

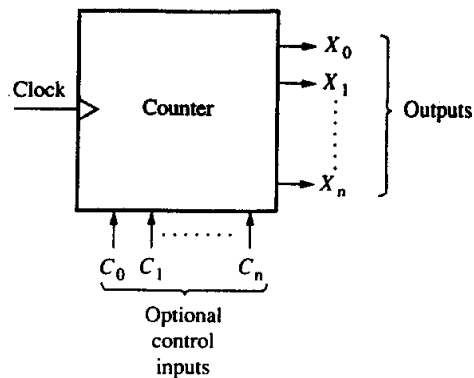
บทที่ 7 เคาน์เตอร์ Counters

บทนำ

วงจรถิบฟลอปเป็นวงจรถิบดิจิทัลซีเคิร์นเรียลเป็นวงจรถิบพื้นฐานในการออกแบบวงจรถิบเคาน์เตอร์ (Counter) เคาน์เตอร์ คือ วงจรถิบที่ใช้ในการนับตัวเลขของแต่ละครั้งที่มีการป้อนสัญญาณเข้ามาเพื่อให้ค่าของฟลิปฟลอปในวงจรถิบเคาน์เตอร์เปลี่ยนแปลงสัญลักษณ์โดยทั่วไปแสดงในรูปที่ 7.1 สัญญาณนาฬิกาที่ป้อนให้กับวงจรถิบเคาน์เตอร์จะทำการเปลี่ยนแปลงสถานะของวงจรถิบหรือเคาน์เตอร์ ฟลิปฟลอปทุกตัวในวงจรถิบจะต้องได้รับสัญญาณอินพุตเหมือนกันที่ป้อนให้กับฟลิปฟลอป แต่ถ้าอินพุตแตกต่างกัน แสดงว่าวงจรถิบมีหลายเอาพุต (ลาเบล X ในรูปที่ 7.1) ฟลิปฟลอปในวงจรถิบแต่ละตัวจะมีเอาพุตเพียงเอาพุตเดียวคือ Q ไม่รวม \bar{Q} ค่าสูงสุดของการนับขึ้นอยู่กับความสัมพันธ์ของ 2^n ขณะที่ n คือจำนวนเอาพุต ในกรณีวงจรถิบขนาด 3 บิต จะทำให้เกิดสถานะได้ 8 สถานะ และวงจรถิบขนาด 4 บิต จะทำให้เกิดสถานะได้ 16 สถานะ ค่าสูงสุดของการนับเราเรียกว่า Modulus ในบทนี้เราจะศึกษาการทำงาน การออกแบบ และการวิเคราะห์วงจรถิบชนิดต่างๆ วงจรถิบคคชทั่วไปแบ่งออกเป็น 2 ประเภทคือ Synchronous และ Asynchronous วงจรถิบแบบ Asynchronous เป็นวงจรถิบที่สามารถออกแบบได้ง่าย ส่วนวงจรถิบแบบ Synchronous โดยใช้เทคนิคในการออกแบบให้สามารถนับได้ตามต้องการ

เคาน์เตอร์เป็นส่วนสำคัญของวงจรถิบดิจิทัลอิเล็กทรอนิกส์ เป็นวงจรถิบชนิด Sequential Circuits มีคุณลักษณะดังต่อไปนี้

1. สามารถนับค่าได้สูงสุด
2. นับขึ้นและนับลง
3. ทำงานได้ทั้งซิงโครนัสและอซิงโครนัส
4. ทำงานได้อิสระ หรือหยุดโดยตัวเอง



รูปที่ 7.1 บล็อกไดอะแกรมของเคาน์เตอร์

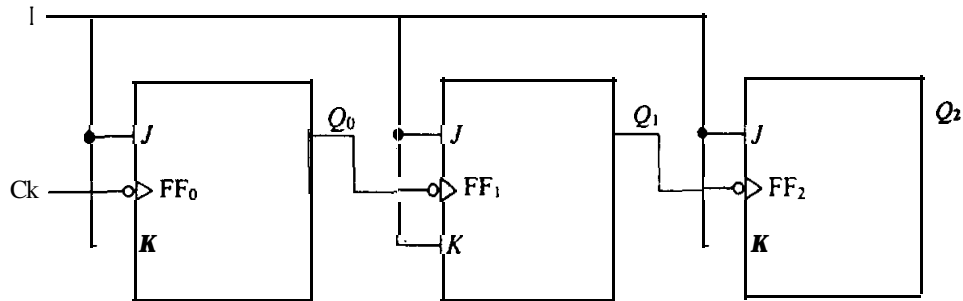
7.1 ASYNCHRONOUS COUNTERS

การนับระบบดิจิตอล จะเป็นการนับแบบไบนารี หรือรหัสไบนารี จากรูป 7.1 แสดงการนับเลขไบนารีเรียงลำดับจาก 0000 ถึง 1111 ถ้าเป็นเลขฐานสิบคือการนับจาก 0 ถึง 15 การนับในระบบดิจิตอลคือการนับ 0000 ถึง 1111 ที่แสดงในรูป เราเรียกว่า Modulo-16 Counter คำว่า Modulo ของตัวนับ คือจำนวนของการนับที่มีการนับถึง บางครั้งก็เรียกสั้นว่า MOD ตัวนับเรียกว่า MOD-16 COUNTER.

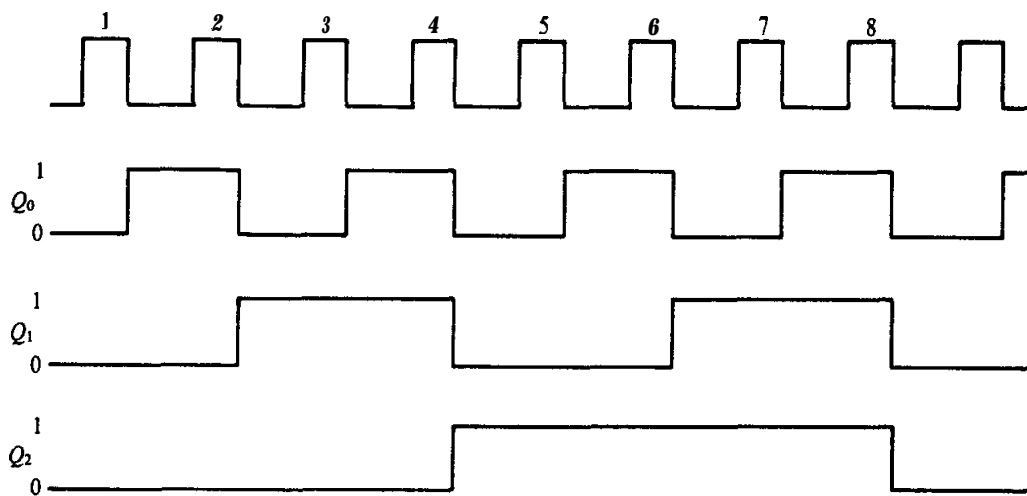
BINARY COUNT					BINARY COUNT				
DECIMAL	8	4	2	1	DECIMAL	8	4	2	1
COUNT	D	C	B	A	COUNT	D	C	B	A
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	0	0	0	12	1	1	0	0
5	0	0	0	1	13	1	1	0	1
6	0	0	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1

ตาราง 7.1 การนับเลขฐานสองขนาด 4 บิต

วงจรนับเลขขนาด 1 บิต โดยใช้ J-K Flip Flop โดยให้อินพุตของ J-K Flip Flop เป็นสภาวะทำงานตรงข้าม (Toggle) คือ $j = 1$ และ $K = 1$ เมื่อมีสัญญาณนาฬิกา 1 ครั้ง แสดงในรูป 7.2 เข้าพุตของ Q จะมีการนับขอบลบของสัญญาณนาฬิกา เข้าพุตของฟลิปฟลอป Q จะทำการเปลี่ยนแปลง 0 เป็น 1 หรือการเปลี่ยนจาก 1 เป็น 0 หรือจากรูปคลื่นที่แสดงจะเป็นการหาร 2 ของสัญญาณระบบ

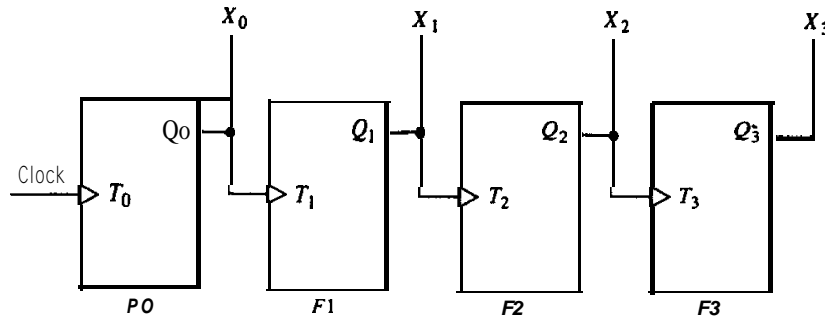


รูปที่ 7.2 วงจรเคาน์เตอร์แบบ RIPPLE (3 bit)



รูปที่ 7.2 (ต่อ) ไคอะแกรมเวลาของวงจรเคาน์เตอร์แบบ RIPPLE

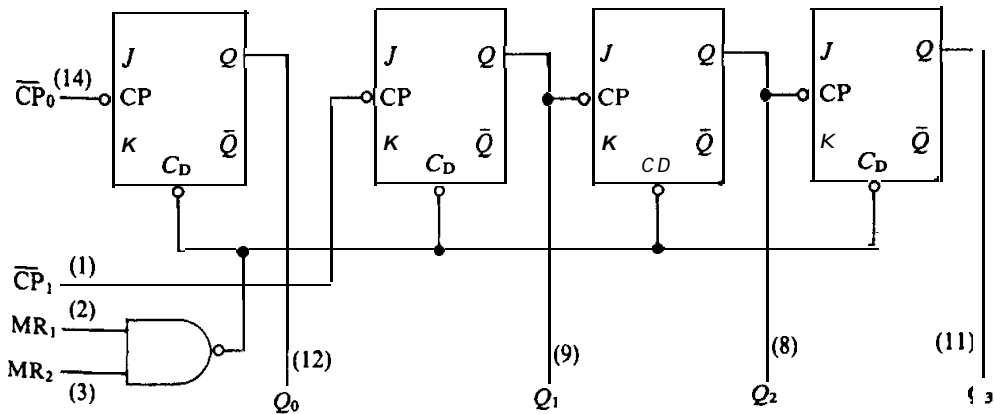
วงจรรนับแบบ RIPPLE โดยการเชื่อมต่อฟลิปฟลอปจำนวน N ตัว ดังแสดงในรูป 7.3 ผลการนับจะนับได้ 8 สถานะ คือ มีฟลิปฟลอป 3 ตัวจะได้ค่า 2 ยกกำลัง 3 จะเป็น 8 สถานะทางเข้าพุท หรือความสามารถในการนับ คือ $2^N - 1$ จากรูป 7.3 เป็นวงจรรนับ 8 สังเกตสัญญาณนาฬิกาที่กระตุ้น FF0 จะทำงานที่ขอบลบ (Negative edge) การทำงานจะได้ตามรูปของไคอะแกรมเวลาตามรูปที่ 15 แสดงรูปคลื่นเรียงลำดับที่สมบูรณ์ ซึ่งมี 3 เข้าพุท คือ $Q_2 Q_1 Q_0 = 000$



รูปที่ 7.3 7493 4 bit Binary Counter

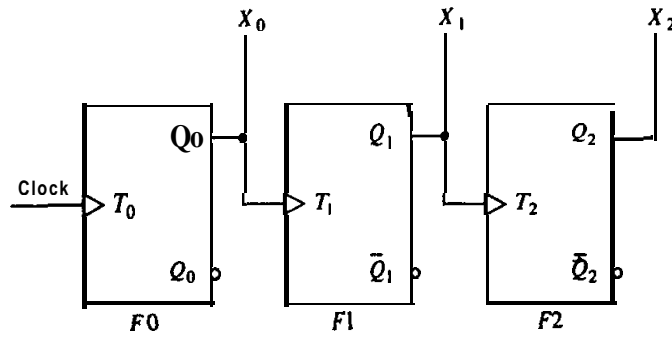
จากวงจรรเราจะเห็นว่าถ้ามีการนับค่า 000 ถึง 111 แล้วเราจะเห็นว่าการทำงานของวงจรรจะเริ่มต้นนับค่า 000 ใหม่ การเริ่มต้นนับใหม่นี้เราเรียกว่า RIPPLE COUNTER

จากรูป 7.4 แสดงการทำงานของวงจรรนับเลขไบนารีแบบ 4 บิต พร้อมไคอะแกรมเวลาที่เกิดขึ้นของเข้าพุทของวงจรร RIPPLE COUNTER

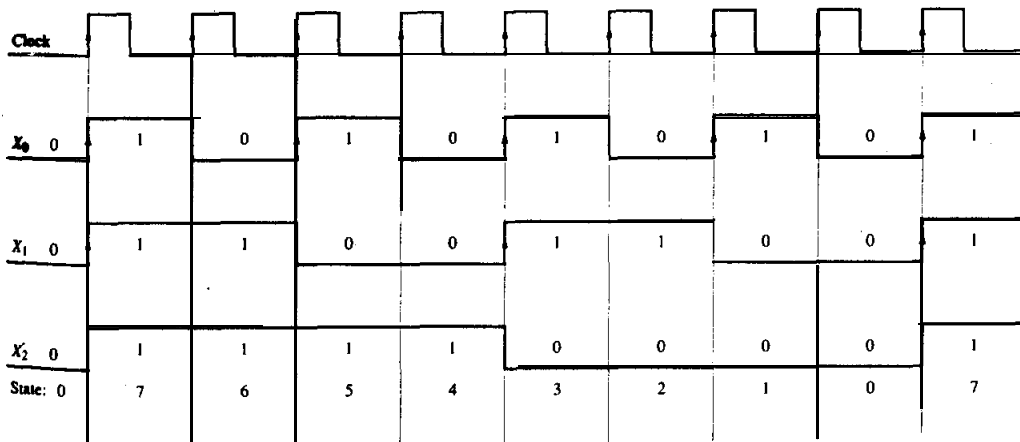


รูปที่ 7.4 A 4 BIT RIPPLE COUNTER

วงจรนับแบบ Asynchronous ที่สร้างขึ้นด้วย T Flip Flop โดยการเชื่อมต่อแบบอนุกรมดัง
แสดงในรูปที่ 7.5 โดยใช้ T Flip Flop แทน J-K และ D Flip Flop

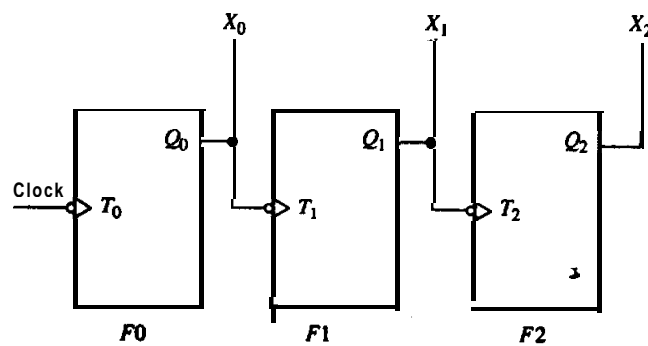


รูปที่ 7.5 3 บิตวงจรนับลง

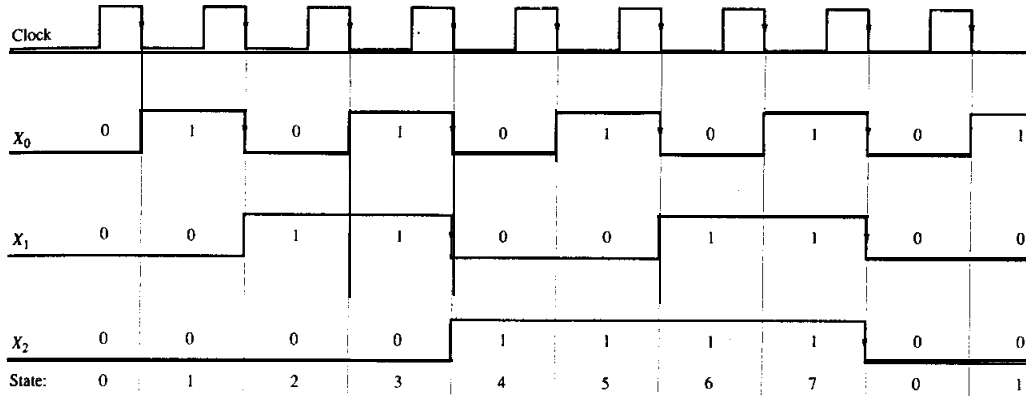


รูปที่ 7.6 ไตอะแกรมเวลาของวงจรนับลง

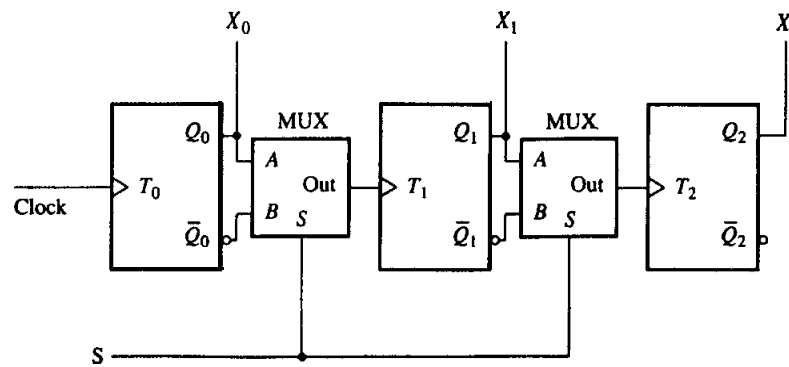
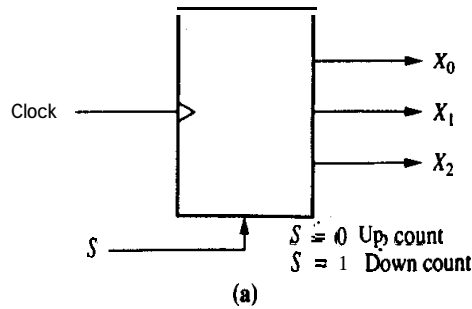
7.2 RIPPLE COUNTER



รูปที่ 7.7 วงจรนับขึ้นชนิด Ripple up Counter



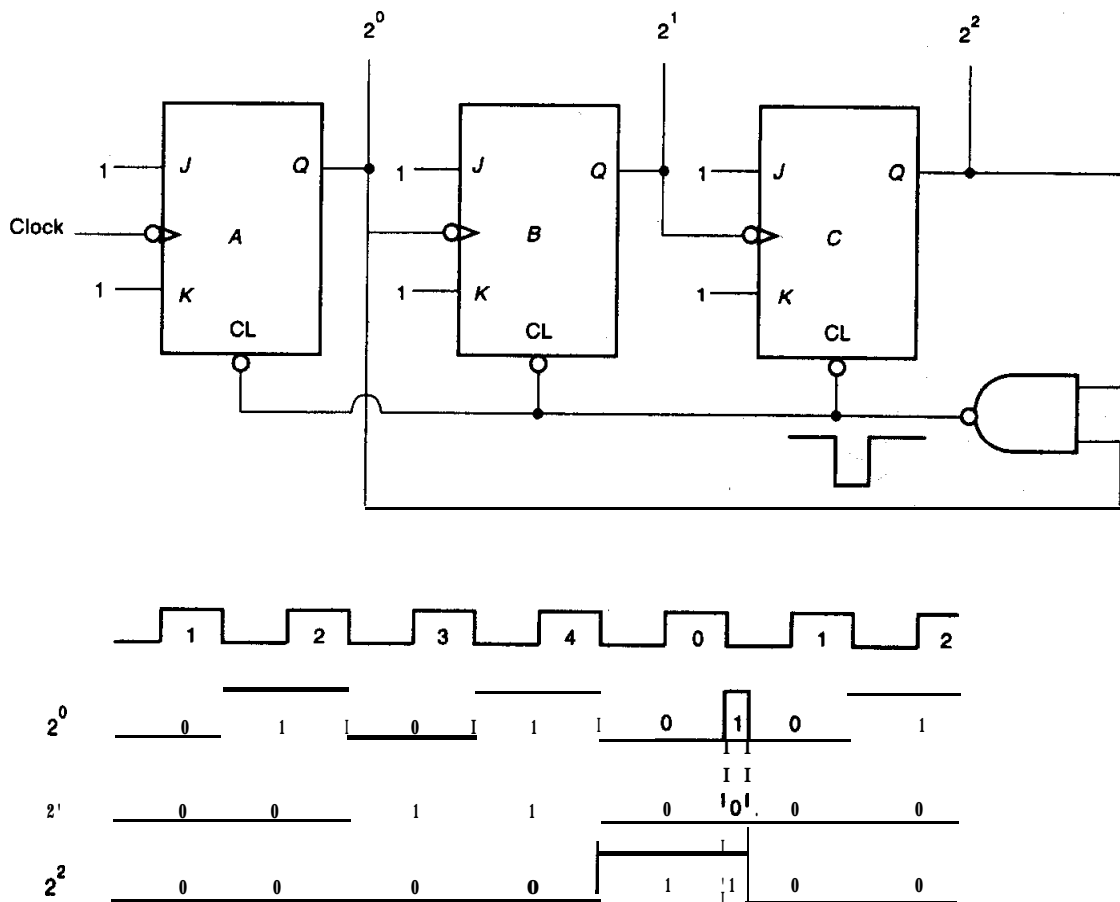
รูปที่ 7.8. ไตอะแกรมเวลาของวงจรนับขึ้นชนิด Ripple
7.3 วงจรนับขึ้นและนับลง (Up/Down Counter)



รูปที่ 7.9 An Asynchronous Up/Down Counter

7.4 ASYNCHRONOUS DECADE COUNTER

การออกแบบแคว้นเตอร์ในรูปที่ 7.10 ซึ่งเป็นแบบ RIPPLE COUNTER แต่เรานำมาทำเป็นตัวนับแบบ DECADE โดยวิธีการเคลียเข้าพุตตามความต้องการที่จะให้นำเท่าไร จากรูปเป็นวงจร Divide by 5 decode and clear counter โดยใช้ลอจิก NAND GATE เป็นตัวทำหน้าที่ให้เข้าพุตเคลียเป็นค่า LOW หรือค่า 0 เมื่อมีการนับจำนวนถึงครั้งที่ 5 จากรูปคลื่นที่แสดงในรูป จำนวนค่า 5 จะเกิดขึ้นเพียงนิดเดียว ก่อนที่แคว้นเตอร์จะมีค่าเป็น 0 กรณีเข้าพุตมีค่าเป็น 0 วงจร NAND GATE จะมีค่าเป็น HIGH และแคว้นเตอร์เริ่มนับ 000 สำหรับการทำงานในพัลส์ต่อไป การทำงานของวงจรมีจะเคลียในพัลส์ที่ 5



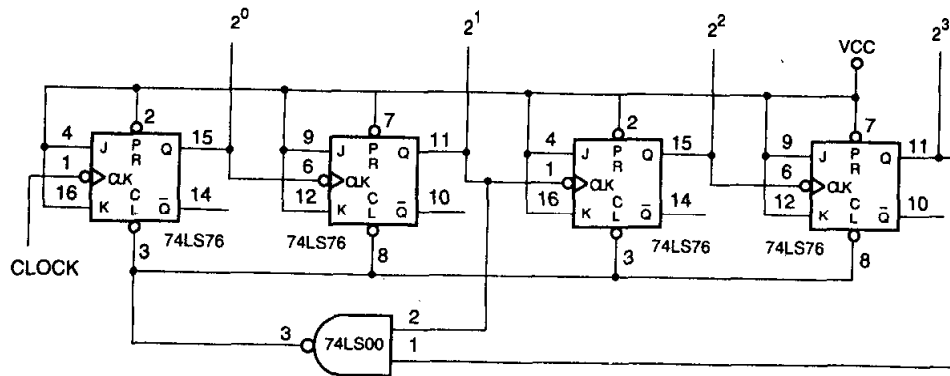
รูปที่ 7.10 Divide by 5 decode and clear counter

ตัวอย่าง 7.1 การออกแบบ Divide by 10 decode and clear counter

การทำงานของวงจรจะมีการนับ 0 ถึง 9 และเริ่มต้นนับที่ 0 ใหม่ เคาน์เตอร์จะทำการเคลียฮั่วทุกของฟลิปฟลอปทุกตัวเมื่อนับค่าที่ 9 แล้ว นักออกแบบจะต้องทำการถอดรหัสค่า 10 ให้ทำการเคลียฟลิปฟลอป ฉะนั้นการนับถึงค่า 10 ต้องใช้ฟลิปฟลอป 4 ตัว สังเกตจากตารางไบนารีในการนับ 0000 ถึง 1010 ในการถอดรหัสจะต้องใช้ NAND GATE ทำการเคลียค่า 10 ให้เป็นค่า 0

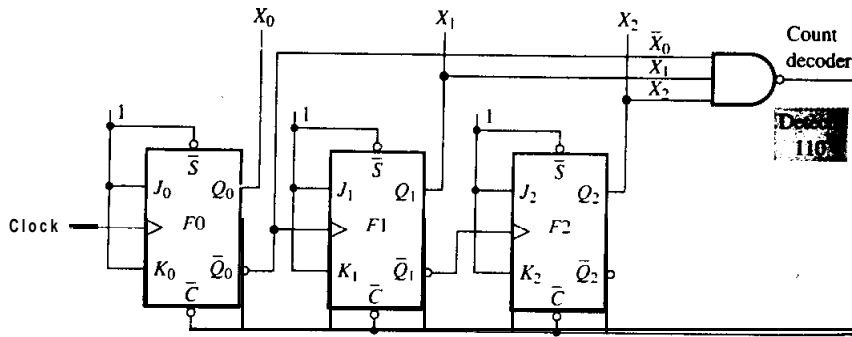
Decimal Binary

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	** 1010

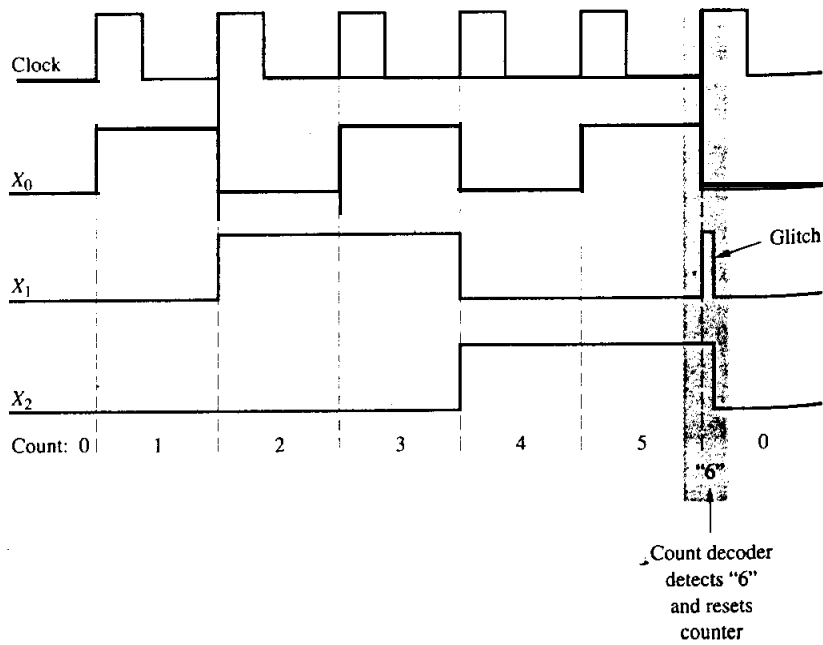


รูปที่ 7.11 วงจร DIVIDE BY DECODE AND CLEAR COUNTER

ตัวอย่าง 7.2 วงจรนับที่แสดงในรูป 7.12 เป็นชนิด 3 บิต Asynchronous Counter ที่นับ 0-5 (000 - 101) และเริ่มนับ 0 ใหม่ โดยใช้ J-K Flip Flop ทำงานแทน T Flip Flop

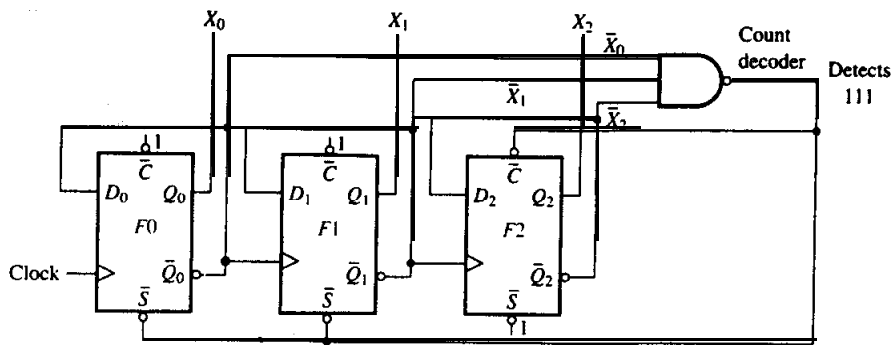


รูปที่ 7.12. วงจรนับชนิด 3 บิต

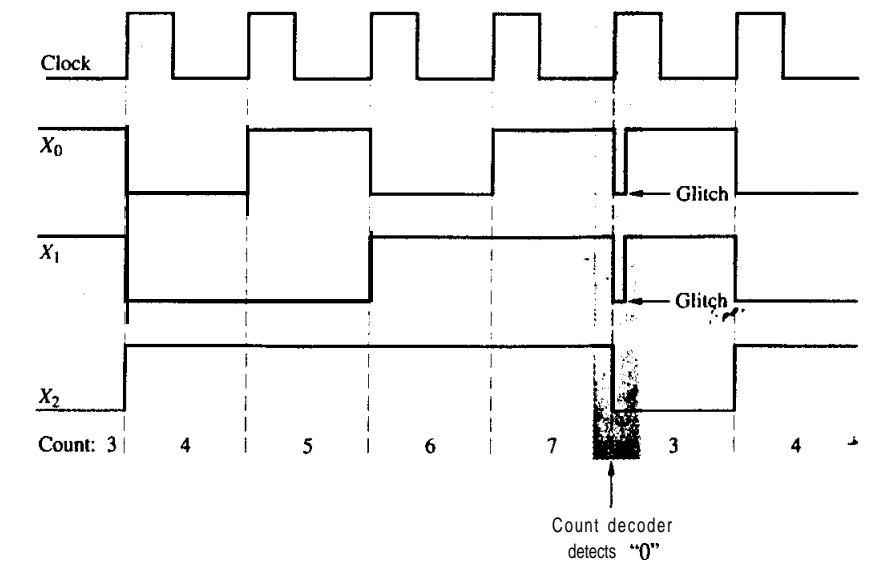


รูปที่ 7.13 ไคอะแกรมเวลาของวงจรถับ 3 บิต

ตัวอย่าง 7.4 วงจรนับที่แสดงในรูป 7.14. เป็นชนิด 3 บิต Asynchronous Counter ที่นับ 3-7 (011 - 111) และเริ่มนับ 0 ใหม่ โดยใช้ D Flip Flop



รูปที่ 7.14 วงจรนับ 3 บิตโดยใช้ D Flip Flop



รูปที่ 7.15 ไคอะแกรมเวลาของวงจรรุ่นชนิด 3 บิต

7.5 Down Counter

วงจรมีเราสามารถนับถอยหลังจาก 9, 8, 7, 6, -----, 2, 1, 0 เป็นการนับจากค่าสูงสุดมายังค่าต่ำ เราเรียกว่า Down Counter จากตัวอย่างของวงจรมี Mod - 8 Asynchronous Down Counter เป็นวงจรมีถอยหลัง ในรูป 7.16 (a) ส่วนในรูป 7.16 (b) เป็นตารางแสดงการนับถอยหลัง การทำงานที่แตกต่างกับการนับขึ้นของวงจรมี Up Counter คือ ตัว Carry จากตัว FF1 ไปยัง FF2 และตัว FF2 ไปยังตัว FF3 ของ Down Counter ใช้ Q bar ส่วน Up Counter ใช้ Q เป็นสัญญาณนาฬิกาที่จะไปกระตุ้นฟลิปฟลอปตัวต่อไป สังเกตว่า Down Counter จะมี Preset ควบคุมในการเซ็ตค่าเอาพุตของฟลิปฟลอปให้มีค่าเป็น 111

วงจรมีน้อยลงชนิด 3 บิตจะนับเลขฐานสองเป็น 111, 110, 101, 100, 011, 010, 001, 000 และวนรอบกลับมานับ 111 ใหม่อีกครั้งหนึ่ง การทำงานของวงจรมีน้อยลงให้สังเกตจากวงจรมีเราจะเห็นว่าตัวกระตุ้นสัญญาณของฟลิปฟลอป (Trigger line) จากฟลิปฟลอปตัวที่ 1 ไปยังตัวที่ 2 เราจะใช้เอาพุต Q bar เป็นสัญญาณกระตุ้นฟลิปฟลอปตัวต่อไป ต่างกับสัญญาณนับขึ้นเราใช้สัญญาณ Q เป็นตัวกระตุ้นการทำงานของฟลิปฟลอปตัวต่อไป การทำงานของ J - K Flip Flop ในวงจรมีขึ้นนับลงเหมือนกัน คือถ้ามีสัญญาณมากระตุ้นฟลิปฟลอปที่มีค่า J - K = 1 จะอยู่ในสภาวะการทำงานที่ทำให้เกิดค่าตรงกันข้าม (Toggle) จะช่วยให้การเข้าใจการทำงานของวงจรมีลง ดูจากรูปคลื่นของสัญญาณ การทำงานในแต่ละช่วงเวลา บรรทัดบนสุดเป็นสัญญาณนาฬิกา (CLK) เป็นอินพุตของ FF1 ส่วนบรรทัดล่างสุดเป็นการนับเลขฐานสอง ซึ่งเริ่มจาก 111 ด้านซ้ายสุดของเอาพุต Q และ Q bar ที่แสดงในฟลิปฟลอป FF1 และ FF2 แต่สำหรับเอาพุต FF3 แสดงเฉพาะ Q เท่านั้น

จากรูปพัลส์รูปที่ 1 จะพบว่าฟลิปฟลอปอยู่ในสภาวะเซ็ต (SET) เอาพุตของฟลิปฟลอปแต่ละตัวจะมีค่าเป็น 111 การเปลี่ยนสถานะของพัลส์ 1 จากสูงไปเป็นค่าต่ำ 0 ทำให้ FF1 เปลี่ยนสถานะจากเดิมคือเปลี่ยนจากสูงไปเป็นค่าต่ำ หรือ 1 ไปเป็น 0 เอาพุตของวงจรมีจะเป็น 110

พัลส์รูปต่อไป การเปลี่ยนแปลงของพัลส์นาฬิกาจาก 1 ไปเป็น 0 FF1 จะเปลี่ยนสถานะเป็นผลให้ค่าเอาพุต Q เปลี่ยนจาก 0 ไปเป็น 1 ส่วน Q bar เปลี่ยนจาก 1 ไปเป็น 0 อันเป็นผลให้ FF2 เปลี่ยนสถานะ โดยเอาพุต Q เปลี่ยนจาก 1 ไปเป็น 0 ฉะนั้นเอาพุตของวงจรมีจะมีค่าเป็น 101

เมื่อพิจารณาการทำงานแล้วพัลส์รูปที่ 3 จะกระตุ้น FF1 ทำให้เอาพุต Q ของ FF1 เปลี่ยนจาก 1 ไปเป็น 0 แต่ Q bar เปลี่ยนจากต่ำไปเป็นสูง หรือ 0 ไปเป็น 1 เอาพุตของวงจรมีจะเป็น 100

พัลส์รูปที่ 4 มากระตุ้นการทำงานของ FF1 FF1 จะถูกเซ็ตและเอาพุตของ Q bar จะเปลี่ยนจาก 1 ไปเป็น 0 เป็นผลทำให้ FF2 เปลี่ยนสถานะ โดย FF2 อยู่ในสภาวะเซ็ต และเอาพุต Q bar เปลี่ยน

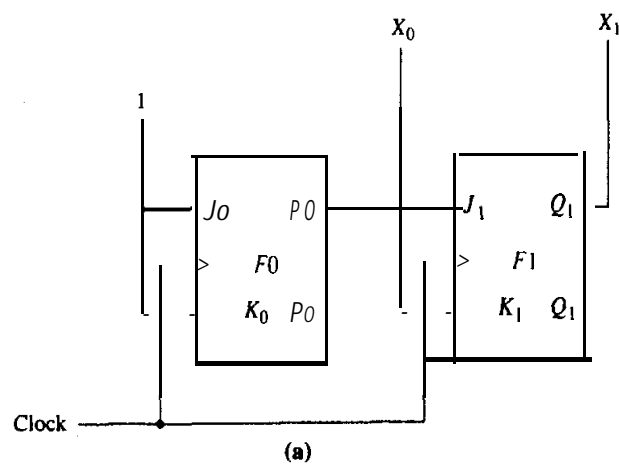
จาก 1 ไปเป็น 0 เป็นผลให้ FF3 เปลี่ยนสถานะเป็นรีเซ็ต (RESET) ให้อุปกรณ์ทำงานของพัลส์ ลูกที่ 4 จะเป็น 011

ส่วนรูปคลื่นที่เหลือจะมีการทำงานในทำนองเดียวกัน ให้อุปกรณ์เปลี่ยนแปลงขึ้นอยู่กับสภาวะ อินพุตของฟลิปฟลอปและสัญญาณกระตุ้นการทำงานของฟลิปฟลอปแต่ละตัว

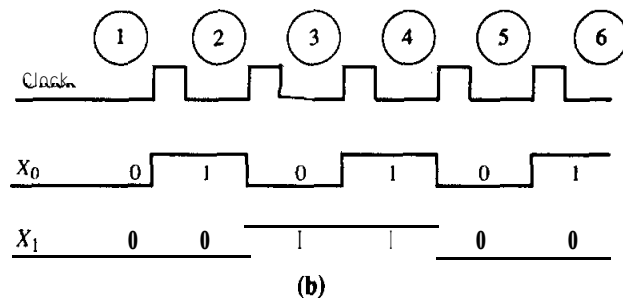
7.7 SYNCHRONOUS COUNTER

วงจรรนับแบบซิงโครนัส (SYNCHRONOUS) การทำงานของฟลิปฟลอปแต่ละตัวจะไม่ถูกกระตุ้นพร้อมกันขณะที่มีการป้อนสัญญาณนาฬิกาของระบบเข้ามา สำหรับการงานในความถี่สูงๆ บางความถี่จำเป็นต้องใช้ฟลิปฟลอปแต่ละตัวของวงจรรนับได้รับการกระตุ้นพร้อมกัน วงจรแบบนี้เป็นวงจรรนับแบบซิงโครนัส จากรูป 7.18 ก เป็นวงจรรนับแบบซิงโครนัส ซึ่งดูแล้วค่อนข้างยุ่งยากเป็นการทำงานของการนับ MOD-8 หรือวงจรรนับขนาด 3 บิต ครั้งแรกให้สังเกตการต่อสัญญาณ CLOCK เข้ากับอินพุตของฟลิปฟลอปแต่ละตัวจะเป็นการต่อ CLOCK แบบขนานซึ่งทุกตัวจะได้รับพร้อมกัน ส่วนในรูป 7.18 ข เป็นตารางแสดงการลำดับการนับของวงจรรนับนี้ คอลัมน์ A เป็นคอลัมน์เลขฐานสอง 1s และ FF1 จะนับเลขคอลัมน์นี้ ส่วนคอลัมน์ B เป็นการนับเลขฐานสอง 2s และ FF2 จะนับเลขคอลัมน์นี้ คอลัมน์ C เป็นหลัก 3s และ FF3 จะนับเลขฐานสองคอลัมน์นี้

ตัวอย่าง 7.5 วงจรรนับแบบ Synchronous ชนิด 2 บิต

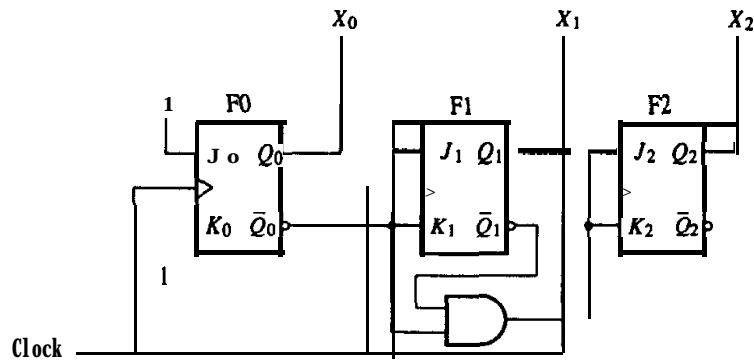


รูปที่ 7.16 a) ลอจิกไดอะแกรม

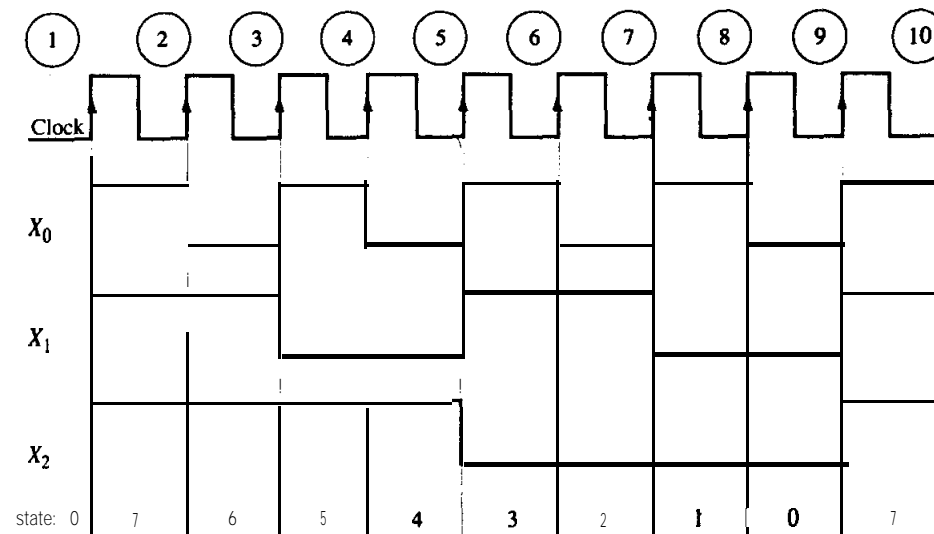


รูปที่ 7.16 b) ไคอะแกรมเวลาของวงจรนับ

ตัวอย่าง 7.6 วงจรนับ 3 bit synchronous binary down counter

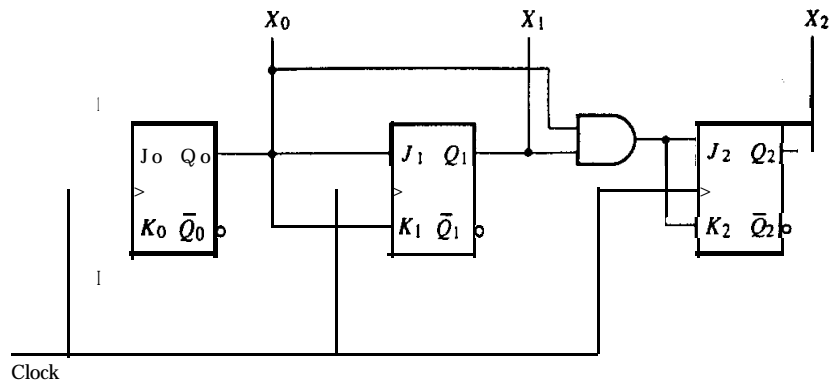


รูปที่ 7.17 ลอจิกไคอะแกรม

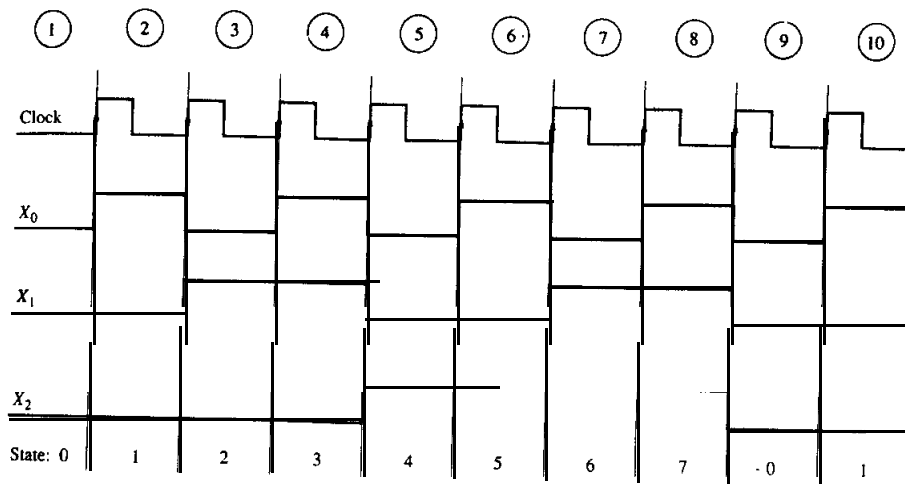


รูปที่ 7.18 ไคอะแกรมเวลาของวงจรนับ 3 บิตแบบ Synchronous

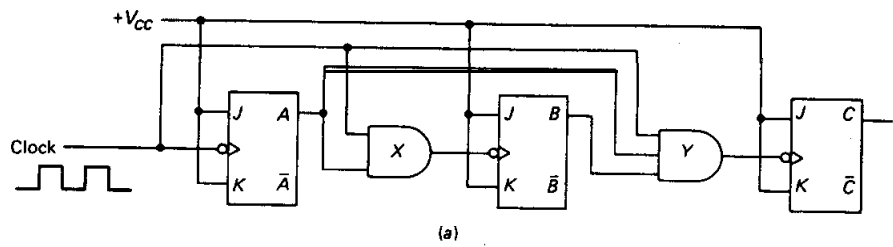
ตัวอย่าง 7.7 วงจรนับ 3 bit synchronous **binary** Up counter



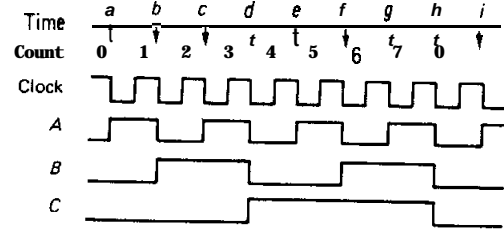
รูปที่ 7.19 ลอจิกไออะแกรม



รูปที่ 7.20 ไคอะแกรมเวลาของวงจรถับ 3 บิตแบบ synchronous



C	B	A	Count
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7
0	0	0	



รูปที่ 7.21 วงจรนับขนาด 3 บิตแบบซิงโครนัส a) วงจรฟลิปฟลอป b) ตารางการนับ

c) ไคอะแกรม

ต่อไปนี้จะตรวจสอบลำดับการนับของวงจรถับแบบ MOD-8 โดยยึดรูป 7.21 a) และ b) เป็นแนวทางในการอธิบายการนับ

พัลส์ 1 แถวที่ 2

การทำงานของวงจรถับ : ฟลิปฟลอปแต่ละตัวได้รับสัญญาณนาฬิกา มีฟลิปฟลอป FF1 เท่านั้นที่มีการเปลี่ยนแปลงเอาพุต เพราะค่า 1 นั้นส่งไปยัง J-K ของอินพุตฟลิปฟลอปตัวที่ 1

เอาพุตของฟลิปฟลอป : จะมีค่า 001

พัลส์ 2 แถวที่ 3

การทำงานของวงจรถับ : ฟลิปฟลอปแต่ละตัวได้รับสัญญาณนาฬิกา มีฟลิปฟลอป 2 ตัวคือ FF1 และ FF2 เท่านั้นที่มีการเปลี่ยนแปลงเอาพุต เพราะค่า 1 นั้นส่งไปยัง J-K ของอินพุตฟลิปฟลอปตัวที่ 1 และตัวที่ 2

FF1 เปลี่ยนจาก 1 ไปเป็น 0

FF2 เปลี่ยนจาก 0 ไปเป็น 1

เอาพุตของฟลิปฟลอป : จะมีค่า 010

พัลส์ 3 แฉวที่ 4

การทำงานของวงจร : ฟลิปฟลอปแต่ละตัวได้รับสัญญาณนาฬิกา มีฟลิปฟลอบ FF1 เท่านั้นที่มีการเปลี่ยนแปลงเอาพุต เพราะค่า 1 นั้นส่งไปยัง J-K ของอินพุตฟลิปฟลอบตัวที่ 1

FF1 เปลี่ยนจาก 0 ไปเป็น 1

เอาพุตของฟลิปฟลอบ : จะมีค่า 011

พัลส์ 4 แฉวที่ 5

การทำงานของวงจร : ฟลิปฟลอปแต่ละตัวได้รับสัญญาณนาฬิกา มีฟลิปฟลอบทุกตัวคือ FF1 FF2 และ FF3 ที่มีการเปลี่ยนแปลงเอาพุต เพราะค่า 1 นั้นส่งไปยัง J-K ของอินพุตฟลิปฟลอบทุกตัว

FF1 เปลี่ยนจาก 1 ไปเป็น 0

FF2 เปลี่ยนจาก 1 ไปเป็น 0

FF3 เปลี่ยนจาก 0 ไปเป็น 1

เอาพุตของฟลิปฟลอบ : จะมีค่า 100

พัลส์ 5 แฉวที่ 6

การทำงานของวงจร : ฟลิปฟลอปแต่ละตัวได้รับสัญญาณนาฬิกา มีฟลิปฟลอบ 1 ตัวคือ FF1 เท่านั้นที่มีการเปลี่ยนแปลงเอาพุต เพราะค่า 1 นั้นส่งไปยัง J-K ของอินพุตฟลิปฟลอบตัวที่ 1

FF1 เปลี่ยนจาก 0 ไปเป็น 1

เอาพุตของฟลิปฟลอบ : จะมีค่า 101

พัลส์ 6 แฉวที่ 7

การทำงานของวงจร : ฟลิปฟลอปแต่ละตัวได้รับสัญญาณนาฬิกา มีฟลิปฟลอบ 2 ตัวคือ FF1 และ FF2 เท่านั้นที่มีการเปลี่ยนแปลงเอาพุต เพราะค่า 1 นั้นส่งไปยัง J-K ของอินพุตฟลิปฟลอบตัวที่ 1 และตัวที่ 2

FF1 เปลี่ยนจาก 1 ไปเป็น 0

FF2 เปลี่ยนจาก 0 ไปเป็น 1

เอาพุตของฟลิปฟลอบ : จะมีค่า 110

พัลส์ 7 แฉวที่ 8

การทำงานของวงจร : ฟลิปฟลอปแต่ละตัวได้รับสัญญาณนาฬิกา มีฟลิปฟลอบ 1 ตัวคือ FF1 ที่มีการเปลี่ยนแปลงเอาพุต เพราะค่า 1 นั้นส่งไปยัง J-K ของอินพุตฟลิปฟลอบตัวที่ 1

FF1 เปลี่ยนจาก 0 ไปเป็น 1

เอาพุตของฟลิปฟลอบ : จะมีค่า 111

พัลส์ 8 แฉวที่ 9

การทำงานของวงจร : ฟลิปฟลอปแต่ละตัวได้รับสัญญาณนาฬิกา มีฟลิปฟลอปทุกตัวคือ FF1 FF2 และ FF3 ที่มีการเปลี่ยนแปลงเข้าชุด เพราะค่า 1 นั้นส่งไปยัง J-K ของอินพุตฟลิปฟลอปทุกตัว

FF1 เปลี่ยนจาก 1 ไปเป็น 0

FF2 เปลี่ยนจาก 1 ไปเป็น 0

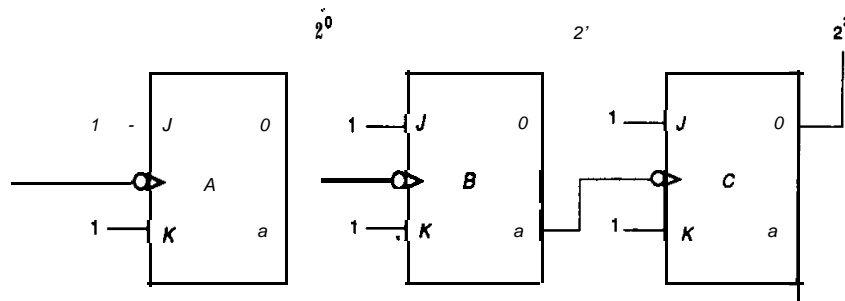
FF3 เปลี่ยนจาก 1 ไปเป็น 0

เข้าชุดของฟลิปฟลอป : จะมีค่า 000

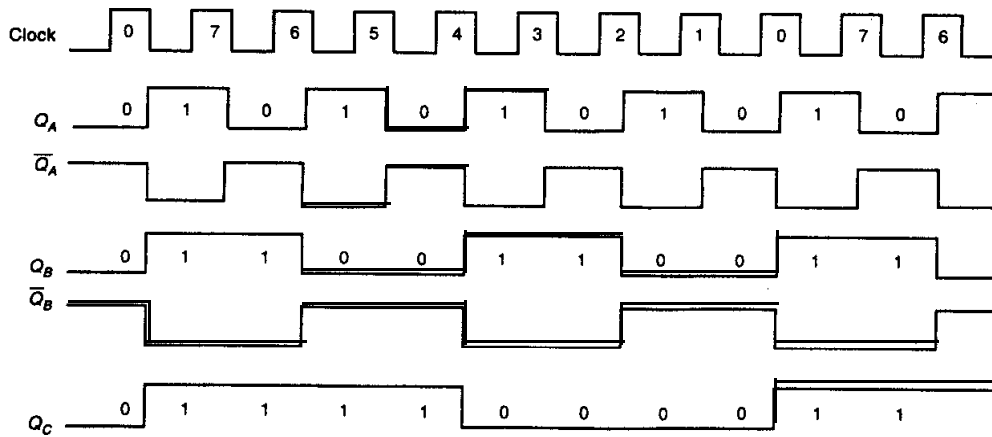
การอธิบายทั้งหมดนั้น เป็นการแสดงการทำงานของวงจรรนับแบบซิงโครนัส สักคำว่าฟลิปฟลอปแต่ละตัวถ้า $J = K = 1$ จะทำให้เข้าชุดฟลิปฟลอปกับค่าเป็นตรงกันข้าม ส่วนกรณีถ้า $J = K = 0$ เข้าชุดของฟลิปฟลอปไม่เปลี่ยนแปลง

7.8 SYNCHRONOUS DOWN COUNTER

วงจรรนับลงของซิงโครนัสแสดงในรูปที่ 7.25 จะมีลจิกเกตในควบคุมการทำงานของฟลิปฟลอป ที่จะทำให้ \bar{Q} ของเข้าชุดฟลิปฟลอปนับลง



รูปที่ 7.22 a) วงจรลอจิกนับลง 3 บิตแบบซิงโครนัส



รูปที่ 7.22 a) ไคอะแกรมเวลาวงจรลอจิกนับลง 3 บิตแบบซิงโครนัส

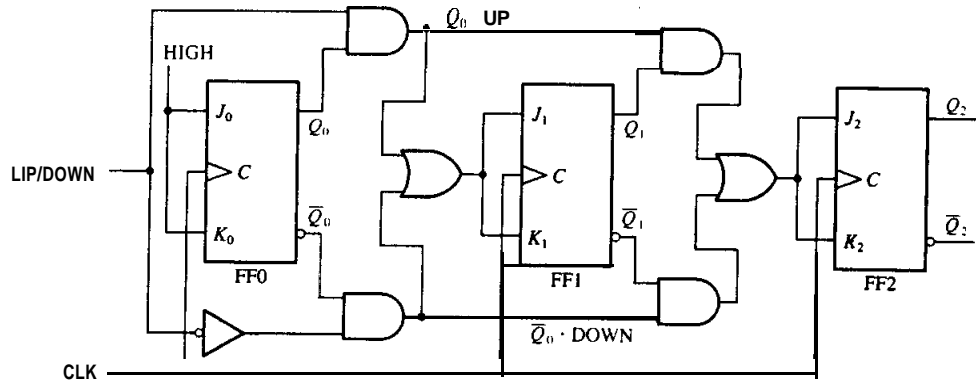
7.9 THE UP - DOWN COUNTER

เป็นวงจรมานับที่สามารถนับขึ้นและนับลงได้ จากรูป 7.23 วงจรมานับขึ้นนับลง จะมีสัญญาณในการควบคุมการนับขึ้นหรือนับลง

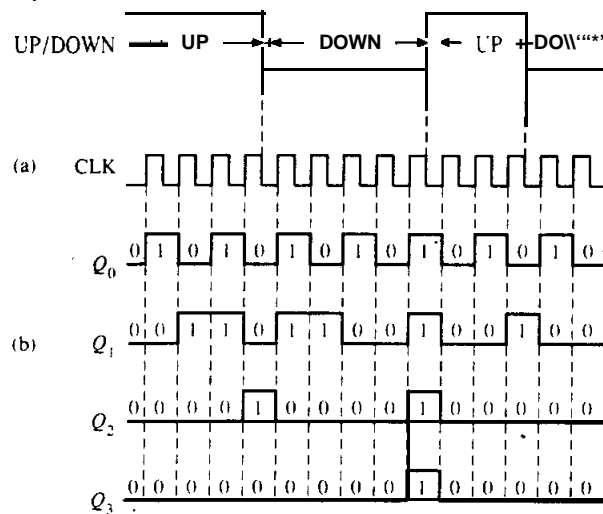
ตารางการนับขึ้นนับลงของวงจรมานับชนิด 3 บิต

up
up
 0, 1, 2, 3, 4, 5, 4, 3, 2, 3, 4, 5, 6, 7, 6, 5, etc.
} down
} down

Clock Pulse	UP	Q_2	Q_1	Q_0	DOWN
0	}	0	0	0	}
1		0	0	1	
2		0	1	0	
3		0	1	1	
4		1	0	0	
5		1	0	1	
6		1	1	0	
7		1	1	1	



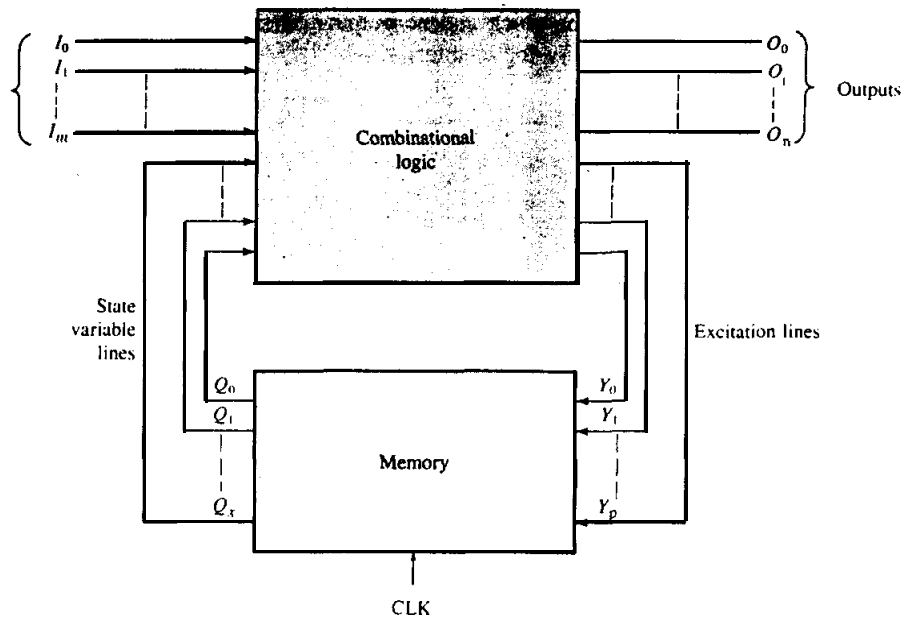
รูป 7.23 SYNCHRONOUS UP - DOWN COUNTER



รูปที่ 7.24 ไคอะแกรมเวลาของวงจรนับ

การออกแบบวงจรซีเควินเชียล

ก่อนที่จะเริ่มทำการออกแบบวงจรซีเควินเชียล เราจะต้องทำความเข้าใจวงจรซีเควินเชียล (Sequential circuit or State machine) วงจรซีเควินเชียลประกอบด้วย 2 ส่วนคือ ส่วนที่หนึ่งเป็นวงจรคอมไบเนชันและส่วนที่สองจะเป็นหน่วยความจำ (Memory) ที่เรียกว่า Flip Flop ที่แสดงในรูปที่ 7.25



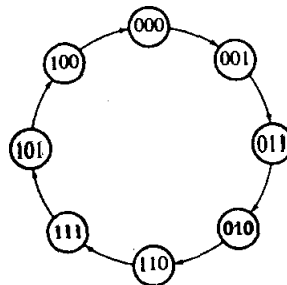
รูปที่ 7.25 General Clocked sequential circuit

การจัดเก็บข้อมูลในส่วนของหน่วยความจำมาจากอินพุตของวงจรถอมไบเนชันลอจิกคือ ($I_0, I_1, I_2, \dots, I_n$) เพื่อให้วงจรทำงานตามความต้องการ ข้อมูลที่อยู่ในส่วนของหน่วยความจำเรียกว่า Present state และสามารถเปลี่ยนแปลงเป็น Next state เมื่อมีสัญญาณนาฬิกาเข้ามากระตุ้นที่ (Y_0, Y_1, \dots, Y_n) Present state ของหน่วยความจำจะแทนที่ด้วย State variables คือ (Q_0, Q_1, \dots, Q_n) ค่าของ State variable เหล่านี้ขึ้นอยู่กับค่าของอินพุต (I_0, I_1, \dots, I_n)

ขั้นตอนการออกแบบมีดังต่อไปนี้

1. เขียน State Diagram

ตัวอย่างการเขียน State Diagram ของ 3 bit Gray code counter โดยการนับดังนี้
000 001 011 010 110 111 101 100 000 001 และวนรอบซ้ำๆกัน



ขั้นตอนที่ 2 Next state table

วงจรถ่ายเลขฐานสิบจะกำหนดการทำงานด้วย State diagram และขั้นตอนที่ 2 จะเขียนตาราง Next state table ที่จะเกิดขึ้นของแต่ละสถานะของวงจรถ่ายเลขฐานสิบจาก Present state เป็น Next state การเปลี่ยนแปลงให้ขึ้นอยู่กับสัญญาณนาฬิกา ดูจากตารางของการเปลี่ยนแปลงรหัส Gray code counter

Present state			Next state		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	0	0	1
0	0	1	0	1	1
0	1	1	0	1	0
0	1	0	1	1	0
1	1	0	1	1	1
1	0	1	1	0	0
1	0	0	0	0	0

ขั้นตอนที่ 3 Transition table

ขั้นตอนที่ 3 เป็นการพัฒนาในส่วนของการเปลี่ยนค่าของ Present state เป็น next state หรือเรียกว่า Transition table ในขั้นตอนนี้จะต้องกำหนดฟลิปฟล็อปที่ต้องการที่จะนำมาใช้วงจรถ่ายเลข

ถ้าฟลิปฟล็อปจะ Reset ($Q = 0$) ในกรณี ถ้าฟลิปฟล็อปมีค่า Present state เป็นค่า 0 และค่าของ Next state ต้องการให้เป็นค่า 0 ($0 \rightarrow 0$) เราจะเห็นว่าค่าเข้าพุตของฟลิปฟล็อปไม่มีการเปลี่ยนแปลง (No change) ($J = 0, K = 0$) หรืออยู่ในเงื่อนไข Reset ($J = 0, K = 1$) เมื่อมีสัญญาณนาฬิกาเข้ามา ก็จะทำให้เปลี่ยนแปลงตามอินพุต จากกรณีเราจะเห็นว่าค่าของ K อาจจะเป็นค่า 1 หรือ ค่า 0 ได้ เราเรียกว่า Do Not care ส่วนค่าของ $J = 0$ เพราะฉะนั้นฟลิปฟล็อปอินพุตจะได้ค่า $J = 0$ ส่วนค่า $K = X$ จะทำให้เกิดค่า $Q = 0$

ถ้าฟลิปฟล็อปจะ Set ($Q = 1$) ในกรณี ถ้าฟลิปฟล็อปมีค่า Present state เป็นค่า 1 และค่าของ Next state ต้องการให้เป็นค่า 1 ($1 \rightarrow 1$) เราจะเห็นว่าค่าเข้าพุตของฟลิปฟล็อปไม่มีการเปลี่ยนแปลง (No change) ($J = 0, K = 0$) หรืออยู่ในเงื่อนไข Set ($J = 1, K = 0$) เมื่อมีสัญญาณนาฬิกาเข้ามา ก็จะ

เปลี่ยนแปลงตามอินพุต จากกรณีเราจะเห็นว่าค่าของ J อาจจะเป็นค่า 1 หรือ ค่า 0 ได้ เราเรียกว่า Do Not care ส่วนค่าของ K = 0 เพราะฉะนั้นฟลิปฟลอปอินพุตจะได้ค่า J = X ส่วนค่า K = 0 จะทำให้เกิดค่า Q = 0

ถ้าฟลิปฟลอป Reset (Q = 0) และจะต้องเปลี่ยนค่าไปเป็น Set (Q = 1) (0 → 1) จะทำให้อินพุตของฟลิปฟลอปมีค่าเป็น Reset คือ J = 1, K = 0 หรืออยู่ในสภาวะ Toggle J = 1, K = 1 เป็นเงื่อนไขที่ต้องการ ฉะนั้นอินพุตของฟลิปฟลอปคือ J = 1, K = X จะทำให้ค่าของ Q เปลี่ยนจาก 0 ไปเป็นค่า 1 ตามต้องการ

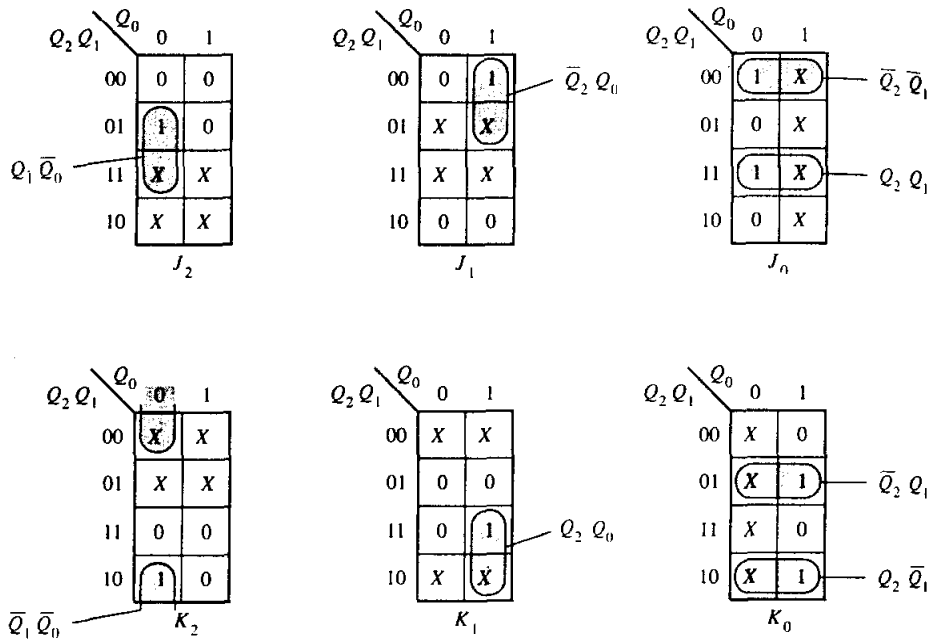
ถ้าฟลิปฟลอป Set (Q = 1) และจะต้องเปลี่ยนค่าไปเป็น Reset (Q = 0) (1 → 0) จะทำให้อินพุตของฟลิปฟลอปมีค่าเป็น Reset คือ J = 0, K = 1 หรืออยู่ในสภาวะ Toggle J = 1, K = 1 เป็นเงื่อนไขที่ต้องการ ฉะนั้นอินพุตของฟลิปฟลอปคือ J = X, K = 1 จะทำให้ค่าของ Q เปลี่ยนจาก 1 ไปเป็นค่า 0 ตามต้องการ

ตาราง Transition table for 3 bit Gray code counter

Output State Transition			Flip Flop Inputs		
Q2	Q1	Q0	(Present state)		
P→N	P→N	P→N	J ₂ K ₂	J ₁ K ₁	J ₀ K ₀
0→0	0→0	0→1	0x	0x	1x
0→0	0→1	1→1	0 x	1 x	x 0
0→0	1→1	1→0	0 x	x 0	x 1
0→1	1→1	0→0	1 x	x 0	0 x
1→1	1→1	0→1	x 0	X 0	1x
1→1	1→0	1→1	X 0	X 1	X 0
1→1	0→0	1→0	x 0	0 x	x 1
1→0	0→0	0→0	x 1	0 x	0 x

ขั้นตอนที่ 4 Karnaugh Maps

ขั้นตอนที่ 4 จะเป็นการเปลี่ยนสภาวะของฟลิปฟลอป JK จากตาราง Transition table เพื่อที่จะนำไปลดรูปโดยใช้คาร์โนแมพของอินพุตฟลิปฟลอปแต่ละตัว การลดรูปแสดงในรูปแบบที่ แสดงค่าของอินพุตแต่ละตัว



รูปที่ 7.26 การลดรูปโดยใช้คาร์โนแมพ

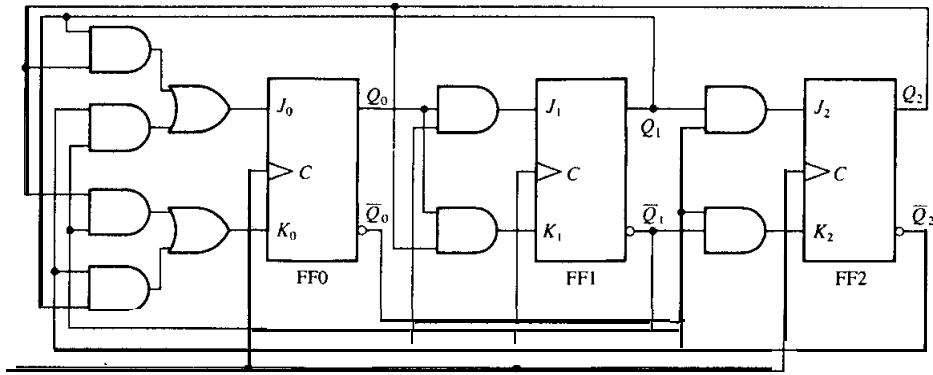
ขั้นตอนที่ 5 Input Logic

จะได้อินพุตของฟลิปฟลอปดังต่อไปนี้

$$\begin{aligned}
 J_2 &= Q_2 Q_1 + \bar{Q}_2 \bar{Q}_1 \\
 K_0 &= Q_2 \bar{Q}_1 + \bar{Q}_2 Q_1 \\
 J_1 &= \bar{Q}_2 Q_0 \\
 K_1 &= Q_2 Q_0 \\
 J_0 &= Q_1 \bar{Q}_0 \\
 K_2 &= \bar{Q}_1 \bar{Q}_0
 \end{aligned}$$

ขั้นตอนที่ 6 Implementation

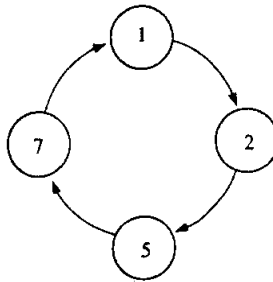
เขียนวงจรลอจิกโดยใช้สมการของอินพุตจากขั้นตอนที่ 5 เพื่อติดตั้งวงจรนับรหัส Gray code



รูปที่ 7.27 วงจรลอจิกที่ได้จากการออกแบบ

ตัวอย่าง 7.8 จงออกแบบวงจรนับเลขต่อไปนี้คือ 1-2-5-7 โดยใช้ J K Flip Flop

1. State Diagram



2. Next'state table

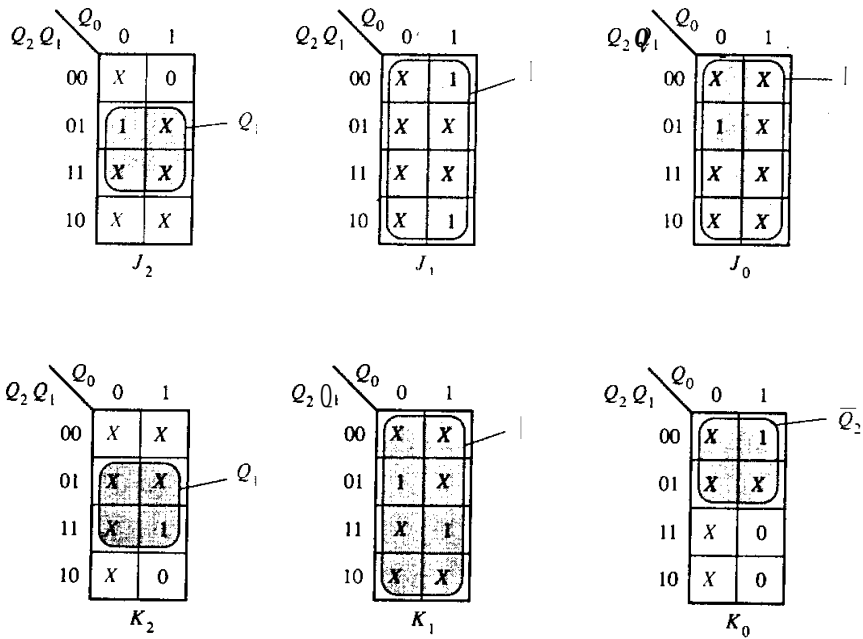
Present State			Next State		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	1	0	1	0
0	1	0	1	0	1
1	0	1	1	1	1
1	1	1	0	0	1

2. Transition table

Transition table.

Output State Transitions			Flip-Flop Inputs (present state)		
Q_2	Q_1	Q_0	J_2K_2	J_1K_1	J_0K_0
Present → Next	Present → Next	Present → Next			
0 → 0	0 → 1	1 → 0	0x	1x	x1
0 → 1	1 → 0	0 → 1	1x	x1	1x
1 → 1	0 → 1	1 → 1	x0	1x	x0
1 → 0	1 → 0	1 → 1	x1	x1	x0

3. การลดรูปโดยใช้คาร์โนแมพ



รูปที่ 7.28 การลดรูปโดยใช้ K-Map

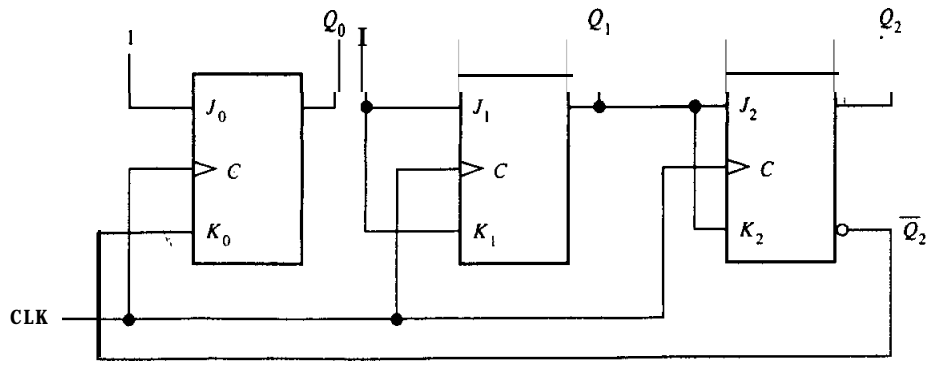
4. Input Logic

$$J_0 = 1, K_0 = \bar{Q}_2$$

$$J_1 = K_1 = 1$$

$$J_2 = K_2 = Q_1$$

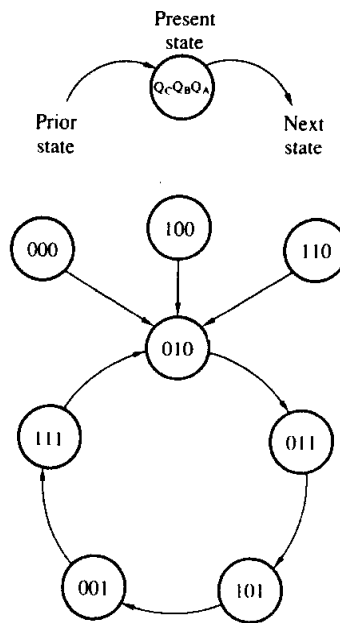
5. วงจรลอจิก



รูปที่ 7.29 วงจรลอจิก

ตัวอย่าง 7.9 จงออกแบบวงจรเคาน์เตอร์แบบ Synchronous MOD 5 ด้วยการนับต่อไปนี้ 2,3,5,1,7, วงจรเคาน์เตอร์จะต้องเริ่มต้นด้วยตัวเอง ถ้าเกิดค่าสถานะ 0,4,6, ให้ไปเริ่มต้นนับที่ค่า 2

1. เขียน State diagram



รูปที่ 7.30 State Diagram

2. Next State Table

Present State				Next State		
Q _C	Q _B	Q _A		Q _C	Q _B	Q _A
0	0	0		0	1	0
1	0	0		0	1	0
1	1	0		0	1	0
0	1	0		0	1	1
0	1	1		1	0	1
1	0	1		0	0	1
0	0	1		1	1	1
1	1	1		0	1	0

3. Transition table

Present State			Next State			Present Inputs					
Q _C	Q _B	Q _A	Q _C	Q _B	Q _A	J _C	K _C	J _B	K _B	J _A	K _A
0	0	0	0	1	0	0	x	1	x	0	x
1	0	0	0	1	0	x	1	1	0	0	x
1	1	0	0	1	0	0	1	x	0	1	x
0	1	0	0	1	1						x
0	1	1	1	0	1	1	x	x	1	x	0
1	0	1	0	0	1	x	1	0	x	x	0
0	0	1	1	1	1	1	x	1	x	x	0
1	1	1	0	1	0	x	1	x	0	x	1

4. การลดรูปโดยใช้ K-Map

	Q_A	0	1
$Q_C Q_B$			
0 0		0	1
0 1		0	1
1 1		x	x
1 0		x	x

$$J_C = Q_A$$

$K_C = 1$ by inspection of table

(a)

	Q_A	0	1
$Q_C Q_B$			
0 0		1	1
0 1		x	x
1 1		x	x
1 0		1	0

$$J_B = Q_C Q_A$$

	Q_A	0	1
$Q_C Q_B$			
0 0		x	x
0 1		0	1
1 1		0	0
1 0		x	x

$$K_B = \overline{Q_C} Q_A$$

(b)

	Q_A	0	1
$Q_C Q_B$			
0 0		0	x
0 1		1	x
1 1		0	x
1 0		0	x

$$J_A = \overline{Q_C} Q_B$$

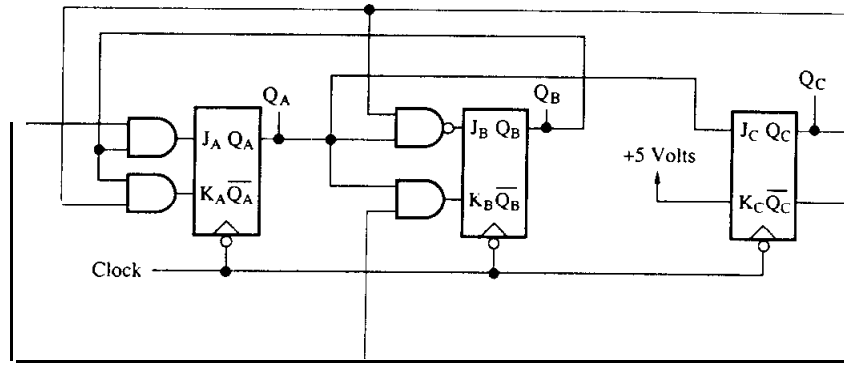
	Q_A	0	1
$Q_C Q_B$			
0 0		x	0
0 1		x	0
1 1		x	1
1 0		x	0

$$K_A = Q_C Q_B$$

(c)

รูปที่ 7.31 การลดรูป

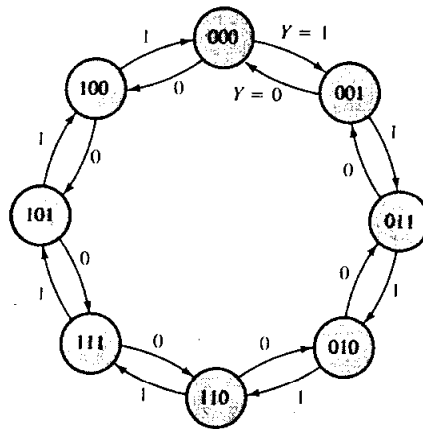
5. การเขียนวงจรลอจิก



รูปที่ 7.32 วงจรลอจิก

ตัวอย่าง 7.10 การออกแบบวงจรนับขึ้นนับลงขนาด 3 บิต การนับให้เรียงลำดับตามรหัส Gray Code วงจรนี้ถ้านับขึ้นในการควบคุมทางอินพุตมีค่าเป็น 1 แต่ถ้านับลงให้มืค่าเป็น 0

ขั้นตอนที่ 1 State Diagram



ขั้นตอนที่ 2 Next State Table

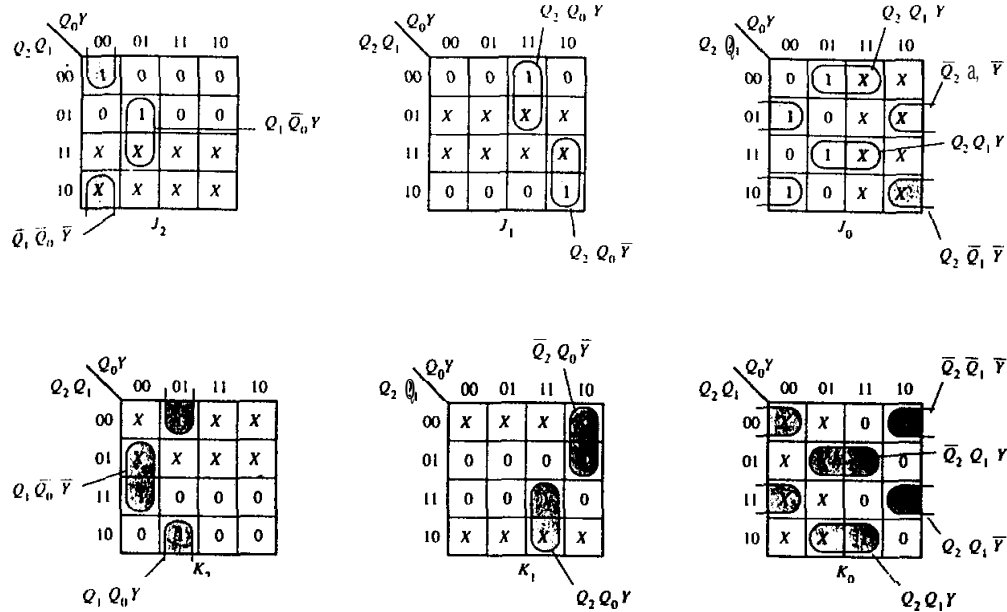
Present State			Next State					
Q_2	Q_1	Q_0	$Y = 0$ (down)			$Y = 1$ (up)		
			Q_2	Q_1	Q_0	Q_2	Q_1	Q_0
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	1	1
0	1	1	0	0	1	0	1	0
0	1	0	0	1	1	0	0	0
1	1	0	0	1	0	1	0	1
1	0	0	0	1	1	1	0	0
1	0	1	0	0	0	1	1	0

$Y = \overline{\text{up/down control input}}$.

ขั้นตอนที่ 3 Transition Table

Output State Transitions			Y	Flip-Flop Input (present state)		
Q_2	Q_1	Q_0		J_2K_2	J_1K_1	J_0K_0
Present → Next	Present → Next	Present → Next	Down = 0 Up = 1			
0 → 1	0 → 0	0 → 0	0	1X	0X	0X
0 → 0	0 → 0	0 → 1	1	0X	0X	1X
0 → 0	0 → 0	1 → 0	0	0X	0X	X1
0 → 0	0 → 1	1 → 1	1	0X	1X	X0
0 → 0	1 → 0	1 → 1	0	0X	X1	X0
0 → 0	1 → 1	1 → 0	1	0X	X0	X1
0 → 0	1 → 1	0 → 1	0	0X	X0	1X
0 → 1	1 → 1	0 → 0	1	1X	X0	0X
1 → 0	1 → 1	0 → 0	0	X1	X0	0X
1 → 1	1 → 1	0 → 1	1	X0	X0	1X
1 → 1	1 → 1	1 → 0	0	X0	X0	X1
1 → 1	1 → 0	1 → 1	1	X0	X1	X0
1 → 1	0 → 1	1 → 1	0	X0	1X	X0
1 → 1	0 → 0	1 → 0	1	X0	0X	X1
1 → 1	0 → 0	0 → 1	0	X0	0X	1X
1 → 0	0 → 0	0 → 0	1	X1	0X	0X

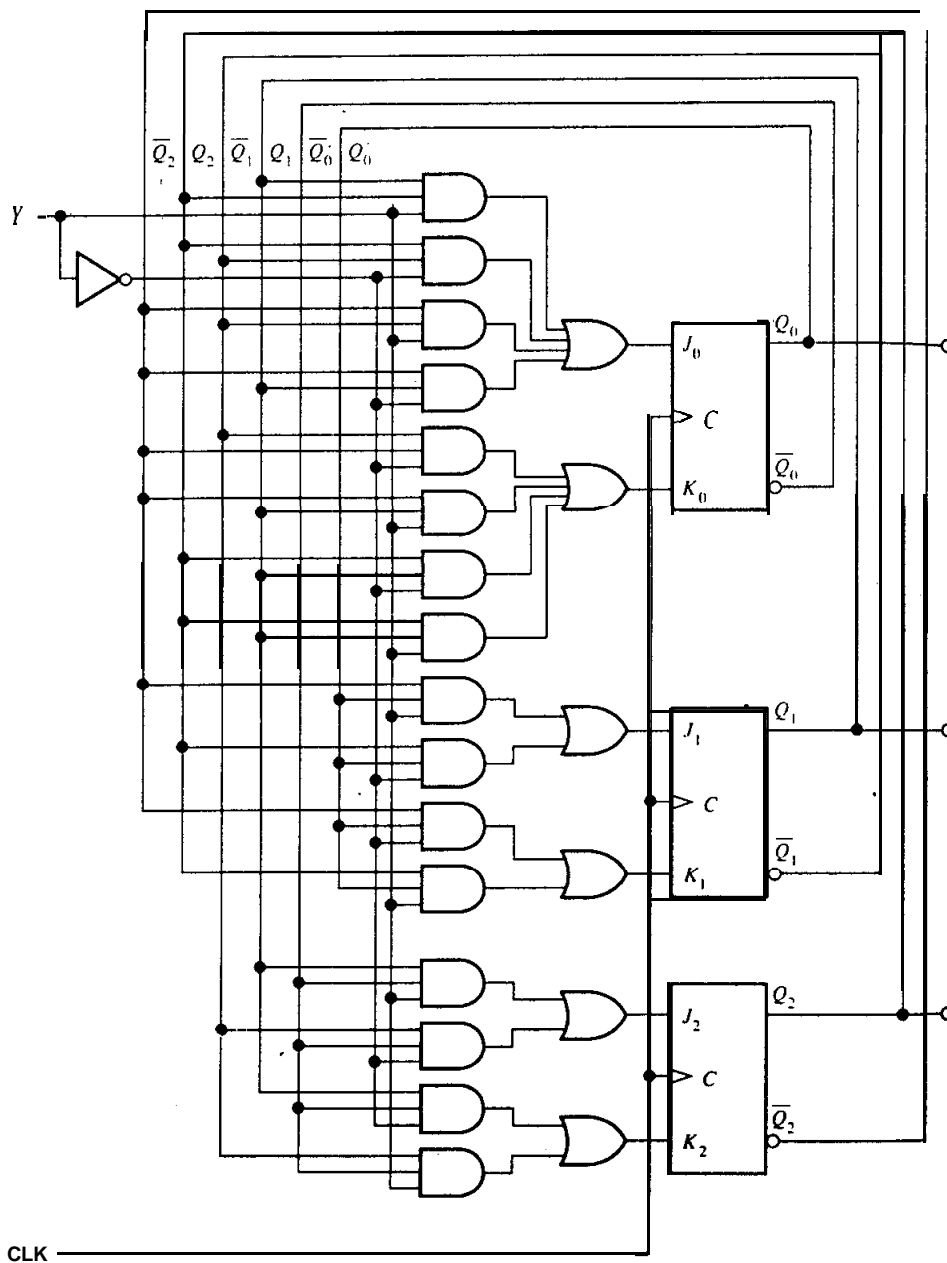
ขั้นตอนที่ 4 การลดรูปโดยใช้ K-Map



4. Input Logic

$$\begin{aligned}
 J_0 &= Q_2 Q_1 Y + Q_2 \bar{Q}_1 Y + \bar{Q}_2 Q_1 \bar{Y} + Q_2 \bar{Q}_1 \bar{Y} \\
 K_0 &= Q_2 Q_1 Y + Q_2 Q_1 \bar{Y} + Q_2 \bar{Q}_1 \bar{Y} + Q_2 \bar{Q}_1 Y \\
 J_1 &= Q_2 Q_0 Y + Q_2 Q_0 \bar{Y} \\
 K_1 &= \bar{Q}_2 Q_0 \bar{Y} + Q_2 Q_0 \bar{Y} \\
 J_2 &= Q_1 \bar{Q}_0 Y + Q_1 Q_0 \bar{Y} \\
 K_2 &= Q_1 Q_0 Y + Q_1 Q_0 \bar{Y}
 \end{aligned}$$

ขั้นตอนที่ 6 วงจรลอจิก



สรุป

วงจรรนับหรือ Counter Circuit นั้นเป็นวงจรรนับเลขฐานสอง เป็นการนับ 2^n ขณะที่ค่า n คือจำนวนของฟลิปฟลอปของวงจรรนับ การนับของวงจรรนับทุก MODULUS สามารถใช้วงจรถลอจิกร่วมกับวงจรรนับเพื่อให้การนับเป็นไปตามความต้องการ วงจรรนับนี้เราแบ่งออกเป็น 12 ชนิด คือ

วงจรรนับแบบซิงโครนัส และวงจรรนับแบบอซิงโครนัส การออกแบบของวงจรรทั้งสองมีเทคนิคที่แตกต่างกัน วงจรรนับที่ MODULUS สูงกว่ามีโครงสร้างง่ายกว่าวงจรรนับที่มี MODULUS ต่ำกว่า

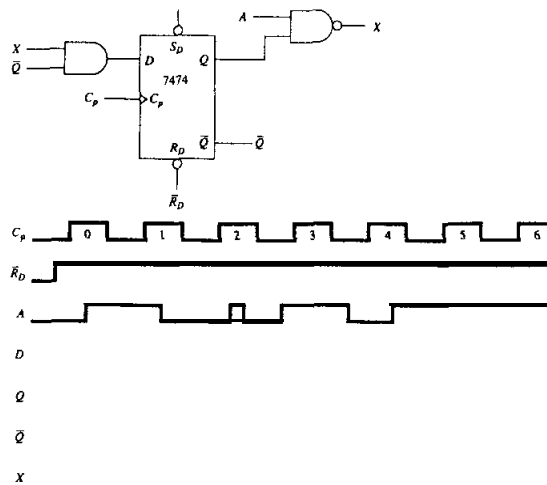
Ripple Counter เป็นวงจรรนับแบบอซิงโครนัส ฟลิปฟลอปแต่ละตัวจะถูกกระตุ้นการทำงานโดยเข้าพุทของฟลิปฟลอปที่อยู่ก่อนหน้า

Parallel Counter เป็นวงจรรนับแบบซิงโครนัส ฟลิปฟลอปแต่ละตัวจะถูกกระตุ้นการทำงานพร้อมกัน

Up - Down Counter เป็นพื้นฐานของวงจรรนับ แบบอซิงโครนัสและซิงโครนัส สามารถนับขึ้นและนับลงได้ 2 ทิศทาง

แบบฝึกหัด

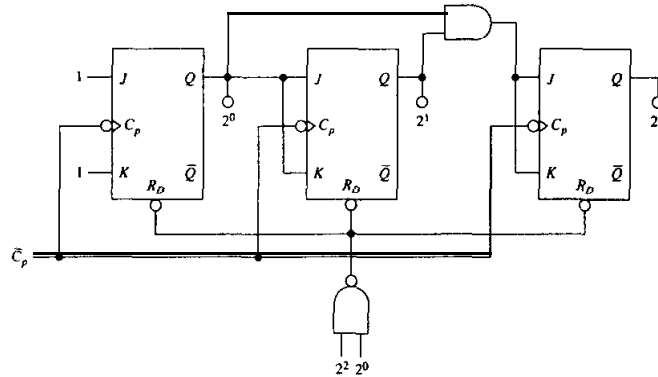
- จงอธิบายการทำงานของวงจรถลอจิกชนิดคอมไบเนชันกับวงจรรซีเคิร์นเขียนแตกต่างกันอย่างไร
- จากรูปคลื่นที่แสดงในรูปของสัญลักษณ์ลอจิกต่อไปนี้ จงเขียนรูปคลื่น D , Q , \bar{Q} และ X



3. ออกทราบดีว่าวงจรนับต่อไปนี้ จะใช้ J-K Flip Flop ก็ดี

3.1 MOD-7 3.2 MOD - 12 3.3 MOD-25 3.4 MOD-9

4. จากวงจรที่กำหนดให้จงเขียนรูปคลื่นของสัญญาณนาฬิกาจำนวน 10 พัลส์ในการทำงานของวงจรต่อไปนี้ โดยให้สถานะเริ่มแรกของเอาต์พุตทั้งหมดมีค่าเป็น 0



5. จงออกแบบวงจรนับแบบซิงโครนัสในการนับ 1-6-4-2 ส่วนสถานะ 0, 3, 5, 7 ให้กลับไปนับที่ 1 โดยใช้ J-K Flip Flop

6. จากข้อที่ 5 ให้ใช้ D Flip Flop