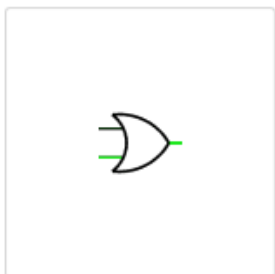


แก้ไขครั้งสุดท้ายเมื่อวันที่ 5 กันยายน 2568

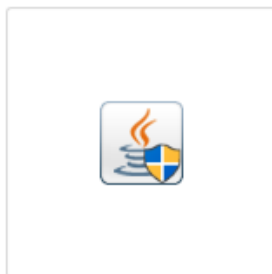
Logisim (2.7.1)

<http://www.cburch.com/logisim>

<http://www.cburch.com/logisim/docs/2.3.0/guide/tutorial/index.html> (คู่มือการใช้งาน)

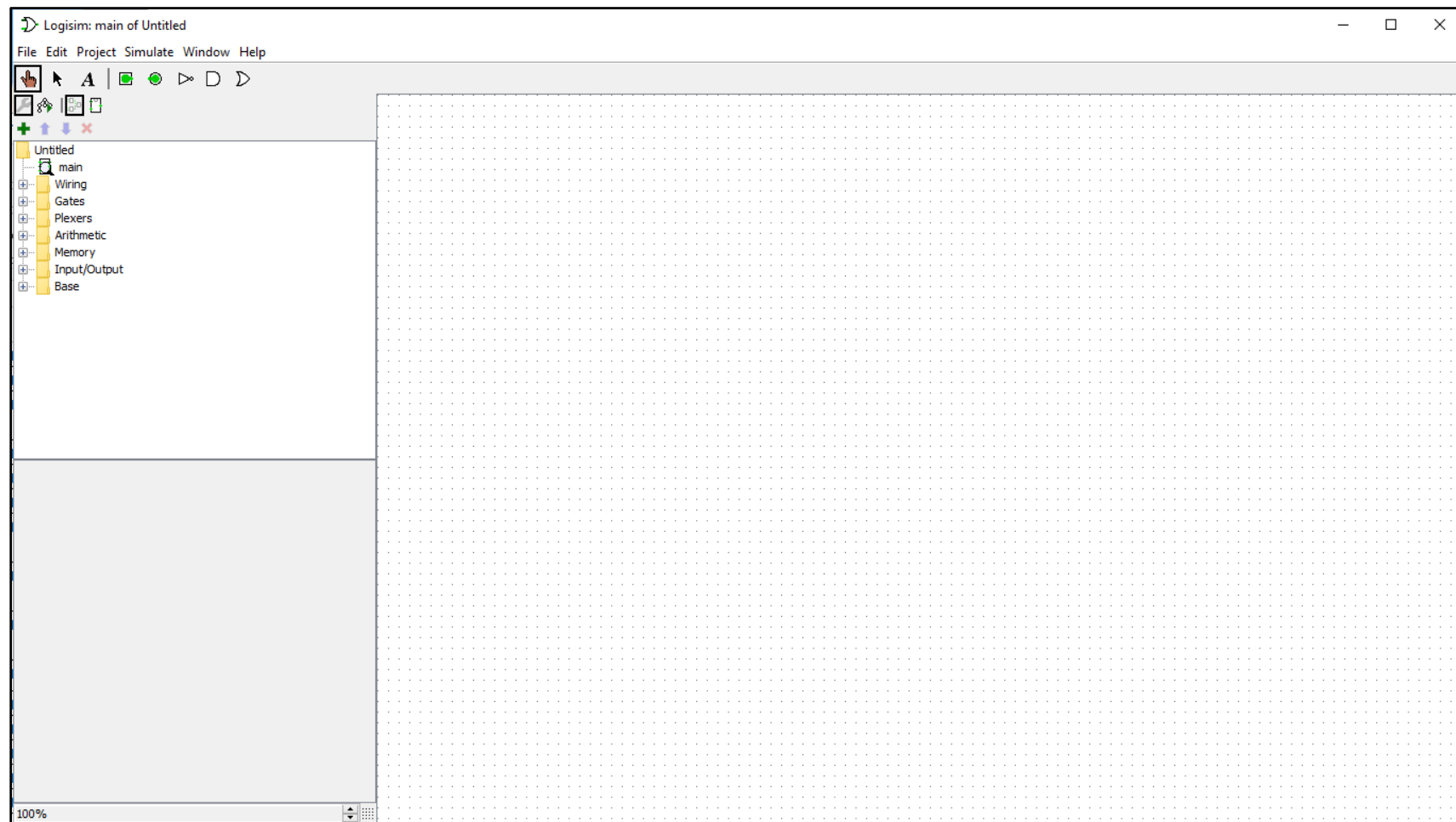


logisim-win-2.7.1



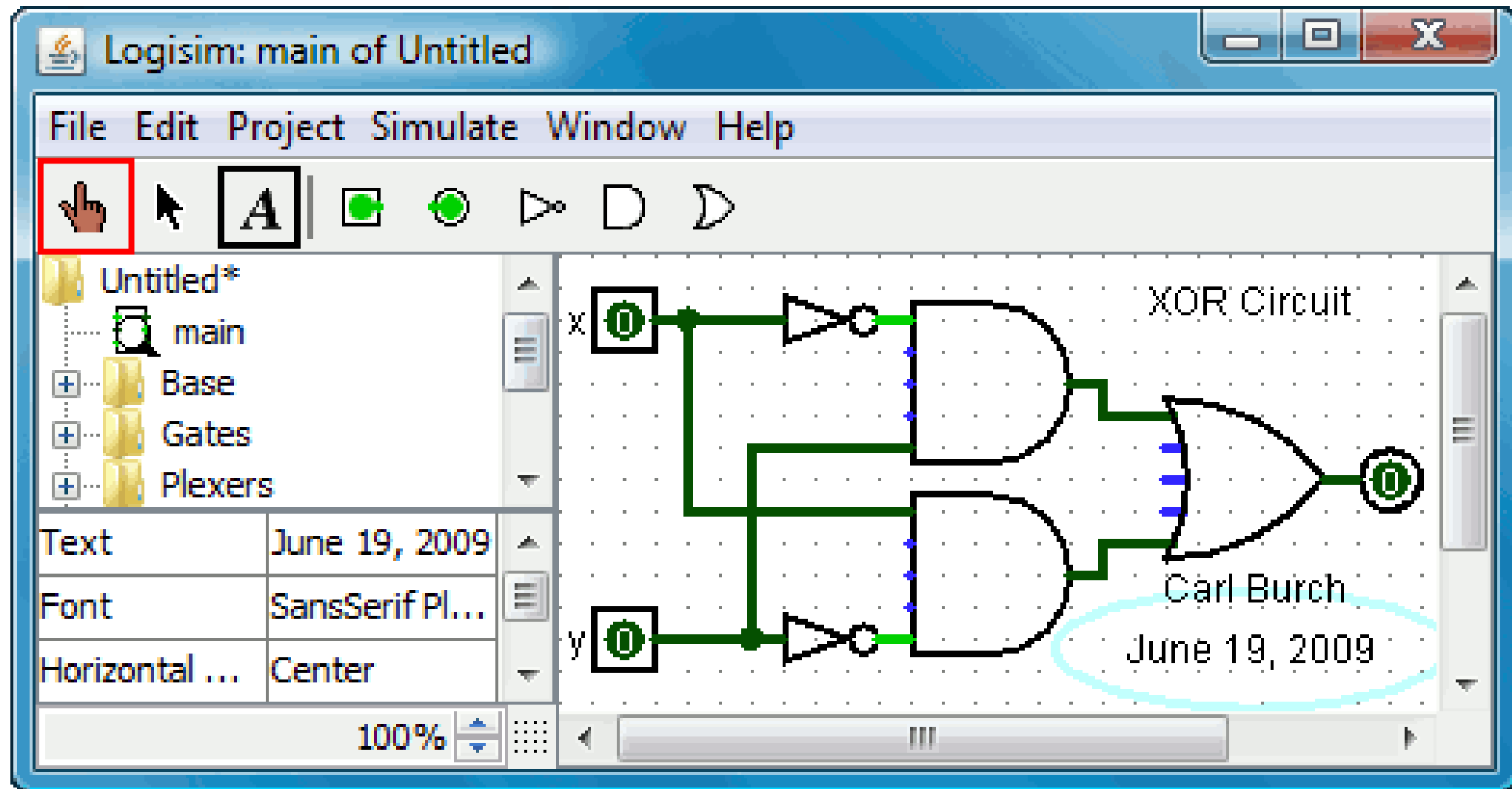
jre-8u351-windows-x64

ติดตั้ง JRE ก่อน

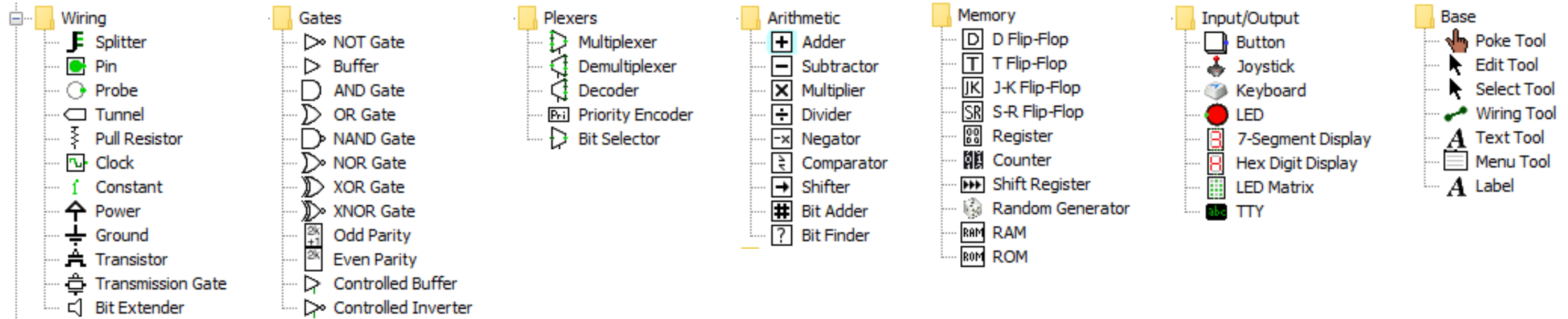


ใช้ Java Runtime Environment (JRE8) ไฟล์ jre-8u351-windows-x64.exe

ทำความรู้จักชิ้นส่วนต่าง ๆ

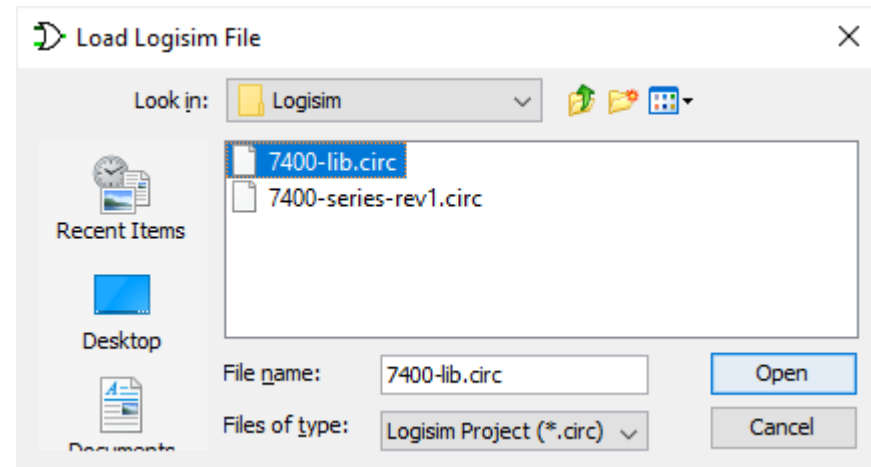
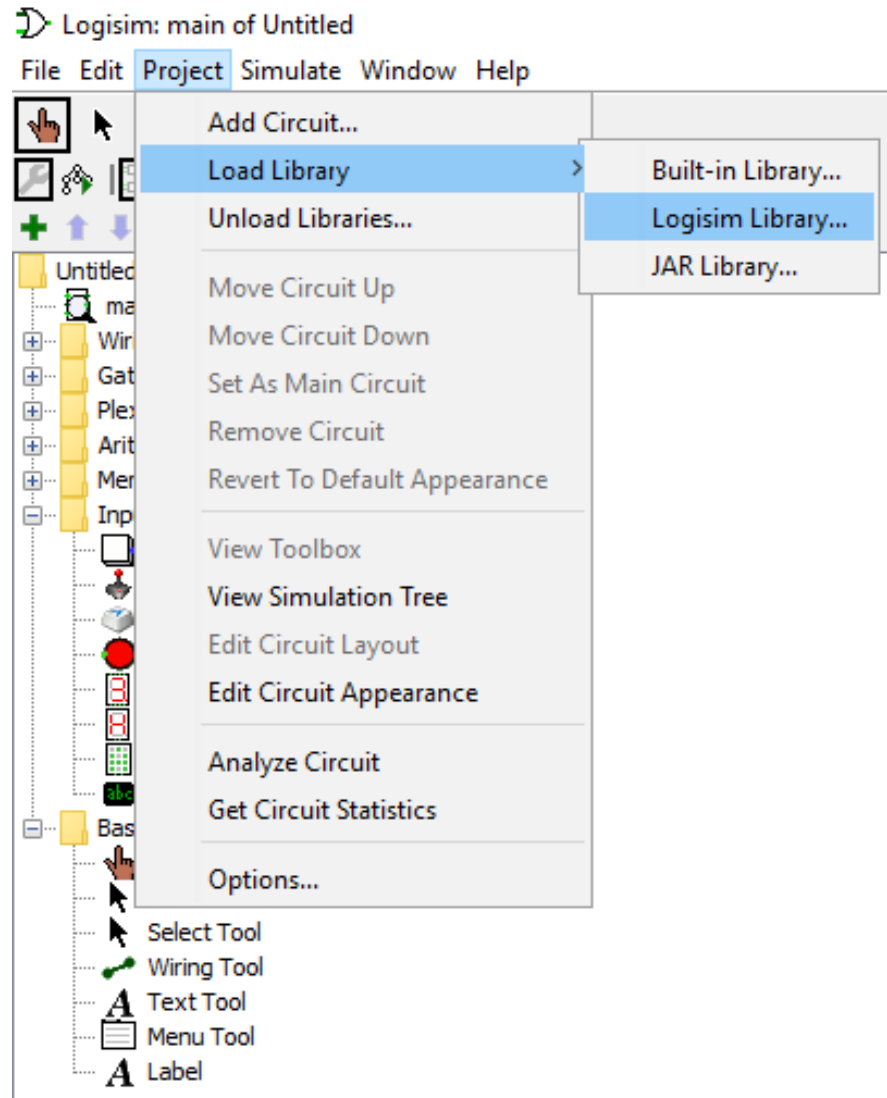


Wiring, Gates, Plexers, Arithmetic, Memory, Input/Output, Base



ติดตั้ง Library

<http://www.cburch.com/logisim/links.html>



7400 series

7400-lib

main

7400

7402

7404

7407

7408

7410

7411

7420

7421

7425

7427

7430

7432

7448

7451

7474

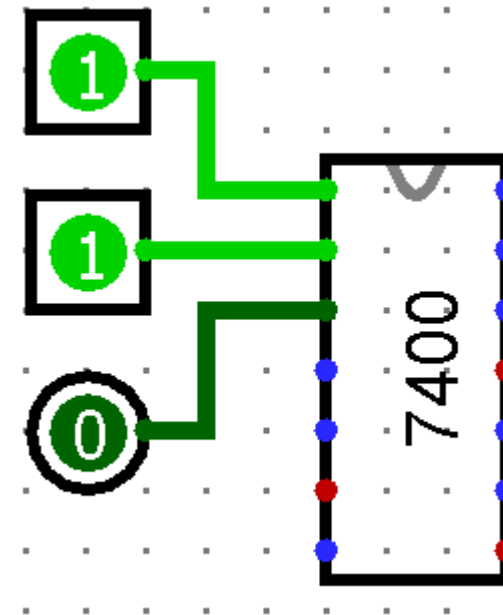
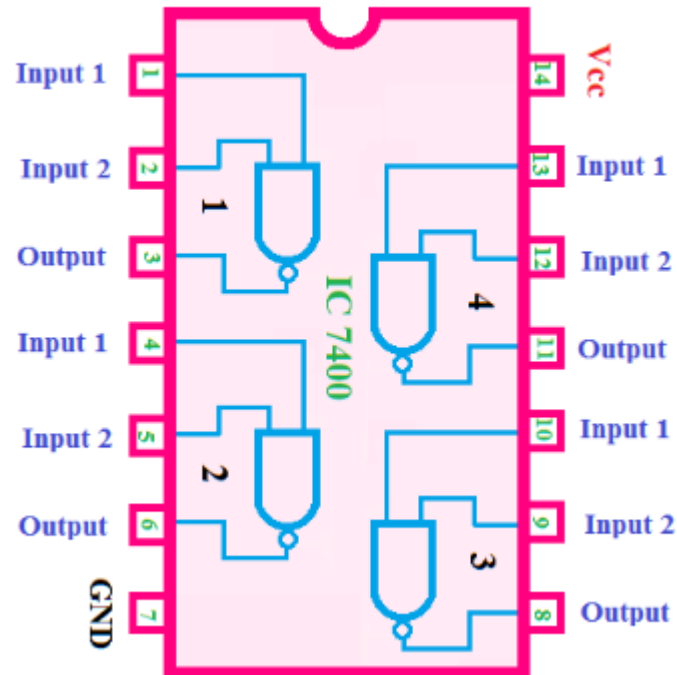
7476

7483

7486

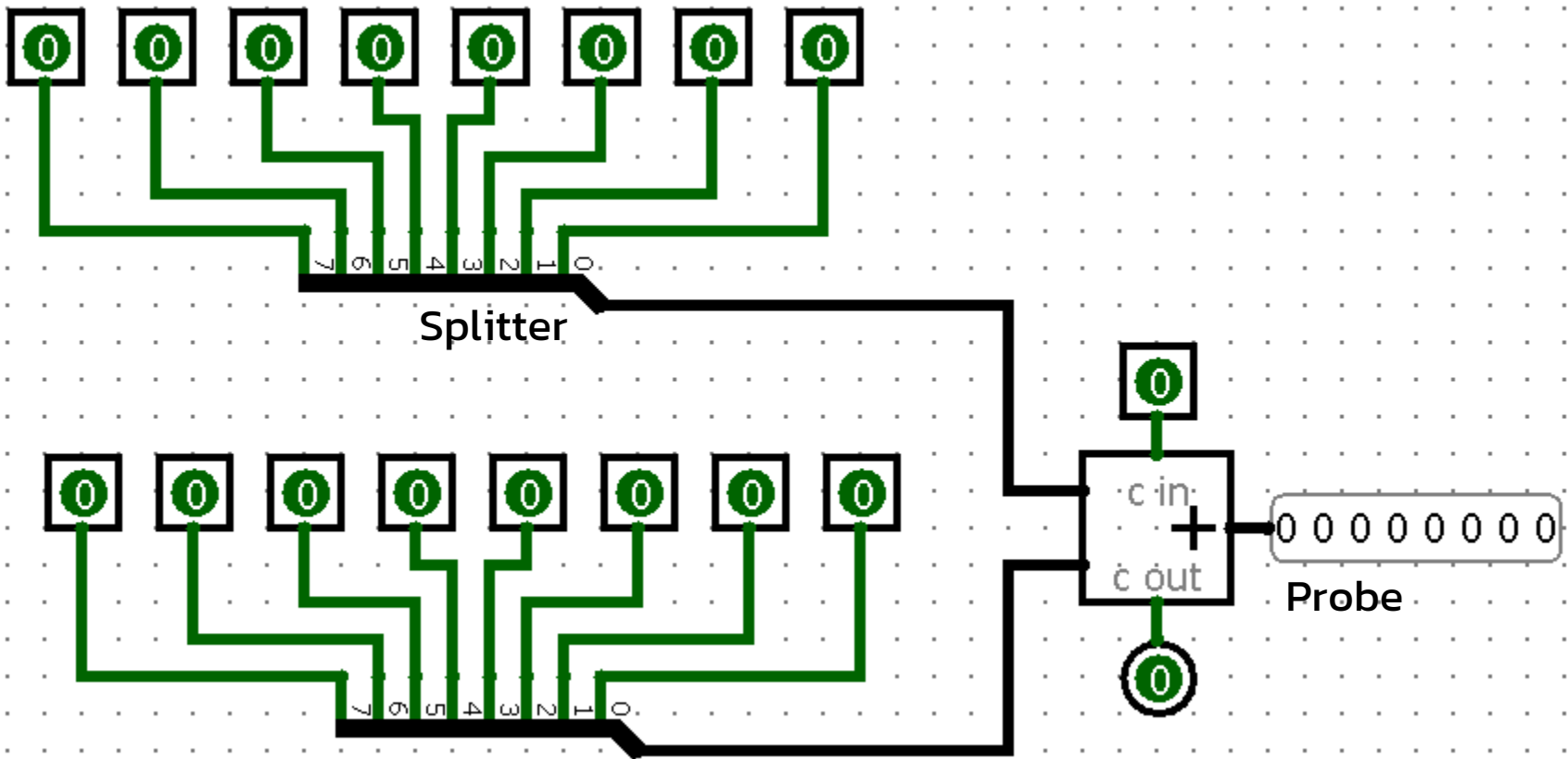
7489

รอยบาก บอกว่าเป็นด้านบน

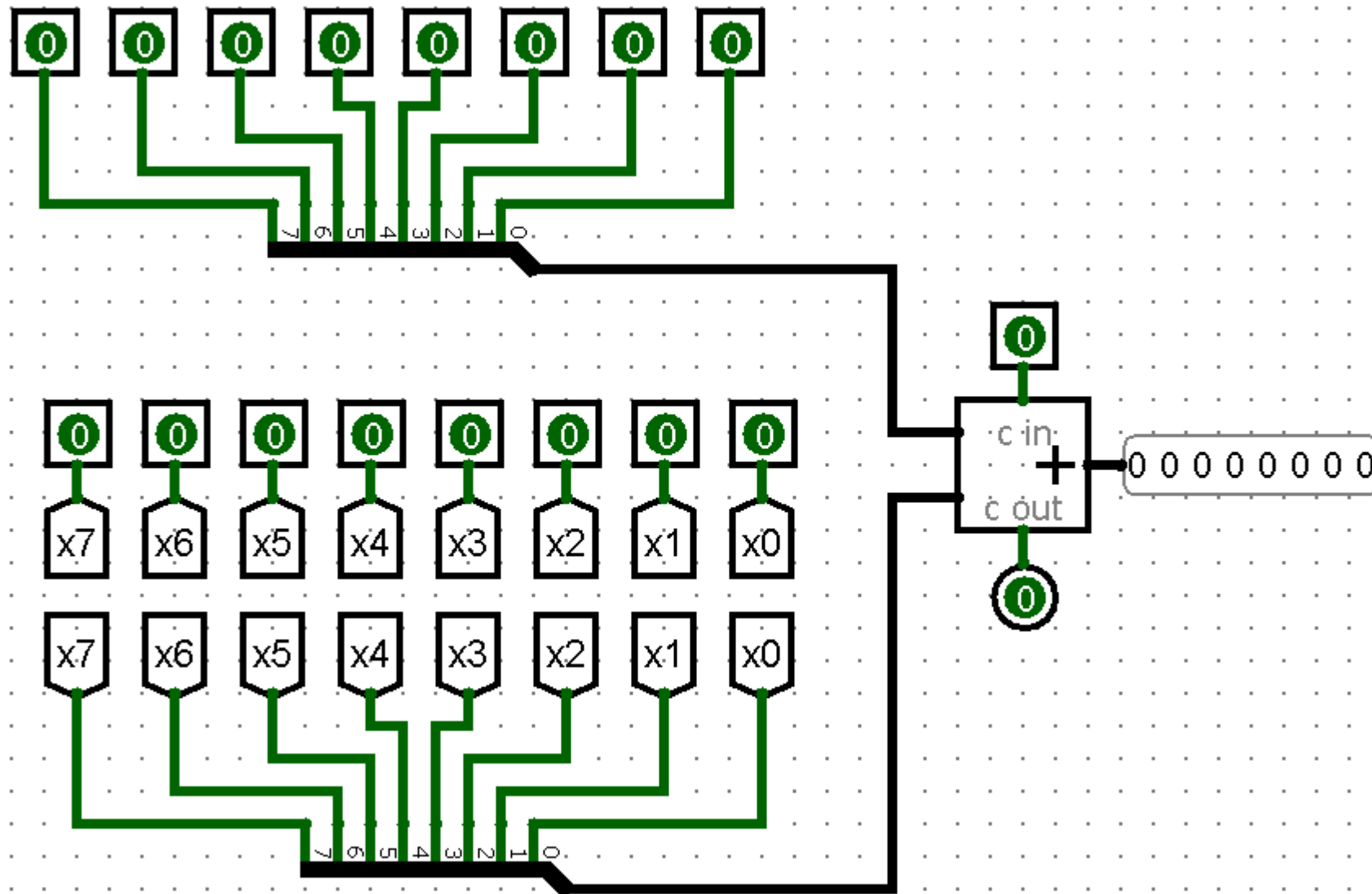


จุดสีน้ำเงิน = อินพุต, จุดสีแดง = เอาพุต
สายสีเขียวอ่อน = 1, สายสีเขียวแก่ = 0
สายสีแดง = short circuit
ไม่ต้องต่อ Vcc กับ GND

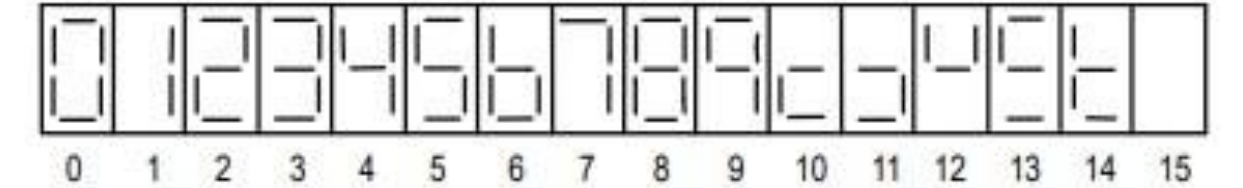
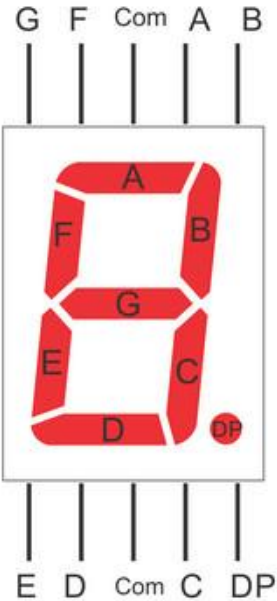
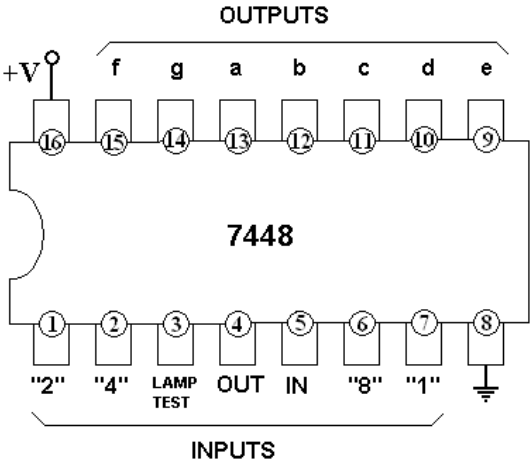
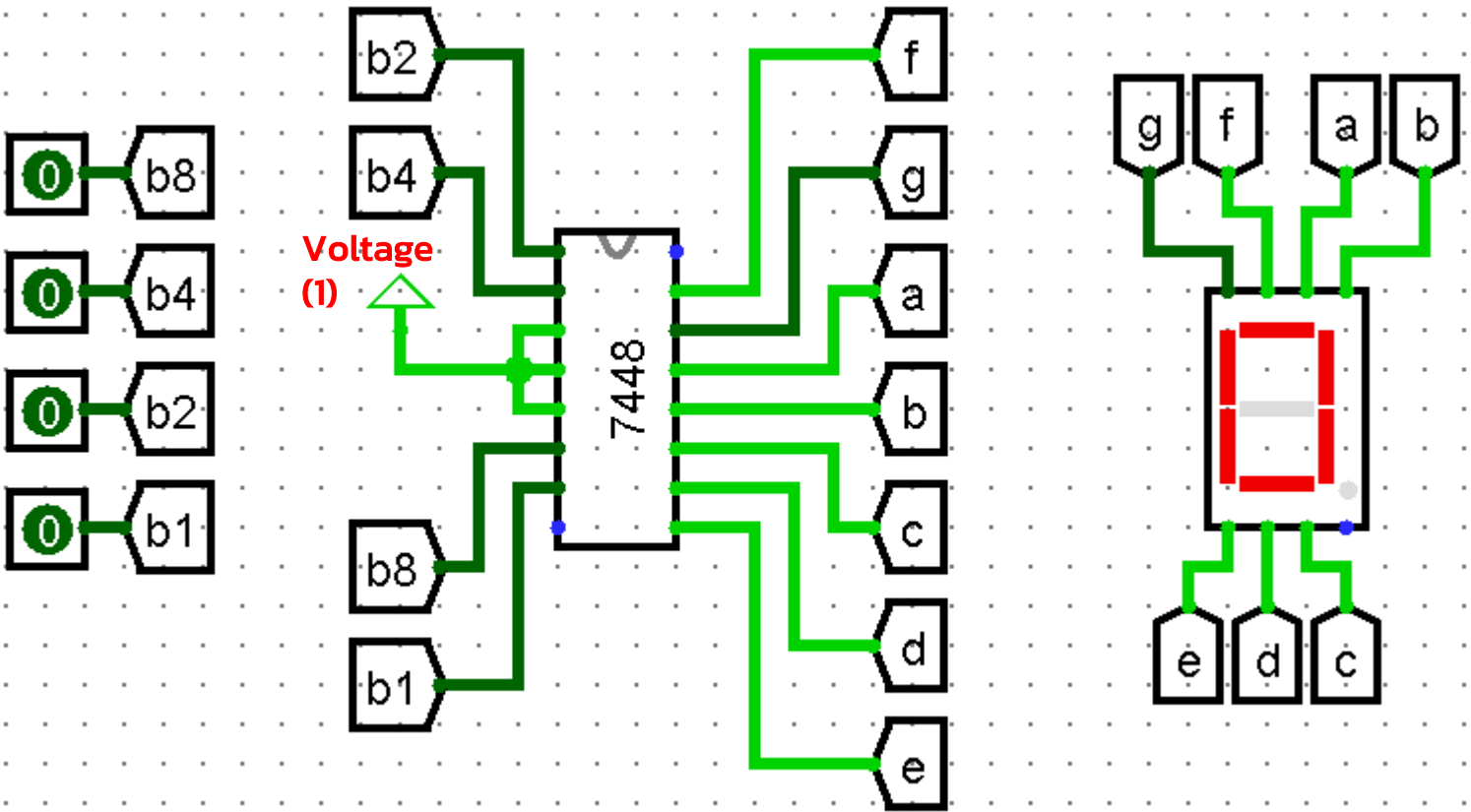
ทดสอบอุปกรณ์ก่อนใช้



ใช้ Tunnel แทนการลากสายไฟยาว ๆ



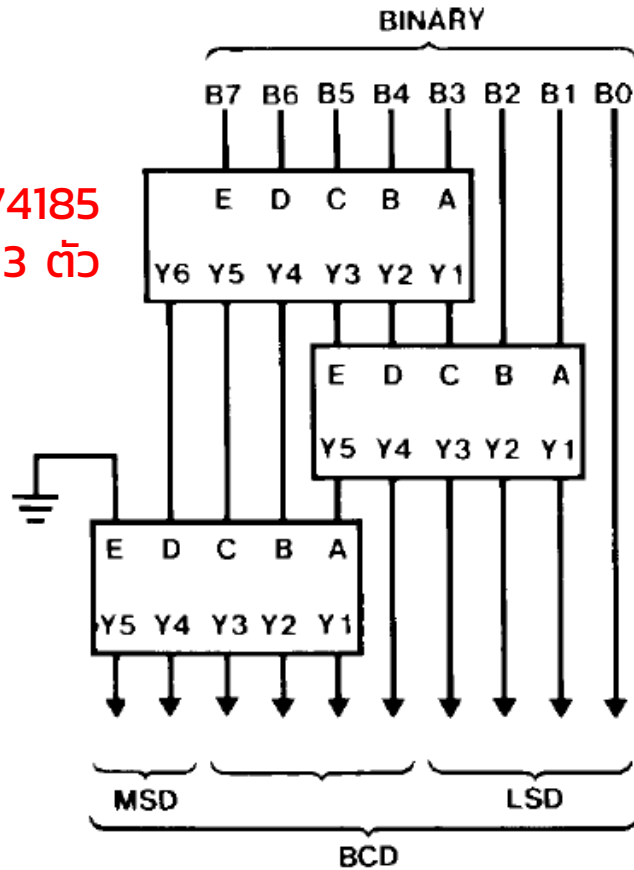
การแสดงผลด้วย 7-segment



NUMERICAL DESIGNATIONS — RESULTANT DISPLAYS

Binary to BCD

ใช้ IC เบอร์ 74185
จำนวน 3 ตัว



TL/F/6561-11

FIGURE 5. 8-Bit Binary-to-BCD Converter

MSD—Most significant decade

LSD—Least significant decade

Note A: Each rectangle represents a DM74185A.

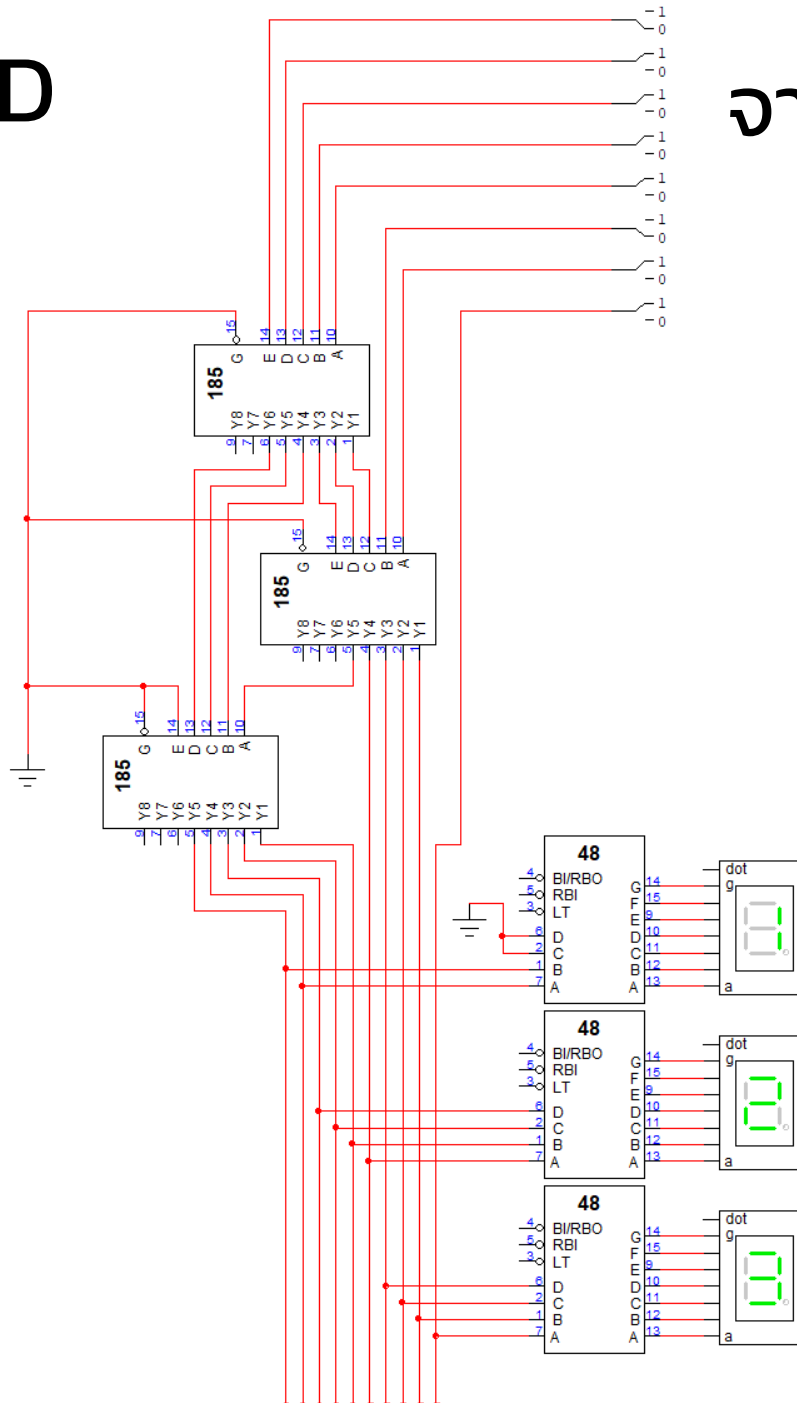
Note B: All unused E inputs are grounded.

8-bit Binary to BCD

รูปนี้เป็นของโปรแกรม [LogicWorks](#)

โปรแกรม [Logisim](#) ยังหา IC เบอร์ **74185** ไม่ได้

ต้องสร้าง 74185 เอง

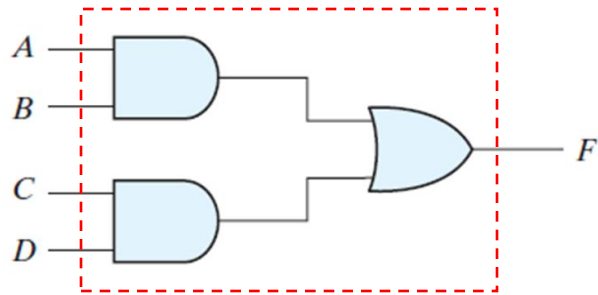


จากบนลงล่าง

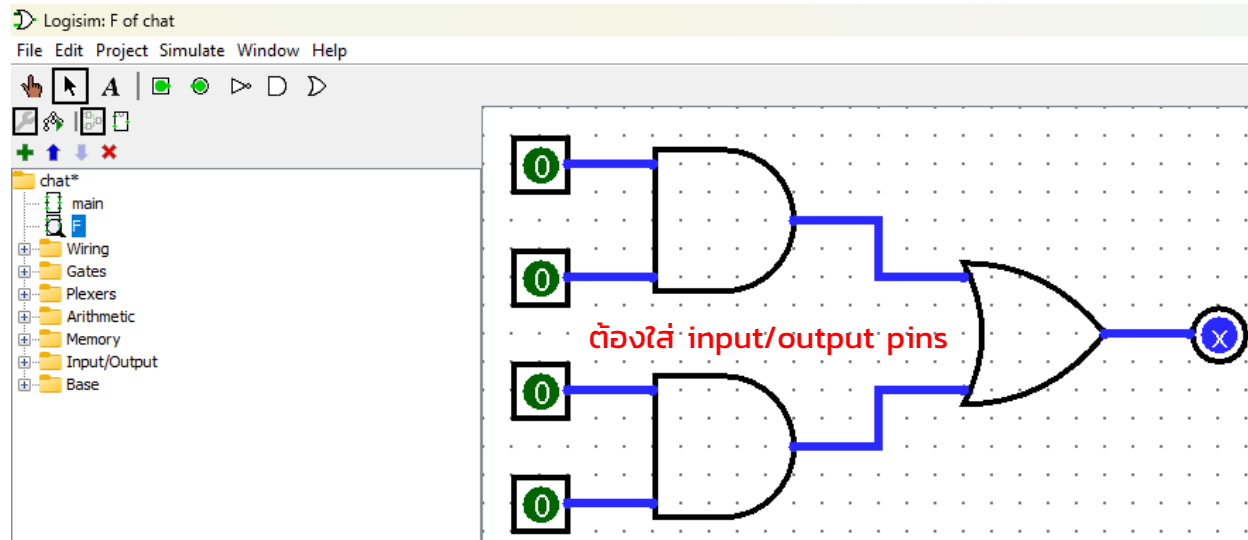
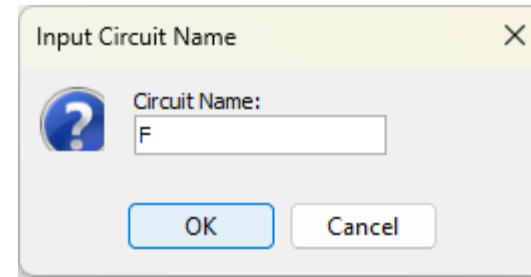
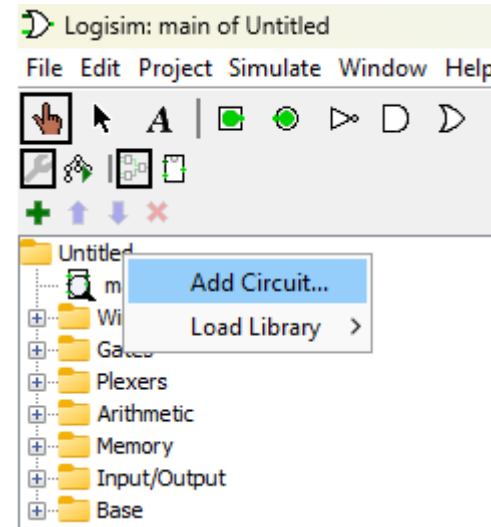
0111 1011

123_{10}

สร้าง IC บน Logisim

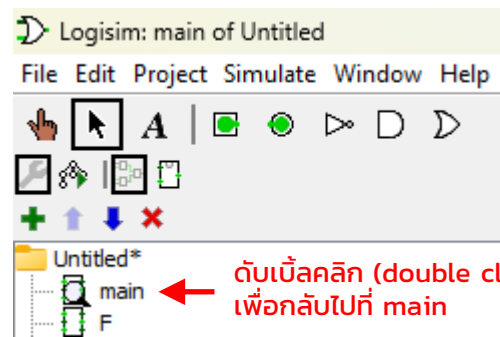
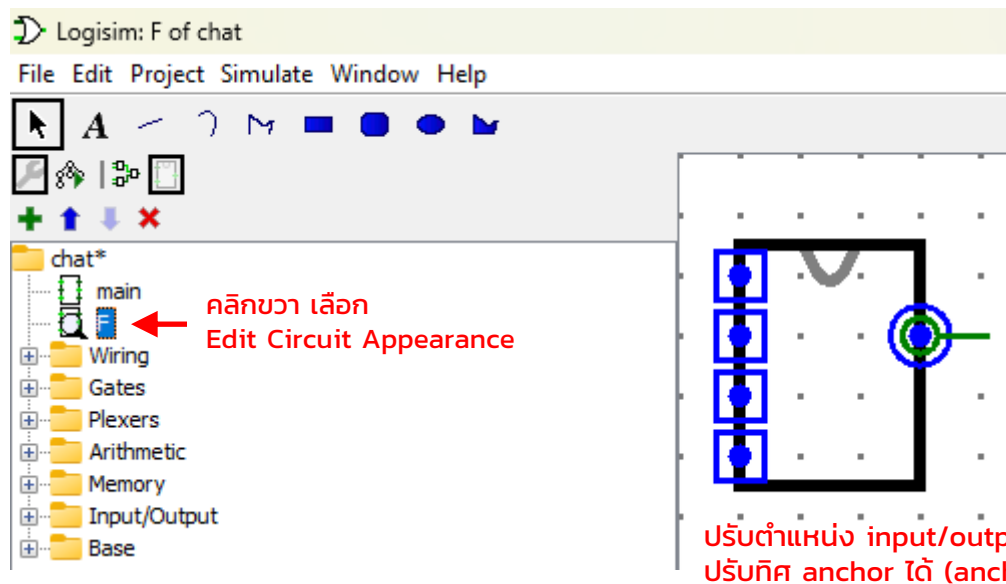


ทำในกรอบสีแดงให้เป็น IC

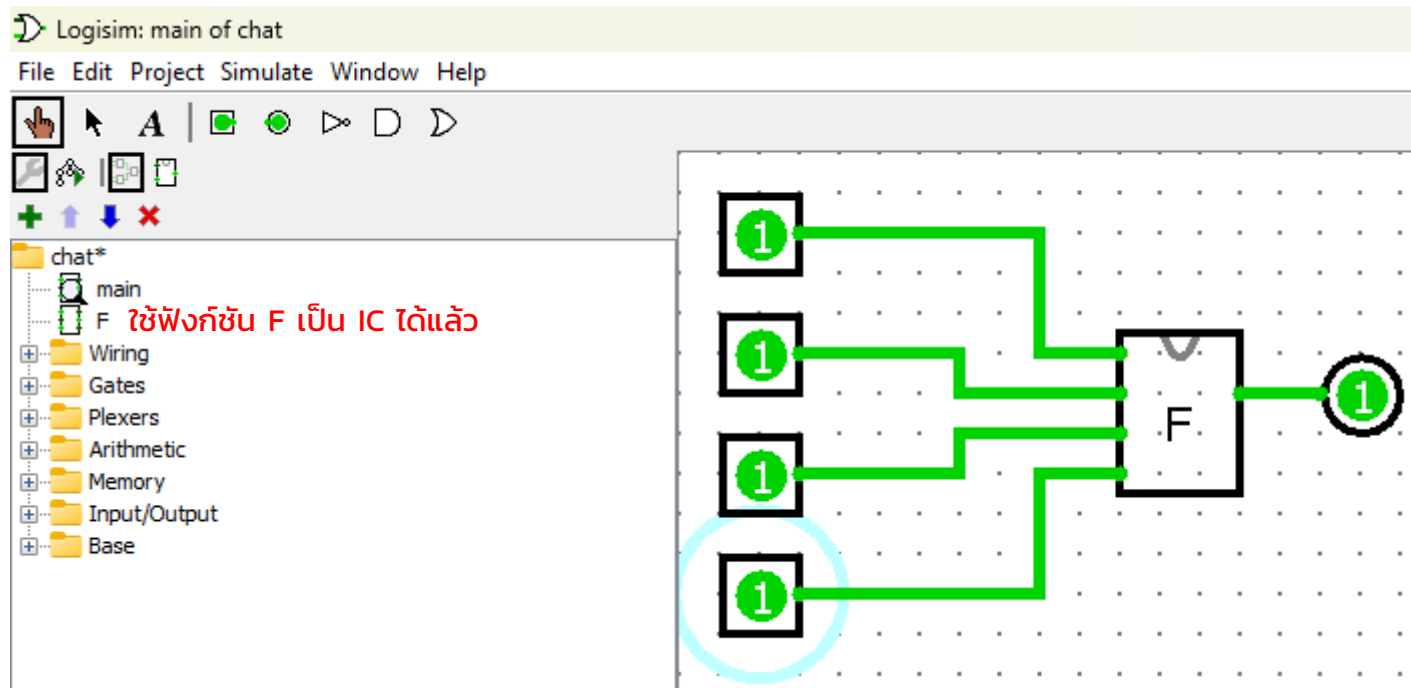


ต้องใช้ input/output pins

ถ้าโปรแกรมค้าง หรือไม่ทำงานตามที่ควร
ให้ปิดแล้วเปิดโปรแกรมใหม่



ปรับตำแหน่ง input/output pins ได้
ปรับทิศ anchor ได้ (anchor = east คือชี้ไปตะวันออก)

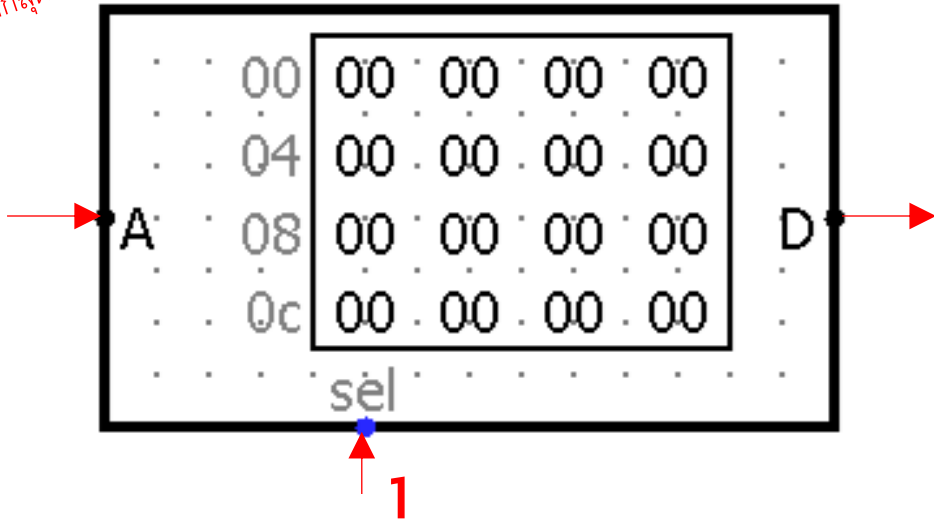


สร้าง IC เบอร์ 74185 แล้วเอามาต่อกันเป็น binary-to-bcd converter

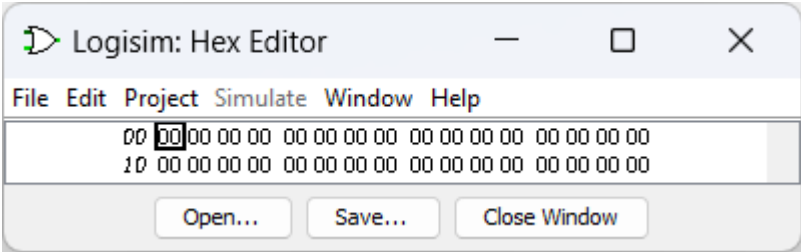
Function Tables															
Binary Words		Inputs						Outputs							
		Binary Select					Enable	Y8	Y7	Y6	Y5	Y4	Y3	Y2	Y1
E	D	C	B	A	G										
0	1	L	L	L	L	L	L	H	H	L	L	L	L	L	L
2	3	L	L	L	L	H	L	H	H	L	L	L	L	L	H
4	5	L	L	L	H	L	L	H	H	L	L	L	L	H	L
6	7	L	L	L	H	H	L	H	H	L	L	L	L	H	H
8	9	L	L	H	L	L	L	H	H	L	L	L	H	L	L
10	11	L	L	H	L	H	L	H	H	L	L	H	L	L	L
12	13	L	L	H	H	L	L	H	H	L	L	H	L	L	H
14	15	L	L	H	H	H	L	H	H	L	L	H	L	H	L
16	17	L	H	L	L	L	L	H	H	L	L	H	L	H	H
18	19	L	H	L	L	H	L	H	H	L	L	H	L	L	L
20	21	L	H	L	H	L	L	H	H	L	H	L	L	L	L
22	23	L	H	L	H	H	L	H	H	L	H	L	L	L	H
24	25	L	H	H	L	L	L	H	H	L	H	L	L	H	L
26	27	L	H	H	L	H	L	H	H	L	H	L	L	H	H
28	29	L	H	H	H	L	L	H	H	L	H	L	H	L	L
30	31	L	H	H	H	H	L	H	H	L	H	H	L	L	L
32	33	H	L	L	L	L	L	H	H	L	H	H	L	L	H
34	35	H	L	L	L	H	L	H	H	L	H	H	L	H	L
36	37	H	L	L	H	L	L	H	H	L	H	H	L	H	H
38	39	H	L	L	H	H	L	H	H	L	H	H	H	L	L
40	41	H	L	H	L	L	L	H	H	H	L	L	L	L	L
42	43	H	L	H	L	H	L	H	H	H	L	L	L	L	H
44	45	H	L	H	H	L	L	H	H	H	L	L	L	H	L
46	47	H	L	H	H	H	L	H	H	H	L	L	L	H	H
48	49	H	H	L	L	L	L	H	H	H	L	L	H	L	L
50	51	H	H	L	L	H	L	H	H	H	L	H	L	L	L
52	53	H	H	L	H	L	L	H	H	H	L	H	L	L	H
54	55	H	H	L	H	H	L	H	H	H	L	H	L	H	L
56	57	H	H	H	L	L	L	H	H	H	L	H	L	H	H
58	59	H	H	H	L	H	L	H	H	H	L	H	H	L	L
60	61	H	H	H	H	L	L	H	H	H	H	L	L	L	L
62	63	H	H	H	H	H	L	H	H	H	H	L	L	L	H
All		X	X	X	X	X	H	H	H	H	H	H	H	H	H

ไม่ได้เขียนหลักสุดท้าย

ไม่ได้เขียนหลักสุดท้าย



Selection: ROM	
Address Bit Width	5
Data Bit Width	6
Contents	(click to edit)



Double-dabble algorithm

HUNDREDS TENS ONES

0000	0000	0000	11110011	Initialization
0000	0000	0001	11100110	Shift
0000	0000	0011	11001100	Shift
0000	0000	0111	10011000	Shift
0000	0000	1010	10011000	Add 3 to ONES, since it was 7
0000	0001	0101	00110000	Shift
0000	0001	1000	00110000	Add 3 to ONES, since it was 5
0000	0011	0000	01100000	Shift
0000	0110	0000	11000000	Shift
0000	1001	0000	11000000	Add 3 to TENS, since it was 6
0001	0010	0001	10000000	Shift
0010	0100	0011	00000000	Shift

2 4 3

BCD

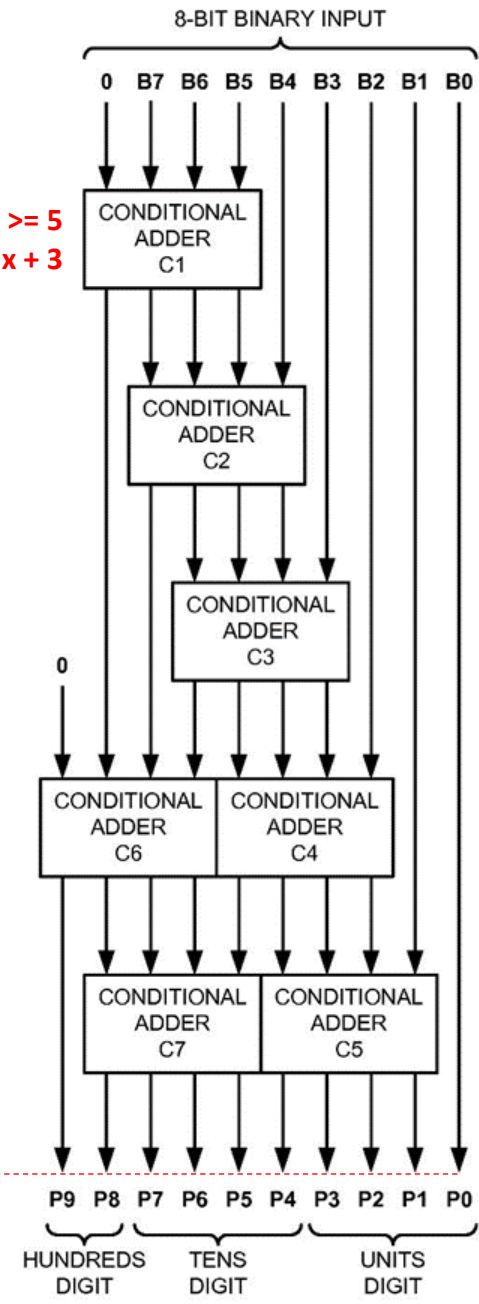
Any BCD digit which is at least 5 is incremented by 3 (ทั้ง carry out)

shift ครั้งสุดท้าย ไม่ต้อง +3

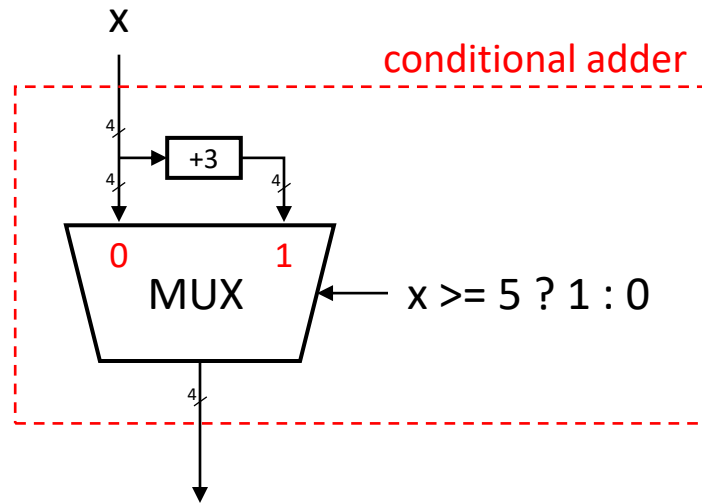
if $x \geq 5$
 $x = x + 3$

==

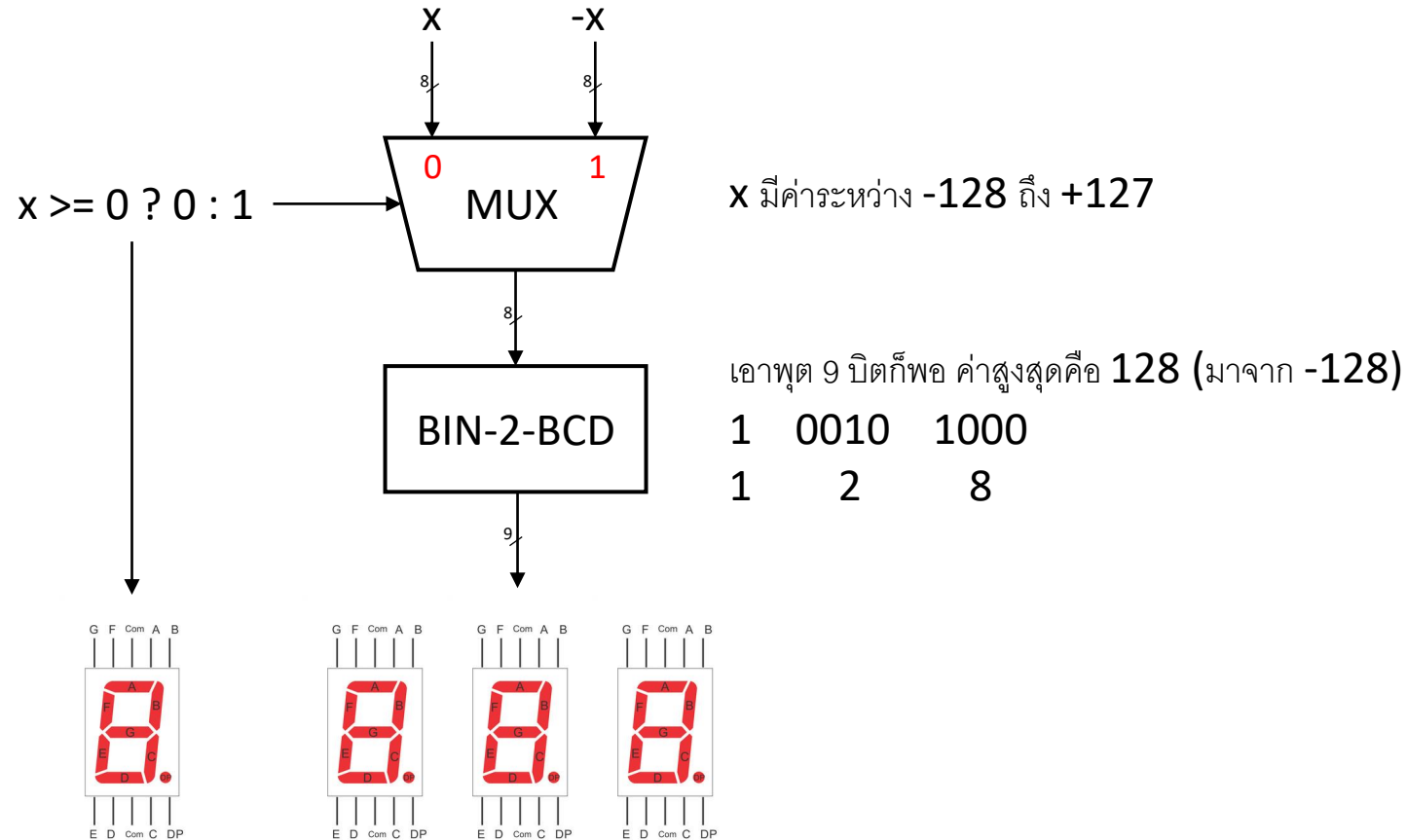
BCD
OUTPUT



Conditional Adder



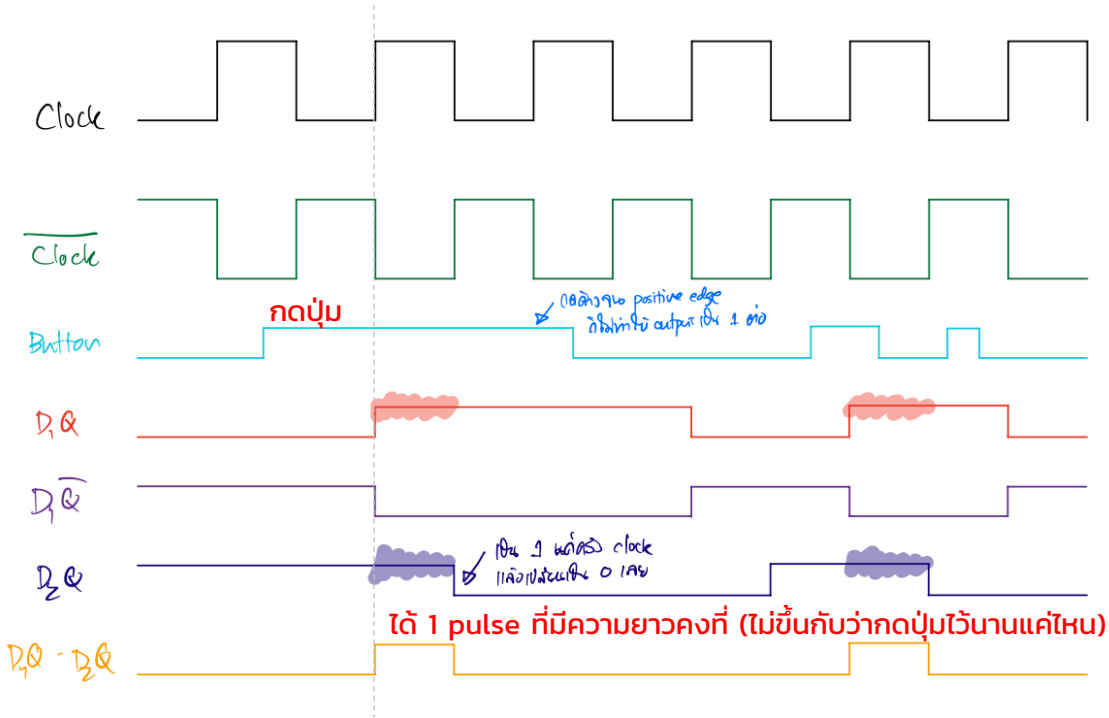
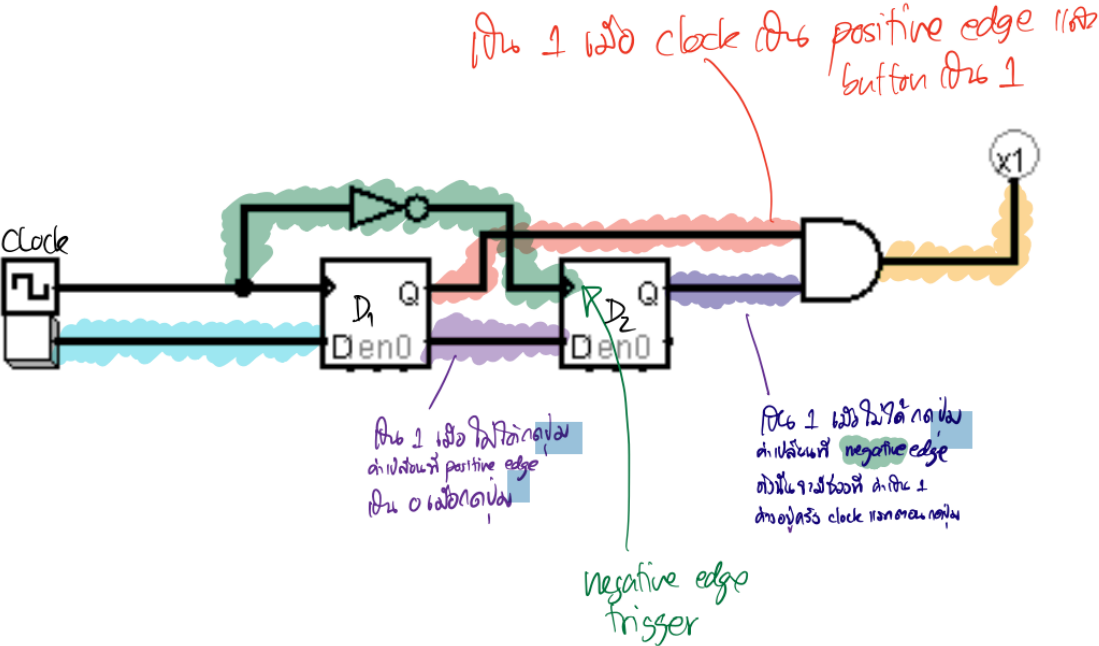
ต่อเติมให้รับอินพุตแบบ 2's complement ได้



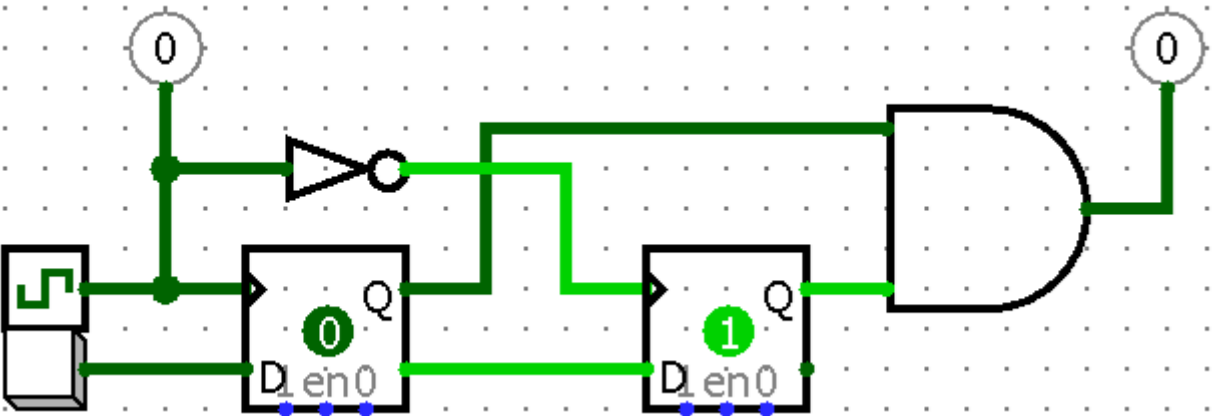
0 ไม่ต้องทำอะไร

1 แสดงเครื่องหมายลบ

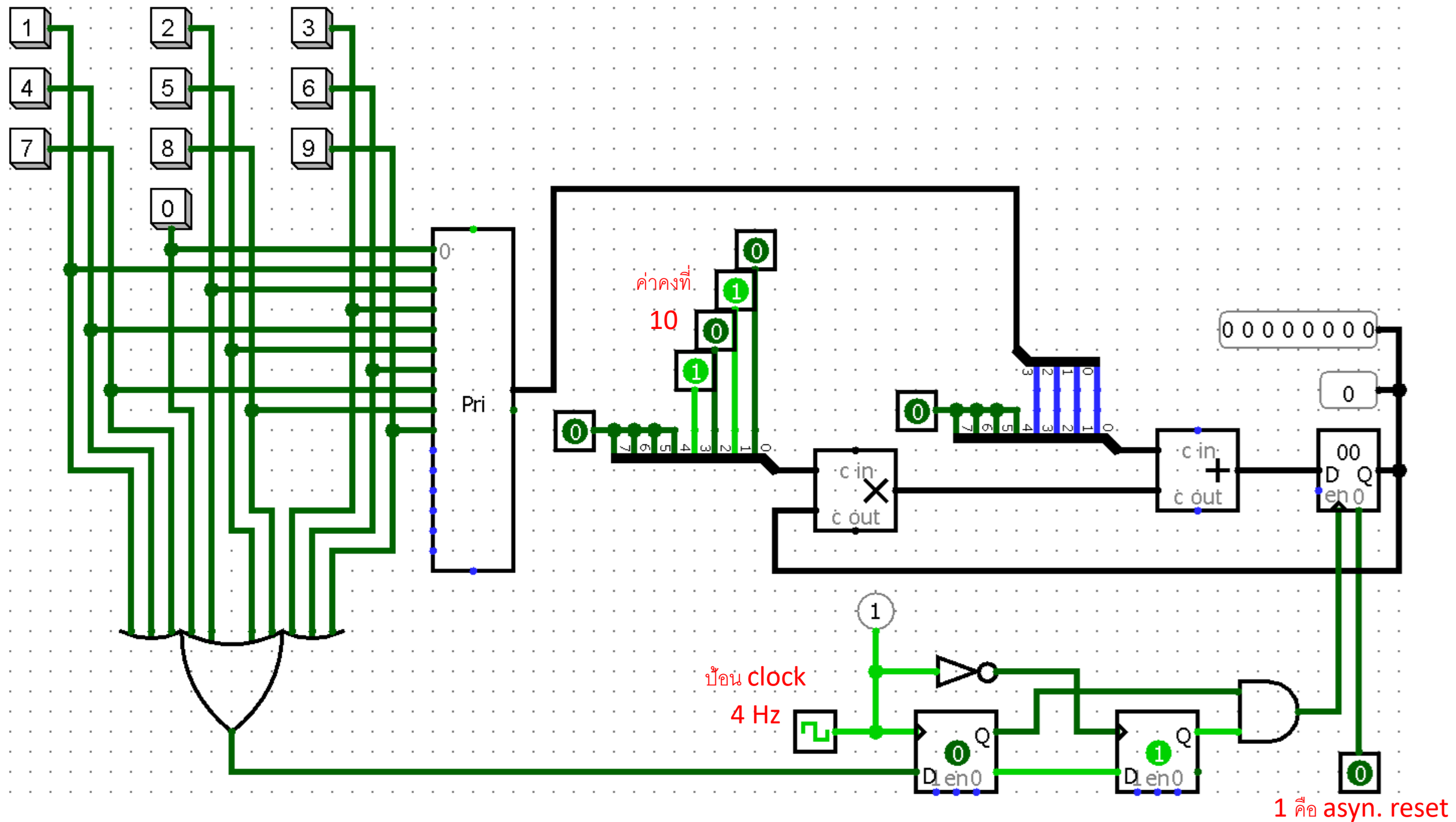
Button (one shot)



ตอน simulate
เลือก Ticks Enabled



กดปุ่ม 1 ครั้ง จะได้ 1 pulse ที่มีความยาวคงที่เสมอ

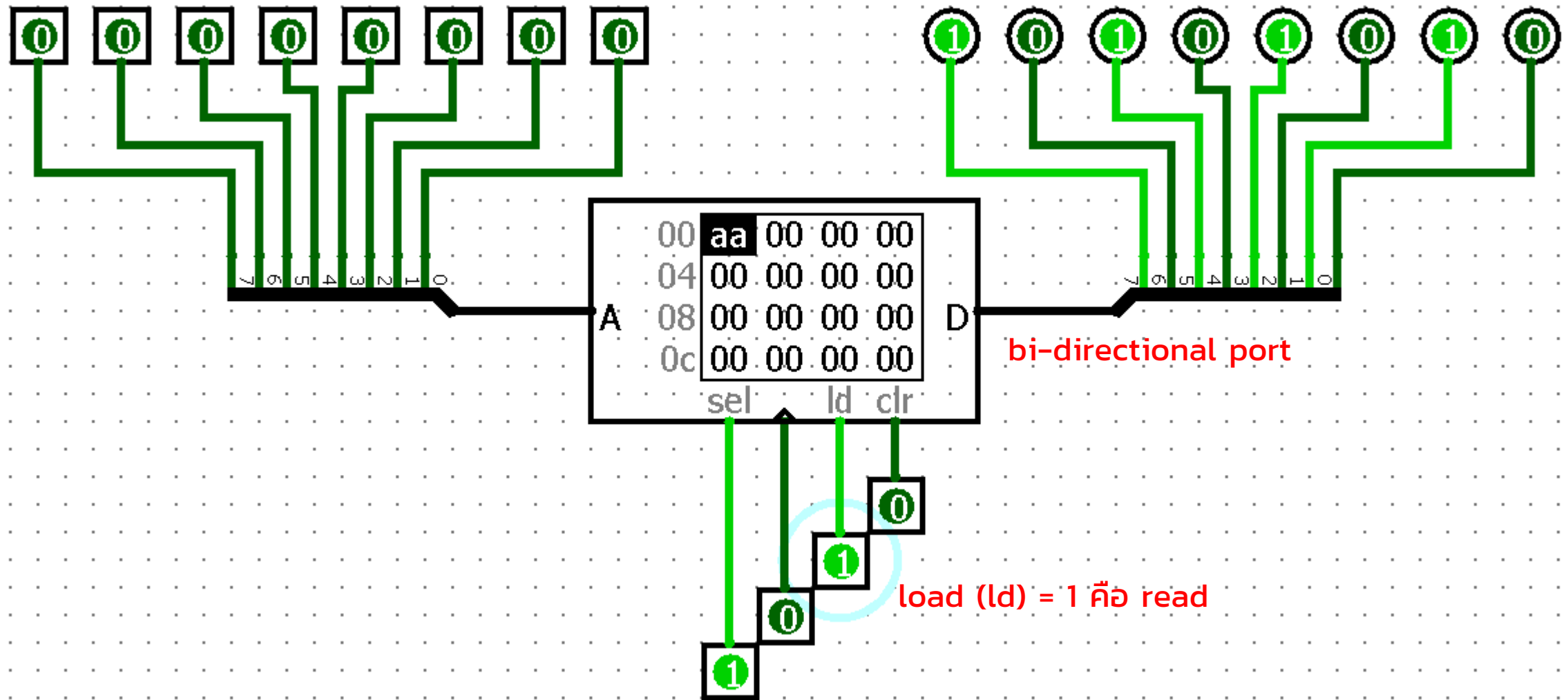


Keyboard

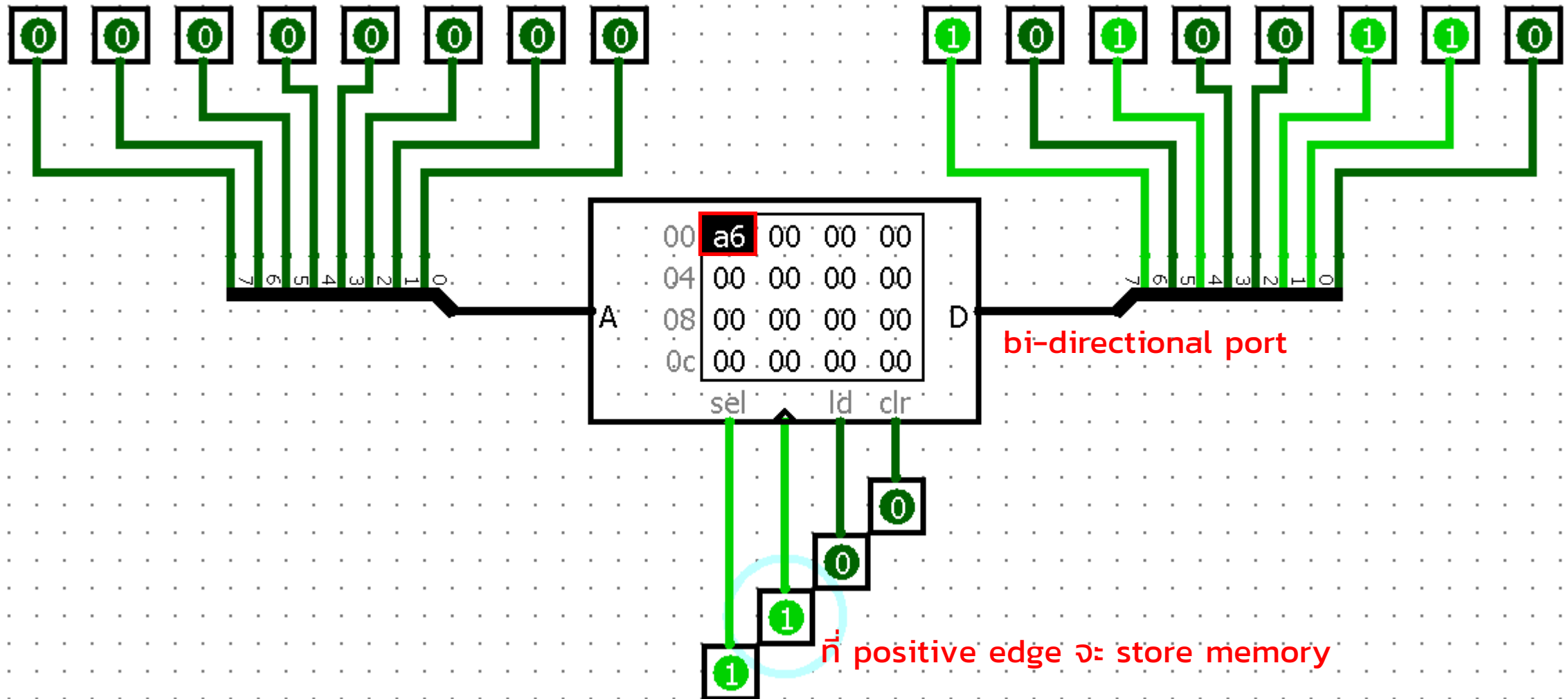
1	2	3	C
4	5	6	CE optional
7	8	9	←
0	+	-	P

- C clear (reset เครื่องคิดเลข)
- CE clear entry
- ← backspace (ลบ 1 หลัก เช่น 123 เป็น 12)
- + add
- subtract
- P push (to stack)

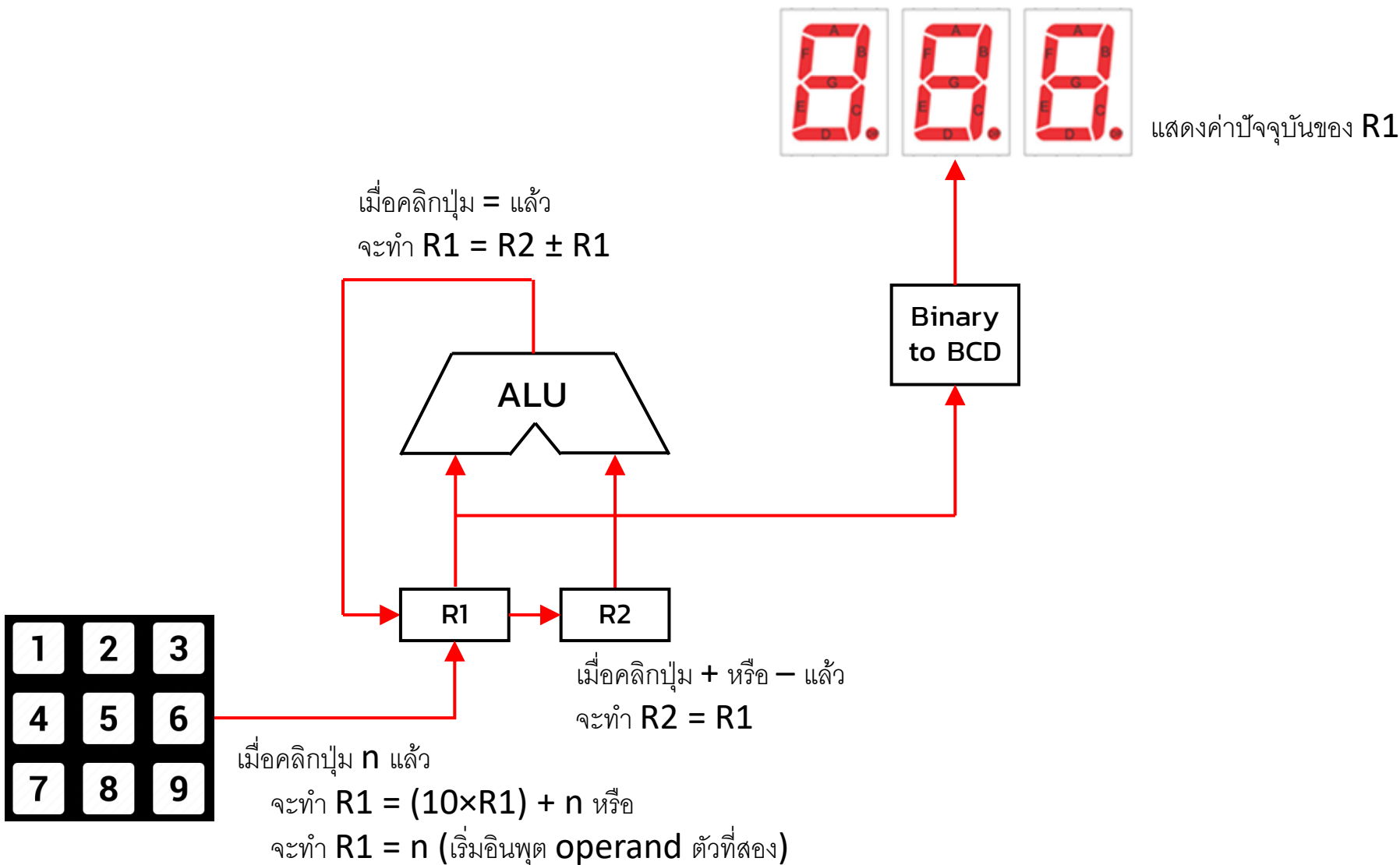
Memory (read)



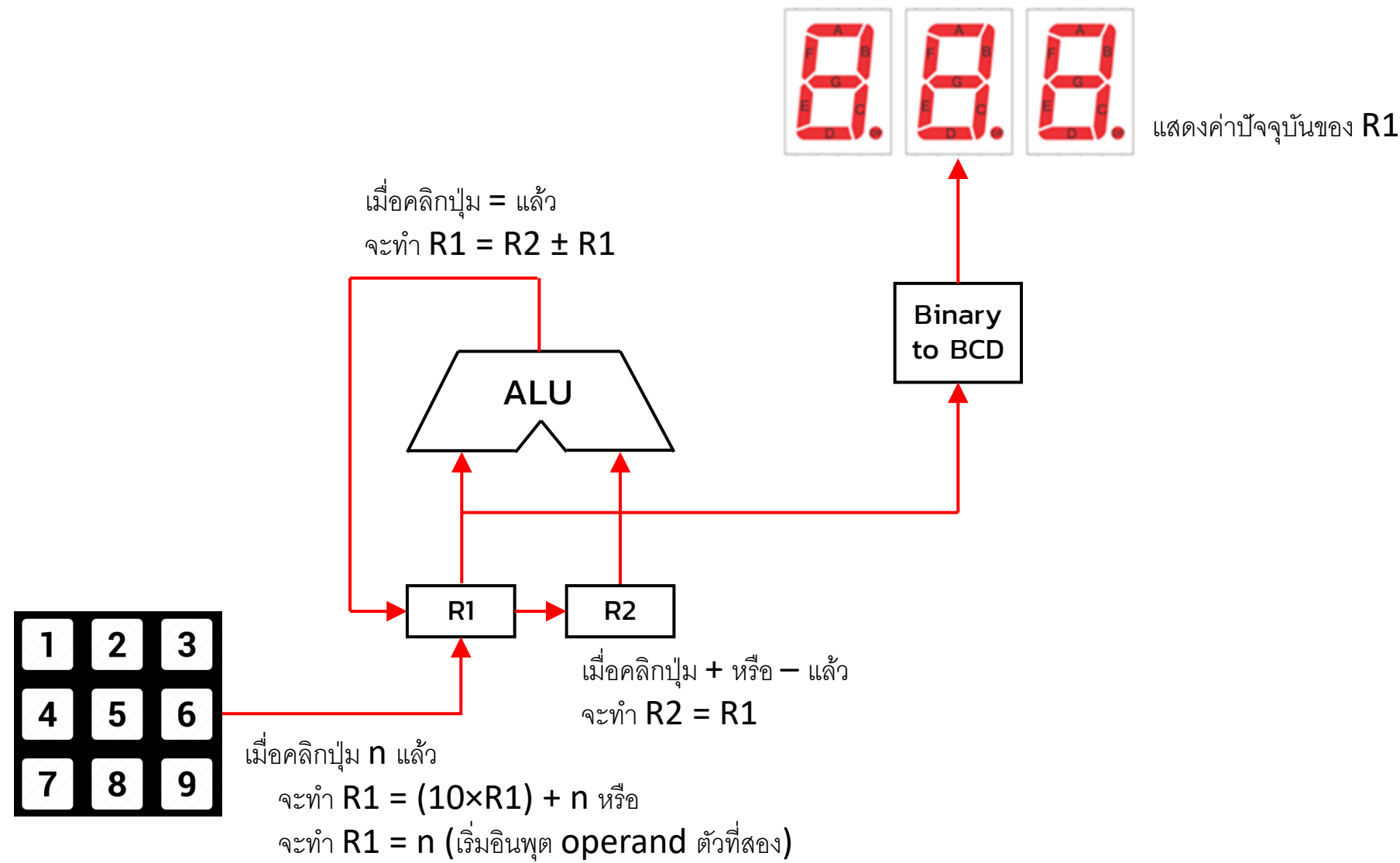
Memory (write)



Data path (infix)



Data path (postfix)



วิดีโอ ความยาวไม่เกิน 10 นาที ส่งลิงก์มาที่ chatchawit.a@chula.ac.th ตั้ง subject ว่า [2301274] ส่งงานเครื่องคิดเลข xxxxxxxxxx (x คือ id นิสิต)

หัวข้อที่ต้องนำเสนอในวิดีโอ

1. data path ข้อมูลมันไหลไปมาระหว่าง module ต่าง ๆ ได้ยังไง มันคำนวณ prefix, infix, postfix ได้ยังไง (3 นาที)
2. การรับอินพุตจากคีย์บอร์ดเป็นเลขฐานสิบ แล้วแปลงเป็นเลขฐานสอง (2 นาที)
3. ออกแบบวงจรบวกและวงจรถ่าย (2 นาที)
4. การแสดงเอาพุตจากเลขฐานสอง แปลงเป็น 7-segment ทำยังไง (2 นาที)
5. ทดลองบวกเลข 2 หลัก เช่น $35 + 71 = 106$ ที่ผลลัพธ์เกิน 100 (20 วินาที)
6. ทดลองลบเลข 2 หลัก เช่น $71 - 35 = 36$ ที่ผลลัพธ์เป็นบวก (20 วินาที)
7. ทดลองลบเลข 2 หลัก เช่น $35 - 71 = -36$ ที่ผลลัพธ์เป็นลบ (20 วินาที)

เวลาที่ใช้ถ่ายคลิปได้นะ ก็อย่าให้รวมแล้วเกิน 10 นาที

Data path ต้องวาดรูปใหม่ ไม่ใช่ตัว circuit

จะได้มี concept ว่ามันทำงานยังไง (ป้องกันลอกกันด้วย)

แยกวิดีโอเป็นตอน ๆ ต้องมีเว็บอัปโหลด

ใช้รูปแบบ (mp4) ที่เร่ง speed ได้

งานกลุ่มแบ่งหน้าที่กัน? แบ่งกันนำเสนอในวิดีโอ

หรืองานเดี่ยวทำ IC มาปลั๊กกับของอาจารย์ ตรวจง่าย

ไม่ต้องมี stack ทำแค่

$1 + 2 = 3$ prefix

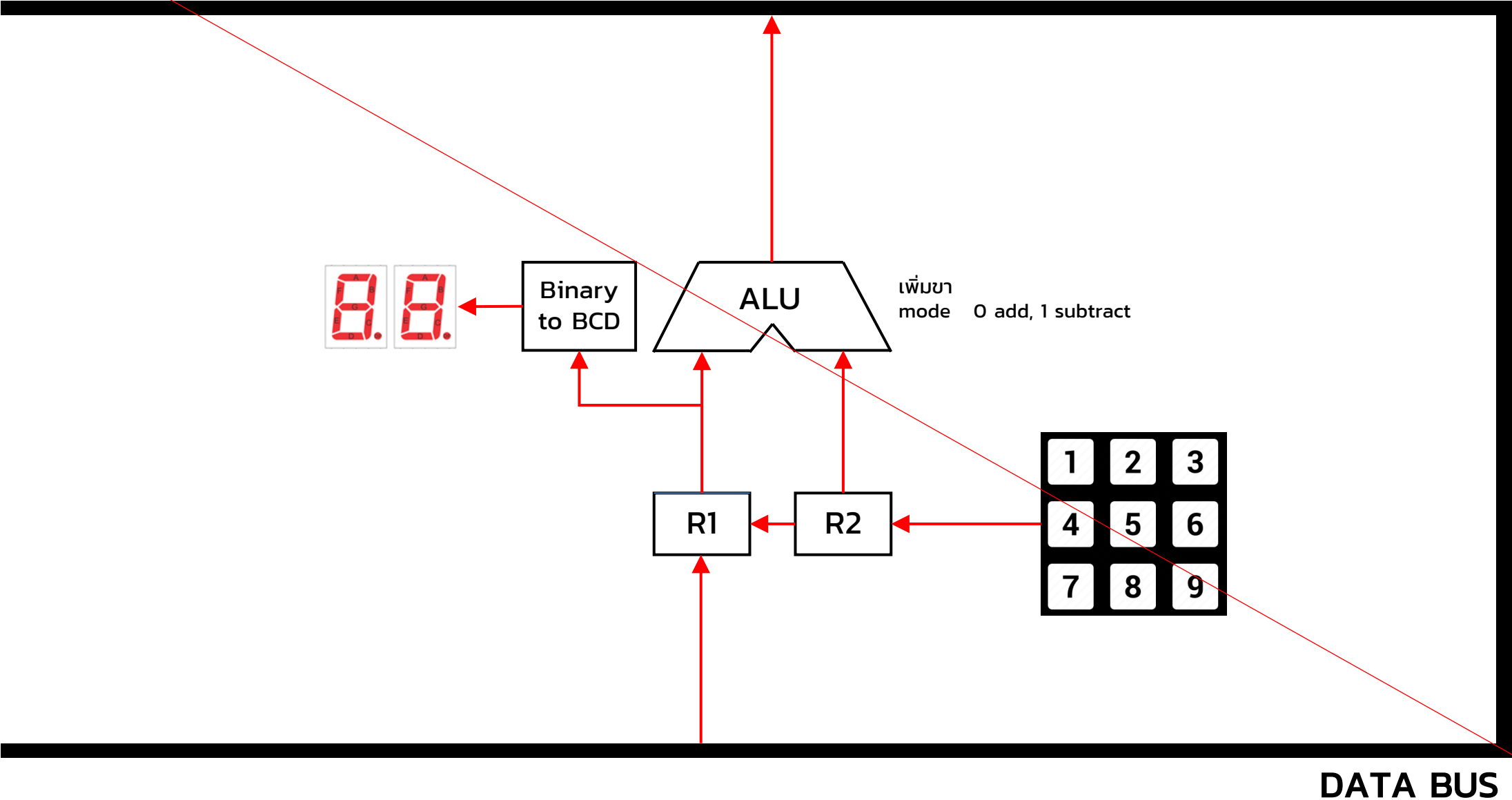
$1\ p\ 2\ p\ +$ postfix (ต้อง p ทุกครั้ง บางคนละครั้งที่สอง)

$+ 1\ p\ 2\ p$ infix (หลังบวกไม่ต้อง push) อธิบายให้ชัดเจน

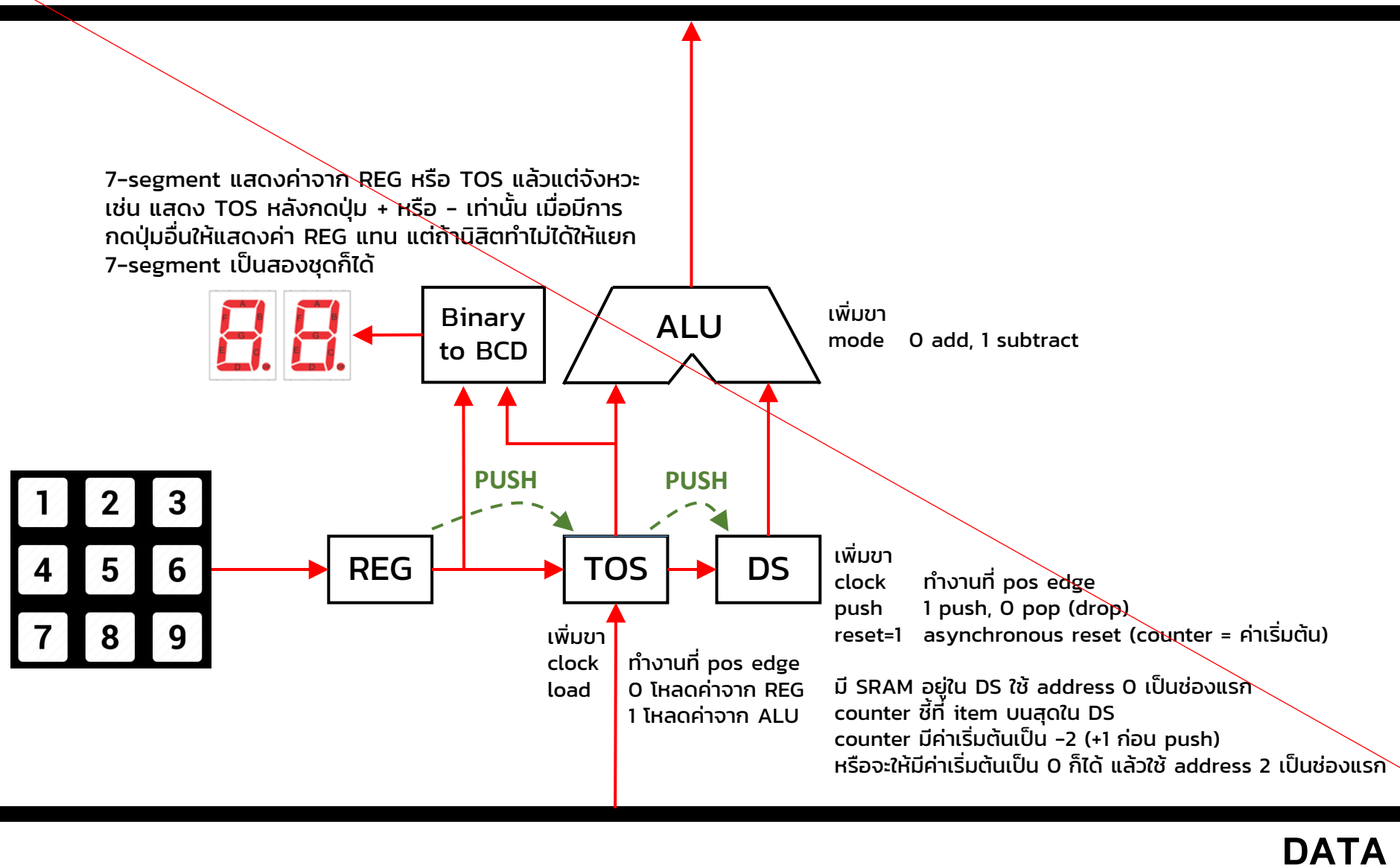
ต้องมีเสียงอธิบายประกอบทุกตอน ห้ามเงียบ

ตรวจแบบไม่มีเสียง ตรวจยากมาก ต้องฟังตามเมาส์

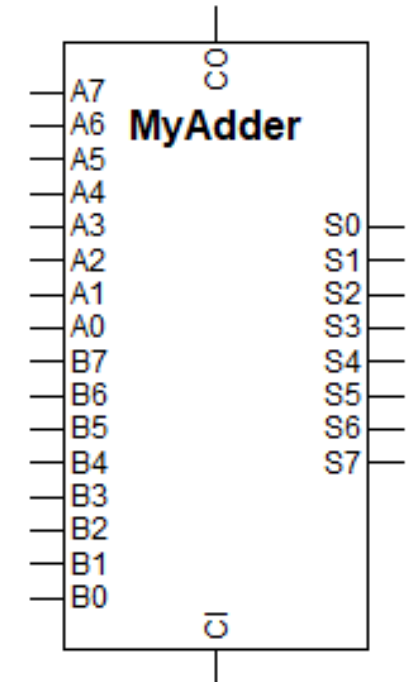
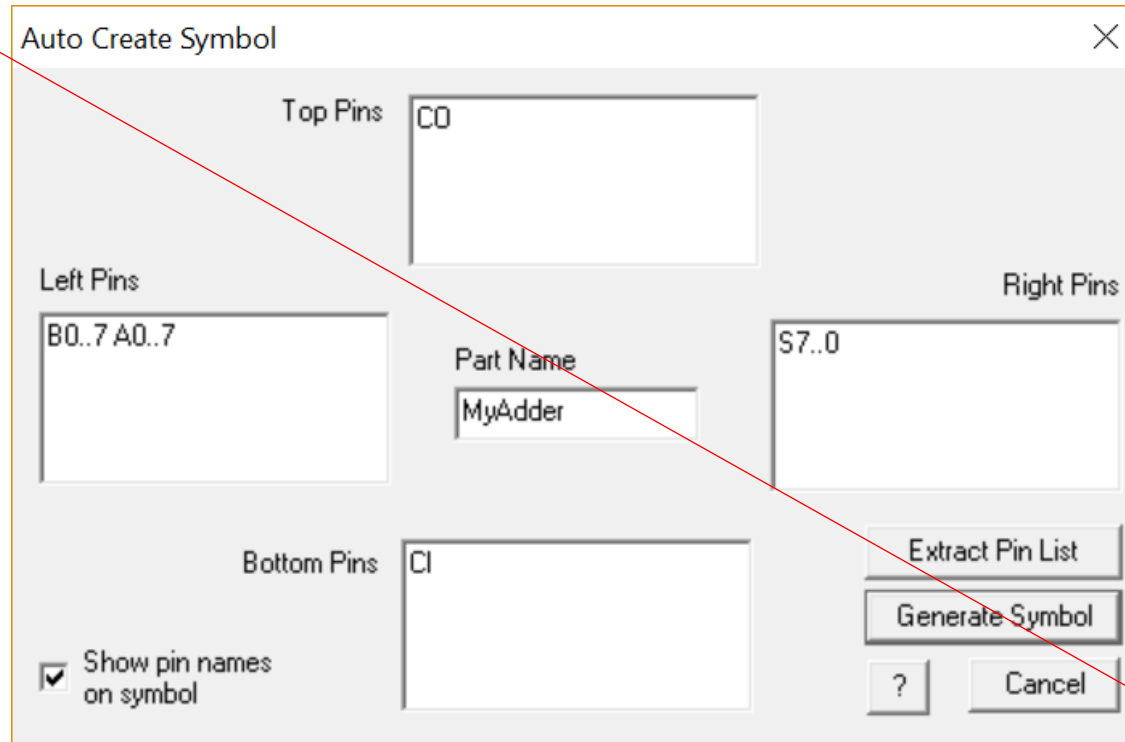
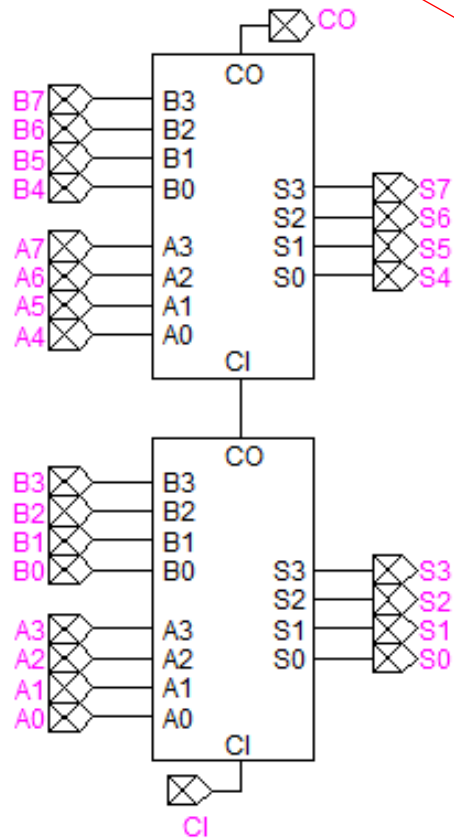
Data path (infix)



Data path (postfix)



สร้าง IC บน LogicWorks



รูปนี้เป็นของโปรแกรม LogicWorks โปรแกรม Logisim ก็ทำได้เหมือนกัน

<http://www.cburch.com/logisim/docs/2.3.0/guide/subcirc/creating.html>

<http://www.cburch.com/logisim/docs/2.3.0/guide/subcirc/using.html>

<http://www.cburch.com/logisim/docs/2.3.0/guide/subcirc/debug.html>