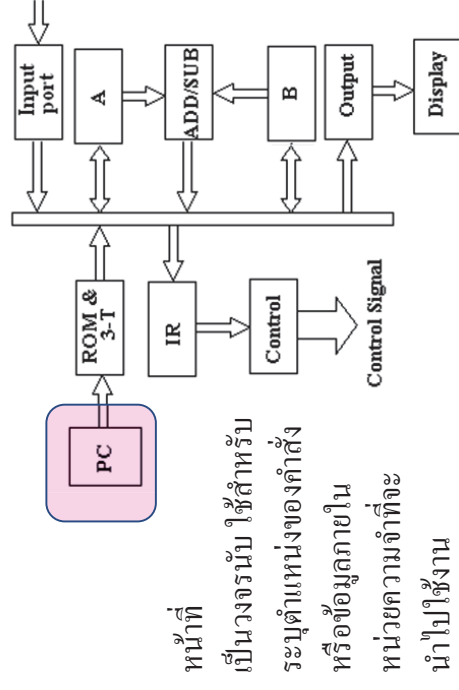


โปรแกรมเคาน์เตอร์ ใน SiCo (Program Counter in SiCo)



▶ 2

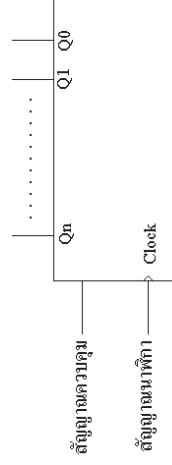
โปรแกรมเคาน์เตอร์และหน่วยความจำรวม
Program Counter และ ROM

1

วงจรมีค่า (Counter)

วงจรมีค่าเป็น FSM แบบหนึ่ง สัญญาณเอาต์พุตของวงจรมีค่าได้จาก
สัญญาณเอาต์พุตของฟลิปฟล็อปโดยตรง วงจรมีค่าแบ่งออกได้เป็น 2 แบบ
คือ Synchronous Counter และ Asynchronous Counter หรือ Ripple Counter

Synchronous Counter ลักษณะวงจรมีค่าจะเป็นตามรูปที่ 8.16 โดยสัญญาณ
นาฬิกาจะป้อนให้แก่ Clock ของฟลิปฟล็อปทุกตัว



▶ 3

การออกแบบวงจรมีค่าก็มีลำดับขั้นเหมือนกับวิธีการออกแบบ FSM คือมี
ขั้นตอนดังนี้

1. เขียน State Diagram
2. เขียน State Table
3. หาจำนวนฟลิปฟล็อป n โดย $2^n \geq$ ค่าสูงสุดที่จะนับได้ และกำหนด
ชนิดของฟลิปฟล็อป
4. หาฟังก์ชันอินพุตของฟลิปฟล็อปแบบแต่ละตัว
5. เขียนวงจร

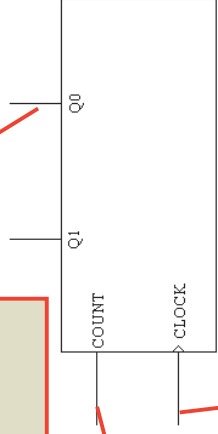
▶ 4

ตัวอย่างที่ 8.6 จงออกแบบวงจรนับจำนวนแบบเลขไบนารี ขนาด 2 บิต

กำหนดให้มีสัญญาณควบคุมการนับ

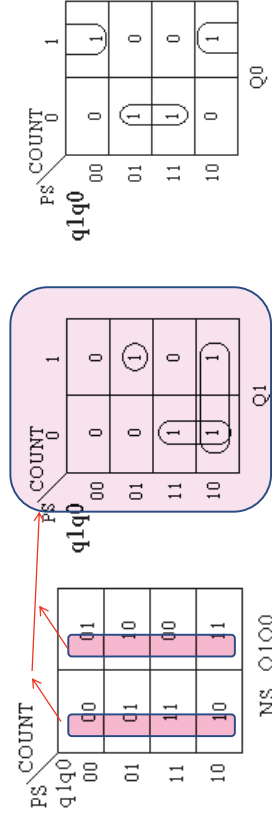
เป็นเอาต์พุตของวงจรถับ โดย Q0 เป็น บิตที่มีนัยสำคัญต่ำ (LSB) บิต Q1 เป็น บิตที่มีนัยสำคัญสูง (MSB)

เป็นสัญญาณควบคุมการนับ ถ้าเป็นลอจิก '1' จะนับสัญญาณนาฬิกา ถ้าเป็นลอจิก '0' จะหยุดนับและคงค่าสถานะเดิม



เป็นสัญญาณนาฬิกา

แยกตารางออกออกเป็น Q1 และ Q0



หาสมการ

$$Q1 = q1.COUNT + q1.q0 + \overline{q1}.q0.COUNT$$

$$Q0 = q0.COUNT + q0.COUNT$$

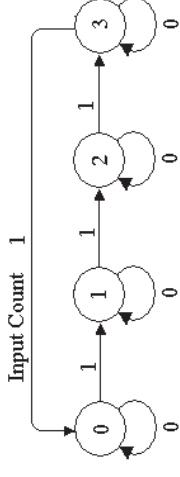
หาสมการ Input

เนื่องจากใช้ D flipflop : Input D = Output Q

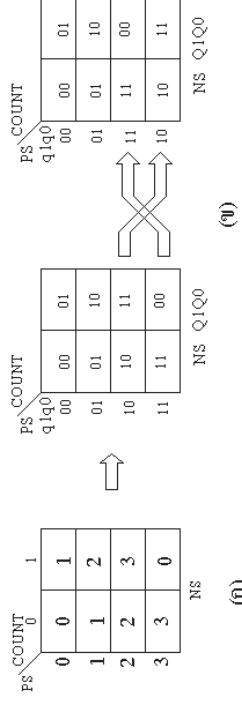
$$D1 = Q1 = q1.COUNT + q1.q0 + \overline{q1}.q0.COUNT$$

$$D0 = Q0 = q0.COUNT + \overline{q0}.COUNT$$

เขียน State Diagram

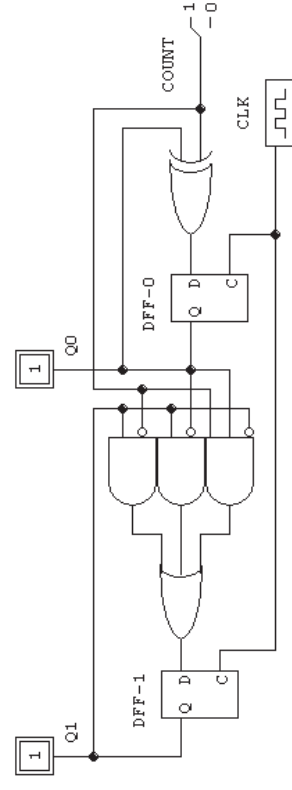


แปลงจาก State Diagram เป็นตารางการทำงาน หรือ State table

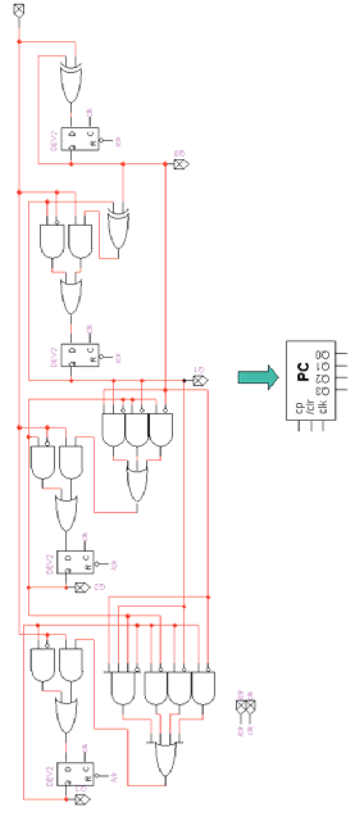


$$D1 = Q1 = q1.COUNT + q1.q0 + \overline{q1}.q0.COUNT$$

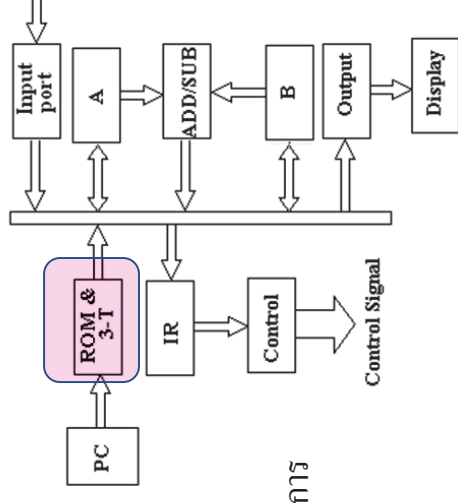
$$D0 = Q0 = q0.COUNT + \overline{q0}.COUNT$$



โปรแกรมเตาเตอร์ใน SiCo Program Counter in SiCo



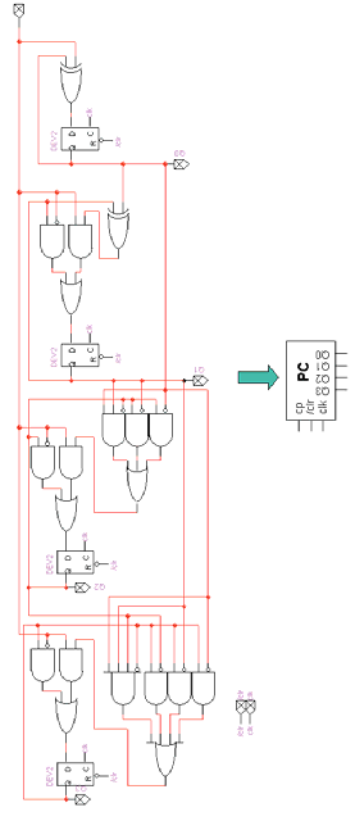
หน่วยความจำ



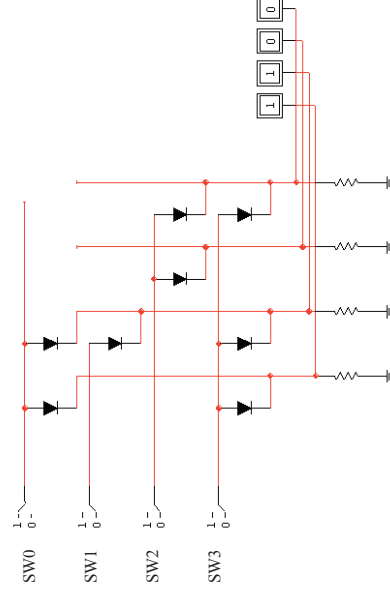
หน้าที่
เก็บคำสั่งและ
ข้อมูลสำหรับการ
ทำงานของ
คอมพิวเตอร์

โปรแกรมเตาเตอร์ใน SiCo

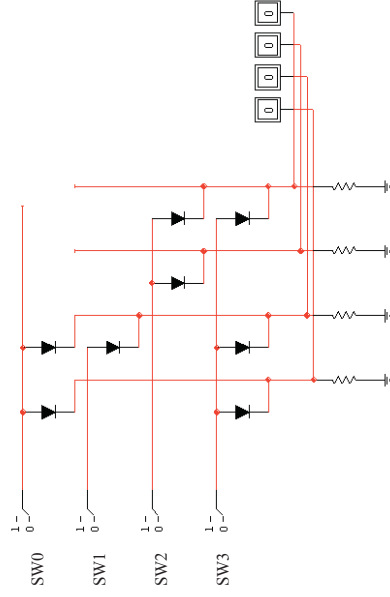
Program Counter in SiCo



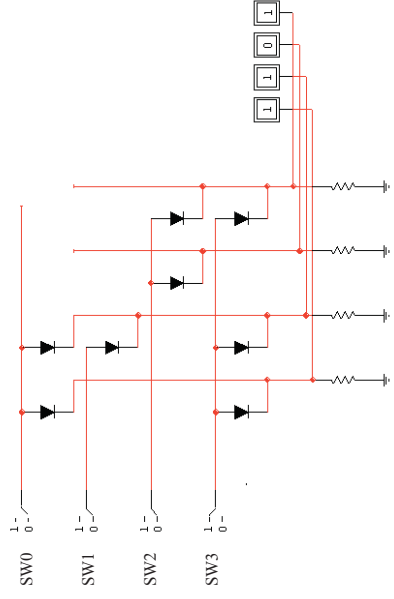
Diode ROM เมื่อ SW0 เป็น Logic 1



Diode ROM

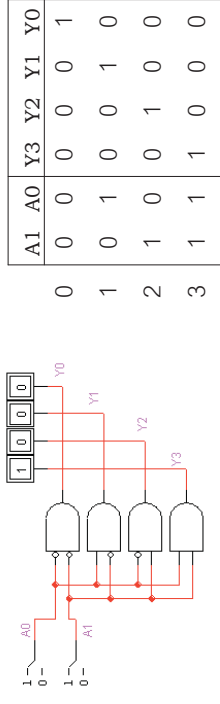


Diode ROM ដើម SW3 រឿង Logic 1



▶ 13

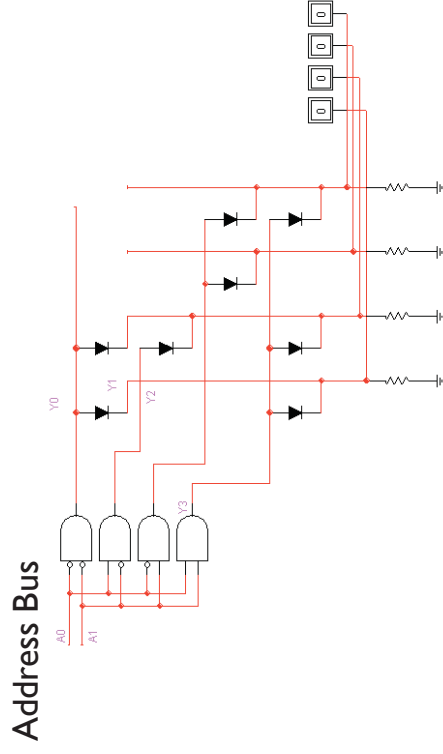
2-to-4 Line Decoder



	A1	A0	Y3	Y2	Y1	Y0
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1	1	0	0	0

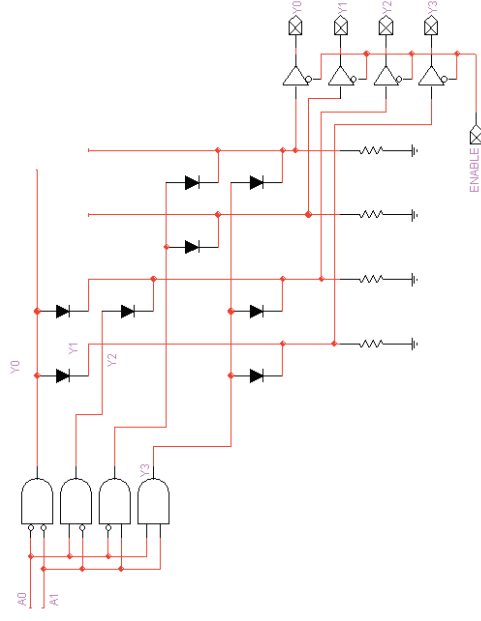
▶ 14

ROM : Address Bus



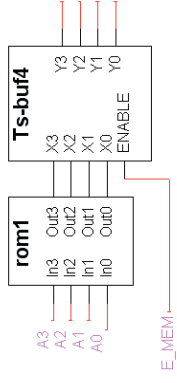
▶ 15

ROM : Output Enable



▶ 16

รอม ใน SiCo (ROM in SiCo)



- Address Bus In3-In0 มีทั้งหมด 2⁴ 16 ตำแหน่ง
- Data Bus Out3-Out0 เก็บข้อมูลตำแหน่งละ 4 บิต
- เอาต์พุตเป็นแบบ 3 สถานะ
- E_MEM = ข้อมูลออกจาก ROM ได้
- E_MEM = 1 Y3-Y0 เป็น Hi-Z