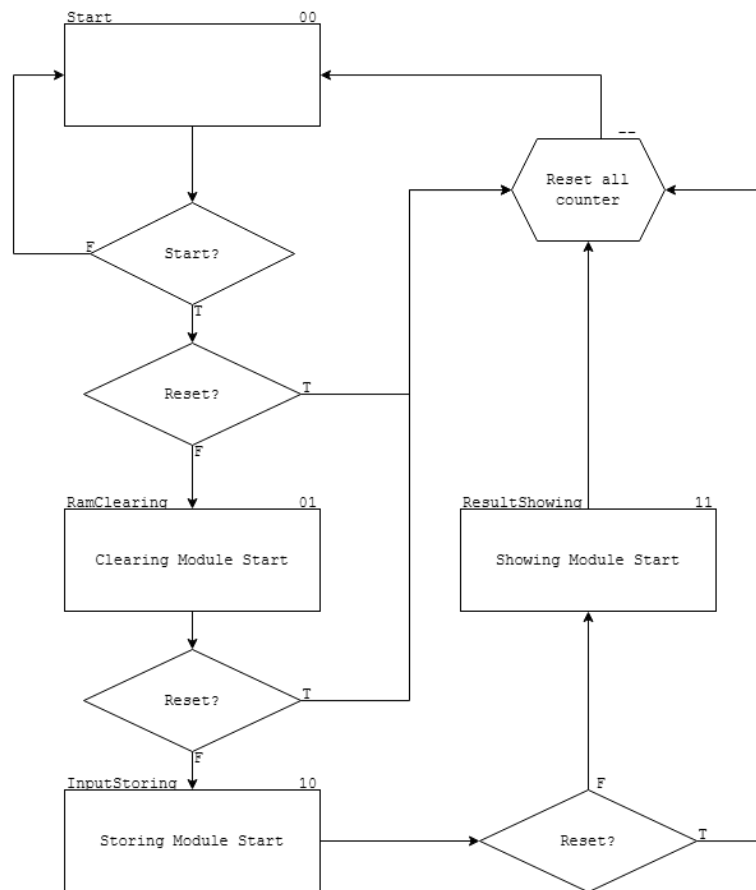


Sorting Circuit

- ASM Chart



- แนวคิด และวิธีทำ

แนวคิด: โจทย์บอกต้องเรียงข้อมูลที่อยู่ใน ROM หากนำจำนวนของเลขนั้นๆ ที่อยู่ใน ROM มาเก็บใน RAM ได้ ก็จะสามารถเรียงข้อมูลได้อย่างง่าย เหมือนหาข้อมูลแล้วเก็บจำนวนลงใน Array แล้วไล่หาทีละ จำนวน ว่ามีกี่ตัวแล้วโชว์ออกมา

วิธีทำ: วงจรในรูปจะใช้ MUX เพื่อใส่อินพุต เพื่อให้ตรงตามจุดประสงค์ของ State และเป็นวิธีที่ง่ายที่สุด ในการแยกหน้าที่แต่ละวงจร

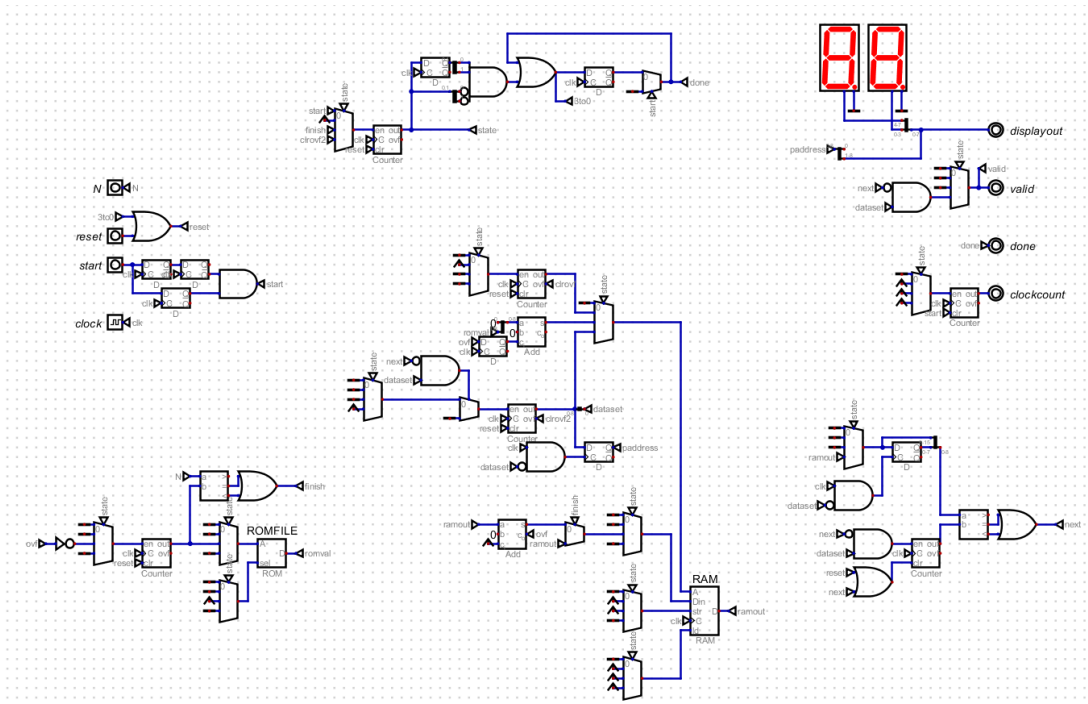
- State 00 คือ State ที่รอ Input Start แล้วเริ่มย้ายไปที่ State 01
- State 01 คือ State ที่ใช้ Clear ค่าแรมทุก Address ให้เป็น 0 (ทราบภายหลังว่าไม่จำเป็น)
- State 10 คือ State ที่ใช้ดูค่าใน ROM ตามจำนวน N แล้วบวกทีละ 1 ใส่ใน RAM
- State 11 คือ State ไล่ดูทีละ Address ว่าแต่ละเลขจำนวนเท่าไรแล้วแสดงผล

ปัญหาที่พบ

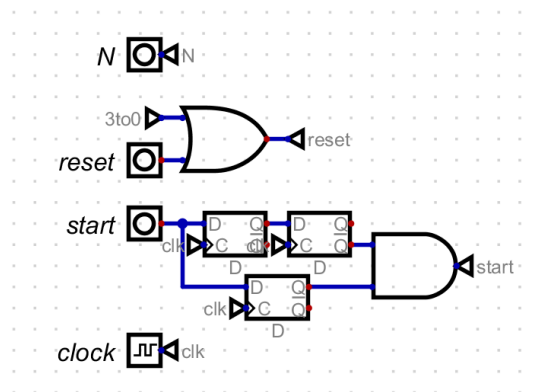
- จำนวนตัวเลขมากที่สุดที่ใส่ได้คือ 511 ตัว แต่แรมเก็บค่าได้สูงสุด 255 แต่โชคดีที่ RAM มี 9 บิต ส่วนเลขสูงสุดที่พบคือ 8 บิต วิธีแก้ไขคือเก็บไปที่ละสองช่อง หากช่องแรกเต็มค่อยไปเติม Overflow ที่ Address ถัดไป

- ตอนแสดงค่า ตอนดึงจำนวนตัวเลขออกจากแรมจะดึงได้ทีละ Address เลยทำให้ต้องอ่าน 2 clock วิธีแก้คือ ใช้ D-FlipFlop เก็บค่าแรกไว้ก่อน พอค่าออกมาครบให้สัญญาณ DataSet จึงค่อยแสดงผล เช่น มี 0 511 ตัว ข้อมูลจะถูกเก็บแยกเป็น 00000001 11111111 จึงต้องอ่าน 11111111 เก็บไว้ก่อนแล้วรอ 00000001 มาตาม หลังจากนั้นค่อยแสดงค่า valid

- Circuit

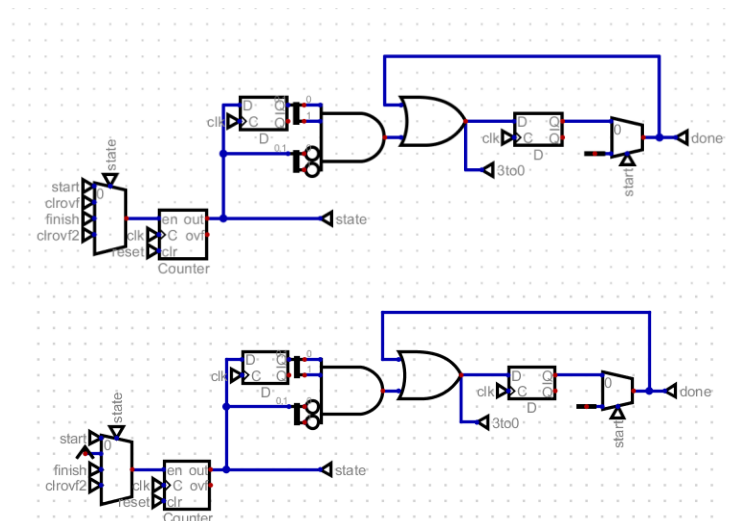


Input



มี Single pulser ที่ Start และหาก พบว่า State จาก 3 ไป 0 ให้ Reset ด้วย

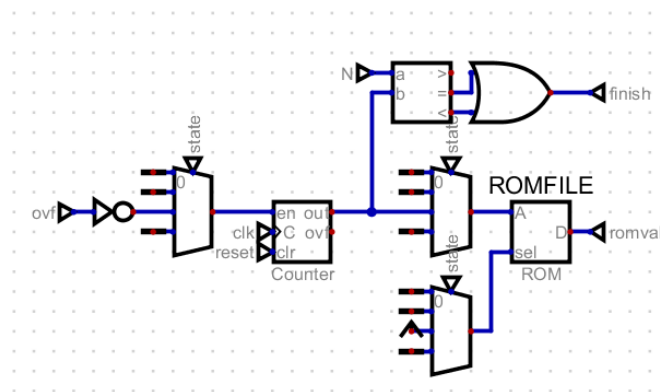
State



ใช้ Counter ในการนับ State และส่วนของข้างบนคือตัวชี้ค่า State 3 ไป 0 ใหม่

ทราบภายหลังว่าไม่ต้อง Clear Ram ก่อนจึงสามารถแทน 01 ด้วย High ได้เพื่อข้ามขั้นตอนตามภาพ

ROM Managing Module

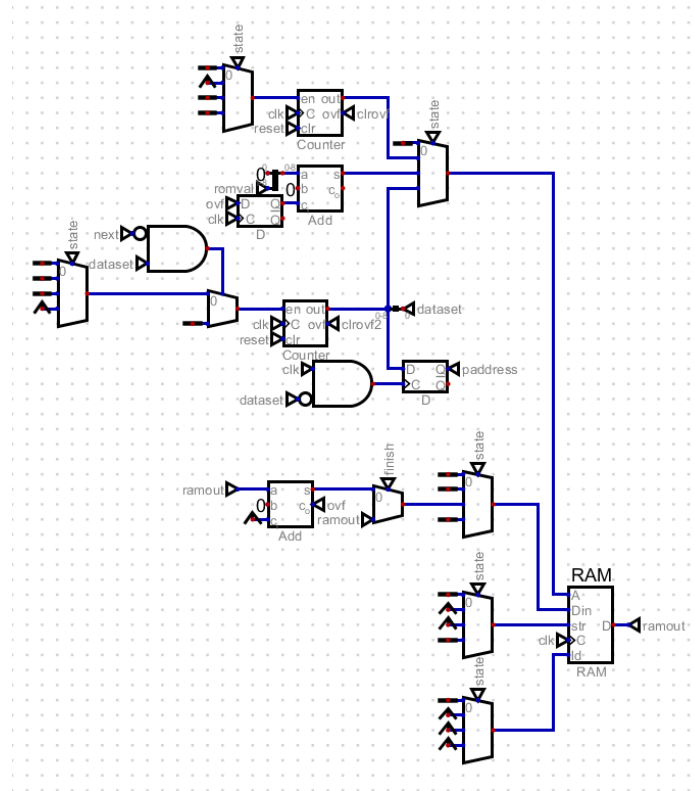


ตรง Mux ตรงตัวเพราะ State 2 มีการดึงค่าจาก ROM State เดียว และหาก Address Overflow ให้หยุดการอ่านค่าต่อไปก่อน ให้บวก Address ของ Ram อันถัดไปให้เสร็จแล้วค่อยค่าต่อไป

ส่วนของข้างบน คือตัวนับว่าอ่านถึงค่าที่ N หรือยัง ถ้าถึงแล้วให้เปลี่ยนเป็น State ถัดไปที่ State

Machine

RAM Managing Module



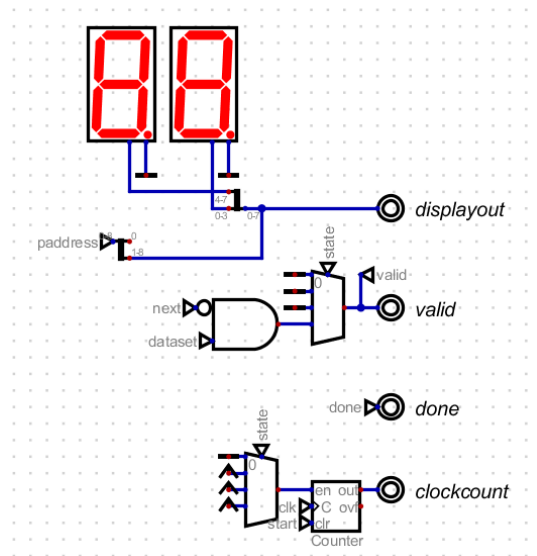
State ที่มีการเขียนคือ 1,2 State ที่มีการอ่านคือ 2,3

Din จะรับค่าจากแรมเข้ามาแล้วบวก 1 จากค่าก่อนหน้าหากค่าก่อนหน้า เป็น FF หมายความว่าหากบวกอีก 1 จะ Overflow ดังนั้นจะให้ Overflow เปลี่ยนเป็น 00 แทนแล้วค่อย บวก Address หน้า โดยการส่งสัญญาณ ovf ไปหาส่วนจัดการสัญญาณ A ให้บวก Address ไป 1

A จะอ่านค่าจะ romval แล้วเติม 0 ด้านหลังเพื่อเขียน Address ที่ละ 2 หาก ovf เข้ามาจะบวก Address ไป 1 ส่วนตอน State 3 แสดงผลตอน Dataset และยังแสดงผลยังไม่ครบจำนวน

paddress คือ ค่า address เอาไว้หารสองจะได้ตัวเลข

ส่วนแสดงผล



displayout เอา paddress มาเอาบิตตัวสุดท้ายออกจะได้เลขตามต้องการ

valid คือเช็คค่า dataset และยังไม่ครบ

done คือใช้ค่าจะ state วิ่งจาก 3 ไป 0 จะแปลว่าเสร็จ

clockcount ให้นับตอนอยู่ State 1,2,3