

**บทที่ 9**  
**รีจิสเตอร์**  
**REGISTER**

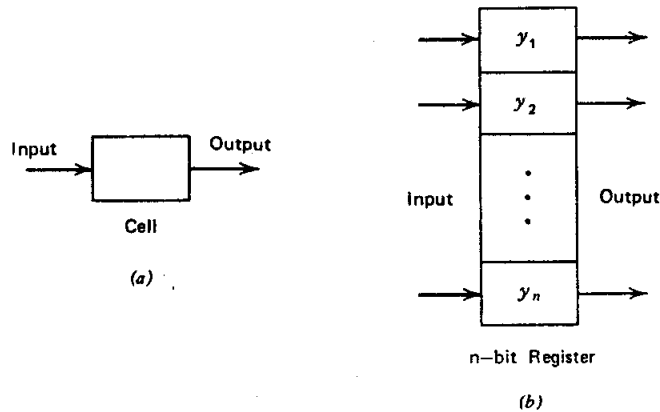
**วัตถุประสงค์**

เมื่อศึกษาจบบทนี้แล้วนักศึกษาสามารถ

1. อธิบายการทำงานของรีจิสเตอร์เลื่อน
2. เขียนวงจรรีจิสเตอร์เลื่อนซ้ายและขวาได้
3. เขียนตารางแสดงการเลื่อนข้อมูลได้ทั้งแบบเลื่อนซ้าย เลื่อนขวา หรือหมุนวน
4. อธิบายการทำงานของรีจิสเตอร์ถ่ายโอนแบบขนาน

## 9.1 ความนำ

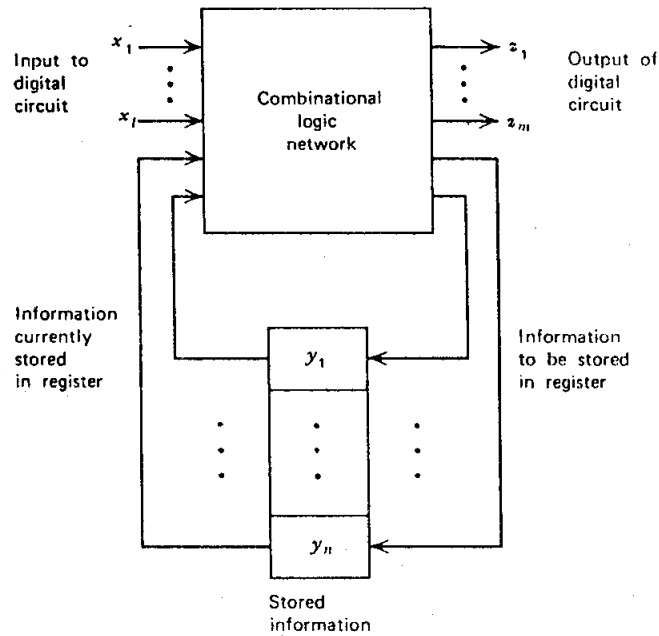
โครงข่ายดิจิทัล (digital network) นอกเหนือจากสามารถดำเนินการกับสัญญาณดิจิทัลในลักษณะตรรกแล้ว ยังต้องมีความสามารถเก็บรายละเอียดข้อมูลเกี่ยวกับพฤติกรรมในอดีต หรืออินพุตในอดีตเข้าไว้ในโครงข่ายด้วย เพื่อให้บรรลุจุดประสงค์นี้จึงได้มีการพัฒนาอุปกรณ์เก็บข้อมูล ในที่นี้จะกล่าวถึงขั้นส่วนพื้นฐานในการเก็บข้อมูลซึ่งเรียกว่าเซลล์ (cell) และสมมุติว่ามันสามารถเก็บสัญญาณได้โดยมีค่า 1 หรือ 0 เอาท์พุทของแต่ละเซลล์มีค่าสอดคล้องกับปริมาณซึ่งเก็บไว้ในเซลล์ เนื้อหาของเซลล์ยังคงเดิมจนกว่าจะได้รับสัญญาณอินพุทเพื่อสั่งให้เซลล์เปลี่ยนแปลงเนื้อหา รูป 9.1 แสดงสัญลักษณ์แทนเซลล์



รูป 9.1 เซลล์และรีจิสเตอร์ (a) เซลล์ (b) รีจิสเตอร์ขนาด  $n$  บิต

เซลล์เก็บข้อมูลเดียวมีความจุในการเก็บข้อมูลค่อนข้างจำกัด เนื่องจากมันสามารถเก็บข้อมูลเพียงบิตเดียว ถ้าต้องการเก็บข้อมูลมากขึ้นเราอาจนำหลาย ๆ เซลล์มาต่อรวมกัน ดังรูป 9.1 (b) เพื่อสร้างเป็นรีจิสเตอร์ รีจิสเตอร์ซึ่งประกอบด้วย  $n$  เซลล์ เรียกว่า รีจิสเตอร์ขนาด  $n$  บิต (n-bit register)

รีจิสเตอร์ถ้าอยู่เดี่ยว ๆ แล้วจะมีคุณค่าน้อย ต้องต่อรวมกับโครงข่าย (วงจร) ตรรกะประสม ถ้าข้อมูลที่เก็บไว้ต้องนำมาใช้ในส่วนการคำนวณ ดังรูป 9.2 ซึ่งเป็นแบบจำลองทั่วไปของวงจรตรรก (รูปนี้ก็คือวงจรซีเควนเขียนนั่นเอง)



รูป 9.2 แบบจำลองทั่วไปของวงจรดิจิทัล

## 9.2 รีจิสเตอร์เลื่อน และรีจิสเตอร์ถ่ายโอนแบบขนาน Shift and Parallel Transfer Register

นอกเหนือจากการดำเนินการในวงจรนับแล้ว ฟลิปฟลอปยังใช้ก่อร่างเป็นรีจิสเตอร์เพื่อเก็บและถ่ายโอนข้อมูลฐานสอง ถ้าข้อมูลถูกเคลื่อนย้ายแบบอนุกรม (ทีละบิต) ฟลิปฟลอปนั้นจะถูกเชื่อมต่อกันเป็นรีจิสเตอร์เลื่อน (shift register) ซึ่งอาจถูกออกแบบให้เลื่อนข้อมูลไปทางขวา หรือไปทางซ้าย นอกจากนี้ข้อมูลยังอาจถูกถ่ายโอนในแบบขนาน (ทุกบิตพร้อมกันในเวลาเดียวกัน) เพื่อให้การดำเนินการรวดเร็วที่สุด แบบนี้เป็นรีจิสเตอร์ถ่ายโอนแบบขนาน (parallel transfer register)

ถ้าใช้ JK ฟลิปฟลอปในการสร้างรีจิสเตอร์เลื่อน อินพุต J และ K จะเป็นอินพุตข้อมูล และคล็อกอินพุตเป็นทริกเกอร์พัลส์หรือพัลส์เลื่อน

ถ้าใช้ JK ฟลิปฟลอปสร้างรีจิสเตอร์ถ่ายโอนแบบขนาน เคลียร์อินพุต (CLR) และพรีเซ็ทอินพุต (PS) จะเป็นอินพุตข้อมูล

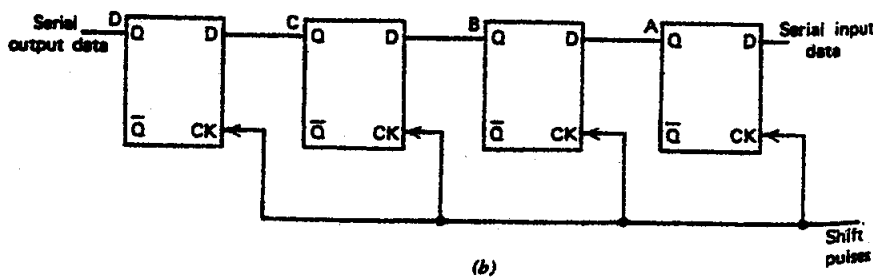
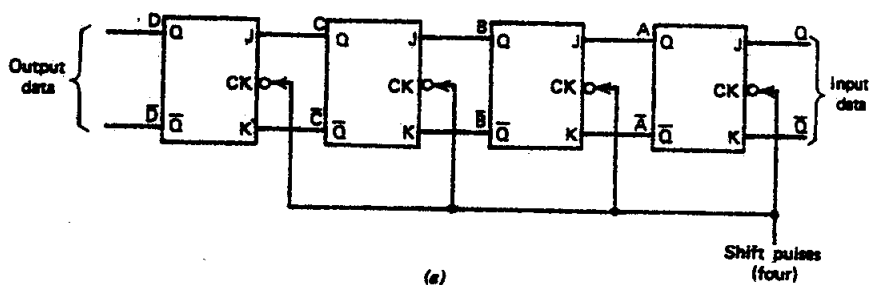
D ฟลิปฟลอปมีความเหมาะสมโดยเฉพาะสำหรับสร้างรีจิสเตอร์เลื่อน

## 9.2.1 รีจิสเตอร์เลื่อน

9.2.1.1 รีจิสเตอร์เลื่อนซ้าย ข้อมูลถูกเลื่อนบิตที่มีนัยสำคัญมากที่สุดเข้าไปก่อน

รีจิสเตอร์เลื่อนซ้ายซึ่งเลื่อนข้อมูลขนาด 4 บิต สร้างโดยใช้ฟลิปฟลอป 4 ตัว แสดงดังรูป 9.3 กรณีใช้ JK ฟลิปฟลอปข้อมูลอินพุตประกอบด้วยสัญญาณฐานสองที่ตรงข้ามกัน โดยสัญญาณข้อมูลอ้างอิงป้อนเข้าทางอินพุต J และสัญญาณข้อมูลตรงข้ามเข้าทางอินพุต K สำหรับ D ฟลิปฟลอปเส้นทางข้อมูลอินพุตมีอันเดียวเข้าทางอินพุต D

พัลส์เลื่อนป้อนสู่แต่ละฟลิปฟลอป ทำให้ดำเนินการพร้อมกัน เมื่อพัลส์เลื่อนปรากฏ ข้อมูลอินพุตจะถูกเลื่อนเข้าสู่ฟลิปฟลอป แต่ละฟลิปฟลอปถูกเซ็ท หรือรีเซ็ทสอดคล้องกับข้อมูลอินพุตที่เวลาพัลส์เลื่อนปรากฏ ดังนั้นบิตของข้อมูลอินพุตถูกเลื่อนเข้าสู่ฟลิปฟลอป A โดยพัลส์เลื่อนตัวแรก ในขณะเดียวกันข้อมูลของฟลิปฟลอป A ถูกเลื่อนเข้าสู่ฟลิปฟลอป B และเป็นเช่นนี้ต่อไป ที่แต่ละพัลส์เลื่อนข้อมูลซึ่งเก็บอยู่ในรีจิสเตอร์จะเลื่อนไปทางซ้าย 1 ระยะฟลิปฟลอป ข้อมูลใหม่ก็จะถูกเลื่อนเข้าสู่ฟลิปฟลอป A ในขณะที่ข้อมูลซึ่งอยู่ที่ฟลิปฟลอป D ถูกเลื่อนออกไปทางซ้าย เพื่อนำไปใช้ในรีจิสเตอร์เลื่อนอื่น หรือหน่วยคอมพิวเตอร์



รูป 9.3 รีจิสเตอร์เลื่อนซ้าย (a) ใช้ JK ฟลิปฟลอป (b) ใช้ D ฟลิปฟลอป

ตัวอย่างการเลื่อนข้อมูล ลองพิจารณาการเลื่อนข้อมูลในตาราง 9.1 เริ่มต้นรีเซ็ตทุก ๆ ฟลิปฟลอป (เอาท์พุท Q ของทุกฟลิปฟลอปเป็น 0) แล้วป้อนข้อมูลตรรกะ 1 คงที่ตลอดเข้าที่อินพุทของฟลิปฟลอป A จากตาราง 9.1 จะเห็นว่าข้อมูลในแต่ละฟลิปฟลอปถูกเลื่อนอย่างไรสำหรับแต่ละพัลส์เลื่อน และจะสังเกตเห็นการเลื่อนข้อมูลอินพุทตรรกะ 1 ตัวแรกที่ป้อนสู่ฟลิปฟลอป A ว่าถูกเลื่อนไปทางซ้ายไปถึงฟลิปฟลอป D เมื่อพัลส์เลื่อนผ่านไป 4 พัลส์

ตาราง 9.1 การเลื่อนข้อมูลอินพุทตรรกะ 1 คงที่ตลอดของรีจิสเตอร์เลื่อนซ้าย

พัลส์เลื่อน	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

ตัวอย่างอีกอันหนึ่ง ให้พิจารณาการเลื่อนข้อมูล 0, 1 สลับกันเข้าไปในรีจิสเตอร์เลื่อนซ้าย ซึ่งมีค่าเริ่มต้นเป็นตรรกะ 1 ทั้งหมด ดังตาราง 9.2

ตาราง 9.2 การเลื่อนข้อมูล 0, 1 สลับกันเข้าไปในรีจิสเตอร์เลื่อนซ้าย ซึ่งมีค่าเริ่มต้นเป็นตรรกะ 1 ทุกฟลิปฟลอป

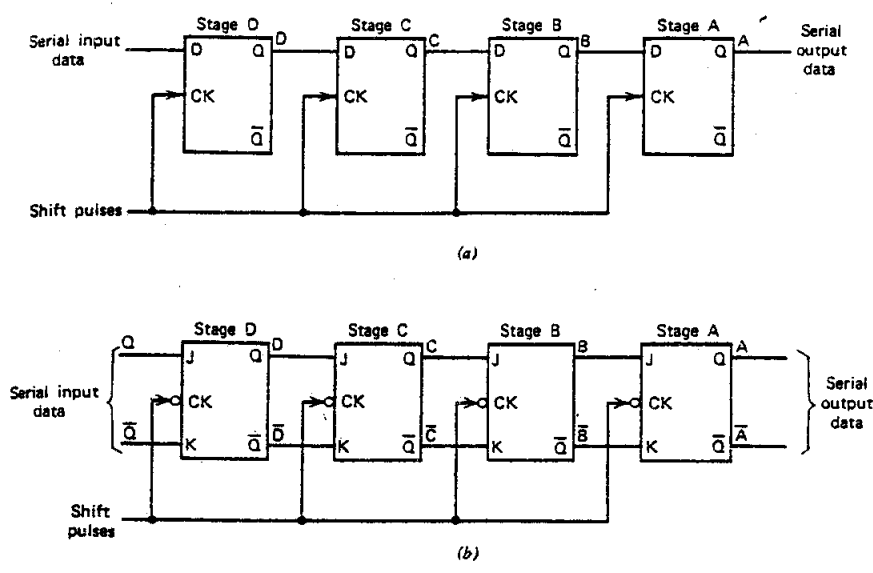
พัลส์เลื่อน	D	C	B	A
0	1	1	1	1
1	1	1	1	0
2	1	1	0	1
3	1	0	1	0
4	0	1	0	1

ข้อสังเกตบางประการของรีจิสเตอร์เลื่อนข้อมูล มีดังต่อไปนี้

1. จำนวนพัลส์เลื่อนต้องเท่ากับจำนวนฟลิปฟลอปในรีจิสเตอร์นั้น
2. จำนวนฟลิปฟลอปเท่ากับจำนวนบิตของข้อมูล
3. การเลื่อนข้อมูลจะเกิดขึ้นพร้อมกัน แต่เกิดเมื่อพัลส์เลื่อนปรากฏเท่านั้น

9.2.1.2 รีจิสเตอร์เลื่อนขวา บิตที่มีนัยสำคัญน้อยที่สุดของข้อมูลถูกเลื่อนเข้าไปก่อน

บางครั้งเรามีความจำเป็นต้องเลื่อนบิตที่มีนัยสำคัญน้อยที่สุดของข้อมูลเข้าไปก่อน เช่นในกรณีที่ทำกรบวกเลข จะใช้รีจิสเตอร์เลื่อนขวาดังรูป 9.4 ข้อมูลอินพุตเข้าที่ฟลิปฟลอป D และถูกเลื่อนไปทางขวา การดำเนินการเลื่อนเช่นเดียวกับรีจิสเตอร์เลื่อนซ้ายต่างกันที่แบบนี้เลื่อนข้อมูลไปทางขวา ตาราง 9.3 แสดงตัวอย่างการเลื่อนข้อมูลตรรกะ 1 คงที่ตลอดเข้าไปในรีจิสเตอร์เลื่อนขวา ซึ่งถูกรีเซ็ตก่อนเริ่มใส่อินพุตเข้าไป



รูป 9.4 รีจิสเตอร์เลื่อนขวา (a) ใช้ D ฟลิปฟลอป (b) ใช้ JK ฟลิปฟลอป

ตาราง 9.3 การเลื่อนข้อมูลตรรก 1 คงที่ตลอดของรีจิสเตอร์เลื่อนขวา

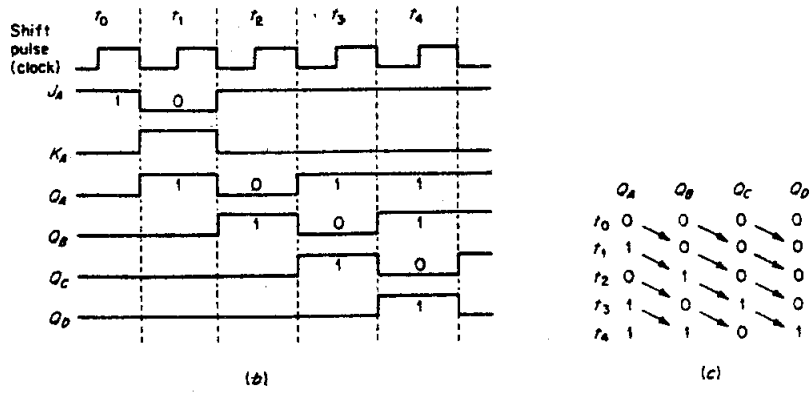
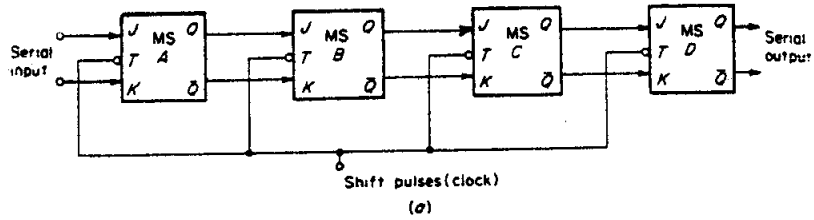
พัลส์เลื่อน	D	C	B	A
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1

นอกจากจะมีการเลื่อนข้อมูลเข้าสู่รีจิสเตอร์แล้ว ก็ยังมีการเลื่อนข้อมูลออกจากรีจิสเตอร์ด้วย ตาราง 9.4 แสดงการดำเนินการของรีจิสเตอร์เพื่อเลื่อนข้อมูล 1101 ออกไป ให้สังเกตว่าเอาต์พุตของฟลิปฟลอป A จะมีเลขฐานสองของอินพุตแต่ละบิต (เริ่มตั้งแต่บิตที่มีนัยสำคัญน้อยที่สุด) ปรากฏอยู่ในแต่ละขั้นตอนของการเลื่อนข้อมูล ในกรณีนี้เสมือนกับว่ามีอินพุตตรรก 0 เป็นข้อมูลที่ถูกลื่อนเข้าสู่รีจิสเตอร์ ดังนั้นหลังจากพัลส์เลื่อนผ่านไป 4 พัลส์ ข้อมูลถูกลื่อนออกจากรีจิสเตอร์ และทิ้งให้รีจิสเตอร์อยู่ในสถานะรีเซ็ตหลังจากพัลส์ที่สี่

ตาราง 9.4 การเลื่อนข้อมูลออกจากรีจิสเตอร์เลื่อนขวา

พัลส์เลื่อน	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

ตัวอย่างรีจิสเตอร์เลื่อนขวาในรูป 9.5 แสดงการเลื่อนข้อมูล 1101 เข้าไปในรีจิสเตอร์ซึ่งเริ่มต้นถูกรีเซ็ตก่อน และแสดงรูปคลื่นเอาต์พุตของฟลิปฟลอปต่างๆ ที่ประกอบเป็นรีจิสเตอร์นี้

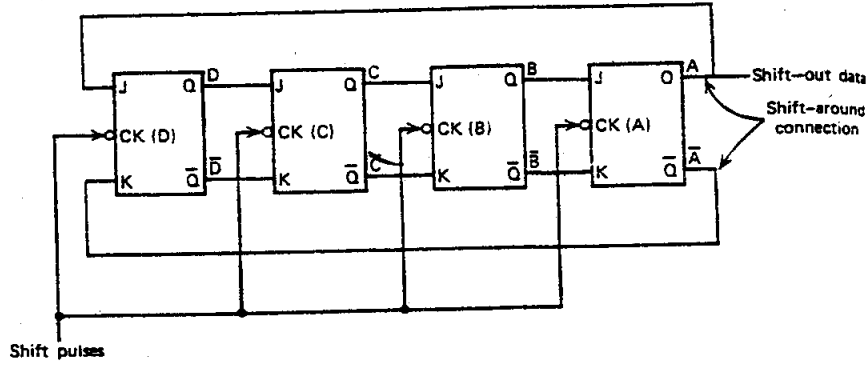


รูป 9.5 รีจิสเตอร์เลื่อนขวา (a) วงจร (b) รูปคลื่น (c) ตารางความจริงแสดงการเลื่อนข้อมูล

### 9.2.1.3 รีจิสเตอร์-หมุนวน (Shift-Around Register)

เมื่อมีความจำเป็นต้องเลื่อนข้อมูลออกจากรีจิสเตอร์โดยไม่ให้เกิดการสูญหายข้อมูลเริ่มต้น เราจะใช้รีจิสเตอร์แบบเลื่อน-หมุนวน รูป 9.6 แสดงรีจิสเตอร์เลื่อนขวาแบบเลื่อนหมุนวน สิ่งจำเป็นที่ต้องกระทำการต่อเอาท์พุทของฟลิปฟลอป A เข้าเป็นอินพุทของฟลิปฟลอป D แล้วจะได้ผลว่าในขณะที่พัลส์เลื่อน 4 พัลส์ เคลื่อนย้ายข้อมูลฐานสองเข้าสู่ฟลิปฟลอป A ข้อมูลจะถูกเลื่อนออกจากฟลิปฟลอป A เข้าสู่ฟลิปฟลอป D และกลับเข้าไปอยู่ในรีจิสเตอร์อย่างเดิม



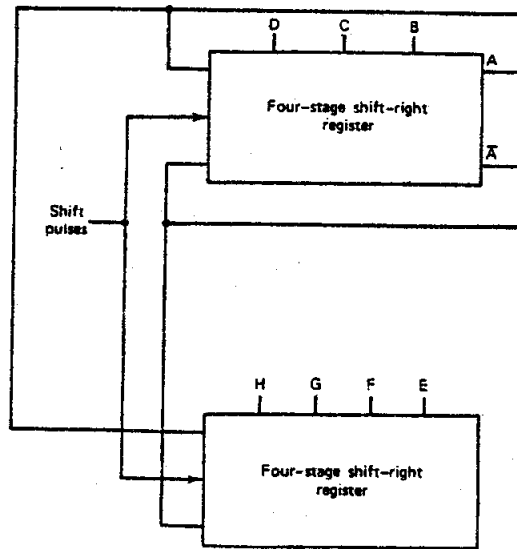


รูป 9.6 รีจิสเตอร์เลื่อนขวา แบบหมุนวน

ตาราง 9.5 การเลื่อนแบบหมุนวนชนิดเลื่อนขวา

พัลส์เลื่อน	D	C	B	A
0	1	1	0	1
1	1	1	1	0
2	0	1	1	1
3	1	0	1	1
4	1	1	0	1

ตาราง 9.5 แสดงผลของการเลื่อนข้อมูลเลขฐานสอง 1101 ผ่าน (และหมุนวน) รีจิสเตอร์เลื่อน ให้สังเกตว่าหลังจากพัลส์เลื่อนเกิดขึ้น 4 พัลส์ ค่าเริ่มต้นจะอยู่ในรีจิสเตอร์อีกครั้ง เพื่อให้เห็นอาการใดๆ ที่เกิดขึ้นนอกเหนือจากการเลื่อนจำนวนเลขหมุนวนรีจิสเตอร์ พิจารณารีจิสเตอร์เลื่อน 2 ตัวต่อกันดังรูป 9.7 โดยในรูปนี้ใช้รูปกล่องแทนรีจิสเตอร์แต่ละตัว ซึ่งประกอบด้วยฟลิปฟลอป 4 ตัวสร้างเป็นรีจิสเตอร์เลื่อนขวา การต่อวงจรภายนอกของเอาต์พุต A และ  $\bar{A}$  ของรีจิสเตอร์ตัวที่ 1 กลับเข้าไปที่อินพุตข้อมูลของรีจิสเตอร์ตัวเดียวกันนี้ ให้ผลเป็นรีจิสเตอร์เลื่อนหมุนวน สัญญาณตรรกซึ่งปรากฏที่เอาต์พุต A (และ  $\bar{A}$ ) ก็เลื่อนเข้าสู่รีจิสเตอร์ตัวที่ 2 ด้วย ตาราง 9.6 แสดงตัวอย่างการเลื่อนข้อมูลโดยเริ่มต้นด้วย 1101 ในรีจิสเตอร์ตัวที่ 1 และ 0000 ในรีจิสเตอร์ตัวที่ 2 ถ้าส่วนที่เป็นการหมุนวนของรีจิสเตอร์ 1 ไม่ได้ถูกใช้และอินพุตข้อมูลไม่มีการเชื่อมต่อ (ตรรก 0) แล้ว หลังจาก 4 พัลส์เลื่อนเกิดขึ้นแล้วข้อมูลเริ่มต้นในรีจิสเตอร์ตัวที่ 1 จะอยู่ในรีจิสเตอร์ตัวที่ 2 โดยรีจิสเตอร์ตัวที่ 1 เป็นรีจิสต์



รูป 9.7 รีจิสเตอร์เลื่อนขวา 2 ตัวต่อเชื่อมโยงกัน

ตาราง 9.6 การดำเนินการของรีจิสเตอร์เลื่อนในรูป 9.7

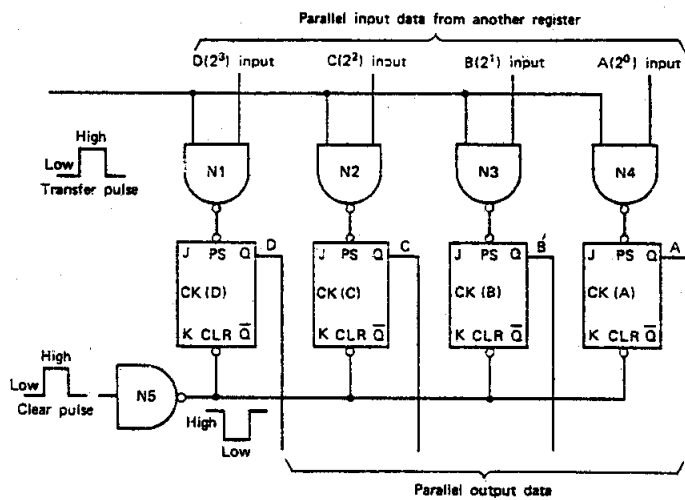
พัลส์เลื่อน	D	C	B	A	H	G	F	E
0	1	1	0	1	0	0	0	0
1	1	1	1	0	1	0	0	0
2	0	1	1	1	0	1	0	0
3	1	0	1	1	1	0	1	0
4	1	1	0	1	1	1	0	1

### 9.2.2 รีจิสเตอร์ถ่ายโอนข้อมูลแบบขนาน (Parallel Data Transfer Register)

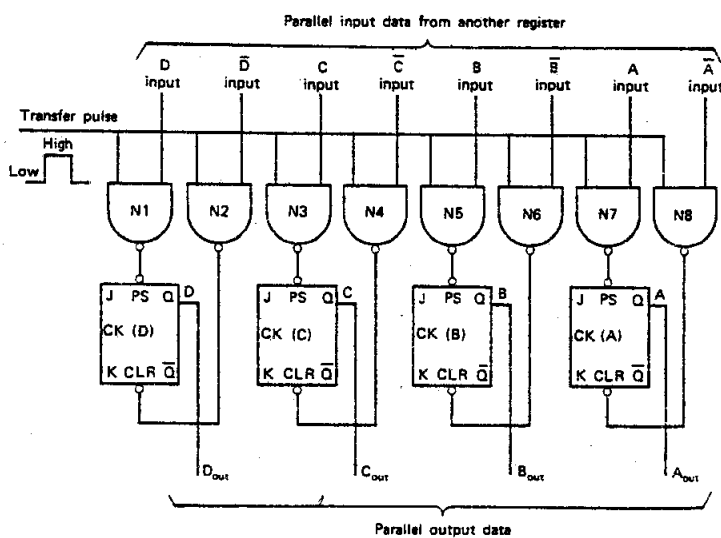
ที่กล่าวมาแล้วเป็นการเคลื่อนย้ายข้อมูลอย่างอนุกรมโดยใช้รีจิสเตอร์เลื่อน ซึ่งเป็นหน่วยธรรมดาในหน่วยอินพุต-เอาต์พุต และหน่วยคอมพิวเตอร์เฉพาะเจาะจงวัตถุประสงค์ อย่างไรก็ตามดิจิทัลคอมพิวเตอร์สมัยใหม่ส่วนมากจัดให้มีการดำเนินการข้อมูลแบบขนาน ข้อมูลทุก ๆ บิตในรีจิสเตอร์ถูกถ่ายโอนในเวลาเดียวกัน ถ้าใช้คัล็อกพัลส์เป็นขั้นตอนพื้นฐานแล้ว รีจิสเตอร์เลื่อนซึ่งมี 20 ภาคว (20 ฟลิปฟลอป) ต้องใช้พัลส์เลื่อน 20พัลส์ จึงจะเคลื่อนข้อมูลซึ่งเก็บไว้ในรูปค่าข้อมูลได้เต็มค่า (20 บิต) ถ้าคัล็อกพัลส์มีช่วงเวลา  $1 \mu \text{ sec}$  เวลาที่ต้อง

ใช้ในการถ่ายโอนค่าขนาด 20 บิตอย่างอนุกรม คือ  $20 \mu \text{ sec}$  ในขณะที่ข้อมูลเดียวกันนี้สามารถถูกถ่ายโอนในแบบขนานเพียงเวลา 1 ช่วงคล็อกพัลส์เท่านั้น คือ  $1 \mu \text{ sec}$  หรือเท่ากับเวลาเคลื่อนย้ายข้อมูล 1 บิตในแบบอนุกรม ค่าใช้จ่ายเพื่อให้ได้การดำเนินการที่รวดเร็วเช่นนี้มาจากสายสัญญาณและจำนวนเกทที่ต้องใช้มากขึ้น

รูป 9.8 (a) แสดงการเชื่อมต่อรีจิสเตอร์ขนาด 4 ภาคว่า (ใช้ฟลิปฟลอป 4 ตัว) เพื่อถ่ายโอนข้อมูลในแบบขนาน การดำเนินการพื้นฐานของรีจิสเตอร์นี้คือ เคลียร์ทุก ๆ ภาคว่าด้วยเคลียร์พัลส์แล้วถ่ายโอนบิตของข้อมูลอินพุตทั้งหมดด้วยพัลส์ถ่ายโอนเดียว



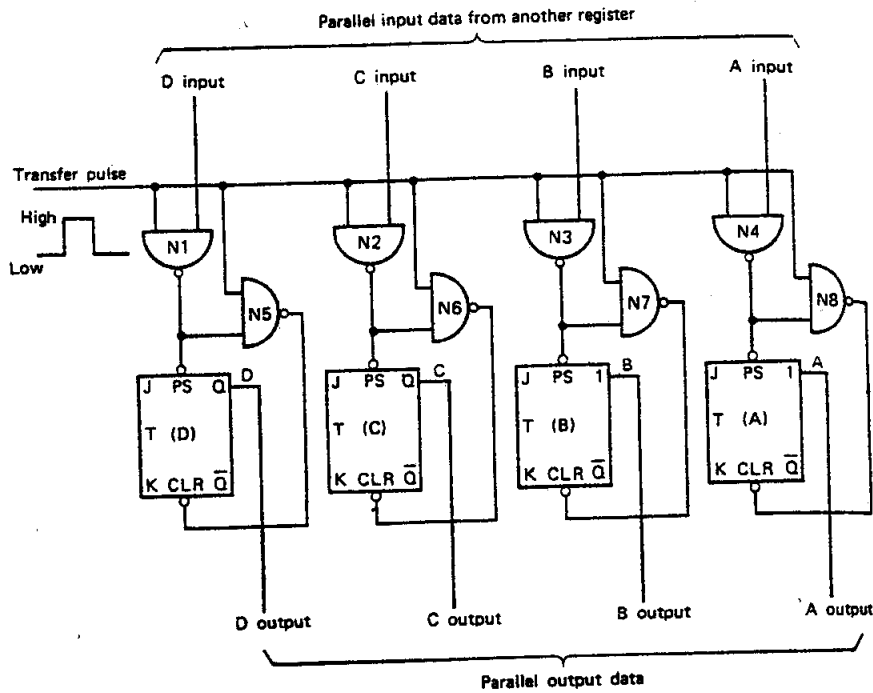
(a)



(b)

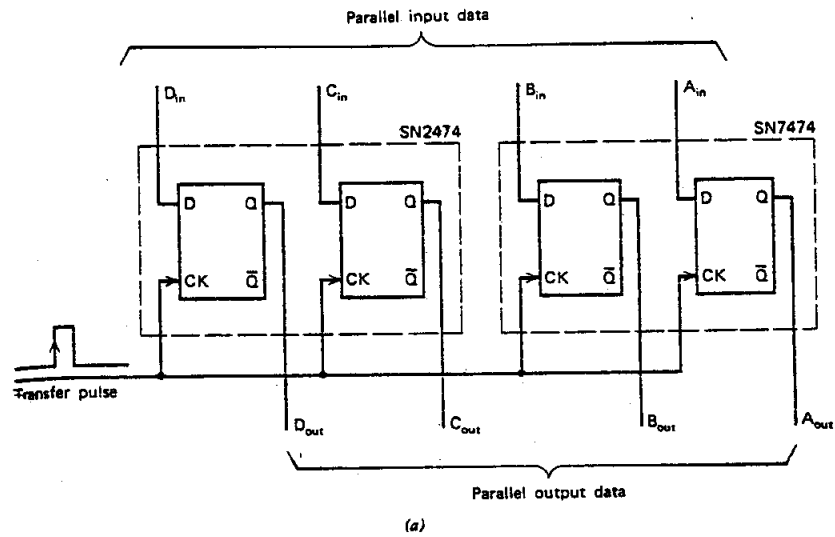
รูป 9.8 รีจิสเตอร์ถ่ายโอนแบบขนาน (a) ใช้เคลียร์พัลส์และพัลส์ถ่ายโอน (b) ใช้พัลส์ถ่ายโอนเดียว

รูป 9.9 แสดงวิธีดัดแปลงทางตรรกเพื่อกำจัดความจำเป็นที่จะต้องใช้สายอินพุตถ่ายโอนถึง 2 สายต่อข้อมูล 1 บิต

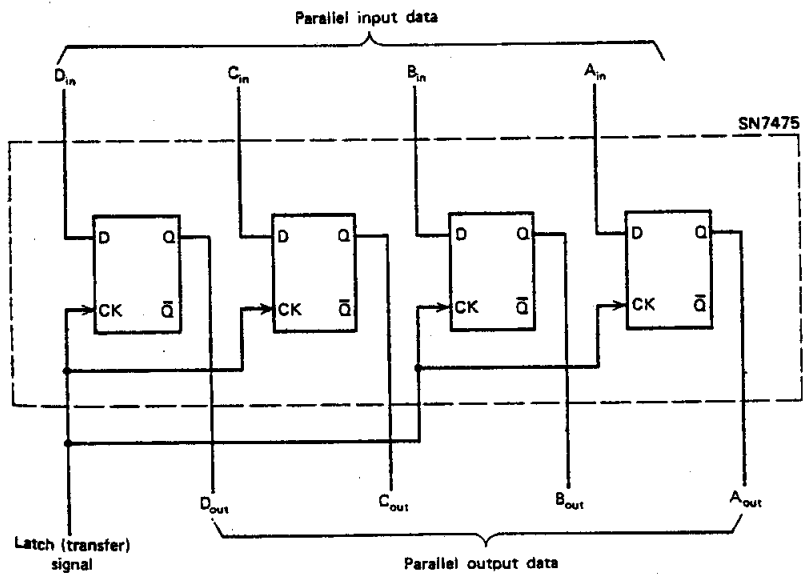


รูป 9.9 การถ่ายโอนแบบขนานแบบดัดแปลงโดยเกท

รีจิสเตอร์สร้างโดย D ฟลิปฟลอปแสดงในรูป 9.10 (a) ซึ่งในที่นี้ใช้ไอซีเบอร์ SN 7474 รีจิสเตอร์แบบนี้เหมาะสมในกรณีที่ต้องการเพียงเก็บข้อมูลแบบขนานเท่านั้น ข้อมูลอินพุตเข้าทางขั้วอินพุต D ของแต่ละภาค และเอาท์พุทเอามาจากขั้ว Q ทั้งหลาย ทุก ๆ ภาคถูกคัลลอคให้ทำงานพร้อมกันด้วยขอบบวกของสัญญาณถ่ายโอน สังเกตว่าก่อนดำเนินการถ่ายโอน ข้อมูลอินพุตอาจแตกต่างจากที่เก็บอยู่ในรีจิสเตอร์ 4 บิต เมื่อพัลส์ถ่ายโอนเกิดขึ้นข้อมูลเอาท์พุทจะเหมือนข้อมูลอินพุต และหลังจากพัลส์ถ่ายโอนแล้ว ข้อมูลอินพุตอาจเปลี่ยนแปลงโดยปราศจากความกระทบกระเทือนต่อข้อมูลเอาท์พุทที่เก็บไว้แล้ว



(a)



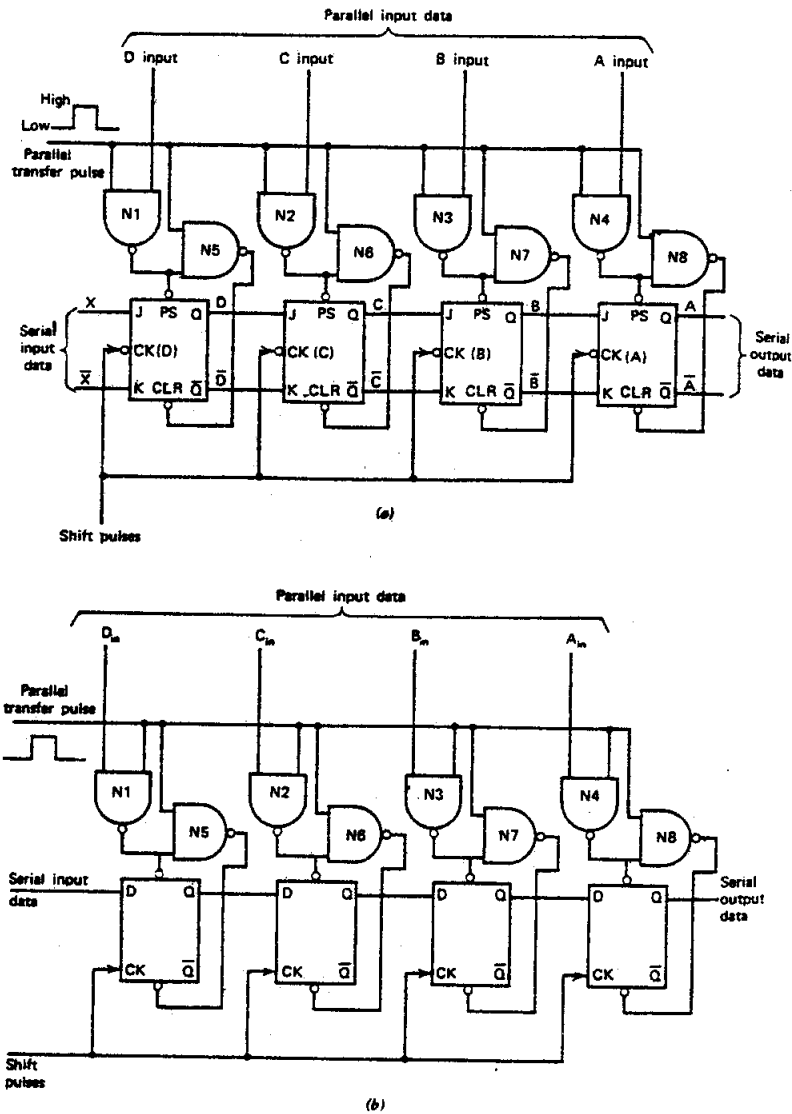
(b)

รูป 9.10 การถ่ายโอนข้อมูลแบบขนานโดยใช้ D ฟลิปฟลอป

ไอซี SN7475 (รูป 9.10 (b)) เป็น D ฟลิปฟลอปจัดเป็นวงจรรีจิสเตอร์สำหรับใช้เป็นแลทช์ข้อมูล (data "latch") สัญญาณคล็อกอินพุตทำให้รีจิสเตอร์ขนาด 4 บิตนี้แลทช์บนบิตของสัญญาณอินพุต เมื่อสัญญาณของแลทช์มีค่าสูง ในทางเป็นจริงแล้ว 7475 จะต่อเชื่อมอินพุตและเอาต์พุตเข้าด้วยกัน จนกระทั่งสัญญาณแลทช์ลงสู่ค่าต่ำ เมื่อสัญญาณนี้ลงต่ำ ข้อมูลอินพุต อาจเปลี่ยนแปลงโดยไม่กระทบต่อข้อมูลเอาต์พุตที่เก็บอยู่ในรีจิสเตอร์ขนาด 4 บิตนี้

### 9.2.3 รีจิสเตอร์ถ่ายโอนข้อมูลแบบอนุกรม-ขนาน (Serial-Parallel Transfer Register)

รีจิสเตอร์ตัวเดียวสามารถดำเนินการเลื่อนข้อมูล (shift or serial) และถ่ายโอนข้อมูลแบบขนาน แสดงดังรูป 9.11 ข้อมูลอินพุตจะถูกถ่ายโอนในแบบขนาน เมื่อมีพัลส์ถ่ายโอนแบบขนานเกิดขึ้น หรือในแบบอนุกรม เมื่อมีพัลส์เลื่อน 4 พัลส์เกิดขึ้น จะต้องเข้าใจโดยถูกต้องก่อนว่า การดำเนินการทั้งสองแบบนี้จะไม่เกิดขึ้นพร้อมกัน เมื่อเปรียบเทียบรีจิสเตอร์ในรูป 9.11 นี้กับรีจิสเตอร์เลื่อนในรูป 9.4 และรีจิสเตอร์ขนานในรูป 9.9 จะเห็นว่า รีจิสเตอร์ในรูป 9.11 นี้ ทำหน้าที่เป็นรีจิสเตอร์ในรูปแบบทั้งสองนั้นนั่นเอง



รูป 9.11 รีจิสเตอร์ถ่ายโอนข้อมูลแบบอนุกรม-ขนาน

## สรุป

รีจิสเตอร์คืออุปกรณ์เก็บข้อมูล เลื่อนข้อมูลแบบอนุกรมหรือถ่ายโอนข้อมูลแบบขนาน สร้างโดยใช้ฟลิปฟลอป

รีจิสเตอร์อาจแบ่งเป็นประเภทใหญ่ๆ ได้เป็น 2 ประเภทคือ แบบอนุกรม (เลื่อน) และแบบขนาน

รีจิสเตอร์เลื่อนทำการเคลื่อนย้ายข้อมูลที่ละบิต มีทั้งรีจิสเตอร์เลื่อนซ้าย และเลื่อนขวา ถ้ามีการป้อนกลับเอาต์พุตของฟลิปฟลอปตัวท้ายสู่อินพุตตัวแรกให้ผลเป็นรีจิสเตอร์หมุนวนไปทางซ้ายหรือไปทางขวา

รีจิสเตอร์แบบขนาน เคลื่อนย้ายข้อมูลทุกบิตพร้อมๆ กัน ทำให้ประหยัดเวลา

รีจิสเตอร์ทุกแบบประกอบด้วยฟลิปฟลอป จำนวนเท่ากับจำนวนบิตของข้อมูลที่ต้องการเคลื่อนย้าย

## แบบฝึกหัด

- 9.1 จงเขียนสถานการณ์ของการเลื่อนข้อมูล 5 บิต 10101 เข้าไปในรีจิสเตอร์เลื่อนซ้าย และขวา
  - 9.2 จงเขียนสถานการณ์ของการเลื่อนข้อมูล 6 บิต 111010 ในรีจิสเตอร์หมุนวนทางซ้าย และขวา
  - 9.3 จงอธิบายการทำงานของรีจิสเตอร์แบบอนุกรม เปรียบเทียบกับรีจิสเตอร์แบบขนาน
  - 9.4 จงเขียนตารางการเลื่อนข้อมูล 1010 ในรีจิสเตอร์รูป 9.7
  - 9.5 จงสร้างรีจิสเตอร์สำหรับเลื่อนข้อมูล 5 บิต เพื่อเลื่อนบิตที่มีนัยสำคัญน้อยที่สุดเข้าไปก่อน
-