หลายแบบ เช่น ใช้การวาดสัญลักษณ์ของ logic gate (schematic capture) หรือ เขียนเป็น text ที่แสดงการเชื่อมต่อของแต่ละองค์ประกอบหรือ การทำงานของวงจร (hardware description languages) จากนั้น simulator จะแสดงผลที่ได้จากการทำงาน หรือ output ของวงจรที่เกิดจาก input ที่ผู้ใช้กำหนด

Simulator มีอยู่หลายประเภทแต่ที่เราจะทดลองใช้ในวิชานี้จะมีสองรูปแบบคือ

- Logic simulator จำลองการออกแบบเป็นการเชื่อมต่อของ logic gate ที่ให้ค่าออกมาเป็น 0 และ 1 (และค่าอื่นๆที่จะพบในตอนหลัง) นิสิตจะใช้โปรแกรมเพื่อตัดสินว่า ตารางความจริง (Truth table) ของวงจรที่สร้างขึ้นตรงกับ input/output ที่ต้องการหรือไม่ สำหรับวงจรง่ายๆ เราสามารถทดลองได้ทุกกรณีของ input แต่ถ้าเป็นวงจรที่ซับซ้อนเราอาจทดสอบได้เฉพาะบางกรณีเท่านั้น
- Timing simulator จะคล้ายกับ Logic simulator แต่จะจำลองการทำงานของ logic gate ให้เหมือนการทำงานจริงยิ่งขึ้น โดยจะยอมให้แต่ละ gate มี delay (ช่วงเวลาที่เกิดขึ้นระหว่างการเปลี่ยนแปลงของ output หลังจากมีการเปลี่ยนแปลงของ input) หรือ อีกนัยหนึ่งโปรแกรมไม่เพียงแต่บอกว่า output จะเป็นอย่างไรจาก input แต่จะบอกด้วย

ว่า output ที่ได้จะได้เมื่อไร โดยทั่วไปผลของ timing simulator จะเป็นในรูปของ waveform (แผนภูมิแสดงค่าของ logic 0 และ 1 เทียบกับเวลา)

การจำลองการทำงานนั้นมีความสามารถได้ถึงระดับหนึ่ง ไม่สามารถจะทำงานได้ “เหมือนจริง” ในทุกๆ ด้าน ผู้ใช้จำเป็นต้องคำนึงถึงขีดจำกัดของโปรแกรมจำลองการทำงานคือ

Logic simulator จะใช้ “แบบจำลอง” (Model) ของวงจร ดังนั้นถึงแม้ว่าวงจรจะทำงานได้อย่างถูกต้องใน simulator แต่เมื่อ “ต่อ” วงจรจริงอาจจะทำงานไม่ถูกต้องก็ได้ เช่น ในวงจรจริงอาจมีปัญหาเกี่ยวกับกระแสไฟฟ้าที่ใช้ไม่เพียงพอ หรือการรบกวนของสัญญาณเกิดขึ้น

ผลที่ได้จากการจำลองการทำงานขึ้นอยู่กับ input ที่ผู้ใช้ทดลองใช้ ในการทำงานจริงอาจจะมี input ที่อยู่นอกเหนือสิ่งที่ทดลองก็ได้ซึ่งวงจรอาจทำงานผิดพลาดสำหรับ input ที่ผู้ใช้ไม่ได้ทดสอบนี้

ทฤษฎี

โปรแกรมที่สามารถให้ผู้ใช้สร้างวงจรและจำลองการทำงานของวงจรตรรกะที่จะใช้ในวิชานี้ นั้นมีชื่อว่า LogicWork™ โดยบริษัท Capilano Computing System, Ltd. (<http://www.capilano.com>) เป็นรุ่นที่มีลิขสิทธิ์ และใช้เพื่อการศึกษาเท่านั้น จึงไม่อนุญาตให้คัดลอกออกไปใช้

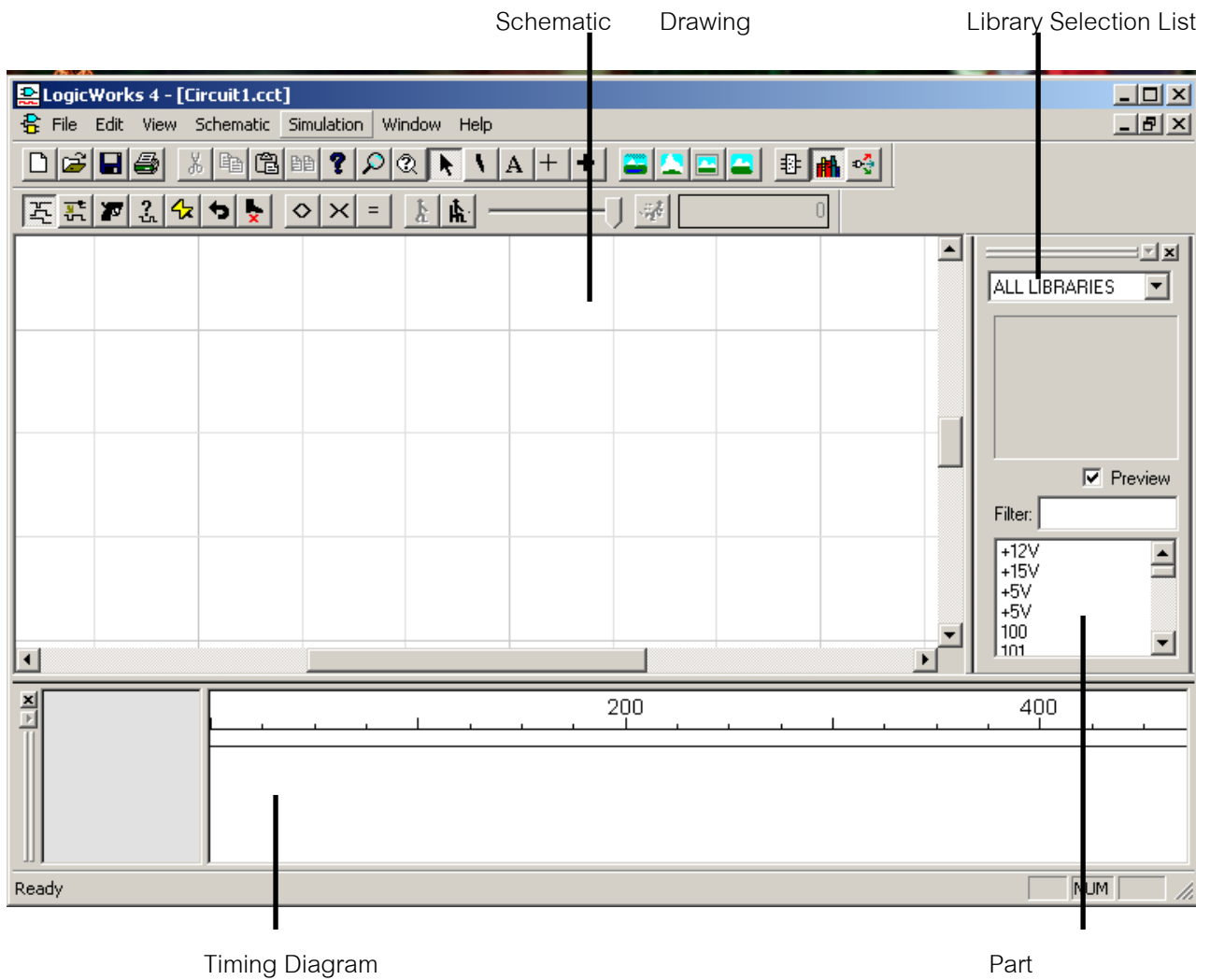
กรณีที่น่าสนใจอยากทดลองใช้โปรแกรมจำลองการทำงานของวงจรตรรกะนี้ภายนอกห้องปฏิบัติการก็สามารถใช้โปรแกรม ชื่อว่า Designworks™ สามารถ download ได้โดยไม่คิดมูลค่าจาก web page ของบริษัท

เรียกใช้โปรแกรมนี้ได้โดย double click ที่ icon LogicWork 4 บนหน้าต่าง Desktop หรือเรียกใช้ผ่าน

START → Programs → LogicWorks 4 → LogicWorks 4

เมื่อเข้าสู่โปรแกรมโดยปกติจะเห็นหน้าจอประกอบด้วยส่วนต่างๆ ดังนี้

1. Schematic Drawing Area บริเวณสำหรับวาดวงจร
2. Library Selection List และ Part Selection List สำหรับเลือกนำอุปกรณ์ต่างๆ มาใช้ในวงจร
3. Timing Diagram แสดงผังเวลาของสัญญาณต่างๆ ที่ต้องการตรวจสอบ



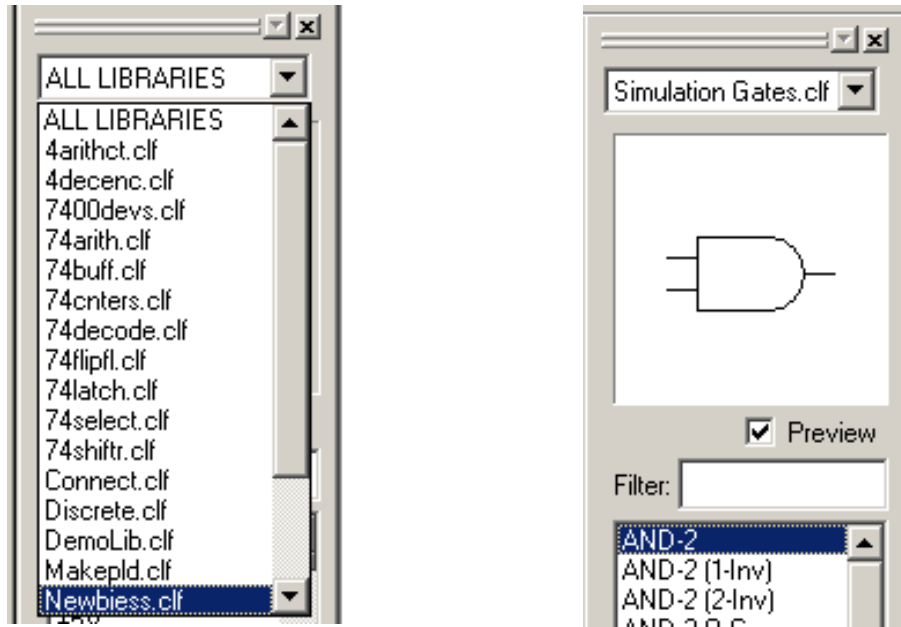
Toolbar มีไว้สำหรับเลือกเครื่องมือต่างๆ โดย ในแถวแรกเป็นเครื่องมือเกี่ยวกับการจัดการไฟล์ และเครื่องมือช่วยในการวาดวงจร ส่วนแถวที่สองเป็นเครื่องมือเกี่ยวกับการทำ Simulator

เริ่มต้นสร้างวงจร

ให้เลือกจากเมนู File → New และเลือก New Design จะปรากฏบริเวณวาดวงจรใหม่ขึ้น (ปกติเมื่อเปิดโปรแกรมใหม่ โปรแกรมจะแสดง New Design ให้เองอยู่แล้ว)

การเลือกและวางอุปกรณ์


หน้าต่าง Part มีไว้สำหรับให้ผู้ใช้เลือกอุปกรณ์ต่างๆ ซึ่งจัดไว้เป็นกลุ่มๆ เรียกว่า Library



หลังจากเลือก Library ที่ต้องการแล้ว สามารถดูรายชื่อของอุปกรณ์ทั้งหมดที่อยู่ใน Library นั้นๆ หากรู้ชื่อของอุปกรณ์ที่จะใช้แล้วก็สามารถพิมพ์ชื่อบางส่วนลงในช่อง Filter ได้ เฉพาะอุปกรณ์ที่มีชื่อมีส่วนเหมือนกับข้อความใน Filter จะปรากฏใน List วิธีนี้สามารถทำให้หาอุปกรณ์ที่ต้องการได้รวดเร็วในกรณีที่ Library มีอุปกรณ์อยู่เป็นจำนวนมาก

สำหรับการทดลองใช้อุปกรณ์ที่ใช้ส่วนใหญ่จะอยู่ใน Library ต่างๆ ดังนี้

- Simulation Gates.clf ซึ่งเป็นกลุ่มของ gate เบื้องต้น
- Simulation IO.clf ซึ่งเป็นกลุ่มของอุปกรณ์ input/output เบื้องต้น
- Simulation Logic.clf ซึ่งเป็นกลุ่มของอุปกรณ์ logic เบื้องต้น


เมื่อผู้ใช้เลือกกลุ่มที่ต้องการ ชื่ออุปกรณ์ในกรอบรายการของ Parts จะเปลี่ยนไป ถ้าต้องการใช้อุปกรณ์ใด ให้ double click ที่ชื่ออุปกรณ์นั้น เมื่อเลื่อน cursor ของ mouse ไปที่ “แผ่นสำหรับออกแบบ” cursor จะกลายเป็นรูปอุปกรณ์นั้น เมื่อต้องการวางอุปกรณ์นี้ไว้ที่ตำแหน่งใด ให้ click ที่บริเวณนั้น cursor ของ mouse จะยังคงเป็นรูปอุปกรณ์นั้นอยู่ ถ้าผู้ใช้ต้องการอุปกรณ์ชนิดเดียวกันอีก จะสามารถทำได้โดย click ที่บริเวณใหม่ต่อไป แต่ถ้าผู้ใช้ต้องการให้ cursor กลับเป็นรูปลูกศรให้ click ที่ปุ่ม  ในหน้าต่าง Palette หรือกด Space Bar

การลากเส้นเชื่อมต่อระหว่างอุปกรณ์

การลากเส้น (คือ wire หรือ สายไป) จะต้องให้ cursor เป็นรูปลูกศรก่อน เส้นที่สร้างจะต้องเริ่มจาก “ขาของอุปกรณ์” (Pin) หรือ ต่อจากเส้นเดิมที่มีอยู่แล้ว ให้นำ cursor ของ mouse มาชี้ที่ขาอุปกรณ์ แล้วกดปุ่มซ้ายของ mouse ค้างไว้พร้อมกับเลื่อน mouse ไปตามทิศทางที่ต้องการจะมีเส้นเกิดขึ้น เมื่อได้เส้นต่ออุปกรณ์ทั้งสองแล้วให้ยกนิ้วออกจากปุ่มเมาส์ การสร้างเส้นนั้นจะทำได้ในแนวนอนหรือแนวตั้ง และจะ “หัก” เส้น (เปลี่ยนทิศทางเป็นมุมฉาก) ได้หนึ่งครั้ง และถ้าต้องการหักเส้นหลายหนก็ต้องทำทีละตอนก่อนเชื่อมโยงกับขาอุปกรณ์

การลบอุปกรณ์หรือสายสัญญาณ

ทำได้ 2 วิธีคือ

- เลือกสิ่งที่ต้องการจะลบ (จะเป็นอุปกรณ์หรือสายสัญญาณ) โดยการ click ที่สิ่งนั้นแล้วกดแป้น delete ที่แป้นพิมพ์ โดยปกติการเลือกทำได้โดยการ click แต่อุปกรณ์บางชนิดเช่น switch เมื่อ click จะเป็นการใช้อุปกรณ์ แทนที่จะเป็นการเลือก สำหรับอุปกรณ์ประเภทนี้การเลือกทำได้โดย drag mouse เป็นบริเวณกว้างคลุมอุปกรณ์ที่ต้องการ
- ใช้ปุ่ม Zap  ในหน้าต่าง Palette จะทำให้ cursor เปลี่ยนรูปเป็นตามที่แสดงที่ปุ่ม นำปลายลางไป click ที่อุปกรณ์หรือสายสัญญาณใด ส่วนนั้นก็จะหายไป


การใช้วิธีทั้ง 2 นี้ สำหรับอุปกรณ์จะมีผลเหมือนกัน แต่ถ้าเป็นเส้นจะได้ผลต่างกัน คือ ในวิธีแรก เส้นทุกเส้นที่เชื่อมต่อกันจะถูกเลือกพร้อมกัน และเมื่อกดปุ่ม delete จะถูกลบหมดทุกเส้น แต่ในวิธีที่สอง เฉพาะเส้น (หรือส่วนของเส้น) ที่ถูก zap เท่านั้นที่จะถูกลบ

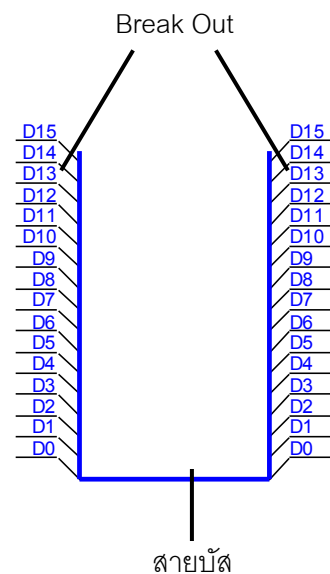
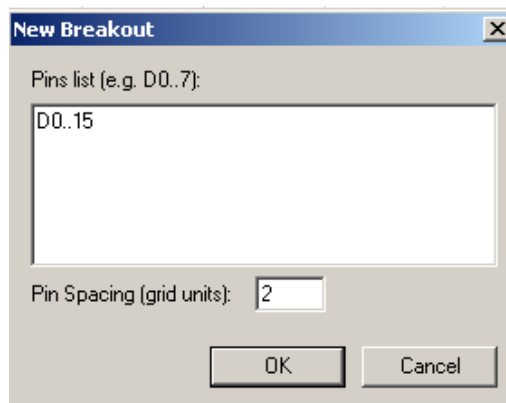
การตั้งชื่อขาสัญญาณและเชื่อมโยงขาสัญญาณโดยชื่อ

ขาต่างๆ ของอุปกรณ์หรือสายสัญญาณ สามารถตั้งชื่อให้ได้ โดยนำเมาส์ไปคลิกที่ขาสัญญาณที่ต้องการด้วยปุ่มขวา จะมีเมนูขึ้นมา ให้เลือกที่ Name... แล้วใส่ชื่อหรือชื่อย่อที่รู้จัก โดยห้ามตั้งชื่อเป็น 0 หรือ 1 และไปติ๊กให้เป็น visible ด้วย

การตั้งชื่อจะให้ประโยชน์ ในกรณีที่อุปกรณ์ที่จะเชื่อมต่อกันนั้นอยู่ห่างกันมาก จะโยงสายอาจไม่สวยงามหรือมีการเชื่อมโยงสัญญาณนั้นมาก เช่น สัญญาณ Clock และการตั้งชื่อนี้จะสามารถดูข้อมูลสัญญาณใน Timing Diagram ได้อีกด้วย


การใช้บัส

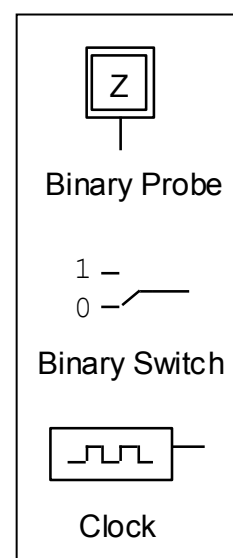
สายสัญญาณที่มีคุณสมบัติเหมือนกัน และมีจำนวนมากหรือเป็นกลุ่ม เราเรียกว่า บัส (Bus) เช่น บัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) เป็นต้น การลากสายสัญญาณเหล่านั้นมักมีอุปสรรคไม่ใช่น้อย และเสียเวลามาก จึงมีการใช้สายสัญญาณบัสแทน ซึ่งทำได้โดยสร้าง Break Out เพื่อแยกสายสัญญาณที่ต้องการต่อก่อน ซึ่งสร้างได้จาก เมนู Schematic แล้วเลือก New Breakout.. หรือ กด Ctrl + B จะได้หน้าต่างดังรูป แล้วจึงกำหนดสายสัญญาณที่ต้องการสร้าง จากนั้นจึงนำสายสัญญาณนั้นๆ ไปต่อกับอุปกรณ์ที่ต้องการ แล้วจึงเชื่อมระหว่าง Break Out ด้วยสายบัส ซึ่งสร้างได้จาก ปุ่ม  เช่นเดียวกับการลากสายสัญญาณปกติ




การจำลองการทำงานของวงจรตรรกะ (Simulation)

สามารถทำ Logic Simulation ได้ในหน้าต่างที่วาดวงจรมานั้นเลย และสามารถดู Timing Simulation ได้ในหน้าต่าง Timing Diagram ซึ่งเป็นลักษณะเด่นของซอฟต์แวร์นี้ โดยอุปกรณ์พื้นฐานที่ใช้ในการจำลองการทำงานของวงจรตรรกะได้แก่ Binary Switch, Binary Probe และ Clock ซึ่งอยู่ใน Library Simulation IO.clf

Binary switch จะทำหน้าที่เป็น input ที่ให้ค่าเป็น 0 หรือ 1 โดยใช้ cursor เป็นลูกศร (โดย click ที่ปุ่ม  ในหน้าต่าง Palette) แล้ว click ที่ Binary switch จะเห็นว่าสามารถเปลี่ยนค่าได้ระหว่าง 0 และ 1












Binary probe จะเป็นอุปกรณ์ที่แสดงค่าของ logic โดยจะแสดงค่าเป็น 0, 1, C, X หรือ Z ตามค่าที่ได้ (C หมายความว่าไม่มีค่าผิดพลาด X หมายความว่ายังไม่ค่า Z หมายความว่า binary probe ไม่ได้ถูกต่อกับอุปกรณ์อื่น หรือเรียกว่าต่อกับค่า High Impedance) นอกจากนี้เราสามารถ ใช้ signal probe วัดค่าของสายสัญญาณใดๆได้อีกด้วยโดยกดปุ่ม  ในหน้าต่าง Paletteจะทำให้ cursor เปลี่ยนเป็นรูปเหมือนกับปุ่ม จากนั้นจึงนำปลายของ cursor ไป click สายสัญญาณที่ต้องการ cursor ก็แสดงค่าของสายเส้นนั้นออกมา

Clock เป็นอุปกรณ์ที่ทำหน้าที่เป็น input ที่ให้ค่า 0 และ 1 อย่างต่อเนื่อง ซึ่งสามารถกำหนดพารามิเตอร์ได้โดยใช้ cursor ลูกศรคลิกที่ Clock อันที่ต้องการ แล้วไปที่ เมนู Simulation เลือก Parameters.. หรือจะใช้ Ctrl + K ก็ได้ ค่า Default ที่กำหนดไว้มีค่า Low = 10 และ High = 10 (Low หมายถึง จำนวนหน่วยเวลาที่เป็นศูนย์ และ High หมายถึง จำนวนหน่วยเวลาที่เป็นหนึ่ง) Clock แต่ละอันจะมีพารามิเตอร์แยกเป็นอิสระจากกัน

ความเร็วในการจำลองการทำงานนี้สามารถควบคุมได้จาก Simulation Toolbar โดยเลื่อนเข็มกำหนดความเร็วมาทางซ้ายเพื่อให้ความเร็วในการทำ Simulation ช้า และเลื่อนเข็มมาทางขวาเพื่อให้เร็วขึ้น นอกจากนี้ยังสามารถบอกให้ Simulator ทำงานทีละ Step, Reset เพื่อเริ่มต้น Simulation ใหม่ และยัง Zoom In, Zoom Out ได้อีกด้วย หากไม่แน่ใจว่าแต่ละปุ่มใน Toolbar ทำหน้าที่อะไร ให้เลื่อนเมาส์ไปแช่อยู่ที่ปุ่ม สักพักหนึ่งจะมี Tool Tip ปรากฏขึ้นเพื่ออธิบายหน้าที่ของปุ่มนั้นๆ



-  ปุ่ม แสดง/ซ่อน Timing Diagram
-  ปุ่ม เพิ่มสัญญาณลงใน Timing Diagram
-  ปุ่ม กำหนด ค่าพารามิเตอร์ของอุปกรณ์
-  ปุ่ม reset Timing Diagram
-  ปุ่ม Zoom in เพื่อขยายขนาดภาพใน Timing Diagram
-  ปุ่ม Zoom out เพื่อย่อขนาดภาพใน Timing Diagram
-  ปุ่ม Normal size เพื่อปรับขนาดภาพ ใน Timing Diagram ให้เป็น ขนาดปกติ
-  ส่วนควบคุมการ run, stop และความเร็วในการ simulate
-  หน่วยเวลาที่ทำกร simulate

Timing diagram เป็นการแสดงค่าของสัญญาณจากที่ใดที่หนึ่งในวงจรเทียบกับเวลา โดยจะแสดงในหน้าต่างอีกหน้าต่างหนึ่ง สัญญาณที่จะแสดงได้นั้นจะต้องมาจากเส้น ให้ click ที่เมนู View แล้ว click ที่ Timing Window จะได้หน้าต่าง Timing เกิดขึ้น (โดยปกติจะเปิดอยู่แล้ว) ขอให้สังเกตว่าจะไม่มีสัญญาณใดๆ อยู่ในหน้าต่างเลย ถ้าต้องการจะดู Waveform ของสัญญาณใด ต้องเลือกให้สัญญาณนั้นลงในหน้าต่าง Timing ซึ่งมีขั้นตอนดังนี้

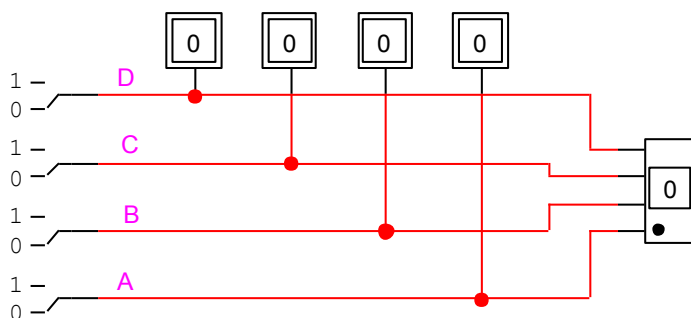
- Click ที่เส้นที่มีสัญญาณที่ต้องการ
- Click ที่เมนู Simulation แล้ว click ที่คำสั่ง add to timing หรือกดปุ่ม Ctrl + T ที่แป้นพิมพ์

ในหน้าต่าง Timing จะปรากฏสัญญาณนั้น

ข้อแนะนำ ให้ทำการตั้งชื่อสัญญาณต่างๆ ให้มีความหมายหรือตรงกับที่ออกแบบไว้ เมื่อดู Waveform แล้วจะได้เข้าใจได้ดี

การทดลอง

1. สร้างวงจรโดยใช้อุปกรณ์ Binary Switch, Binary Prob, Hex Display จาก Simulation IO.clf Library แล้วเชื่อมสายสัญญาณเข้าด้วยกัน และตั้งชื่อสัญญาณ ดังรูป



แล้วบันทึกผลการทดลองให้ครบตามตัวอย่างตาราง

D	C	B	A	ผลที่แสดง
0	0	0	1	
0	0	1	0	
0	0	1	1	
.	.	.	.	
.	.	.	.	
1	1	1	0	
1	1	1	1	

D	C	B	A	ผลที่แสดง
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

0

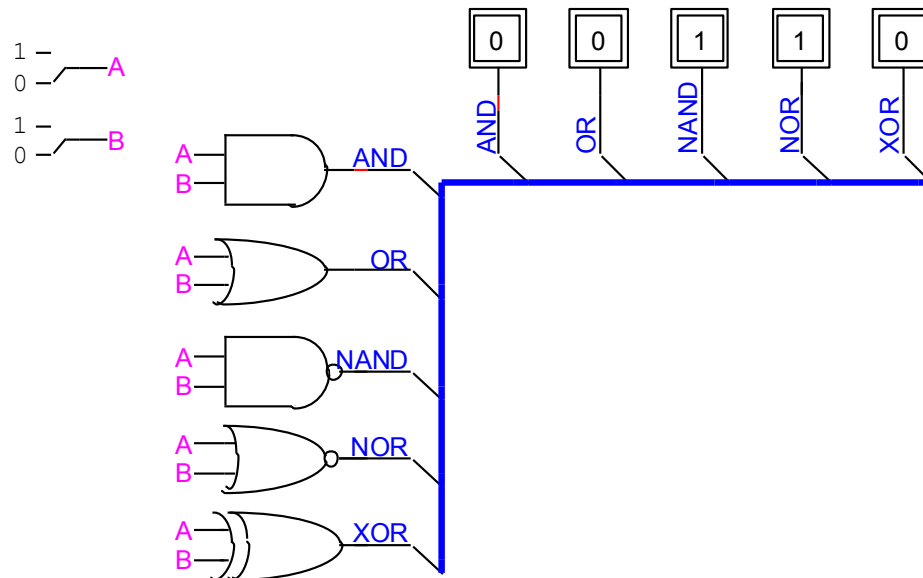
0

0

0

0

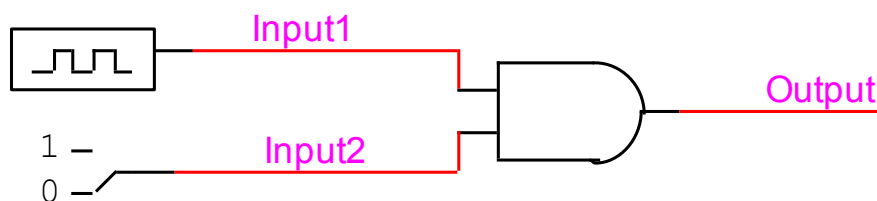
2 สร้างวงจรโดยใช้อุปกรณ์ AND-2, OR-2, NAND-2, NOR-2, XOR-2 จาก Simulation Gates.clf Library แล้วใช้วิธีการเชื่อมโยงสัญญาณด้วยชื่อโดยใช้ cursor จากการคลิกที่ A ใน Toolbar หมายถึงการสร้าง Breakout สามารถกรอกชื่อสัญญาณที่ต้องการ เช่น AND OR NAND NOR XOR และกำหนด Pin Spacing ให้สวยงามตามต้องการได้ลงใน New Breakout dialog box หลังจากกด OK แล้วให้ใช้ arrow key ในการปรับทิศของ Breakout



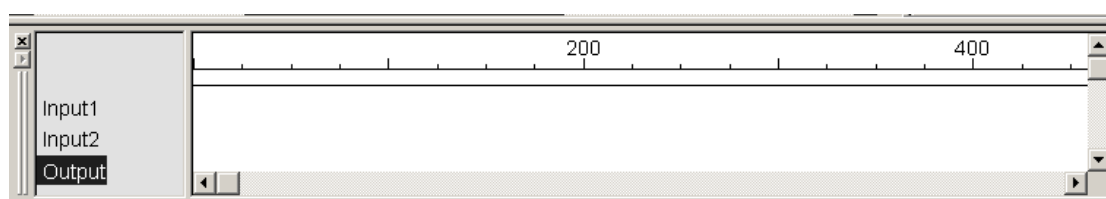
แล้วบันทึกผลการทดลองให้ครบตามตาราง

A	B	AND	OR	NAND	NOR	XOR
0	0	0	0	1	1	0
0	1	0	1	1	0	1
1	0	0	1	1	0	1
1	1	1	1	0	0	0

3. สร้างวงจร

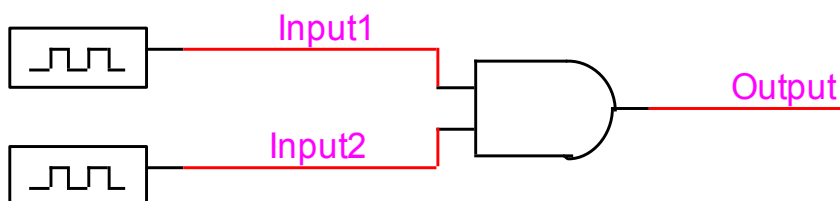


ทำการทดลองเปลี่ยนค่าของ switch บันทึกผลการทดลองของ waveform ที่เกิดขึ้น (ถ้าไม่มี waveform ให้คลิกปุ่ม reset แล้วคลิกปุ่ม run ที่ simulation toolbar)



นอกจากนี้จะต้องทดลองปรับความเร็วที่ระดับต่างๆ และลองใช้ปุ่ม Reset, Run, Step, Zoom in, Zoom Out, Normal Size แล้วสรุปในผลการทดลองด้วย

4. สร้างวงจร



ทำการทดลอง run simulator และบันทึกผล ให้สังเกตว่า clock ทั้งสองทำงานเหมือนกันมีความถี่เท่ากัน แล้วเอาท์พุทจะเหมือนกับอินพุทหรือไม่ (ลองขยายขนาดสัญญาณออกดู)

ทำการเปลี่ยนพารามิเตอร์ clock ของ input2 เป็น Low = 20 High = 20 ทำการบันทึกผลการทดลอง

ข้อสังเกต เมื่อขยายสัญญาณออกไปเรื่อยๆ จนไม่สามารถขยายได้อีก จะพบว่าฟังก์ชันของเอาท์พุทที่ควรจะเป็นนั้นล่าช้ากว่าอินพุทนิดหน่อย ซึ่งเป็นจุดเด่นของซอฟต์แวร์นี้ที่สามารถจำลอง Delay ที่เกิดขึ้นกับเกทได้ด้วย

3)



ເມື່ອ switch ເປັນ true ກໍ່ຈະເຮັດ And ກັບ clock ກໍ່ຈະສົ່ງຄືນເມື່ອ clock ເປັນ true ແຕ່ຈະ delay ເມື່ອ switch ເປັນ false ເຮັດ And ກັບ clock ກໍ່ຈະສົ່ງຄືນເປັນ false ຈຶ່ງໄດ້

4)



ໝາຍ: ກໍ່ input 2 ເປັນ true ເມື່ອ input 1 ເປັນ true And ກັບ input 1 ຈຶ່ງ true ເມື່ອ input 1 ເປັນ false ເມື່ອ input 2 ເປັນ false ຈຶ່ງໄດ້ output ເປັນ false