หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

แบบต่าง ๆ ของวงจรที่ใช้ในการวิจัย

การที่จะออกแบบวงจรของการทำงานแต่ละส่วนในวงจรเครื่องคำนวณอีเล็กโทรนิคขนาดเล็ก ขั้นแรกควรจะได้ศึกษาเรื่องราวและหลักเกณฑ์ของสิ่งต่อไปนี้คือ

1. ระบบเลขฐานสองหรือไบนารี่ (Binary Number System) ตลอดจนการเปลี่ยนเลขทศนิยมเป็นเลขฐานสองและโค๊ดของเลขฐานสองต่าง ๆ เช่น ระบบโค๊ด BCD (Binary CodedDecimal), โค๊ดเลขตั้งแต่สาม (Excess Three Code) และโค๊ดอื่นเป็นต้น เพราะในการหาหลักการออกแบบเครื่องคำนวณอีเล็กโทรนิค เราอาจจะใช้โค๊ดใด ๆ หรืออาจจะเป็นโค๊ดที่เราตั้งขึ้นมาเอง (Random Code) ก็ได้

2. หลักเกณฑ์ต่าง ๆ ของพีชคณิตบูลีน (Boolean Algebra) ซึ่งต้องใช้วิชาตรรกวิทยา (Logic) เข้ามาช่วย ตลอดถึงการสร้างฟังก์ชันบูลีนจากตารางจริง (Truth Table) และการทำฟังก์ชันให้เป็นรูปที่ง่าย (Simplification) โดยใช้หลักการของการเขียนตาราง (Map) ต่าง ๆ ซึ่งจากฟังก์ชันบูลีนเราสามารถเขียนวงจรโลยิคได้ตามที่ต้องการ

3. ศึกษาการทำงานและโครงสร้างของไอซีและเกตต่าง ๆ เช่น เกต AND, OR, NAND, NOR และ NOT เป็นต้น

สองหัวข้อแรกจะไม่กล่าวถึง เพราะจะหาอ่านได้จากตำราคอมพิวเตอร์โลยิค (Computer Logic) ทั่ว ๆ ไป แต่จะอ้างถึงบางครั้งเมื่อนำไปใช้ในตอนต่อไป จะกล่าวถึงรายละเอียดของวงจรต่าง ๆ ในหัวข้อที่ 3 ดังต่อไปนี้

2.1 วงจรเกต (Gating Circuits)

วงจรเกต หมายถึงวงจรที่มีอินพุทหลายอัน แต่มีเอาพุทเพียงอันเดียว เอาพุทที่ได้ออกมาก็เฉพาะเมื่อมีอินพุทใดอินพุทหนึ่งป้อนเข้าไปเท่านั้น เกตทั่ว ๆ ไปแบ่งเป็น 2 ประเภท คือ

ทรานสมิสชั่นเกต (Transmission Gates) และ

สวิทชิ่งหรือโลยิคเกต (Switching or Logic Gates)

ทรานสมิสชั่นเกต จะส่งสัญญาณอินพุทไปยังเอาพุทโดยไม่มีการเปลี่ยนแปลง จะเกิดการเปลี่ยนแปลงก็ต่อเมื่อมีสัญญาณควบคุม (Control Signals) ถูกป้อนเข้าไปในช่วงระยะเวลาใดเวลาหนึ่งเท่านั้น

สวิทชิ่งหรือโลยิคเกต จะมีเอาพุทเป็นพัลซ์หรือระดับ ดี.ซี. (D.C. Levels) ซึ่งไม่จำเป็นจะต้องเหมือนกับอินพุทก็ได้ เกตแบบนี้เป็นเช่นเดียวกับแบบที่แล้ว คือเวลาที่ปรากฏของสัญญาณเอาพุทจะถูกกำหนดหรือควบคุมโดยสัญญาณอินพุทหลายอัน

เกตุที่ใช้ในการออกแบบวงจรเครื่องคำนวณอีเล็กโทรนิคขนาดเล็กนี้เป็นแบบโลยิคเกต ซึ่งจะทำงานโดยอาศัยสัญญาณ 2 ระดับคือ "TRUE" กับ "FAISE" หรือ "ON" กับ "OFF" และแทนด้วยเลขไบนารี่ (เลขฐานสอง) คือ 1 กับ 0 โดยกำหนดให้

ระดับที่เป็นบวกที่สุดของสัญญาณกำหนดให้เป็น 1 หรือ "TRUE"

ระดับที่เป็นลบที่สุดของสัญญาณกำหนดให้เป็น 0 หรือ "FALSE"

โลยิคเกตมีหลายประเภท คือแบ่งตามชนิดของอุปกรณ์อีเล็กโทรนิคส์ที่นำมาใช้สร้างเป็นเกต ได้แก่

ไดโอดโลยิคเกต (Diode Logic Gates)

ทรานซิสเตอร์-ทรานซิสเตอร์โลยิคเกต (Transistor-Transistor Logic Gate หรือ TTL)

ไดโอด-ทรานซิสเตอร์โลยิคเกต (Diodo - Transistor Logic Gates หรือ DTL)

รีซิสเตอร์-ทรานซิสเตอร์โลยิคเกต (Resistor - Transistor Logic Gates หรือ RTL)

ทรานซิสเตอร์-อินเวอร์เตอร์ (Transistor-Inverter) หรือ NOT เกต

เกตต่าง ๆ เหล่านี้สามารถสร้างให้เป็นแบบของโลยิค AND, OR, NAND และ NOR ได้ ซึ่งชื่อเหล่านี้เป็นชื่อตามลักษณะการทำงานของมัน

ก่อนที่จะถึงแบบของวงจรต่าง ๆ ขอกล่าวถึงโค๊ดของการเปลี่ยนเลขทศนิยม (Decimal Code) เป็นเลขฐานสองซึ่งนำไปใช้ในวงจรเหล่านั้น

2.2 โค๊ดของเลขทศนิยม (Decimal Code)[1]

การเปลี่ยนเลขทศนิยมเป็นเลขฐานสอง ต้องการบิทอย่างน้อย 4 บิท โดยจัดเป็นโค๊ดต่าง ๆ กัน 10 คอมบิเนชัน (Combinations) จากเลข 0 ถึง เลข 9 และอาจจัดเป็นโค๊ดต่าง ๆ ดังเช่นแบบของโค๊ดในตารางที่ 1

โค๊ดบีซีดี (Binary Coded Decimal) เป็นโค๊ดที่นิยมใช้กันมาก เป็นโค๊ดของการเปลี่ยนเลขฐานสองตามตำแหน่งโดยตรง

---

1. Mano, M. Moris; 1972. Computer Logic Design. (New Jersey: Prentice-Hall), p. 16-18.

เลขทศนิยม	บี.ซี.ดี. 8421	เอ็กเซส-3	84-2-1	2421	ไบควินารี่ 5043210
0	0000	0011	0000	0000	0100001
2	0001	0100	0111	0001	0100010
2	0010	0101	0110	0010	0100100
3	0011	0110	0101	0011	0101000
4	0100	0111	0100	0100	0110000
5	0101	1000	1011	1011	1000001
6	0110	1001	1010	1100	1000010
7	0111	1010	1001	1101	1000100
8	1000	1011	1000	1110	1001000
9	1001	1100	1111	1111	1010000

ตารางที่ 1 การโค๊ดเลขทศนิยมเป็นเลขฐานสอง

ตำแหน่งของโค๊ดบีซีดี คือ 8,4,2,1 เช่นบิทที่มีค่าเป็น 0110 สามารถเปลี่ยนเป็นเลขทศนิยมได้คือ 0x8+1x4+1x2+0x1 = 6 การวิจัยครั้งนี้ใช้โค๊ดบีซีดีเป็นโค๊ดเปลี่ยนเลขทศนิยมเป็นเลขฐานสอง

สมัยก่อนการโค๊ดเลขทศนิยมในเครื่องคอมพิวเตอร์ใช้การโค๊ดระบบเอ็กเซส-3 (Excess - 3 Code) โค๊ดนี้ไม่สามารถเปลี่ยนเป็นเลขทศนิยมได้โดยตรงเหมือนโค๊ดบีซีดี เพียงแต่เป็นค่าเพิ่มค่าจากโค๊ดบีซีดีด้วยการบวกค่าอีก 3 โค๊ดอื่น ๆ ในตารางก็เช่นเดียวกัน คือเราสามารถออกแบบวงจรเปลี่ยนเลขทศนิยมเป็นเลขฐานสอง และเปลี่ยนกลับโดยได้ค่าเลขทศนิยมที่ถูกต้องตามโค๊ดนั้น ๆ ได้

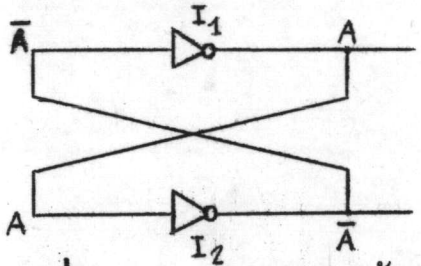
จำนวนเลขฐานสอง (Binary Number System) กับจำนวนเลขในโค๊ดบีซีดี

จะไม่เหมือนกัน ตัวอย่างเช่น

เลขทศนิยม 395 เมื่อเปลี่ยนเป็นเลขฐานสอง คือค่า 110001011 ซึ่งประกอบด้วยเลขฐานสอง 9 ตัว แต่ตัวเลขจำนวนเดียวกันนี้ถ้าเขียนตามโค้ดบีซีดี ซึ่งเลขทศนิยมแต่ละตัวต้องใช้ 4 บิท รวมเป็น 12 บิทเรียงกันคือ 001110010101, 4 บิทแรกแทนเลข 3, 4 บิทต่อมาแทนเลข 9 และ 4 บิทหลังแทนเลข 5 เป็นการนำโค้ดของเลขทศนิยมแต่ละตัวมาต่อเรียงกันตามตำแหน่งเท่านั้น

2.3 ไบสเตเบิลมัลติไวเบรเตอร์ หรือวงจร ฟลิพ-ฟล็อพ[2]
(Bistable Multivibrator or Flip-Flop Circuit)

วงจรไบสเตเบิลมัลติไวเบรเตอร์หรือเรียกชื่อทั่ว ๆ ไปว่าวงจร "ฟลิพ-ฟล็อพ" เป็นวงจรมัลติไวเบรเตอร์ที่สำคัญที่สุด แบบง่าย ๆ ของการทำงานของวงจรมีแสดงในรูป



ในรูปใช้อินเวอร์เตอร์ (Inverters) 2 ตัวต่อกันเป็นแบบอนุกรม มีเอาพุทอยู่ 2 ทางคือ เอาพุท 1 และเอาพุท 0

รูปที่ 1 วงจรฟลิพ-ฟล็อพใช้อินเวอร์เตอร์แสดงการทำงาน

ถ้าเอาพุท A เป็นโลยิค - 1 อินเวอร์เตอร์ I_2 จะทำให้ $\bar{A}$ เป็นโลยิค -0 และอินเวอร์เตอร์ I_1 ทำให้อินพุทจาก $\bar{A}$ เป็นเอาพุทที่ A เป็นโลยิค -1 ดังนั้นภาวะที่เกิดขึ้นจึงเป็น

A มีเอาพุท เป็นโลยิค -1
$\bar{A}$ มีเอาพุท เป็นโลยิค -0

---

2. Nashelsky, Louis; 1972. Introduction to Digital Computer Technology. (New York: John Wiley and Sons), p. 271.

ภาวะเช่นนี้เรียกว่า ภาวะ "เซ็ท" (Set State) ถ้ามีแฟคเตอร์ภายนอกมาทำให้ A เปลี่ยนภาวะเป็นโลยิค -0 เมื่อผ่านอินเวอร์เตอร์ I_2 A จะถูกเปลี่ยนเป็นโลยิค -1 และอินเวอร์เตอร์ I_1 จะทำให้อินพุท $\bar{A}$ ซึ่งมีโลยิค -1 เปลี่ยนเป็นโลยิค -0 ดังนั้นวงจรจะอยู่ในภาวะคือ

A มีเอ๋าพุท เป็นโลยิค -0

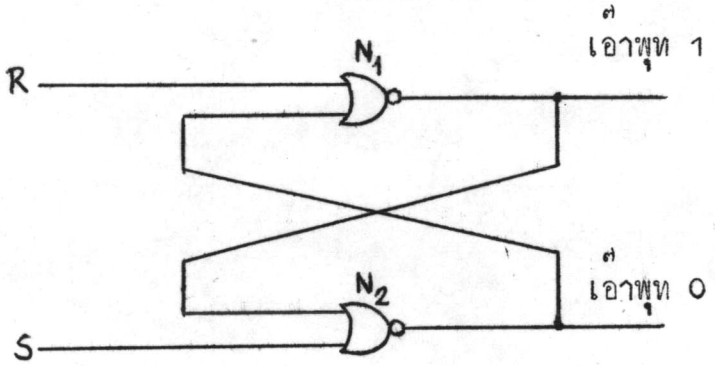
$\bar{A}$ มีเอ๋าพุท เป็นโลยิค -1

ภาวะเช่นนี้เรียกว่า ภาวะ "รีเซ็ท" (Reset State)

สรุปแล้ว วงจรจะมีการทำงานอยู่ 2 ภาวะคือ

1. ภาวะ "เซ็ท" คือเอ๋าพุท 1 มีโลยิคเป็น 1 และเอ๋าพุท 0 มีโลยิคเป็น 0

2.ภาวะ "รีเซ็ท" คือเอ๋าพุท 1 มีโลยิคเป็น 0 และเอ๋าพุท 0 มีโลยิคเป็น 1



R	S	Q	$\bar{Q}$
0	0	ไม่เปลี่ยน	
0	1	1	0
1	0	0	1
1	1	ไม่ต้องการ	

รูปที่ 2 แสดงการเซ็ทและรีเซ็ทของวงจรฟลิพ-ฟล็อพและตารางจริง

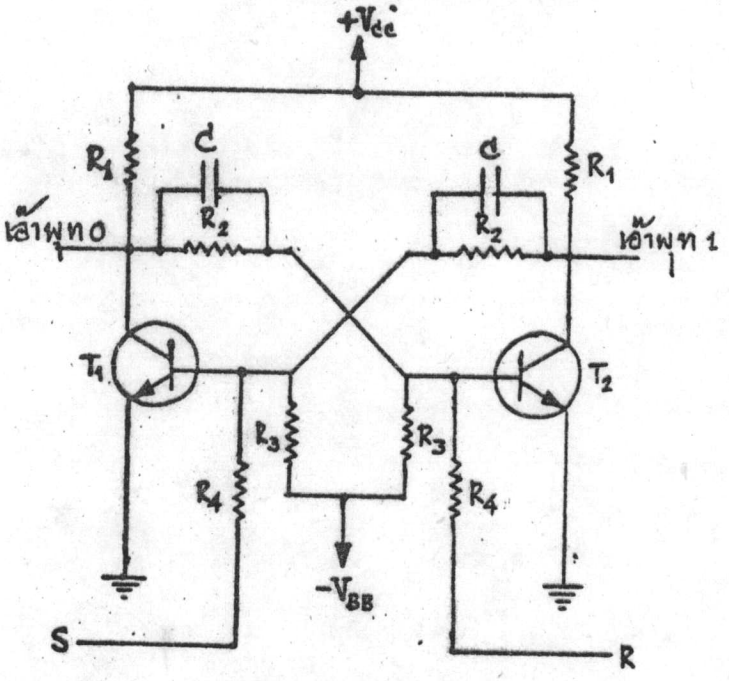
ตามรูปที่ 2 ต่อ NOR เกต เป็นแบบอนุกรมกับอินพุท R (รีเซ็ท) และ S (เซ็ท) เนื่องจาก NOR เกตจะทำให้เอ๋าพุทเป็นโลยิค -0 เมื่อมีอินพุทอันใดอันหนึ่งเป็นโลยิค -1 เมื่อปลาย S มีอินพุทเป็นโลยิค -1 จะทำให้เอ๋าพุทของ N_2 เป็นโลยิค -0 สมมติว่าขณะนี้ไม่มีอินพุทต่อทางปลาย R ดังนั้นอินพุทของ N_1 จะเป็นโลยิค -0 ทั้งคู่ ซึ่งจะทำให้ได้เอ๋าพุทของ N_1 เป็นโลยิค -1 ดังนั้น ผลของการเซ็ทจะทำให้วงจรอยู่ในภาวะ

"เซ็ท" (คือ เอาพุท 1 มีโลยิค –1 และ เอาพุท 0 มีโลยิค –0)

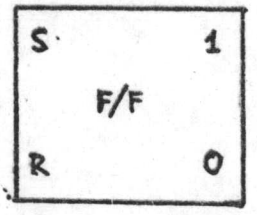
แต่ถ้าทางอินพุท R มีโลยิค –1 จะทำให้เอาพุท 1 เป็นโลยิค –0 และเอาพุท 0 เป็นโลยิค –1 . ซึ่งเป็นภาวะ "รีเซ็ท" ของวงจร

ฟลิพ–ฟล็อพ RS (RS Flip-Flop)

วงจร ฟลิพ–ฟล็อพ เบื้องต้น คือ RS หรือ รีเซ็ท–เซ็ท ฟลิพ–ฟล็อพ



(ก)



(ข)

รูปที่ 3 ฟลิพ–ฟล็อพ RS (ก) วงจรฟลิพ–ฟล็อพ RS
(ข) บล็อคไดอะแกรม

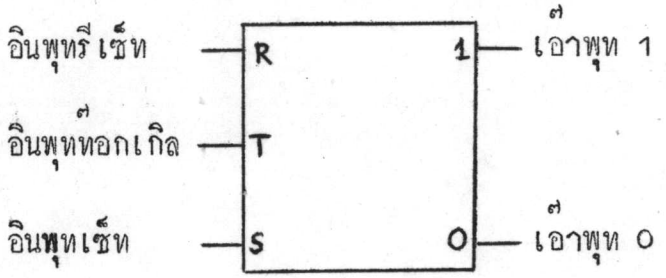
ในรูป (ก) เมื่อป้อนสัญญาณอินพุทพัลซ์ ซึ่งมีโลยิค –1 เข้าที่ปลายอินพุท S จะทำให้วงจรอยู่ในภาวะ "เซ็ท" คือ เอาพุท –1 เป็นโลยิค –1 และ เอาพุท 0 เป็นโลยิค –0 แต่ถ้าป้อนสัญญาณอินพุทพัลซ์ โลยิค –1 เข้าทางปลายอินพุท R จะไดร์ฟให้ทรานซิสเตอร์ T_2 คอนดัคท์ (Conduct หรือ ON) และทำให้ทรานซิสเตอร์ T_1 คัตออฟ (Cutoff หรือ OFF) ทำให้วงจรอยู่ในภาวะ "รีเซ็ท" คือ เอาพุท 1 มีโลยิค –0 และ เอาพุท 0 มีโลยิค –1

ภาวะแรกคือ ภาวะเซ็ท เป็นภาวะที่ทรานซิสเตอร์ T_1 คอนดัคท์ และทรานซิสเตอร์ T_2 คัตออฟ เรียกว่า "ฟลิพสเตท" (Flip State)

ภาวะที่สองคือ ภาวะ รีเซ็ท เป็นภาวะที่ทรานซิสเตอร์ T_2 คอนดัคท์ และทรานซิสเตอร์ T_1 คัตออฟ เรียกว่า "ฟล็อพสเตท" (Flop State)

ฟลิพ-ฟล็อพ RST (RST Flip-Flop)

วงจรมัลติไวเบรเตอร์ที่รวมการทำงานของการ เซ็ท (Set) รีเซ็ท (Reset) และทอกเกิล (Toggle) อยู่ด้วยกัน เรียกว่า ฟลิพ-ฟล็อพ RST



หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

รูปที่ 4 บล็อคไดอะแกรมของ ฟลิพ-ฟล็อพ RST

ในรูปที่ 4 อินพุท R ทำให้ฟลิพ-ฟล็อพอยู่ในภาวะ รีเซ็ท และอินพุท S ทำให้ฟลิพ-ฟล็อพอยู่ในภาวะเซ็ท และอินพุท T ทำให้ฟลิพ-ฟล็อพ อยู่ในภาวะ ทอกเกิล คือมีเอาพุทเปลี่ยนไปเป็นตรงข้ามจากเดิม (Complement or Toggle)

เมื่อป้อนพัลซ์อินพุทโลยิค -1 เข้าทางปลายอินพุท R วงจรจะอยู่ในภาวะรีเซ็ทตลอดเวลาที่สัญญาณอินพุทโลยิค -1 ยังคงอยู่ที่ปลาย R เรียกว่า วงจรล็อค (Locked) อยู่ในภาวะรีเซ็ท และถึงแม้จะมีสัญญาณทอกเกิลป้อนที่อินพุท T อยู่ด้วยก็ตาม วงจรก็ยังคงอยู่ในภาวะรีเซ็ทเหมือนเดิม ทำนองเดียวกัน ถ้ามีสัญญาณอินพุทโลยิค -1 เข้าทางปลาย S ทางเดียวเท่านั้น วงจรก็จะล็อคอยู่ในภาวะเซ็ท

แต่ถ้าปลาย R และ S ได้รับสัญญาณอินพุทโลยิค -0 ทั้งคู่ สัญญาณอินพุท T จึงจะทำงานได้ คือ อินพุท ทอกเกิล ทางปลาย T จะเปลี่ยนโลยิคของเอาพุทของฟลิพ-ฟล็อพเป็นค่าตรงข้าม

001579

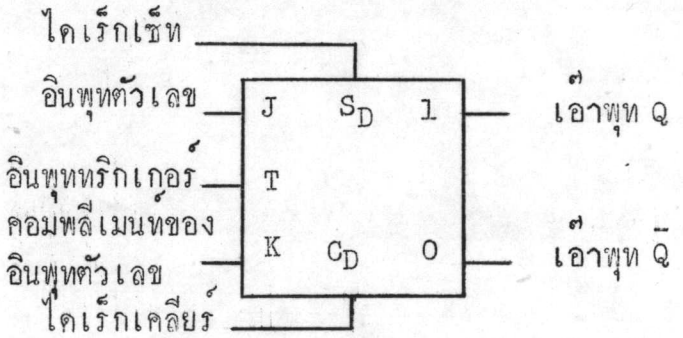
ฟลิพ-ฟล็อพ JK แบบโลยิค TTL (หรือ DTL)

(Transistor-Transistor Logic Type Flip-Flop)

ฟลิพ-ฟล็อพแบบนี้เป็น ฟลิพ-ฟล็อพ ICs ที่นิยมใช้กันมาก และสามารถทำงานได้ 2 แบบ คือ

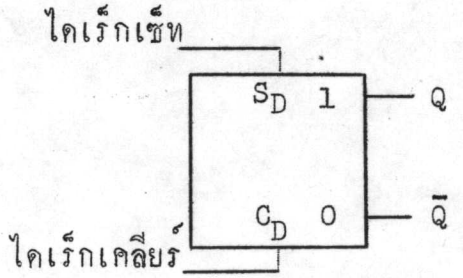
1. แบบอซินโครนัส (Asynchronous Operation) คือฟลิพ-ฟล็อพสามารถจะเปลี่ยนภาวะเมื่อเปลี่ยนโลยิคของสัญญาณ อินพุท

2. แบบซินโครนัส (Synchronous Operation) คือ ฟลิพ-ฟล็อพ จะเปลี่ยนภาวะเมื่อเกิดการซินโครไนซ์ (Synchronize) หรือ เมื่อมีพัลซ์ทริกเกอร์ (Trigger Pulse) เท่านั้น



รูปที่ 5 บล็อคไดอะแกรมของฟลิพ-ฟล็อพ JK

ตามรูปที่ 5 อินพุททุกทางจะไม่ใช้พร้อม ๆ กัน รูปที่ 6 แสดงการทำงานแบบอซินโครนัส (Asynchronous Operation)



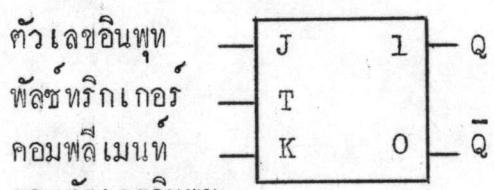
S_D	C_D	Q	$\bar{Q}$	ภาวะ
Lo	Lo	Hi	Hi	ภาวะไม่ต้องการ
Lo	Hi	Hi	Lo	ภาวะเซ็ท
Hi	Lo	Lo	Hi	ภาวะเคลียร์
Hi	Hi	-	-	ไม่เปลี่ยนภาวะ

รูปที่ 6 การทำงานแบบอซินโครนัสของฟลิพ-ฟล็อพ JK

เมื่ออินพุท S_d มีค่า โวลเตจ ต่ำ (Lo คือมีโลยิค - 0) จะทำให้เอาพุท Q มีโวลเตจสูง (Hi คือมีโลยิค -1) และ $\bar{Q}$ มีโวลเตจต่ำ (Lo) ทำให้วงจรอยู่ในภาวะเซ็ท แต่เมื่ออินพุท C_d มีโวลเตจต่ำ (Lo) จะทำให้ เอาพุท $\bar{Q}$ มีโวลเตจสูง (Hi) และ Q มีโวลเตจต่ำ (Lo) ทำให้วงจรอยู่ในภาวะรีเซ็ท หรือเคลียร์

ถ้าอินพุท S_d และ C_d มีโวลเตจสูง (Hi) ทั้งคู่ วงจรจะไม่เปลี่ยนภาวะ ถ้าอินพุทที่เข้าปลายทั้งสองมีโวลเตจต่ำทั้งคู่ ชั่วขณะหนึ่ง แล้วโวลเตจขึ้นสูง เป็นสภาพที่วงจรอยู่ในลักษณะ "ฟรี" (Free) ซึ่งเป็นภาวะที่ไม่ต้องการ เพราะเอาพุท Q และ $\bar{Q}$ มีโวลเตจสูงทั้งคู่

รูปที่ 7 แสดงการทำงานแบบซินโครนัส (Synchronous Operation) ของวงจร



J	K	Q	$\bar{Q}$	ภาวะ
Lo	Lo	-	-	ไม่เปลี่ยนภาวะ
Lo	Hi	Lo	Hi	ภาวะเคลียร์
Hi	Lo	Hi	Lo	ภาวะเซ็ท
Hi	Hi	ทอกเกิล (เอาพุทเปลี่ยนภาวะ)		

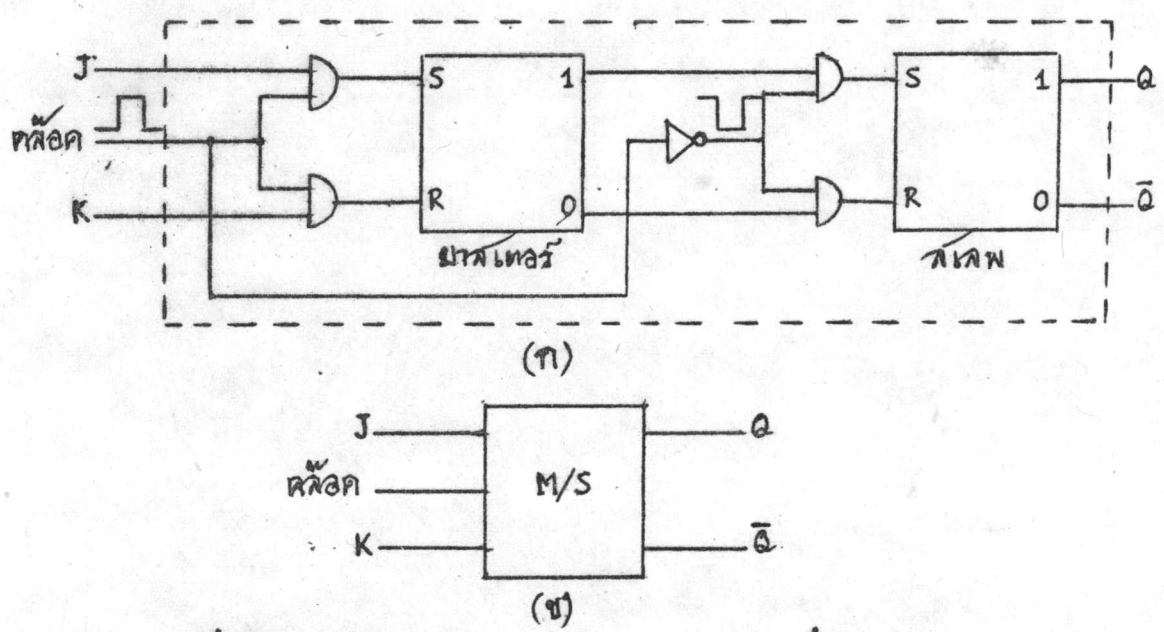
รูปที่ 7 การทำงานแบบซินโครนัสของฟลิพ-ฟล็อพ JK

ตามรูปตัวเลขอินพุท เข้าทางปลาย J และ K ของฟลิพ-พล็อพ ถ้าอินพุททาง J มีโวลเตจต่ำ (Lo) ทาง K มีโวลเตจสูง (Hi) เอาพุทจะทำให้ฟลิพ-ฟล็อพ มีภาวะเคลียร์ หรือรีเซ็ท เมื่อมีพัลซ์ทริกเกอร์ เกิดขึ้น แต่ถ้าอินพุทของ J มี โวลเตจสูง และ K มีโวลเตจต่ำ เอาพุทจะทำให้ฟลิพ-ฟล็อพ อยู่ในภาวะเซ็ท ขณะเมื่อมีพัลซ์ทรีกเกอร์ เช่นเดียวกัน

ถ้าอินพุท J และ K มีโวลเตจต่ำทั้งคู่ พัลซ์ของทริกเกอร์จะไม่มีผลในการเปลี่ยนภาวะของฟลิพ-ฟล็อพ แต่ถ้าอินพุท J และ K มีโวลเตจสูงทั้งคู่ พัลซ์ของทริกเกอร์จะมีผลในการเปลี่ยนภาวะ (ท็อกเกิล) ของฟลิพ-ฟล็อพทุก ๆ ครั้งที่มีพัลซ์ของทริกเกอร์เกิดขึ้น

ฟลิพ-ฟล็อพ มาสเตอร์-สเลพ (Master-Slave Flip-Flop)[3]

วงจร ICs ฟลิพ-ฟล็อพ ที่นิยมใช้กันมากคือ มาสเตอร์-สเลพ (M/S) ซึ่งต่อกันเป็นทั้ง ฟลิพ-ฟล็อพ RS หรือ JK วงจร M/S ประกอบด้วยฟลิพ-ฟล็อพ 2 อย่างต่างชนิดกัน อันหนึ่งรับตัวเลขอินพุทให้ชื่อเป็น ฟลิพ-ฟล็อพ มาสเตอร์ และอีกอันหนึ่งซึ่งเป็นตัวทำงานต่อจาก ฟลิพ-ฟล็อพ มาสเตอร์ ให้ชื่อเป็น ฟลิพ-ฟล็อพ สเลพ เอาพุท ของวงจรออกทาง ฟลิพ-ฟล็อพ สเลพ



รูปที่ 8 ฟลิพ-ฟล็อพ มาสเตอร์-สเลพ (ก) การต่อวงจร
(ข) โลยิคไดอะแกรม

---

[3] Nashelsky, Louis; 1972. Introduction to Digital Computer Technology. (New York : John Wiley and Sons), p. 289.

ตามรูปที่ 8 มีอินพุท J และ K และสัญญาณคล็อค เอ้าพุทเป็น Q และ $\bar{Q}$ สมมติว่า เอ้าพุท Q และ $\bar{Q}$ มีค่าเป็นโลยิค -0 และโลยิค -1 ตามลำดับ ซึ่งฟลิพ-พล็อพจะอยู่ในภาวะรีเซ็ท หรือเคลียร์ อินพุท J และ K มีค่าเป็นโลยิค -1 และโลยิค -0 ตามลำดับ ซึ่งเมื่อมีพัลซ์คล็อคเกิดขึ้นจะทำให้วงจรอยู่ในภาวะ "เซ็ท" ดังนั้น สภาพก่อนที่จะมีพัลซ์คล็อคเกิดขึ้น คือ

ฟลิพ-ฟล็อพมาสเตอร์ อยู่ในภาวะรีเซ็ท

ฟลิพ-พล็อพสเลพอยู่ในภาวะรีเซ็ท

อินพุท J เป็น โลยิค -1 และอินพุท K เป็นโลยิค -0

เมื่อพัลซ์คล็อคเป็นระดับโลยิค -1 เกต And ที่ต่อกับอินพุท ซึ่งมีโลยิค -1 จะทำให้ฟลิพ-พล็อพมาสเตอร์อยู่ในภาวะเซ็ท แต่เมื่อพัลซ์คล็อคผ่านอินเวอร์เตอร์ จะเป็นโลยิค -0 ผ่านไปยังฟลิพ-พล็อพสเลพ แต่ไม่เกิดผลอะไร ดังนั้น ฟลิพ-ฟล็อพสเลพ จะยังคงอยู่ในภาวะ รีเซ็ท ตามเดิม

เมื่ออินพุทพัลซ์คล็อคลดระดับลงเป็น โลยิค -0 ฟลิพ-พล็อพมาสเตอร์ คงอยู่ในภาวะเซ็ท อินเวอร์เตอร์จะเปลี่ยนสัญญาณพัลซ์คล็อคเป็นระดับโลยิค -1 ทำให้ฟลิพ-ฟล็อพสเลพอยู่ในภาวะเซ็ท จึงเท่ากับเป็นการเลื่อน (Shift) ภาวะเซ็ทจากฟลิพ-ฟล็อพมาสเตอร์ไปยังฟลิพ-ฟล็อพสเลพ

เอ้าพุทของฟลิพ-ฟล็อพสเลพ ถือว่าเป็นเอ้าพุทของฟลิพ-ฟล็อพมาสเตอร์-สเลพ ดังนั้นจะเห็นว่าการเปลี่ยนภาวะของฟลิพ-ฟล็อพมาสเตอร์-สเลพ เกิดขึ้นเมื่ออินพุทของคล็อคเปลี่ยนจากระดับโลยิค -1 เป็นระดับโลยิค -0 หรือกล่าวได้ว่าการทริกเกอร์ ของฟลิพ-ฟล็อพมาสเตอร์-สเลพ เกิดขึ้นหลังจาก อินพุทคล็อคเปลี่ยนระดับโลยิค-0

ฟลิพ-ฟล็อพ ชนิดนี้นิยมใช้ในการสร้างวงจรนับ (Counter Circuit) และ ชีพ-รียิสเตอร์ (Shift-Register)

2.4 วงจรบวกเต็ม (Full-Adder Circuit)[4]

วงจรโลยิคที่สำคัญและเป็นหัวใจของการทำงานในหน่วยคำนวณ (Arithmatic Unit) ของเครื่องคอมพิวเตอร์ คือ วงจรบวกเต็ม (Full-Adder Circuit)

เมื่อจะบวกเลขฐานสอง 2 จำนวน คือ จำนวน X และ Y ผลบวกที่ได้ในแต่ละตำแหน่งของบิท ขึ้นอยู่กับค่าของเลขฐานสองที่อยู่ในจำนวน X และ Y ที่ตำแหน่งของบิทนั้น ๆ บวกกับจำนวนที่ทดจากตำแหน่งของบิทที่อยู่ถัดลงไป จะมีบิทตำแหน่งเดียวเท่านั้น ที่ไม่มีจำนวนทด คือ ตำแหน่งแรกของบิท (Unit - Position 2^{0})

ดังนั้น การบวกเต็ม (Full-Adder) จึงมีอินพุทอยู่ 3 ตัว คือ X,Y และตัวทดเข้า C_i (Carry-In) เอาพุทมีอยู่ 2 ตัว คือ เอาพุทของผลบวก S และ ตัวทดออก C_o (Carry-Out) ซึ่งจะต้องนำไปบวกเข้ากับตำแหน่งถัดขึ้นไป

X	Y	C_i	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

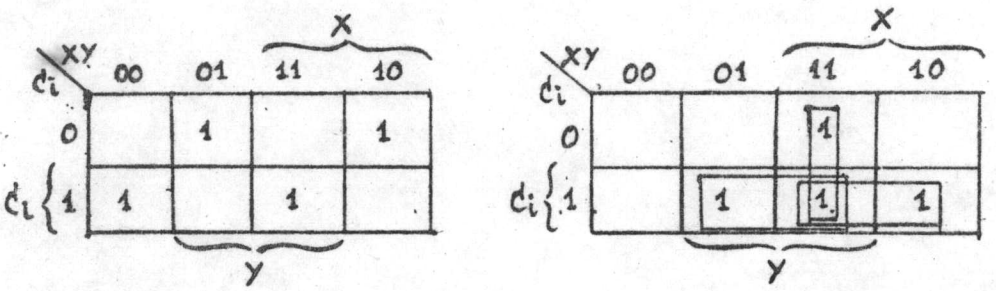
ตารางที่ 2 ตารางจริงของการบวกเต็ม

---

4 Mano M. Moris; 1972. Computer Logic Design. (New Jersey: Prentice-Hall), p. 92 - 95.

จากตารางที่ 2 จะพบว่า ค่าของผลบวก S จะเป็น 1 เมื่อจำนวนบิทที่เป็นค่า 1 ของอินพุททั้งสามมีจำนวนเป็นคี่ และค่าของตัวทดออก C_o จะเป็น 1 เมื่อมีจำนวนบิทที่เป็น 1 ของอินพุททั้งสามเป็นจำนวน 2 หรือ 3

ขั้นต่อไปคือการหา Karnaugh Maps ของวงจรบวกเต็มจากตารางจริงที่ได้จากเอาพุท 2 ค่าจะได้ Karnaugh Maps 2 รูป คือ รูปของ S และของ C_o



(ก) รูปของผลบวก S (ข) รูปของตัวทดออก C_o

รูปที่ 9 Karnaugh Maps ของการบวกเต็ม

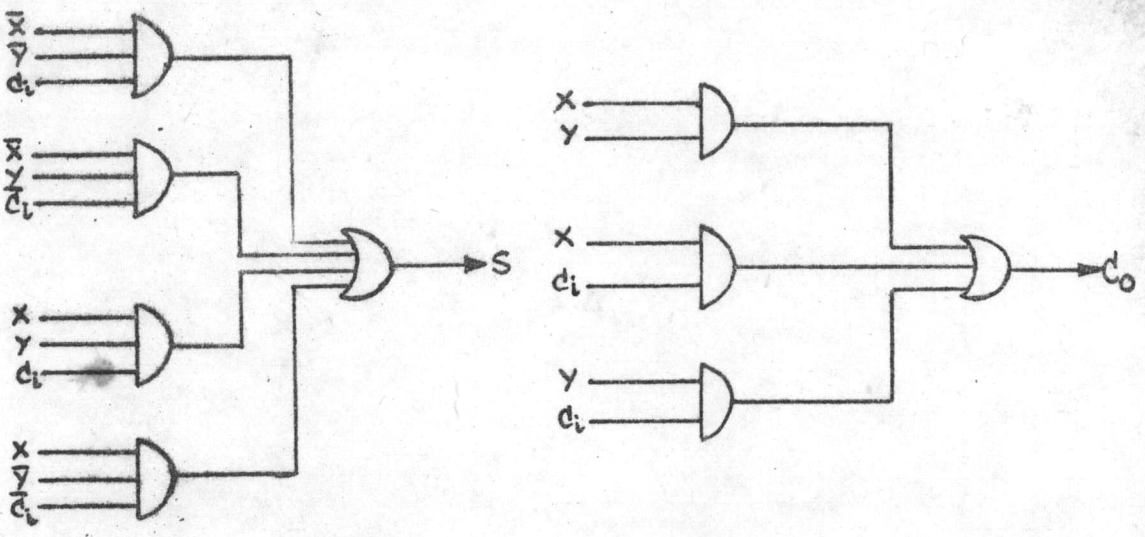
จาก MAP ของผลบวก เราไม่สามารถจะทำฟังก์ชั่น S ให้ง่ายได้ด้วยการลดรูปตามหลักของด้านประชิดของค่า 1 ได้ ดังนั้นฟังก์ชั่นของ S จากการ MAP จะได้เป็น

$$S = \overline{X}\overline{Y}C_i + \overline{X}Y\overline{C}_i + XYC_i + X\overline{Y}\overline{C}_i$$

แต่จากรูปของตัวทดออก เราสามารถใช้หลักการลดรูปของด้านประชิดได้เป็น

$$\begin{aligned} C_o &= XY\overline{C}_i + \overline{X}YC_i + XYC_i + X\overline{Y}C_i \\ &= \overline{X}YC_i + X\overline{Y}C_i + XY \\ &= YC_i + XC_i + XY \end{aligned}$$

ซึ่งสามารถเขียนเป็นโลยิคไดอะแกรมได้เป็นรูปที่ 10



(ก) โลยิคไดอะแกรมของ S (ข) โลยิคไดอะแกรมของ C_o

รูปที่ 10 โลยิคไดอะแกรมของวงจรบวกเต็ม

เมื่อนำวงจรทั้งสองมารวมกันเป็นวงจรของการบวกเต็ม ไม่นิยมนำมารวมกันตรง ๆ แต่ดัดแปลงโดยนำเอาวงจรของตัวทดออกมาใช้ในวงจรผลบวกดังนี้

เพราะว่า $C_o = YC_i + XC_i + XY$

ดังนั้น $\bar{C}_o = \overline{(YC_i + XC_i + XY)}$

เมื่อนำมา OR กับ $XYC_i = \overline{(YC_i + XC_i + XY)} + XYC_i$

และนำมา AND กับ $X+Y+C_i = \left[\overline{(YC_i + XC_i + XY)} + XYC_i\right] \quad (X+Y+C_i)$

ซึ่งเราสามารถแสดงได้ว่า ฟังก์ชันที่ได้ใหม่นี้มีค่าเท่ากับฟังก์ชัน S เดิม ดังนี้

จาก $S = \left[\overline{(YC_i + XC_i + XY)} + XYC_i\right] \quad (X+Y+C_i)$

$$= \overline{\left[\overline{(YC_i + XC_i + XY_i)} + XYC_i\right]} \quad (X+Y+C_i)$$

$$= \overline{\left[\overline{(YC_i + XC_i + XY)} + XYC_i\right]} + \overline{(X+Y+C_i)}$$

$$= \overline{(YC_i + XC_i + XY) \quad (\overline{XYC_i}) + (\overline{X+Y+C_i})}$$

$$= \overline{\left[(YC_i+XC_i+XY)\ (\bar{X}+\bar{Y}+\bar{C}_i)\right]} + \overline{(\bar{X}+Y+C_i)}$$

$$= \overline{(\bar{X}YC_i+X\bar{Y}C_i+XY\bar{C}_i)} + \overline{(\bar{X}+Y+C_i)}$$

$$= \overline{(\bar{X}YC_i+X\bar{Y}C_i+XY\bar{C}_i)}\ (X+Y+C_i)$$

$$= \overline{(\bar{X}YC_i)}\ \overline{(X\bar{Y}C_i)}\ \overline{(XY\bar{C}_i)}\ (X+Y+C_i)$$

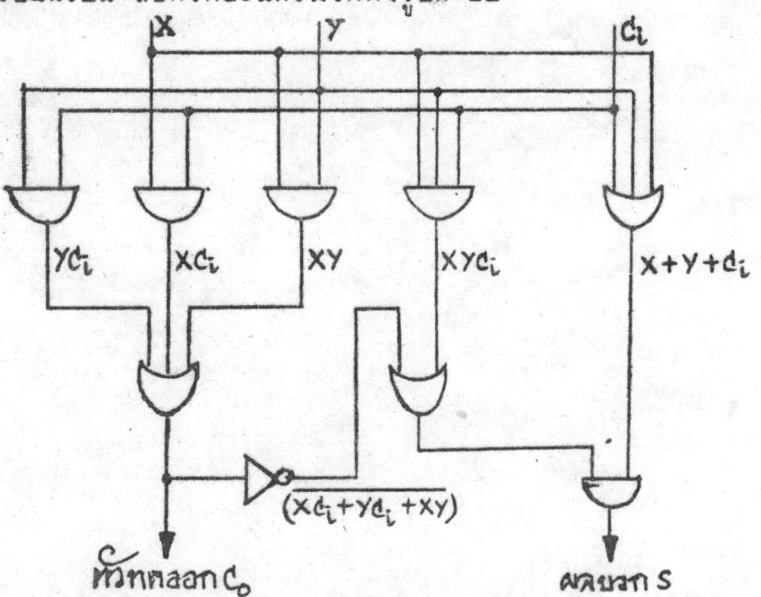
$$= (X+\bar{Y}+\bar{C}_i)\ (\bar{X}+Y+\bar{C}_i)\ (\bar{X}+\bar{Y}+C_i)\ (X+Y+C_i)$$

$$= \left[\bar{C}_i+(X+\bar{Y})\ (\bar{X}+Y)\right]\ \left[C_i+(X+Y)\ (\bar{X}+\bar{Y})\right]$$

$$= (\bar{C}_i+\bar{X}\ \bar{Y}+XY)\ (C_i+\bar{X}Y+X\bar{Y})$$

$$= \bar{X}\ \bar{Y}C_i+\bar{X}Y\bar{C}_i+XYC_i+X\bar{Y}\ \bar{C}_i \quad = \quad S \text{ เติม}$$

ซึ่งสามารถเขียนเป็นโลยิคไดอะแกรมได้ดังรูปที่ 11



รูปที่ 11 วงจรบวกเต็มของ IBM 704 - 709[5]

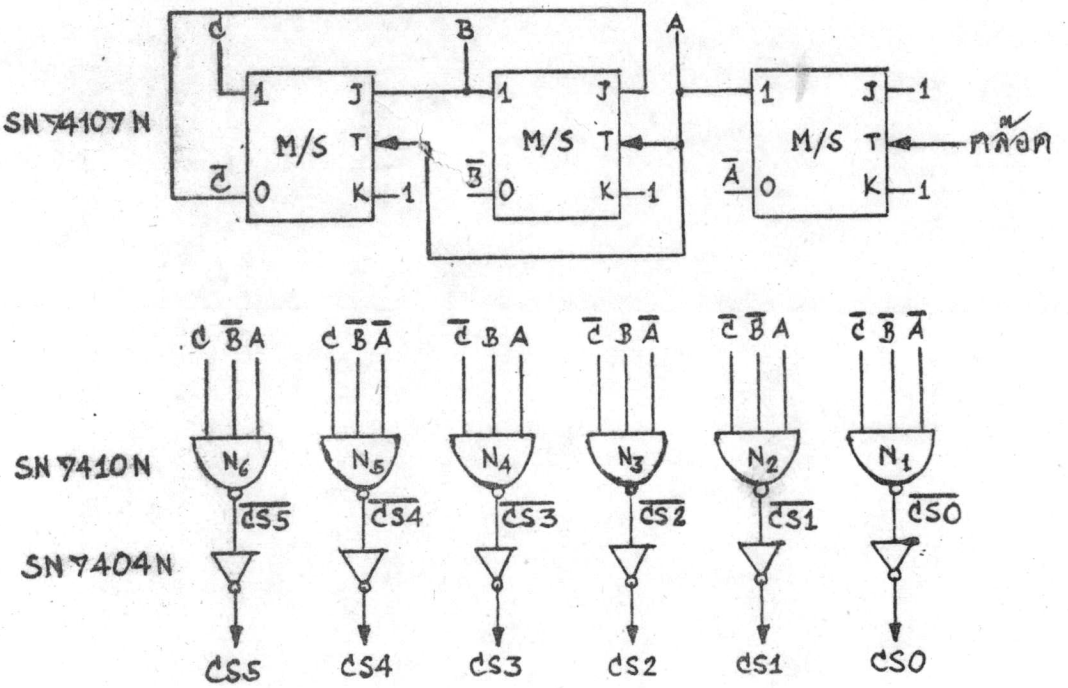
---

[5] Bartee, THOMAS C. 1966. Digital Computer Fundamentals. (2nd.ed. McGraw - Hill Book Co.), p. 175.

2.5 วงจรนับและการเปลี่ยนโค้ดการนับ

(Counter Circuit and Decoding Technique)[6]

เมื่อใช้วงจรนับเป็นวงจรควบคุมลำดับ (Steps) ของการทำงานจำเป็นที่จะต้องมีพัลซ์เป็นตัวกระตุ้นในแต่ละลำดับของการทำงาน ถ้าใช้วงจรนับโมดูลัส –6 (วงจรนับที่ครบรอบใน 6 พัลซ์) เป็นวงจรที่ทำให้เกิดพัลซ์ในแต่ละลำดับของการทำงาน ก็ต้องต่อเอาพุทของวงจรนับเข้ากับเกต NAND เพื่อจะเปลี่ยนโค้ดเป็นลำดับของการนับ รูปที่ 12 แสดงวงจรนับโมดูลัส –6 กับ NAND เกตที่ใช้ในการเปลี่ยนโค้ด 6 เกต



รูปที่ 12 วงจรนับโมดูลัส –6 กับ NAND เกต ที่ใช้เปลี่ยนโค้ด

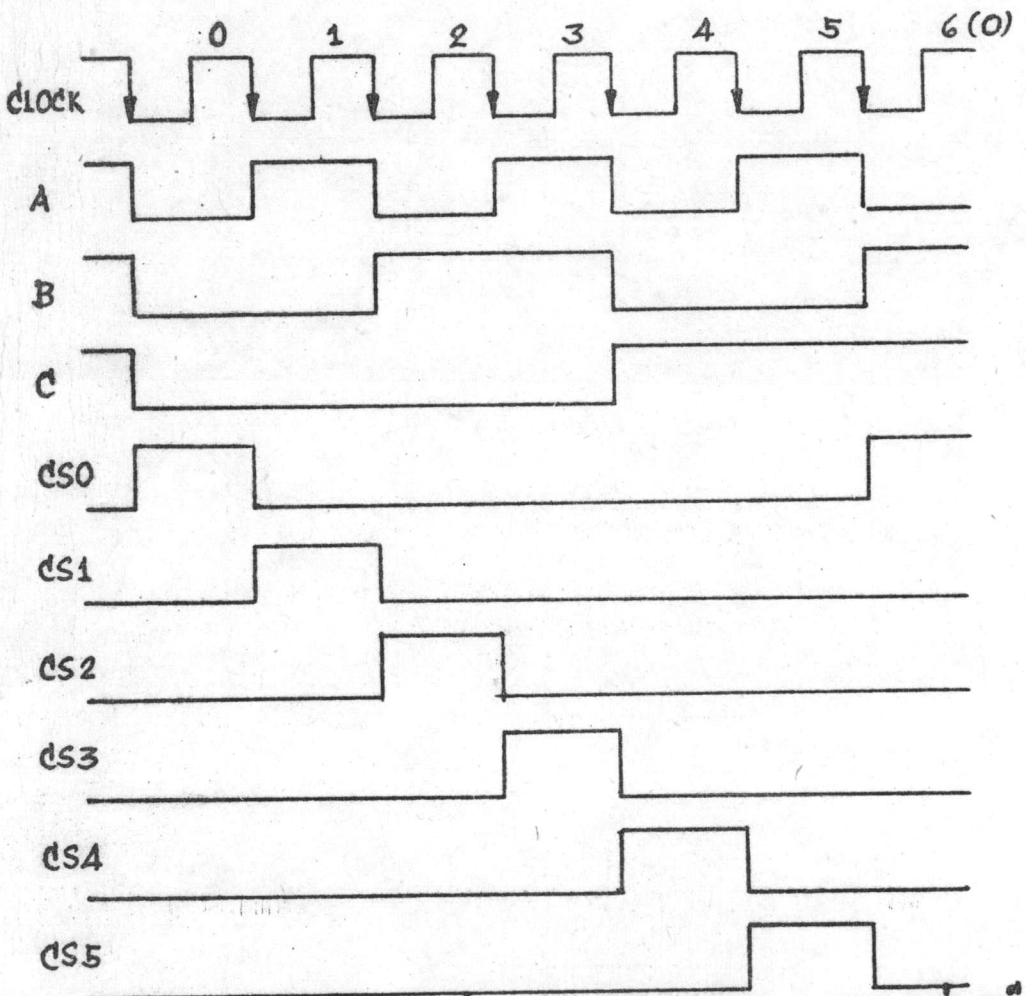
เกต NAND แต่ละอันมีอินพุทอยู่ 3 ค่า เมื่อค่านับเป็น 0 (000) อินพุท $\bar{A}$, $\bar{B}$, $\bar{C}$ จะมีโวลเตจสูง และสัญญาณเอ้าพุท CS0 (ลำดับของการนับที่ 0) จะมีค่า

---

[6] Nashelsky, Louis; 1972. Introduction to Digital Computer Technology. (New York : John Willey and Sons), p. 354 - 355.

เป็นโลยิค -1 ในขณะนี้เมื่อตรวจดูเกตอื่น ๆ จะพบว่ามีอินพุทอย่างน้อย 1 ค่า เป็นโลยิค -0 และเอ้าพุท CS 1 ถึง CS 5 จะเป็นโลยิค -0 ทั้งหมด เมื่อเคาน์เตอร์ได้รับสัญญาณคล็อคและเปลี่ยนเป็น 001 เอ้าพุท CS 1 จะมีโวลเตจสูง คือ มีค่าเป็นโลยิค - 1 แต่เอ้าพุท CS 2 จะมีโวลเตจต่ำ คือมีค่าเป็นโลยิค -0 และ เอ้าพุทอื่น ๆ คือ CS 3, CS 4 และ CS 5 ก็มีโลยิค -0 ด้วย

ผลที่ได้จากการต่อวงจรนับและตัวเปลี่ยนโค้ด จะได้เป็นรูปของคลื่นดังรูปที่ 13

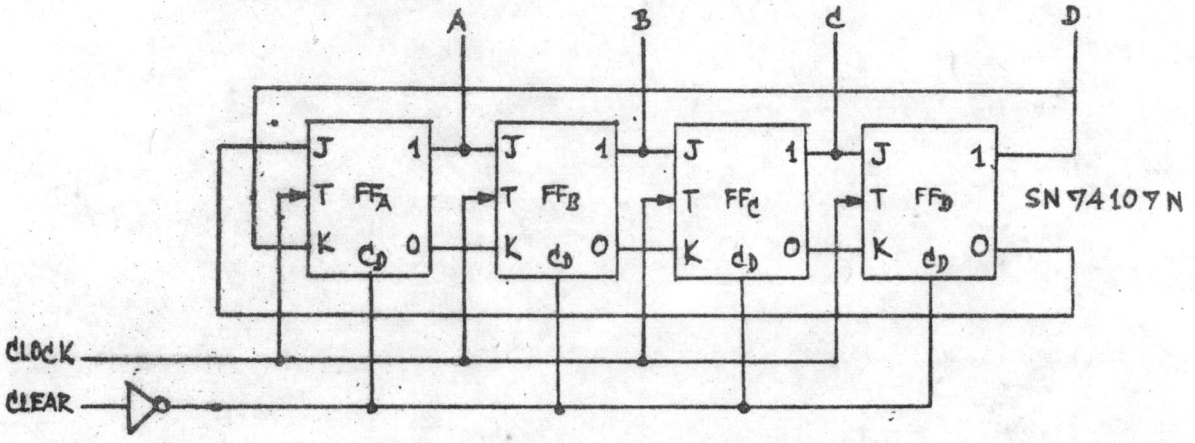


รูปที่ 13 รูปคลื่นของวงจรนับโมดูลัส - 6 กับตัวเปลี่ยนโค้ด

จากรูปที่ 13 จะเห็นว่าแต่ละสัญญาณคล็อคจะทำให้เอ้าพุทของลำดับการนับต่าง ๆ มีค่าโวลเตจเพิ่มขึ้น เรียงกันตามลำดับ เริ่มตั้งแต่ CS0 เพิ่มขึ้น ในช่วงของการนับครั้งแรก และ CS 1 เพิ่มขึ้นในช่วงของการนับครั้งต่อไป เรื่อยไปจนถึง CS 5

วงจร Switch Tail Ring Counter[7]

วงจร Switch Tail Ring Counter หรือ Inverting Ring Counter มีแสดงในรูปที่ 14



รูปที่ 14 Switch Tail Ring Counter

จากรูปที่ 15 ตอนแรกเมื่อกำหนดให้เอาพุทของ A, B, C และ D เป็น 0 หมด คือ พลิพ-ฟล็อพทุกตัวอยู่ในภาวะรีเซ็ท (1- เอาพุท เป็นโลยิค -0, และ 0- เอาพุท เป็นโลยิค -1)

ดังนั้นอินพุทของ FF_a ที่ J ซึ่งได้จากการ Feed-Back ของเอาพุทที่ 0 ของ FF_d จะเป็นโลยิค -1 และอินพุทที่ K ของ FF_a จะเป็นโลยิค -0

ในการคล็อคครั้งแรก เอาพุทของ FF_a จะเปลี่ยนเป็นโลยิค -1 ที่ 1 และเป็นโลยิค -0 ที่ 0 แต่เอาพุทของ FF_b, FF_c และ FF_d ยังคงเดิม ดังนั้นผลของการคล็อคครั้งแรก เอาพุท A จะเป็น 1, ส่วน B,C และ D เป็น 0 หมด

จากการคล็อคครั้งแรกจะทำให้อินพุทของ FF_b เปลี่ยนไป คือ FF_b อยู่ในภาวะเซ็ท ดังนั้นเมื่อคล็อคครั้งที่สอง จะทำให้ได้เอาพุทที่ A และ B เป็น 1, ส่วน C และ D ยังคงเป็น 0

---

[7] Nashelsky, Louis; 1972. Introduction to Digital Computer Technology. (New York : John Wiley and Sons), p. 358.

ทำนองเดียวกัน เมื่อทำการคล็อคครั้งที่สาม จะทำให้ได้เอาพุทของ A, B,C เป็น 1 ส่วน เอาพุทของ D ยังเป็น 0 และ เมื่อทำการคล็อคครั้งที่สี่ เอาพุทของฟลิพ-ฟล็อพทั้ง 4 จะเป็น 1 หมด ตามตารางที่ 3

พัลซ์อินพุท	A	B	C	D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
8	0	0	0	0
9	1	0	0	0

ตารางที่ 3 ตารางจริงของ Switch Tail Ring Counter

เมื่อทำการคล็อคครั้งที่ห้า เอาพุทของ A จะเปลี่ยนเป็น 0 เพราะปลาย J ได้รับอินพุทโลยิค -0 และปลาย K ได้รับอินพุทโลยิค -1 ทำให้ FF_a อยู่ในภาวะรีเซ็ท จึงมีเอาพุทเป็น 0 และเมื่อคล็อคครั้งต่อ ๆ ไป จะได้เอาพุทของฟลิพ-ฟล็อพทั้ง 4 ดังค่าในตารางที่ 3 วนเวียนไปเรื่อย ๆ

2.6 วงจรแก้ผลบวก (Correct-Digit Sum Circuit)

ในการบวกเลข 2 จำนวน ในโค้ด บีซีดี ค่าของตัวตั้งและตัวบวกที่เป็นอินพุท จะอยู่ระหว่าง 0000 ถึง 1001 (จาก 0 ถึง 9) แต่เมื่อบวกกันแล้ว ช่วงของผลบวก จะเป็นเลขฐานสองที่อยู่ระหว่าง 0000 ถึง 10011 (หรือจาก 0 ถึง 19 ในเลขทศนิยม) เพราะว่าค่าสูงสุดของผลบวกจะเป็นได้ถึง 9+ 9 + 1 = 19 (1 เป็นตัวทดมาจากหลักที่แล้ว)

ผลบวกที่ได้จากการบวก					ผลบวกที่ถูกต้อง					เลขทศนิยม
C_4	Z_4	Z_3	Z_2	Z_1	C_n	S_4	S_3	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

ผลบวกที่ถูกต้องแล้ว

ผลบวกที่ต้องแก้

ตารางที่ 4 ตารางจริงของผลบวกและการแก้ผลบวกของเลขโค้ด บีซีดี

จากตารางที่ 4 แสดงเอาพุทของจำนวนเลขที่ไม่ถูกต้อง (Uncorrected Digit) ในสดมภ์ทางซ้ายมือของตาราง ตั้งแต่จำนวน 1010 ลงมา ดังนั้นเมื่อเวลาเปลี่ยนโคดเป็นเอาพุทของเลขทศนิยมออกมา ค่าเหล่านี้จะไม่ถูกต้อง จึงต้องเปลี่ยนจำนวนเหล่านี้ให้ถูกต้อง จำนวนที่ถูกต้องมีอยู่ในสดมภ์ทางขวามือ พร้อมกับตัวทดในหลักต่อไปอีก 1

หลักในการแก้ก็คือ นำเลข 6 หรือ 0110 มาบวกเข้ากับจำนวนที่ไม่ถูกต้องจะทำให้ได้ผลบวกที่ถูกต้อง

วงจรโลยิคที่ใช้ในการนี้ได้จากตารางจริงของการแก้ผลบวก จากตารางที่ 4 จะเห็นว่าเราจะต้องแก้ตัวเลขเพื่อจะให้ได้ตัวทด $C_n = 1$ เมื่อ

1. บิทที่ 5 ของค่าของผลบวกจำนวนเดิม (C_4) เป็น 1 หรือเมื่อ
2. $Z_4Z_3 = 1$ หรือเมื่อ
3. $Z_4Z_2 = 1$

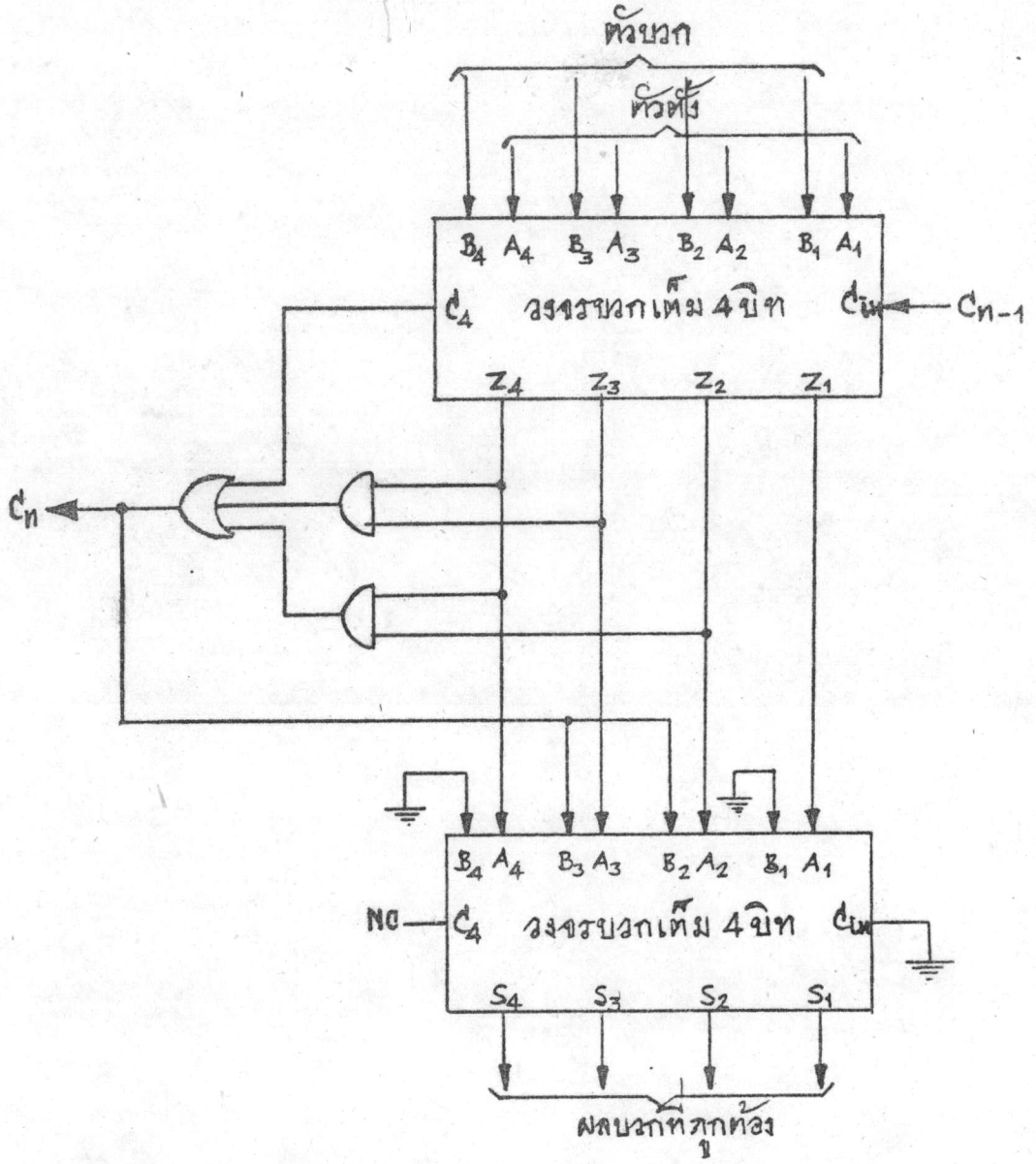
ดังนั้นเราจะได้ฟังก์ชั่นบูลีนที่ให้ค่าของตัวทดออก (C_n) เป็น 1 เป็น

$$C_n = C_4 + Z_4Z_3 + Z_4Z_2$$ [8]

ในการบวกเลขฐานสอง 0110 (เลข 6) เข้ากับตัวเลขผลบวกจำนวนเดิม เราใช้วงจรบวกเต็ม 4 บิท 2 วงจร ดังแสดงในรูปที่ 15

---

[8] Mano, M. Moris; 1972. Computer Logic Design. (New Jersey: Prentice-Hall Inc.), p. 393.

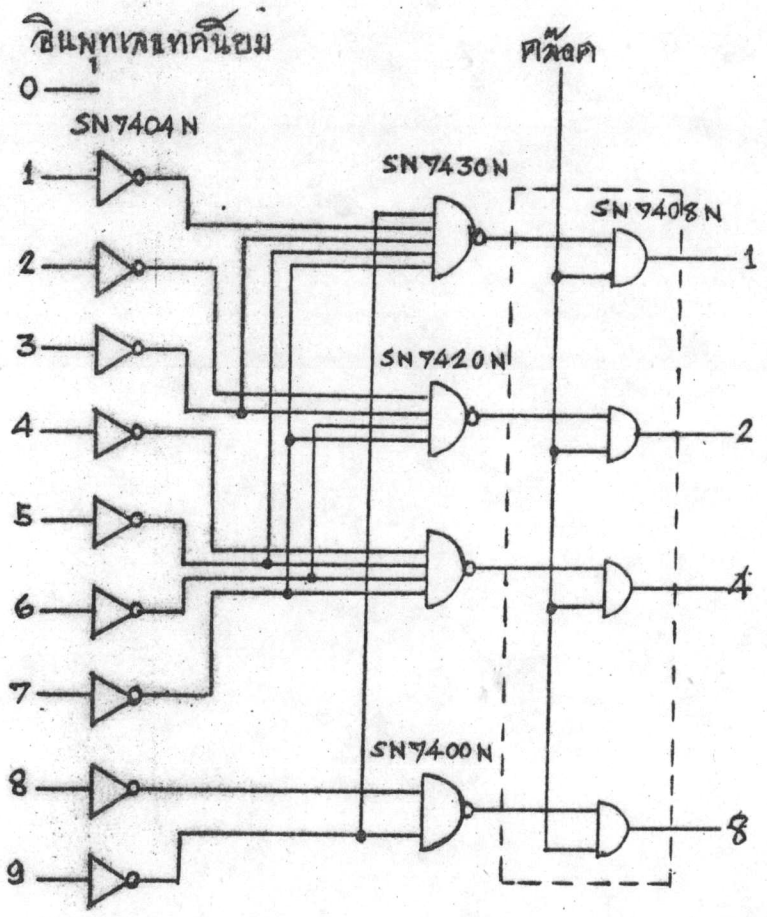


รูปที่ 15 โลยิคไดอะแกรมของการบวกและการแก้ผลบวก

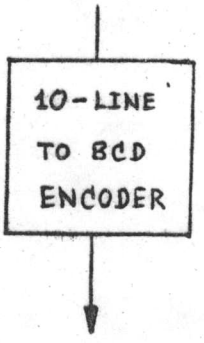
จากรูปที่ 15 จะเห็นว่า เมื่อค่าของตัวทด C_n เป็น 1 ผลบวก Z_1, Z_2, Z_3 และ Z_4 จะถูกบวกด้วย 6 (0110) และได้ผลบวก S_1, S_2, S_3 และ S_4 ที่ถูกต้อง ตามตารางที่ 4

2.7 การเข้าโค้ดและการถอดโค้ด (Encoding and Decoding)[9]

การเข้าโค้ดหรือการโค้ด (Encoding or Coding) หมายถึงวิธีการเปลี่ยนอินพุทจากเลขฐานอื่นเป็นเอาพุทของเลขฐานสอง ตัวเปลี่ยนเรียกว่า ตัวเข้าโค้ด(Encoder) ในการวิจัยนี้ ใช้ตัวเข้าโค้ดที่เปลี่ยนเลขทศนิยมเป็นเลขโค้ดบีซีดี (Decimal to-BCD encoder) วงจรของการเข้าโค้ดมีแสดงในรูปที่ 16



เลขทศนิยม	8	4	2	1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1



รูปที่ 16 โลยิคไดอะแกรมของตัวเข้าโค้ดจากเลขทศนิยมเป็น บีซีดี

---

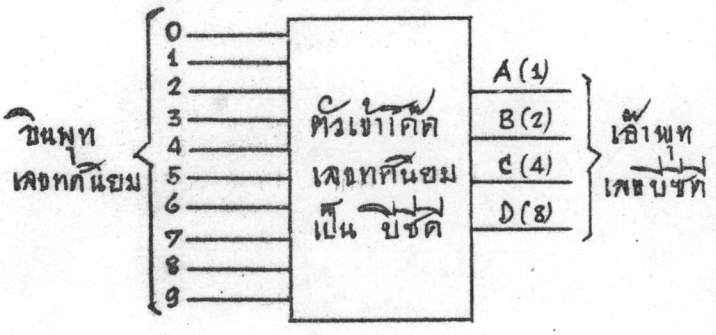
[9] Marcus Abraham; Lenk, John D.; 1973. Computer for Techniciang. (New Jersey : Prentice-Hall, Inc.) p. 192 - 194.

การทำงานของวงจร ตัวอย่างเช่น เมื่อกดคีย์เลข 3 ที่คีย์บอร์ด (Keyboard) อินพุท 3 จะเป็น 1 (true) อินเวอร์เตอร์เกตทำให้เอ้าพุทเป็น 0 (false) ขณะเมื่อเอ้าพุทของอินเวอร์เตอร์เกตอื่น ๆ เป็น 1 (true), ในรูปจะเห็นว่า อินเวอร์เตอร์เกตของ 3 ต่ออยู่กับโลยิค NAND เกต 1 และ 2 เราทราบแล้วว่า เมื่ออินพุทใดอินพุทหนึ่งของ NAND เกต เป็น 0 จะทำให้เอ้าพุทของ NAND เกต เป็น 1 ดังนั้นเอ้าพุทของ NAND เกตุ 1 และ 2 จะมีค่าเป็น 1 แต่อินพุทที่เข้ายัง NAND เกต 4 และ 8 เป็น 1 ทั้งหมด ทำให้เอ้าพุทของ NAND เกต 4 และ 8 เป็น 0 ดังนั้น เอ้าพุทเลขไบนารี่ที่ได้จึงเป็น 0011 คือเลขทศนิยม

ในรูปจะสังเกตเห็นว่า อินพุทของเลขทศนิยม 0 ไม่ได้ต่อกับอะไร เมื่ออินพุทของเลขทศนิยมเป็น 0 (กดคีย์) ไลน์ของเลข 1 – 9 จะเป็นโลยิค –0 เอ้าพุทของอินเวอร์เตอร์เกตจะเป็นโลยิค –1 ทั้งหมด ทำให้เอ้าพุทของ NAND เกตเป็นโลยิค –0 และได้เลขไบนารี่ 0000 หรือเลขทศนิยม 0

AND เกตที่ต่ออยู่ในเส้นไข่ปลา เป็น Open gates ที่ต่อกับสัญญาณคล็อค (อาจเป็นพัลซ์หรือโวลเตจ–คิซี) เมื่อมีสัญญาณคล็อคเกิดขึ้น จึงจะมีเอ้าพุทเกิดขึ้น

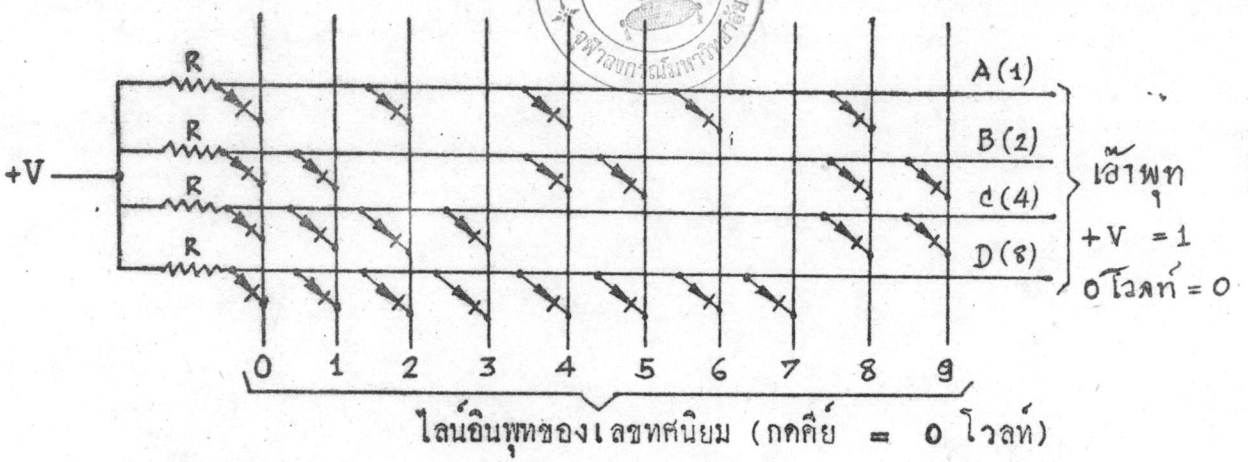
การเข้าโค้ดอาจจะใช้เมทริกไดโอด (Diode Matrices) รูปที่ 17 แสดงบล็อคไดอะแกรมของการเข้าโค้ดของเลขทศนิยมเป็นเลขโค้ด บีซีดี.



(ก)

เลขทศนิยม	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

(ข)



(ค)

รูปที่ 17 เมทริกไดโอดของตัวเข้าโค๊ดจากเลขทศนิยมเป็น บีซีดี[10]

(ก) บล็อคไดอะแกรม (ข) ตารางจริงของการเปลี่ยน

(ค) เมทริกไดโอดของตัวเข้าโค๊ด

อินพุทอาจจะเข้าทางคีย์บอร์ด (Key - Board)หรือสวิทช์จากเลข 0 ถึง 9 โดยการกดคีย์บอร์ดตัวใดตัวหนึ่ง ตัวเข้าโค๊ดจะเปลี่ยนเลขทศนิยมเป็นเลขโค๊ด บีซีดี ตามแบบของรูปที่ 17 (ข)

การทำงานของเมทริกไดโอดมีแสดงในรูป 17 (ค) เมื่อกดคีย์ของเลขทศนิยม ก็เป็นการใส่อินพุท 0 โวลท์ในเมทริก เช่น เมื่อกดคีย์ของอินพุท หมายเลข 3 จะทำให้ไลน์เอาพุทของ C และ D มีโวลท์เดจผ่านไดโอดเป็น 0 แต่ไลน์ A และ B โวลเตจ จะสูง โค๊ดที่ออกทางไลน์ บีซีดี จึงเป็น 0011 ถ้าไม่กดคีย์เลย ไลน์ของเอาพุทจะสูงทั้งหมด

---

[10] Nashelsky, Louis; 1972. Introduction to Digital Computer Technology. (New York : John Wiley and Sons, Inc.), p. 416.

การถอดโค๊ดและการแสดงตัวเลขโดยใช้ภาพ 7 ส่วน

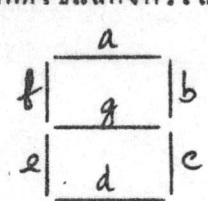
(Decoding and Seven-Segment Display)

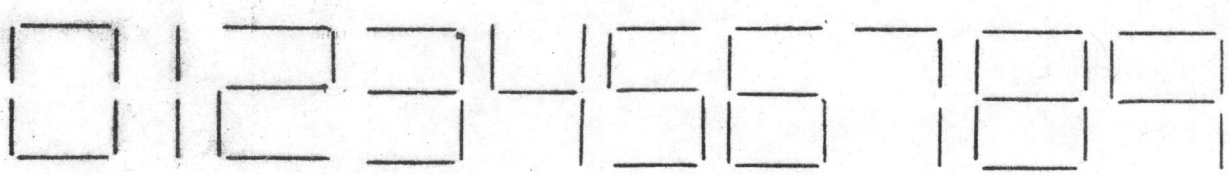
ตารางจริงของการเปลี่ยนเลขโค๊ด บีซีดี เป็นไลน์ 7 ไลน์ เพื่อนำไปไดร์พ (drive) ส่วนต่าง ๆ ของหลอดภาพออกมาเป็นตัวเลขมีในตารางที่ 5

D	C	B	A	เลขทศนิยม	g	f	e	d	c	b	a
0	0	0	0	0	0	1	1	1	1	1	1
0	0	0	1	1	0	0	0	0	1	1	0
0	0	1	0	2	1	0	1	1	0	1	1
0	0	1	1	3	1	0	0	1	1	1	1
0	1	0	0	4	1	1	0	0	1	1	0
0	1	0	1	5	1	1	0	1	1	0	1
0	1	1	0	6	1	1	1	1	1	0	1
0	1	1	1	7	0	0	0	0	1	1	1
1	0	0	0	8	1	1	1	1	1	1	1
1	0	0	1	9	1	1	0	0	1	1	1

ตารางที่ 5 ตารางจริงของการเปลี่ยนโค๊ด บีซีดี เป็นไลน์ 7 ไลน์[11]

ภาพเดิมของหลอดภาพที่ใช้แสดงตัวเลขมีอยู่ 7 ส่วน ดังแสดงในรูปที่ 18





รูปที่ 18 แสดงตัวเลขทศนิยมโดยใช้หลอดภาพ 7 ส่วน

---

[11] Nashelsky, Louis; 1972. Introduction to Digital Computer Technology. (New York : John Wiley and Sons, Inc.), p. 418.

โดยการเลือกคอมบิเนชั่น (Combinations) ต่าง ๆ ของส่วนของภาพทั้ง 7 จะสามารถแสดงตัวเลขจาก 0 - 9 เช่น ถ้าจะแสดงเลขทศนิยม 1 ก็ต้องใช้ส่วน b และ c และโลยิคของ b และ c จะเป็น 1 ในตารางที่ 5 หรือแสดงเลข 5 ก็ต้องใช้ส่วน a,c,d,f และ g ซึ่งค่าโลยิคเป็น 1

จากตารางจริงของการเปลี่ยนเลขโค๊ด บีซีดี เป็นไลน์ 7 ไลน์ เราจะหาฟังก์ชั่นบูลีน[12] ได้เป็น

$$\bar{a} = A\bar{B}\bar{C}\bar{D} + \bar{A}C$$

$$\bar{b} = A\bar{B}C + \bar{A}BC$$

$$\bar{c} = \bar{A}B\bar{C}$$

$$\bar{d} = \bar{A}\bar{B}C + ABC + A\bar{B}\bar{C}$$

$$\bar{e} = \bar{A} + \bar{B}C$$

$$\bar{f} = A\bar{D}\bar{C} + BA + B\bar{C}$$

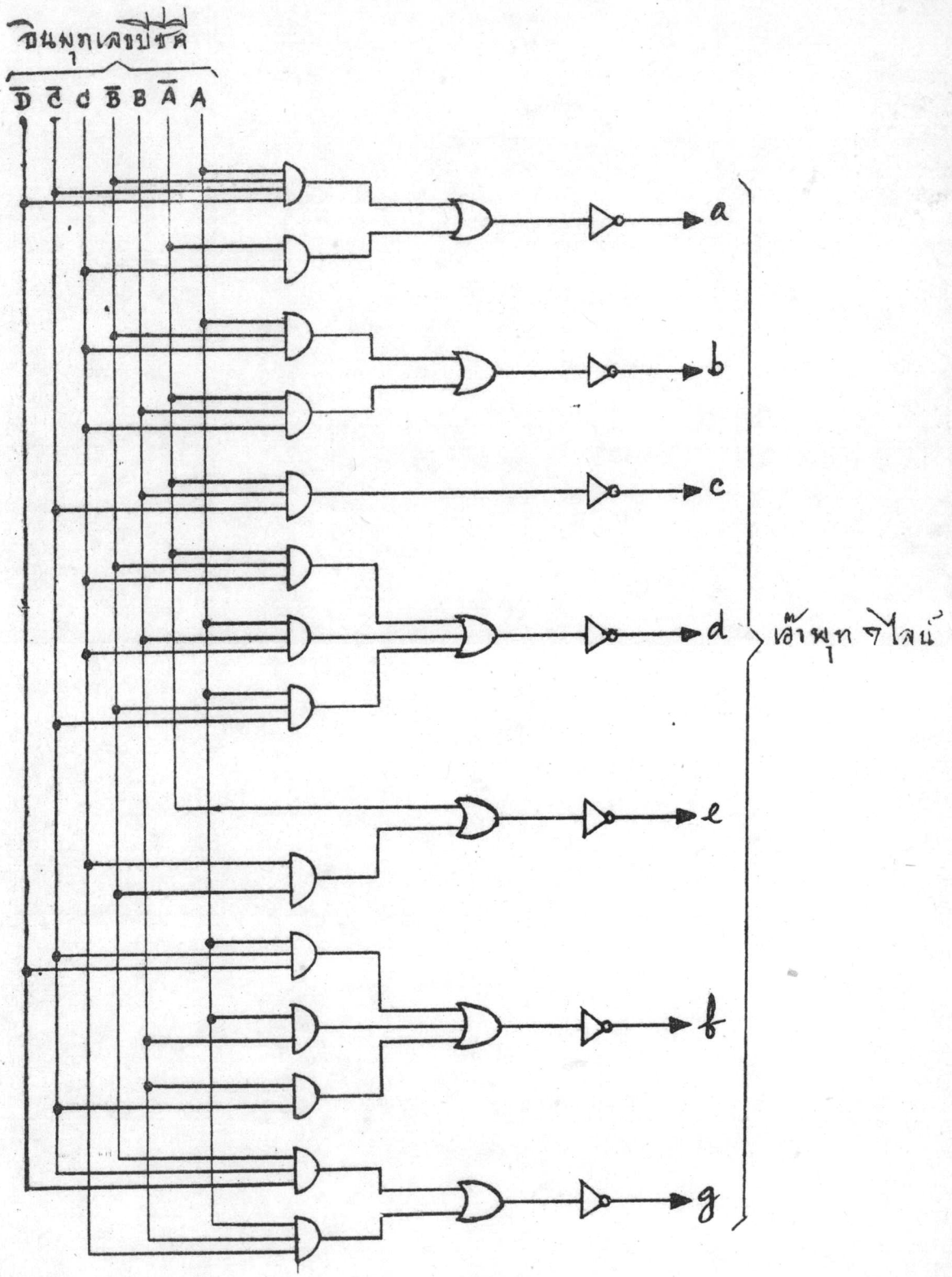
$$\bar{g} = \bar{B}\bar{D}\bar{C} + BAC$$

จากฟังก์ชั่น บูลีน จะเขียนเป็นโลยิคไดอะแกรมได้เป็น BCD-to-7-segment decoder ดังรูปที่ 19

---

[12] Texas Instruments Incorporated; 1971. Desig ning with TTL Integrated Circuits. (McGraw-Hill Book Co.) p. 192

I 16411365



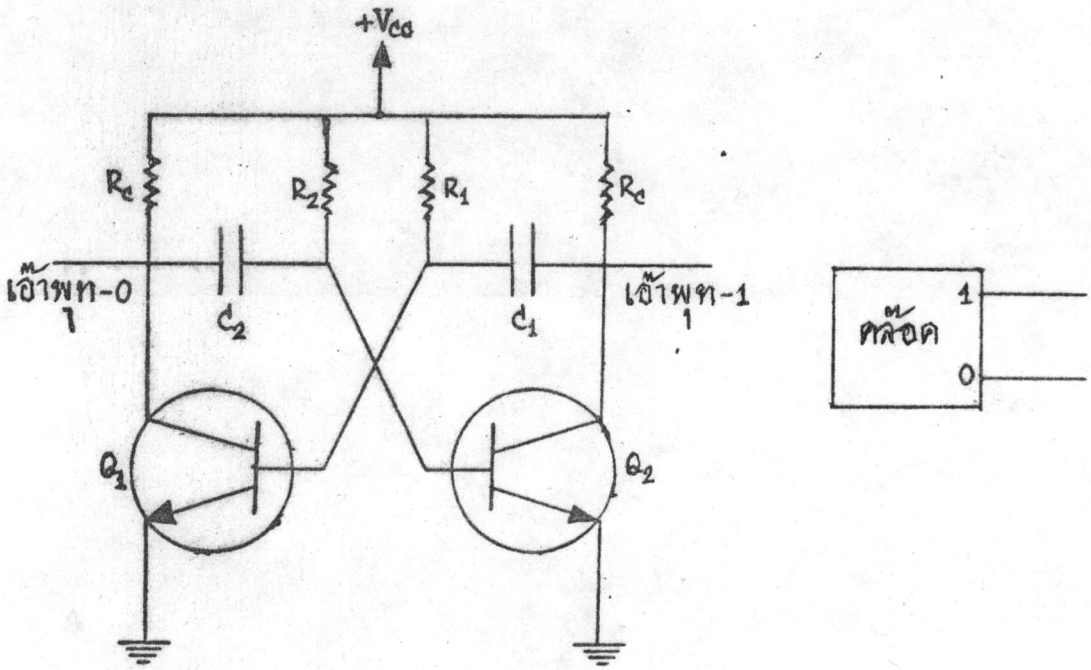
รูปที่ 19 โลยิคไดอะแกรมของตัวถอดโค้ด บีซีดี เป็นไลน์ 7 ไลน์

2.8 อสเตเบิลมัลติไวเบรเตอร์ หรือ คล็อค[13]

(Astable Multivibrator or Clock)

อสเตเบิลมัลติไวเบรเตอร์ หรือ คล็อค เป็นมัลติไวเบรเตอร์ แบบหนึ่งที่มีสภาพการทำงานไม่คงที่ (สเตเบิล) คือ จะเปลี่ยนกลับไปกลับมา (oscillate) "รีเซ็ท" และ "เซ็ท"

วงจรมัลติไวเบรเตอร์ชนิดนี้ จะทำให้เกิดสัญญาณ "คล็อค" (Clock) และนำไปใช้เป็นพัลซ์กระตุ้นให้วงจรอื่น ๆ ทำงานในช่วระยะเวลาใดเวลาหนึ่ง



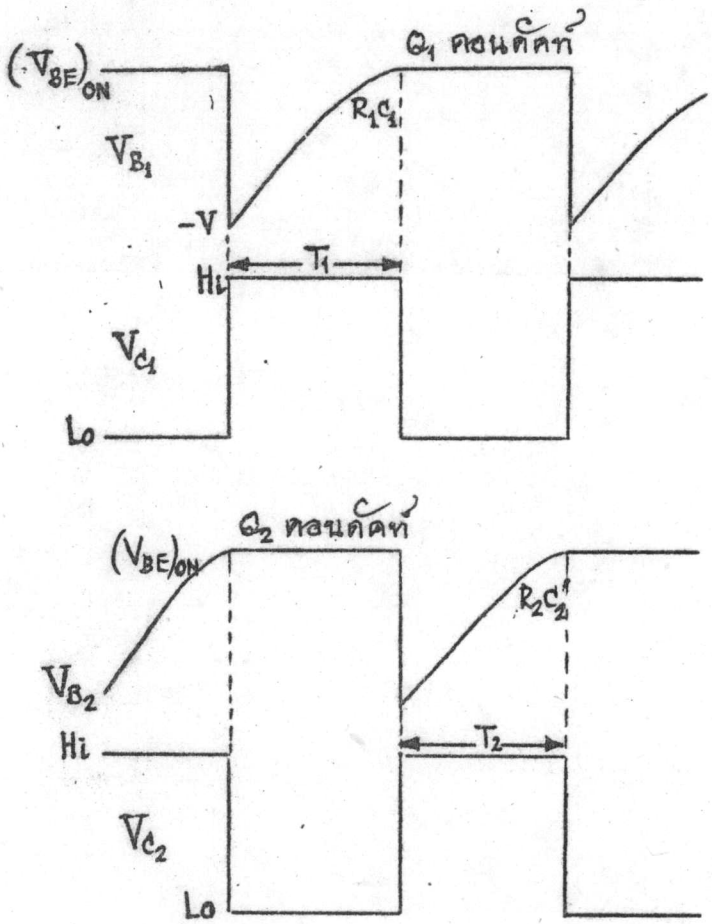
รูปที่ 20 วงจรอสเตเบิลมัลติไวเบรเตอร์ (คล็อค) และโลยิคไดอะแกรม

รูปที่ 20 แสดงวงจรอสเตเบิลมัลติไวเบรเตอร์ (คล็อค) ให้สังเกตุว่าใช้ คาพาซิเตอร์ (Capacitors) C_1 และ C_2 เป็นตัวควบคุม-คลัพปิ้ง วงจรจึงอยู่ในภาวะไม่สเตเบิล

---

[13] Nashelsky, Louis; 1972. Introduction to Digital Computer Technology. (John Wiley and Sons, Inc.), p. 304 - 306.

การทำงานของวงจร ขณะที่ทรานซิสเตอร์ Q_1 คอนดัคท์ (ON) และทรานซิสเตอร์ Q_2 คัตออฟ (OFF) โวลเตจของเบสของ Q_2 จะสูงขึ้นเป็นโค้งเอ็กโปเนนเชียล (exponential curve) จากโวลเตจลบใกล้ค่า $-V_{cc}$ ขึ้นไปถึง $+V_{cc}$ เมื่อโวลเตจขึ้นไปถึงระดับ $(+V_{be})_{on}$ จะทำให้ทรานซิสเตอร์ Q_2 คอนดัคท์ (ON) ค่าของ R_2 และ C_2 จะเป็นตัวกำหนดช่วงเวลาที่ Q_2 คัตออฟอยู่



รูปที่ 21 รูปคลื่นโวลเตจของเบสและคอลเล็คเตอร์ของวงจรอสเตเบิลมัลติไวเบรเตอร์ หรือ คล็อค

เมื่อ Q_2 คอนดัคท์ โวลเตจของคอลเลคเตอร์ Q_2 จะลดลงจาก $+V_{cc}$ จนมีค่าเกือบเท่ากับ 0 โวลท์ โวลเตจของเบสของ Q_1 จะลดลงจาก $(V_{be})_{on}$ ทำให้ Q_1 คัตออฟ และตัวที่กำหนดเวลาที่ Q_1 คัตออฟก็คือ R_1 และ C_1

หลังจากโวลเตจของเบสของ Q_1 สูงขึ้นถึง $(V_{be})_{on}$ จะทำให้ Q_1 คอนดัคท์อีก และเปลี่ยนเป็นคัตออฟอีก การทำงานจะเปลี่ยนกลับไปกลับมาอยู่ตลอดเวลา ที่ยังมีโวลเตจซัพพลาย $+V_{cc}$ อยู่

ความถี่ของสัญญาณคล็อค (clock signal) ของวงจร หาได้จากสูตร

$$f = \frac{1}{T_1+T_2} = \frac{1}{0.7R_1C_1+0.7R_2C_2} = \frac{1.4}{R_1C_1+R_2C_2}$$

ถ้ากำหนดให้รีซิสเตอร์และคาพาซิเตอร์ทั้งสองมีค่าเดียวกัน ความถี่จะได้เป็น

$$f = \frac{1}{T} = \frac{1}{2(0.7)\,RC} = \frac{1}{1.4RC} = \frac{0.7}{RC}$$

ในการวิจัยนี้ generate คล็อคจาก Crystal - Controlled ซึ่งให้เอาพุท เป็น square -wave ความถี่ 100 KHz. รูปจะดูได้จากภาคผนวก

2.9 วงจร 9's คอมพลีเมนท์

(Natural Binary Coded Decimal Complement Circuit)

ในการลบเลข 2 จำนวนในโค๊ด บีซีดี ที่เปลี่ยนมาจากเลขทศนิยมแล้ว เราใช้วิธีทำตัวลบเป็น 9's คอมพลีเมนท์ ซึ่งมีรายละเอียดดังนี้

9's คอมพลีเมนท์ หรือ Natural Binary Coded Decimal Complement ถ้าให้ตัวเลขในโค๊ด บีซีดี ประกอบด้วยค่าของบิทในแต่ละบิท เป็น X_4, X_3, X_2 และ X_1 ตามลำดับ และให้ 9's คอมพลีเมนท์ ประกอบด้วยค่าแต่ละบิทเป็น C_4,C_3,C_2 และ C_1 ตามลำดับ ค่าของคอมพลีเมนท์จะได้ดัง ตารางที่ 6

เลขทศนิยม	โค๊ด BCD				คอมพลีเมนท์ของโค๊ด BCD				คอมพลีเมนท์
	X_4	X_3	X_2	X_1	C_4	C_3	C_2	C_1	
0	0	0	0	0	1	0	0	1	9
1	0	0	0	1	1	0	0	0	8
2	0	0	1	0	0	1	1	1	7
3	0	0	1	1	0	1	1	0	6
4	0	1	0	0	0	1	0	1	5
5	0	1	0	1	0	1	0	0	4
6	0	1	1	0	0	0	1	1	3
7	0	1	1	1	0	0	1	0	2
8	1	0	0	0	0	0	0	1	1
9	1	0	0	1	0	0	0	0	0

ตารางที่ 6 9's คอมพลีเมนท์ของเลขโค๊ด บีซีดี

จากตารางเราจะพบว่า ค่าในสดมภ์ของ C_1 จะกลับกับในสดมภ์ของ X_1 เราเขียนได้ว่า $C_1 = \bar{X}_1$

และค่าของ C_2 และ X_2 จะเหมือนกัน, $C_2 = X_2$

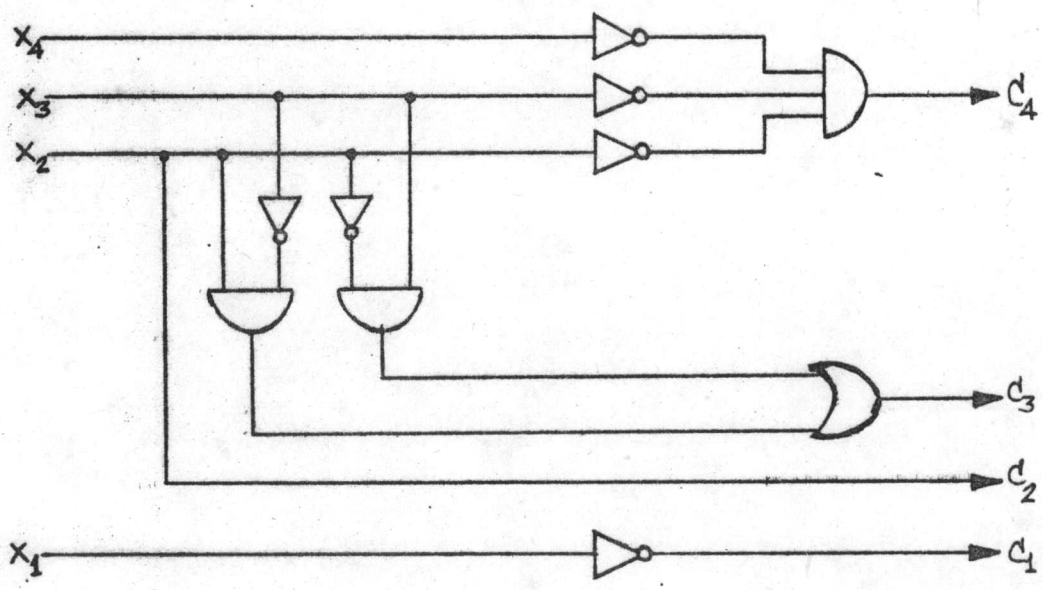
และค่าคอมพลีเมนท์บิทที่ 3 คือ C_3 จะเป็น 1 ก็ต่อเมื่อ ค่าของบิทใน X_3 และ X_2 มีค่ากลับกัน คือ $X_3 = 0, X_2 = 1$ หรือ $X_3 = 1, X_2 = 0$ เราจะเขียนฟังก์ชัน C_3 ได้เป็น

$$C_3 = X_2 \bar{X}_3 + \bar{X}_2 X_3$$

ค่าคอมพลีเมนท์บิทที่ 4 คือ C_4 จะเป็น 1 ต่อเมื่อค่าของบิทใน X_2, X_3, X_4 เป็น 0 ทั้ง 3 ตัว แต่ถ้ามีบิทในบิทหนึ่ง หรือ 2 บิท เป็น 1 บิทของ C_4 จะเป็น 0 เราจะเขียนฟังก์ชันของ C_4 ได้เป็น

$$C_4 = \bar{X}_4 \bar{X}_3 \bar{X}_2$$

จากฟังก์ชันบูลีนของ C_1, C_2, C_3 และ C_4 เราจะเขียนโลยิคไดอะแกรมได้ดังรูปที่ 22



รูปที่ 22 โลยิคไดอะแกรมของการหา 9's คอมพลีเมนท์ของเลขโค้ด บีซีดี

เพื่อให้ใช้ ICs ที่มีอยู่ได้ เราสามารถดัดแปลงวงจร 9's คอมพลีเมนท์ โดยการสังเคราะห์เกตต่าง ๆ แทนเกตที่ใช้อยู่ ในรูปที่ 22 ได้ดังนี้

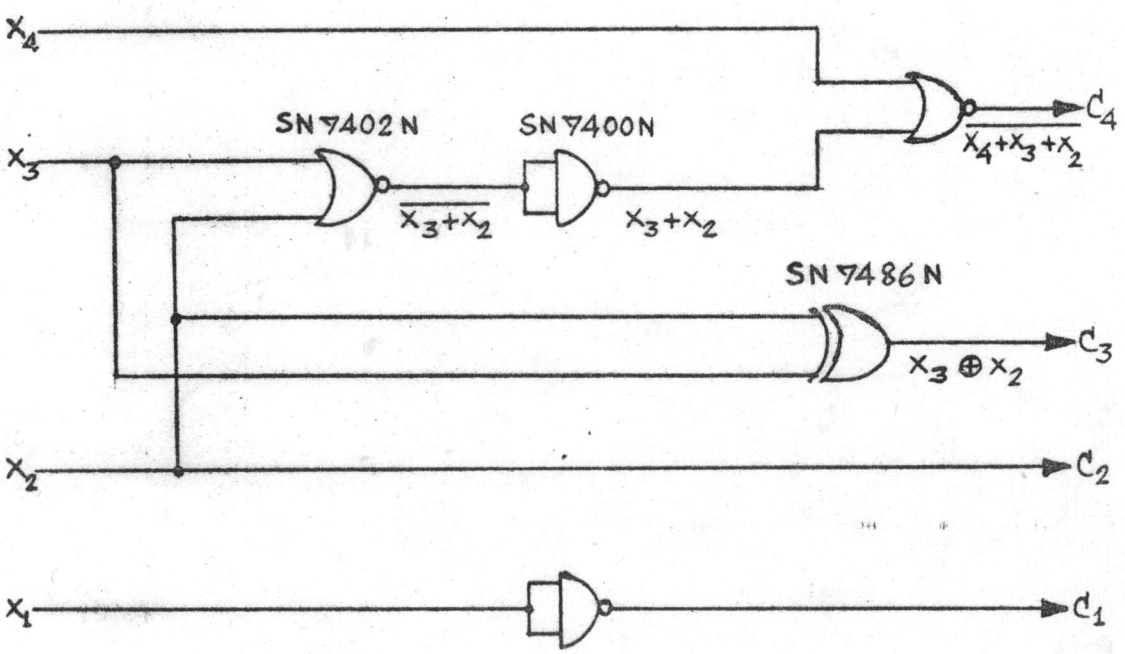
จาก $C_1 = \bar{X}_1$ ใช้เกต 2 - input NAND แทนเกต Inverter ได้

จาก $C_3 = X_2\bar{X}_3 + \bar{X}_2X_3 = X_2 \oplus X_3$ เป็นฟังก์ชั่น Exclusive - OR ใช้เกต Exclusive - OR ได้

จาก $C_4 = \bar{X}_4\bar{X}_3\bar{X}_2$ เนื่องจากไม่มี IC ที่เป็น 3 - input AND gate เราดัดแปลงเป็น

$$C_4 = \bar{X}_4\bar{X}_3\bar{X}_2 = \overline{\bar{\bar{X}}_4\bar{\bar{X}}_3\bar{\bar{X}}_2} = \overline{X_4 + X_3 + X_2}$$

ดังนั้น วงจร 9's คอมพลีเมนท์จะเขียนใหม่ได้เป็นรูปที่ 23



รูปที่ 23 วงจร 9's คอมพลีเมนท์ที่ดัดแปลงแล้ว

2.10 การส่งผ่านข้อมูลในรียิสเตอร์แบบอนุกรม (Shift Registers)

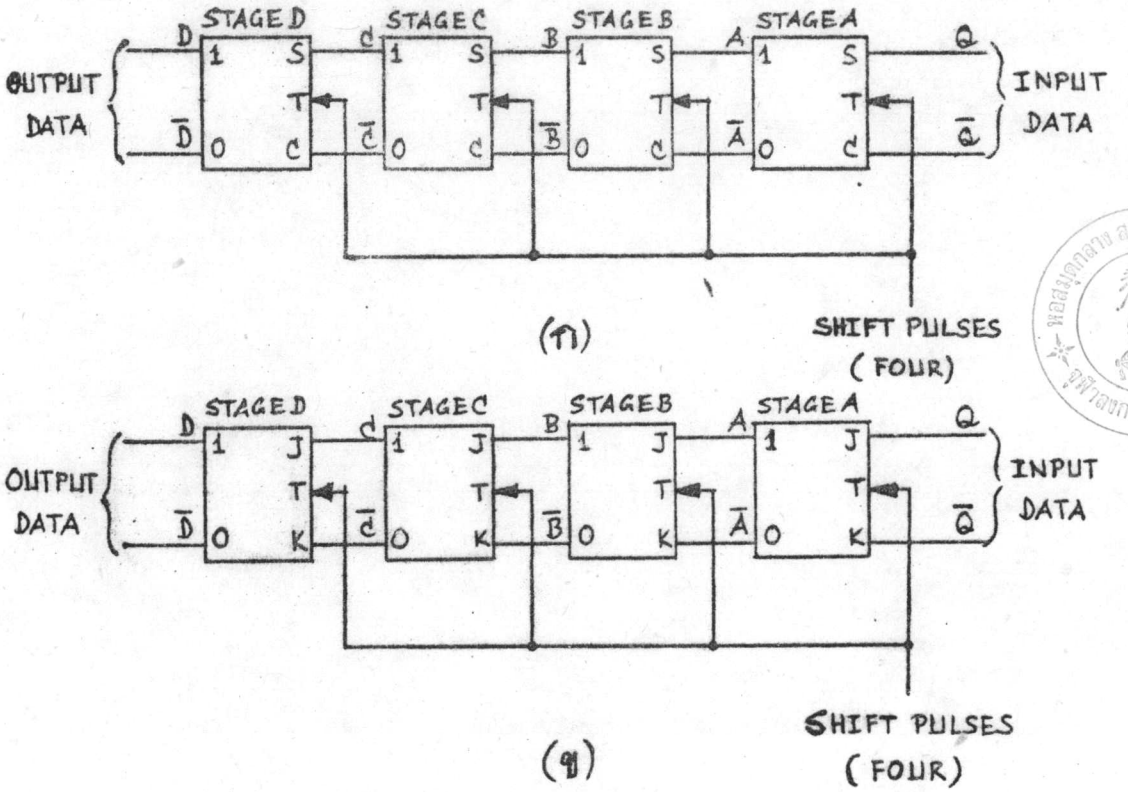
การส่งผ่านข้อมูลเลขฐานสอง ถ้าต้องการให้ตัวเลขเคลื่อนที่ไปทีละบิทต่อเนื่องกัน เราต้องต่อ flip - flop เป็นรูป Shift Register

Shift Register อาจจะออกแบบให้ตัวเลขเคลื่อนที่ไปทางขวาหรือทางซ้ายก็ได้ (คือจากตัวเลขที่มีนัยสำคัญมากที่สุดไปยังตัวเลขที่มีนัยสำคัญน้อยที่สุด หรือกลับกัน) ถ้าใช้ ฟลิพ-ฟล็อพ JK สร้าง Shift Register โดยที่ J และ K เป็นปลายที่ตัวเลขเข้า (data inputs) และ T เป็นทางเข้าของพัลซ์ทริกเกอร์ หรือ shift pulse

การส่งผ่านข้อมูลในรียิสเตอร์แบบอนุกรมทางซ้าย (Shift - Left Register)

รูปที่ 24 แสดงการ shift-left ของ register ที่มี 4 stages รูป ก. ใช้ Resister - Transister Logic (RTL units) รูป ข. ใช้ Transister-Transister Logic (TTL units) หรือ Diode - Transister Logic (DTL units)

ให้สังเกตุว่าในรูปทั้งสอง ตัวเลขอินพุทเป็นสัญญาณเลขฐานสอง ซึ่งมีค่าตรงกันข้าม เอาพุท True (1) มาจากอินพุท S (หรือ J) และเอาพุท False (0) มาจากอินพุท C (หรือ K) ดังนั้นสัญญาณอินพุททั้งสอง จะเป็นค่าโลยิค - 1 เมื่ออินพุท S หรือ J เป็นโลยิค -1 และจะเป็นโลยิค -0 เมื่ออินพุท C หรือ K เป็นโลยิค -1



รูปที่ 24 Shift - Left Registers (ก) RTL units, (ข) TTL หรือ DTL units[14]

shift pulse จะถูก apply เข้าไปแต่ละ stage ต่อเนื่องกันไป เมื่อมี shift pulse เกิดขึ้น ตัวเลขอินพุทที่ปลาย S หรือ J และ C หรือ K จะถูก shift เข้าไปใน stage นั้น ๆ แต่ละ stage จะอยู่ในสภาพ SET หรือ RESET แล้วแต่ตัวเลขอินพุท ขณะเมื่อมี shift pulse

ดังนั้นเมื่อมี shift pulse เกิดขึ้นครั้งแรก ตัวเลขอินพุท (Q) จะถูก shift เข้าใน stage A และขณะเดียวกัน ตัวเลขของ stage A จะ shift เข้าใน stage B จาก stage B ไปยัง stage C ต่อกันไปเรื่อย ๆ

---

[14]Nashelsky, Louis; 1972. Introduction to Digital Computer Technology. (New York: John Willey and Sons), p. 343.

ดังนั้นในแต่ละ shift pulse ตัวเลขจะถูก shift ไปทางซ้ายโดยเลื่อนจาก stage เดิมไปทางซ้ายอีกหนึ่ง stage ใน stage A จะมีตัวเลขใหม่ที่ shift เข้ามา ขณะเดียวกับที่ตัวเลขใน stage D จะ shift ออกเพื่อนำไปใช้ในการคำนวณ หรือการอย่างอื่นต่อไป

ตัวอย่างเช่น ถ้าเริ่มต้นด้วยทุก stages อยู่ในภาวะ RESET (คือ เอาพุท -1 เป็นโลยิค -0) เมื่อ apply อินพุทโลยิค -1 เข้าใน stage A จะได้ตารางของค่าของโลยิคในแต่ละ stage เมื่อมี shift pulses 4 ครั้ง ดังตารางที่ 7

Shift Pulse	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	1	1	1
4	1	1	1	1

ตารางที่ 7 การทำงานของ Shift - Left Register

ต่อไป ถ้าเปลี่ยนเป็น apply อินพุทโลยิค -0 และ 1 สลับกัน เข้าใน stage A และเริ่มต้นทุก stages อยู่ในภาวะ SET (คือ เอาพุท -1 เป็นโลยิค -1) การทำงานจะเป็นดังตารางที่ 8

Shift Pulse	D	C	B	A
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
3	1	0	1	0
4	0	1	0	1

ตารางที่ 8 การทำงานของ Shift - Left Register
เมื่อ apply pulse สลับกัน

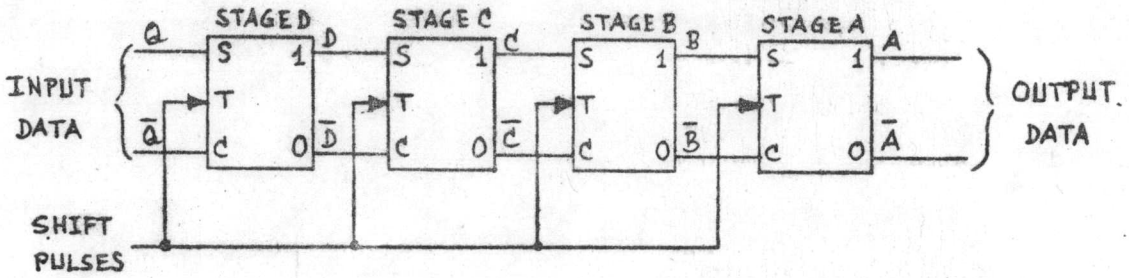
การส่งผ่านข้อมูลในรีจิสเตอร์แบบอนุกรมทางขวา (Shift - Right Register)

บางครั้งเราจำเป็นต้อง shift ตัวเลขของตำแหน่งที่มีนัยสำคัญ น้อยที่สุด (least significant digit) ก่อน ดังเช่นในการบวกหรือลบตัวเลขซึ่งจะต้องมีตัวทดออก (Carry out) ไปยังตำแหน่งถัดไป

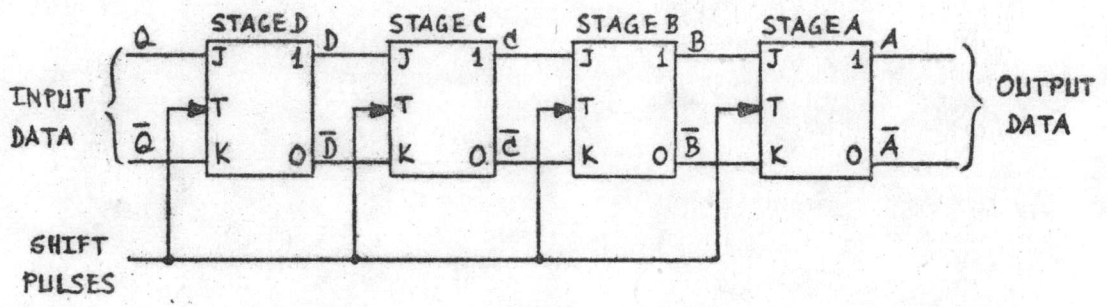
ในรูปที่ 25 ตัวเลขอินพุทจะถูก apply เข้าใน stage D และ shift ไปทางขวา การทำงานจะเหมือนกับการ shift ไปทางซ้าย

ในตารางที่ 9 (ก) แสดงการ shift อินพุท โลยิค -1 เข้าไปในรีจิสเตอร์ที่อยู่ในภาวะ RESET

ในตารางที่ 9 (ข) เป็นการ shift ตัวเลขจากเดิมมีค่าเป็น 1101 โดยอินพุทตัวเลขเป็นโลยิค -0 เข้าใน stage D จนกระทั่งทุก stage อยู่ในภาวะ RESET



(ก)



(ข)

รูปที่ 25 Shift - Right Registers (ก) RTL units, (ข) TTL หรือ DTL units [15]

Shift Pulse	D	C	B	A
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1

(ก)

Shift Pulse	D	C	B	A
0	1	1	0	1
1	0	1	1	0
2	0	0	1	1
3	0	0	0	1
4	0	0	0	0

(ข)

ตารางที่ 9 การทำงานของ Shift - Right Register

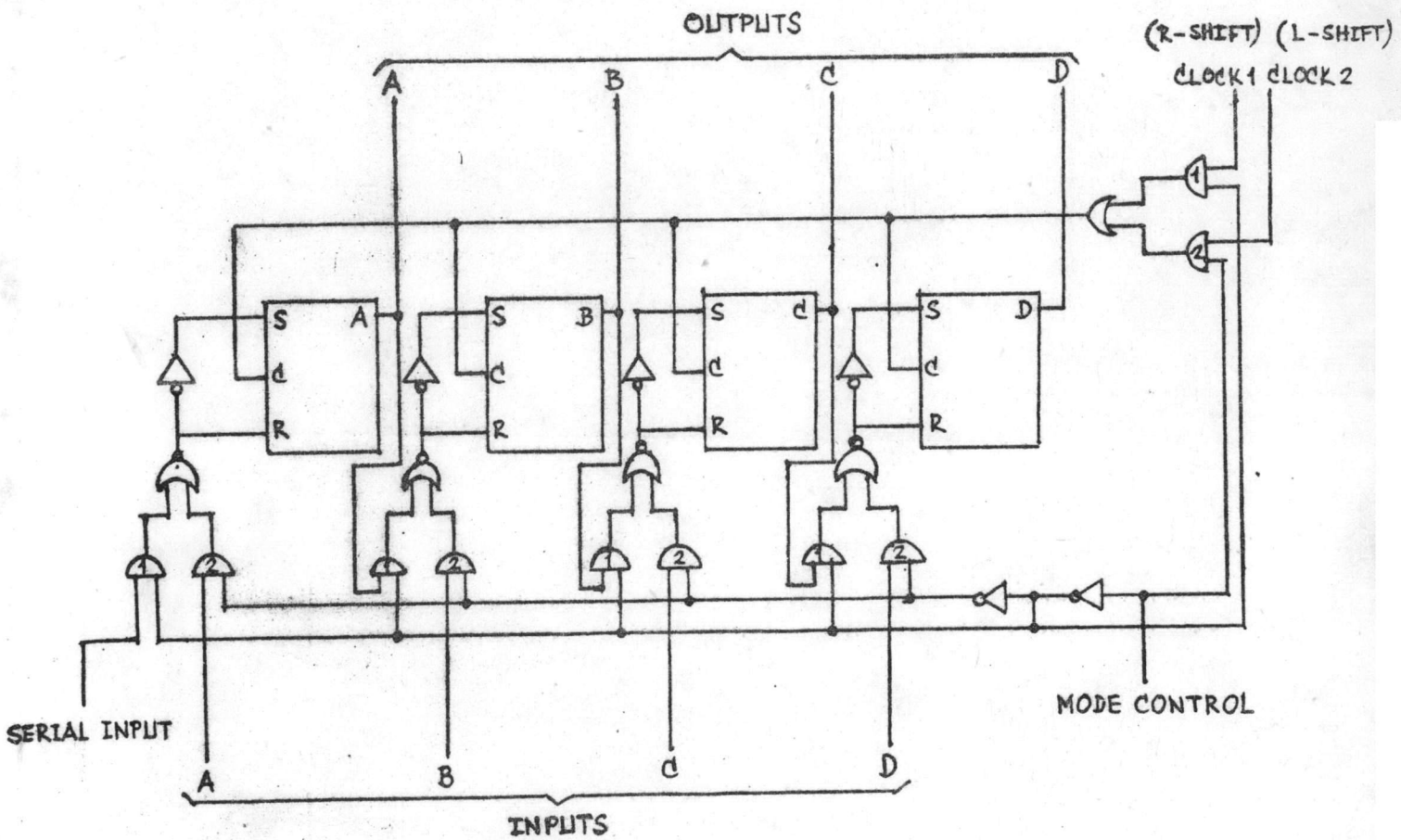
---

[15] Nashelsky, Louis; 1972. Introduction to Digital Computer Technology (New York: John Willey and Sons), p.345

2.11 วงจรรีจิสเตอร์ 4 บิท [16]

(4-Bit Right-Shift Left-Shift Register)

วงจรนี้ประกอบด้วย ฟลิพ-ฟล็อพ R-S มาสเตอร์-สเลพ และเกต AND-ORINVERT อย่างละ 4 และมีเกต AND-OR เกต Inverter ดังแสดงในรูปที่ 26



รูปที่ 26 วงจรรีจิสเตอร์ 4 บิท

การที่ข้อมูลจะ shift ทางขวาหรือทางซ้าย ขึ้นอยู่กับโลยิคอินพุทของตัวควบคุม mode (mode control) รีจิสเตอร์นี้เป็น IC เบอร์ SN 7495 N สามารถจะต่อแบบอนุกรมเป็นรีจิสเตอร์หลาย ๆ บิทได้ ทั้งยังใช้เป็นรีจิสเตอร์แบบ paralleled-in, parallel-out ได้ด้วย

---

[16] Texas Instruments Incorporated; 1969. TTL Integrated Circuits Catalog from Texas Instruments. p. 9-25.

เมื่อโลยิคอินพุทของ mode control เป็น 0 เกต AND เบอร์ 1 จะทำงานได้ (enable) แต่เกต AND เบอร์ 2 จะทำงานไม่ได้ (inhibit) ทำให้เกิดการ shift ทางขวา (right-shift) ตามลำดับ จากฟลิพ-ฟล็อพ A,B,C และ D การ shift ทางขวานี้เกิดจากอินพุทของคลอค 1 (Clock 1) โดยที่อินพุทเข้าทาง serial input

แต่เมื่อโลยิคอินพุทของ mode control เป็น 1 เกต AND เบอร์ 2 จะทำงานได้ และจากอินพุทของคลอค 2 (Clock 2) จะทำให้อินพุท A,B,C, D load เข้ารียิสเตอร์แบบขนาน และจะเกิดการ shift ทางซ้าย ถ้าต่อเอาพุทของแต่ละฟลิพ-ฟล็อพเข้ากับอินพุท แบบขนานของฟลิพ-ฟล็อพตัวก่อนหน้า (เช่น เอาพุทของ D ต่อกับอินพุทของ C เป็นต้น)

2.12 วงจรนับ บี ซี ดี [17]

(BCD or Synchronous 4-bit up/down Counter)

วงจรนับ บีซีดี หรือวงจรนับขึ้น/ลง 4 บิท เป็นวงจรนับที่มีโมดูลัส -10 (Decade Counter) เป็น IC เบอร์ SN 74192 N ในการวิจัยนี้นำมาใช้เป็นรีจิสเตอร์ 4 บิท ของ I/O รีจิสเตอร์เพราะต้องการให้รีจิสเตอร์เพิ่มตัวเลขได้ในการลบซึ่งมีตัวทดในบิทสุดท้าย (end-around carry) เกิดขึ้น

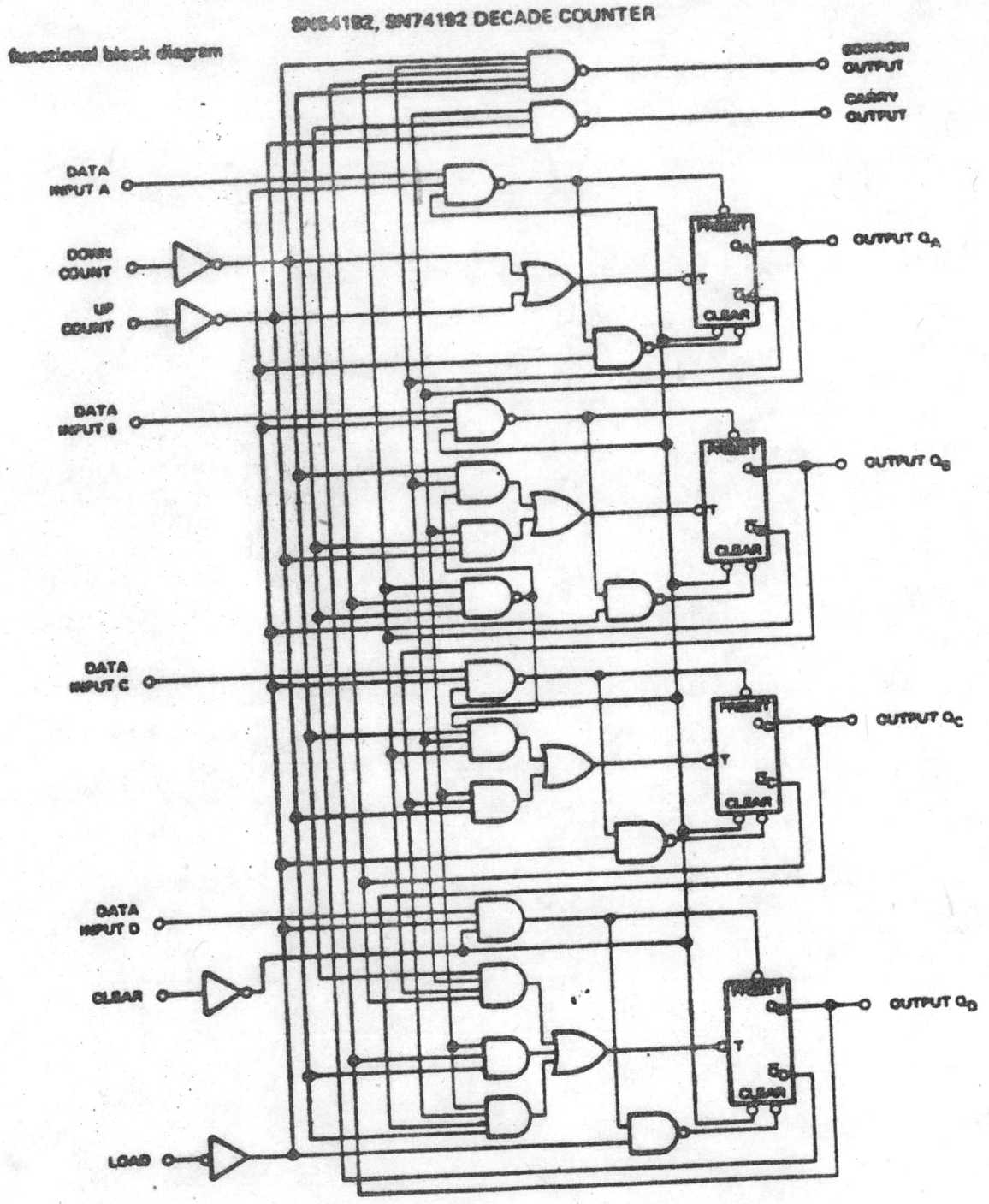
รูปของวงจรนับ บีซีดี มีแสดงในรูปที่ 27

วงจรนับนี้มีการทำงานแบบซินโครนัส (Synchronous Operation) คือฟลิพ-ฟล็อพ ทั้ง 4 จะถูกคล็อกในเวลาเดียวกัน

วงจรนับนี้จะนับขึ้น (up-count) จาก 0000 ถึง 1001 เมื่อมีพัลซ์อินพุทเข้าทาง up-count ในการวิจัยนี้ใช้การนับขึ้นและเมื่อตัวเลขในวงจรนับเป็น 1001 และมีพัลซ์อินพุทเข้าทาง up-count ตัวเลขจะเปลี่ยนเป็น 0000 และมีตัวทด (carry) เป็น 1 ออกทาง carry output และไปเป็นพัลซ์อินพุทในวงจรนับที่อยู่ถัดขึ้นไป ถ้าเป็นการนับลงถึง 0000 จะ มีการยืม (borrow) ทาง borrow output เกิดขึ้น

---

[17] Texas Instrument Incorporated, 1970. TTL Catalog Supplement from Texas Instruments. p. 8-1.



รูปที่ 27 วงจรนับบีซีดี (BCD Counter) SN 74192 N