

สาระสำคัญ

1. อุปกรณ์ที่นำมาเก็บข้อมูลขนาดใหญ่จะถูกเรียกว่า หน่วยความจำ
2. หน่วยความจำแบบที่ง่ายที่สุดคือ หน่วยความจำแบบรอม ซึ่งเป็นแบบที่อ่านอย่างเดียว โครงสร้างภายในของมันมักเป็นไดโอด หรือทรานซิสเตอร์ ที่จะทำหน้าที่เก็บข้อมูลในรูปเลขฐานสอง หน่วยความจำแบบรอมแบ่งออกเป็นประเภทย่อย ๆ คือ มาสค์รอม(Mask ROM) พร็อม(PROM) , อีพร็อม(EPROM) , และอีอีพร็อม(EEPROM)
3. หน่วยความจำที่อ่านและเขียนได้คือ หน่วยความจำแบบแรม โครงสร้างพื้นฐานคล้ายกับหน่วยความจำแบบรอมเพียงแต่มีอินพุตที่สามารถเขียนข้อมูลในรูปเลขฐานสองไปได้ หน่วยความจำแบบแรมแบ่งออกเป็นประเภทใหญ่ ๆ คือ เอสแรม (SRAM) , ดีแรม(DRAM) , ชูโดสแตติกแรม (Pseudostatic RAM) , เอ็นวีแรม (NVRAM) , เอฟแรม(FRAM) และแบบแกนแม่เหล็ก

เนื้อหา

- 8.1 กล่าวนำ
- 8.2 หน่วยความจำแบบรอม (ROM)
- 8.3 หน่วยความจำแบบแรม (RAM)

จุดประสงค์การเรียนรู้

เมื่อจบการศึกษานี้แล้ว นักศึกษาสามารถ

1. อธิบายกลไกที่ใช้ในการเก็บข้อมูลในรูปเลขฐานสองลงในอุปกรณ์หน่วยความจำแบบสารกึ่งตัวนำ ประเภทรอมและแรมได้
2. อธิบายถึงโครงสร้างภายในของหน่วยความจำแบบรอมและแรมได้
3. อธิบายถึงข้อดีและข้อเสียของหน่วยความจำแบบรอมและแรมประเภทต่าง ๆ

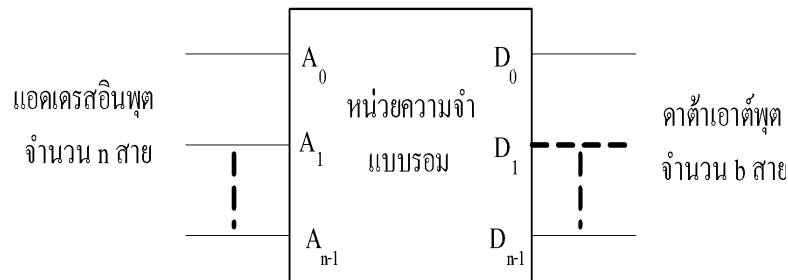
8.1 กล่าวนำ

ฟิลิปฟลิปที่เราได้ศึกษาไปแล้วนั้นสามารถนำมาใช้เป็นตัวเก็บข้อมูลหรือจดจำข้อมูลขนาด 1 บิตได้เมื่อนำฟิลิปฟลิปหลาย ๆ ตัวมาต่อกันเป็นรีจิสเตอร์ทำให้เราสามารถใช้นั่งจดหรือจำข้อมูลยาว ๆ ขนาดหลายบิต (เราเรียกข้อมูลขนาด n บิตว่าคำหรือ Word) แต่บ่อยครั้งที่เราต้องการเก็บคำเป็นจำนวนมาก อาจเป็นพันหรือมากกว่า อุปกรณ์ที่จะเก็บข้อมูลมาก ๆ นี้จะเรียกว่า หน่วยความจำ(Memory) หน่วยความจำที่เราจะศึกษาถึงคือ หน่วยความจำแบบอ่านอย่างเดียว หรือหน่วยความจำแบบรอม(ROM หรือ Read-Only

Memory) และหน่วยความจำแบบที่อ่านและเขียนได้ หรือหน่วยความจำแบบ แรม(RAM หรือ Random Access Memory)

8.2 หน่วยความจำแบบรอม (ROM)

หน่วยความจำแบบรอมถือเป็นหน่วยความจำแบบที่ง่ายที่สุด และอาจพิจารณามันให้เป็นวงจรคอมบิเนชันแบบหนึ่งที่มีอินพุต n สาย และเอาต์พุต b สาย โครงสร้างพื้นฐานของมัน สามารถแสดงได้ดังรูปที่ 8.1



รูปที่ 8.1 โครงสร้างพื้นฐานของหน่วยความจำแบบรอมขนาด $2^n \times b$ บิต

จากรูปที่ 8.1 จะเห็นได้ว่าอินพุตที่มีอยู่ n สายนั้นจะถูกเรียกว่า แอดเดรสอินพุต(Address input) หรืออินพุตตำแหน่งที่อยู่ ซึ่งมีชื่อเป็น A_0, A_1, A_2 จนถึง A_{n-1} ส่วนเอาต์พุตที่มีอยู่ b สายนั้นจะถูกเรียกว่า ดาต้าอินพุต(Data input) หรือเอาต์พุตข้อมูลซึ่งมีชื่อเป็น D_0, D_1, D_2 จนถึง D_{n-1}

ตัวอย่างที่ 8.1 ตารางค่าความจริงดังตารางข้างล่างเป็นตารางค่าความจริงของฟังก์ชันลอจิกแบบคอมบิเนชันที่มี 3 อินพุต 4 เอาต์พุต ข้อมูลในตารางสามารถไปเก็บไว้ในหน่วยความจำแบบรอมขนาด $2^n \times b = 2^3 \times 4 = 32$ บิต ถ้าสมมุติให้การเดินทางของสัญญาณไม่มีดีเลย์(ความล่าช้า) แล้ว ดาต้าเอาต์พุตของหน่วยความจำแบบรอมในทุกเวลาจะมีค่าเท่ากับบิตเอาต์พุตในแถวบนของตารางค่าความจริง ส่วนจะมีค่าเป็นบรรทัดใดแถวนั้นก็ขึ้นกับแอดเดรสอินพุต ที่เลือก เช่นดาต้าเอาต์พุต $D_3 D_2 D_1 D_0 = 0010$ ถ้าแอดเดรสอินพุตที่เราเลือกคือ $A_2 A_1 A_0 = 101$ เป็นต้น

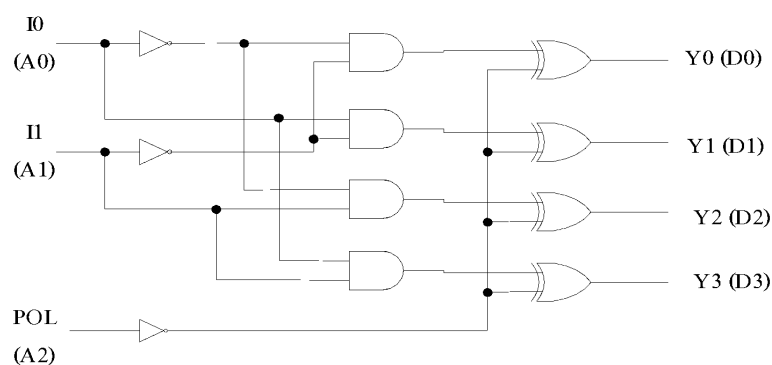
อินพุต			เอาต์พุต			
A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	1	1	1	0
0	0	1	1	1	0	1

0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

ตารางค่าความจริงสำหรับฟังก์ชันลอจิกแบบคอมบินเนชันที่มี 3 อินพุต 4 เอาต์พุต

ข้อสังเกต เนื่องจากหน่วยความจำแบบรอมเป็นวงจรคอมบินเนชัน (เพราะว่าเอาต์พุตขึ้นกับอินพุตอย่างเดียว) ดังนั้นถ้ามองในแง่ของการทำงานเชิงดิจิทัลแล้วเราอาจเห็นว่ามันไม่ได้เป็นหน่วยความจำ มันเป็นเพียงวงจรลอจิกคอมบินเนชันแบบหนึ่งเท่านั้น แต่อย่างไรก็ตามถ้ามองในแง่หนึ่งคือ ข้อมูลได้ถูกเก็บไว้ในหน่วยความจำแบบรอมในขณะที่มันถูกผลิตขึ้น หรือถูกโปรแกรมไว้ว่ามันเป็นหน่วยความจำชนิดหนึ่งซึ่งมันมีความแตกต่างหลักอยู่ประการหนึ่งที่ไม่เหมือนกับหน่วยความจำแบบวงจรรวมหรือแบบไอซีอื่นๆ คือ หน่วยความจำแบบรอมเป็นหน่วยความจำแบบนอนโวลาทิล (Nonvolatile Memory) นั่นคือข้อมูลที่เก็บไว้จะไม่สูญหายแม้ว่าไม่มีไฟเลี้ยงป้อนเข้า

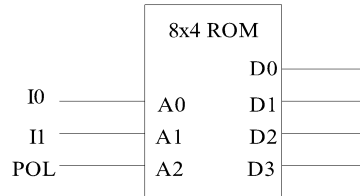
ตารางค่าความจริงข้างต้นนั้นถ้าพิจารณาให้ดีมันก็คือตารางค่าความจริงของวงจรถรหัสแบบเข้า 2 ออก 4 ที่มีเอาต์พุตโพลาริตีคอนโทรล (Output Polarity Control) ซึ่งมันสามารถสร้างขึ้นได้โดยใช้ลอจิกเกตเดี่ยว ๆ มาต่อกัน ดังรูปที่ 8.2



รูปที่ 8.2 วงจรถรหัสแบบเข้า 2 ออก 4 ที่มีเอาต์พุตโพลาริตีคอนโทรล

ดังนั้นอาจกล่าวได้ว่า เรามีวิธีการวงจรลอตรหัสดังกล่าวอยู่ 2 วิธีคือ

1. สร้างโดยใช้ลอจิกเกตเดี่ยว ๆ ดังรูปที่ 8.2
2. สร้างโดยใช้หน่วยความจำแบบรวมที่มีข้อมูลตามตารางข้างต้นบรรจุอยู่ ดังรูปที่ 8.3 (หรืออาจมองได้ว่าหน่วยความจำแบบรวมเป็นตัวแทนตารางค่าความจริงในรูปฮาร์ดแวร์(Hardware))



รูปที่ 8.3 การสร้างวงจรลอตรหัสแบบเข้า 2 ออก 4 โดยใช้หน่วยความจำแบบรวมขนาด 8×4 บิต

จากข้อมูลในตารางความจริงข้างต้น ทำให้เราสามารถกำหนดรูปแบบของอินพุต และเอาต์พุตของวงจรลอตรหัสให้กับอินพุตและเอาต์พุตของหน่วยความจำแบบรวมได้ ดังรูปที่ 8.3 วงจรลอตรหัสที่สร้างโดยหน่วยความจำแบบรวมอันนี้ไม่ได้มีลักษณะเฉพาะเจาะจง นั่นคือเราสามารถเขียนแถว และคอลัมน์ของตารางค่าความจริงในลำดับที่ต่างกันได้ และเราสามารถใชหน่วยความจำแบบรวมที่แตกต่างกันออกไปในการทำหน้าที่ฟังก์ชันลอจิกเดียวกันได้ เพียงแค่เรากำหนดสัญญาณของวงจรลอตรหัสให้กับอินพุตและเอาต์พุตของหน่วยความจำแบบรวมในลักษณะที่แตกต่างกันออกไป ตัวอย่างเช่นถ้าเราสลับบิตข้อมูลในคอลัมน์ D_0 กับ D_2 ของตารางข้างต้น จะทำให้เราได้ ตารางค่าความจริงอันใหม่และหน่วยความจำแบบรวมใหม่ขึ้นมา แต่เราก็ยังใชหน่วยความจำแบบรวมอันใหม่นี้มาใช้สร้างเป็นวงจรลอตรหัสแบบเข้า 2 ออก 4 ได้โดยเพียงแค่สลับ Y_0 ให้เป็น Y_2 เช่นกันถ้าเราสลับบรรทัดแนวนอนของตารางค่าความจริงข้างต้นให้ออกมาเป็นดังตารางข้างล่าง

อินพุต			เอาต์พุต			
A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	1	1	1	0
0	0	1	0	0	0	1
0	1	0	1	1	0	1
0	1	1	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	1	0	0	1	1	1

1	1	1	1	0	0	0
---	---	---	---	---	---	---

ตารางค่าความจริงที่ข้อมูลในแวนอนถูกสลับ

เราจะได้หน่วยความจำแบบบรอมอันใหม่ แต่อันใหม่นี้ก็ยังคงสามารถใช้เป็นวงจรถอดรหัสแบบเข้า 2 ออก 4 ได้ ด้วยการจัดแอดเดรสอินพุตใหม่ โดยให้ $A_0 = \text{POL}$, $A_1 = I_0$ และ $A_2 = I_1$ เมื่อเราสร้างหน่วยความจำแบบบรอมเพื่อเก็บข้อมูลตามตารางที่ให้มานั้นอินพุต $A_2A_1A_0$ มักถูกกำหนดให้เป็นแอดเดรสอินพุต และเอาต์พุต $D_3D_2D_1D_0$ มักถูกกำหนดให้เป็นดาต้าเอาต์พุต วิธีการอ่านคือ จากขวาไปซ้ายและบนลงล่าง โดยทั่วไปเรามักใช้ไฟล์ข้อมูล (Data File) อันหนึ่งเพื่อระบุถึงตารางค่าความจริงที่เก็บไว้ในหน่วยความจำแบบบรอมในขณะที่มันถูกโปรแกรมหรือถูกผลิตขึ้น โดยไฟล์ข้อมูลมักบอกแอดเดรส(ตำแหน่งที่อยู่) และข้อมูลที่อยู่ในแอดเดรสเป็นเลขฐานสิบหก เช่น ตารางข้างต้น อาจมีไฟล์ข้อมูลที่บอกว่าแอดเดรส 0 ถึง 7 จะมีข้อมูล คือ E, 1, D, 2, B, 4, 7 8 ซึ่งถ้ากลับไปดูในตารางพบว่า

บรรทัดแรกมีแอดเดรสเป็น 0 ($A_2A_1A_0 = 000$) มีข้อมูล = 14 หรือ E

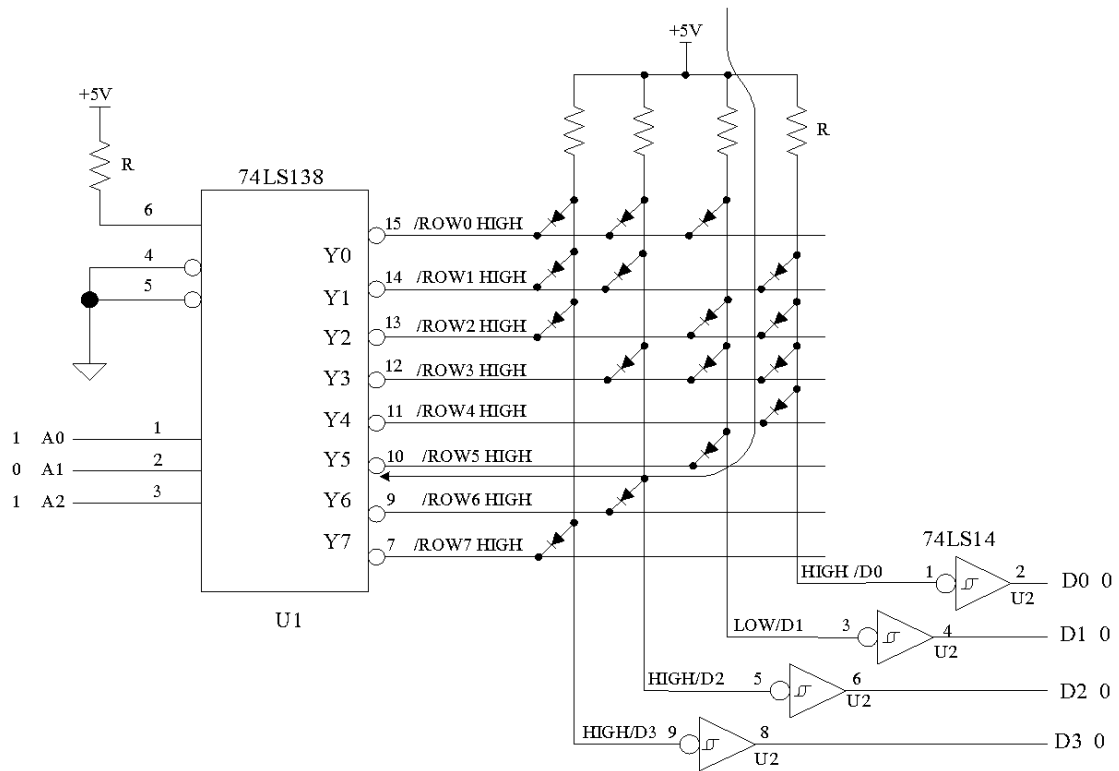
($D_3D_2D_1D_0 = 1110$)

บรรทัดสองมีแอดเดรสเป็น 1 ($A_2A_1A_0 = 001$) มีข้อมูล = 1 = ($D_3D_2D_1D_0 = 0001$)

และบรรทัดต่อ ๆ ไปจะมีค่าเป็น B, 4, 7 และ 8 เป็นต้น

โครงสร้างภายในของหน่วยความจำแบบบรอม

กลไกที่หน่วยความจำแบบบรอมใช้เก็บข้อมูลนั้นมีหลายแบบโดยขึ้นกับเทคโนโลยีว่าจะเป็นแบบใด แต่ส่วนใหญ่แล้วมักใช้ไดโอดหรือทรานซิสเตอร์เพื่อทำหน้าที่แยกระหว่าง 1 กับ 0 รูปที่ 8.4 เป็นไดอะแกรมของหน่วยความจำแบบบรอมขนาด 8×4 บิต แบบพื้นฐานซึ่งสร้างขึ้นโดยใช้วงจรถอดรหัส 74LS138 กับไดโอด



รูปที่ 8.4 วงจรลอจิกของหน่วยความจำแบบรอมขนาด 8×4 บิตแบบพื้นฐาน

จากรูปที่ 8.4 พบว่าแอดเดรสอินพุต ($A_2 A_1 A_0$) จะเป็นตัวเลือกเอาต์พุตของวงจรถอดรหัส (Y_0 ถึง Y_7) ขึ้นมาหนึ่งสายตามต้องการโดยเอาต์พุตของวงจรถอดรหัสแต่ละสายจะถูกเรียกว่า เวิร์ดไลน์ (Word Line) เพราะมันจะเป็นตัวเลือกแถวในแนวนอน 1 แถว (หรือคำ 1 คำ) ของตารางค่าความจริงที่ได้ถูกเก็บไว้ในหน่วยความจำแบบรอม ในรูปที่ 8.4 ได้แสดงให้เห็นถึงสถานการณ์ที่แอดเดรสอินพุต $A_2A_1A_0 = 101$ และแถวที่ 5 ได้ถูกเลือก (ในรูปที่ 8.4 ใช้สัญลักษณ์ /ROW โดย / หมายถึงแอกทีฟโลว์ และ ROW แปลว่าแถว) ส่วนเส้นในแนวดิ่งแต่ละเส้นในรูปจะถูกเรียกว่า บิตไลน์ (Bit Line) เพราะมันตรงกับบิตเอาต์พุต 1 บิต ของหน่วยความจำแบบรอม เวิร์ดไลน์ที่ถูกเลือกเส้นหนึ่งจะเป็นตัวดึงบิตไลน์เส้นหนึ่งให้เป็นโลว์ ถ้ามีไดโอดเชื่อมต่อระหว่างเวิร์ดไลน์กับบิตไลน์ในรูปจะเห็นว่ามีไดโอดแค่หนึ่งตัวในแถวแนวนอนที่ 5 (/ROW5) ดังนั้นบิตไลน์ที่สัมพันธ์กัน (ซึ่งก็คือ /D1) จะถูกดึงเป็นโลว์ ในรูปจะเห็นว่าบิตไลน์จะถูกต่อเข้ากับขั้วอินพุตของเวอร์เตอร์ 74LS14 เพื่อเป็นการปรับระดับสัญญาณรบกวนของวงจร และท้ายสุดเราก็จะได้เอาต์พุตของหน่วยความจำเป็นรอม $D_3 D_2 D_1 D_0$ เป็น 0010 สำหรับกรณีในรูปที่ 8.4 ซึ่งสัมพันธ์กับค่าในตารางค่าความจริงข้างต้น

จะเห็นได้ว่าจุดตัดระหว่างเวิร์ดไลน์ (แนวนอน) กับบิตไลน์ (แนวตั้ง) แต่ละจุดเท่ากับ 1 บิต ของหน่วยความจำ ถ้ามีไดโอดปรากฏอยู่ที่จุดตัดเท่ากับว่า 1 บิต ได้ถูกเก็บไว้ในหน่วยความจำ ถ้าหากไม่มีไดโอดอยู่หมายถึง

ถึง มีบิต 0 ได้ถูกเก็บไว้ในหน่วยความจำ ดังนั้นถ้าเราสร้างหน่วยความจำแบบบรอมลักษณะที่นี้เราต้อง “โปรแกรม” มันโดยการแทรกหรือดึงไดโอดเข้าไปหรือออกจากจุดตัดแต่ละจุด วิธีนี้เป็นวิธีที่ใช้ในคอมพิวเตอร์สมัยปี 1970 ซึ่งการเก็บข้อมูลลงในหน่วยความจำแบบบรอมลักษณะนี้ทำโดยการดึงเอาไดโอดออกจากจุดตัดที่ต้องการเก็บบิต 0 ไว้

ชนิดของหน่วยความจำแบบบรอม

หน่วยความจำแบบบรอมในลักษณะที่กล่าวมานั้นจะได้พบเห็นกันแล้ว ปัจจุบันจะพบในรูปของไอซีชิปซึ่งรวมไอซีชิปตัวหนึ่งสามารถเก็บข้อมูลได้เป็นล้านบิตและมีราคาไม่แพงทั้งนี้ก็เป็นไปตามคำแนะนำของ นาย กอร์ดอน มอร์ ประธานบริษัทอินเทล ซึ่งกล่าวไว้ว่า “จำนวนบิตของหน่วยความจำที่จะมีได้ในไอซีชิป 1 ตัวนั้นจะเพิ่มขึ้น 4 เท่าในเวลา 2 ปี” ดังนั้นในอนาคตหน่วยความจำก็จะมีขนาดเล็กลงแต่สามารถเก็บข้อมูลได้มากนั่นเอง ประเภทของหน่วยความจำแบบบรอมอาจสรุปได้ดังนี้

1. รอม (Read-Only Memory) เป็นหน่วยความจำที่ถูกโปรแกรมในช่วงที่ผลหรือจากโรงงานผู้ผลิตไม่สามารถเปลี่ยนแปลงได้ จัดเป็นหน่วยความจำประเภทนอนโวลตาไทล์

2. มาสค์รอม (Mask ROM) เป็นหน่วยความจำแบบบรอมประเภทไอซีสมัยแรก ๆ มักผลิตในกรณีที่มีจำนวนขึ้นมาก ๆ และไม่ต้องการเปลี่ยนแปลงข้อมูลในภายหลัง เพราะราคาจะถูกในการผลิตแต่สำหรับงานที่มีจำนวนขึ้นไม่มาก รอมประเภทนี้จะไม่คุ้มค่าในการผลิตและจัดเป็นหน่วยความจำประเภทนอนโวลตาไทล์

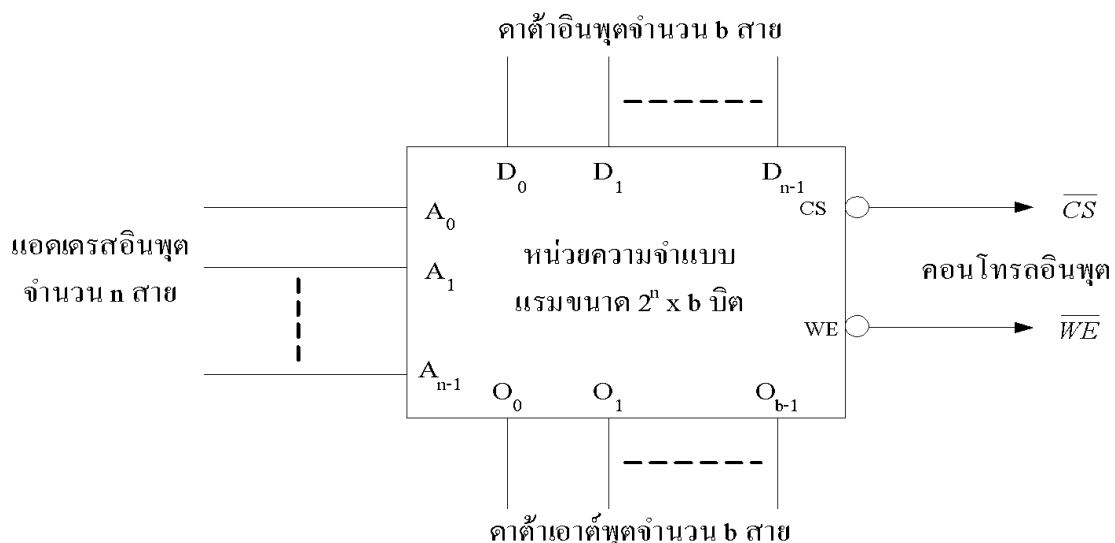
3. พรอม (PROM หรือ Programmable Read-Only Memory) จะคล้ายกับมาสค์รอม แต่ต่างกันที่ผู้ใช้สามารถเก็บข้อมูล (หรือ “โปรแกรม” มัน) ได้ และใช้เวลาเพียงไม่กี่นาทีโดยใช้พรอมโปรแกรมเมอร์ ในไอซีชิปของหน่วยความจำแบบพรอมนี้จะไม่ไดโอดหรือทรานซิสเตอร์ที่ต่อเข้าด้วยกันซึ่งก็เท่ากับว่ามีบิต 1 เก็บอยู่ ผู้ใช้สามารถโปรแกรมโดยใช้พรอมโปรแกรมเมอร์ในการตั้งค่าที่ต้องการให้เป็นบิต 0 ได้โดยการติดฟิวส์ภายในวงจรออก ดังนั้นมันจึงไม่สามารถนำมาโปรแกรมใหม่ได้ (เขียนได้ครั้งเดียว) จัดเป็นหน่วยความจำประเภทนอนโวลตาไทล์

4. อีพรอม (EPROM หรือ Erasable Programmable Read-Only Memory) อีพรอมสามารถโปรแกรมได้เหมือนกับพรอมโดยการเก็บประจุ (Charge) ไว้บนเกตมีฉนวนในตัวมัน แต่สามารถลบข้อมูลได้ด้วยไฟฟ้าแทนที่จะเป็นแสง อัลตราไวโอเลตหลังลบข้อมูลออกแล้วสามารถโปรแกรมใหม่ได้ จัดเป็นหน่วยความจำประเภทนอนโวลตาไทล์

8.3 หน่วยความจำแบบแรม (RAM)

หน่วยความจำแบบรวมที่ศึกษานั้นเป็นหน่วยความจำที่อ่านได้อย่างเดียว หน่วยความจำแบบแรมนั้นเป็นแบบที่อ่านและเขียนได้ แรม ย่อมาจากคำว่า **Random Access Memory** ซึ่งหมายถึง เวลาที่มันใช้อ่านหรือเขียนข้อมูล 1 บิต จะไม่ขึ้นกับตำแหน่งที่อยู่ของบิตนั้นในหน่วยความจำ หรือกล่าวอีกนัยหนึ่งคือเราสามารถอ่านและเขียนข้อมูลในตำแหน่งที่อยู่ใดก็ได้ในหน่วยความจำ (แทนที่จะเลื่อนบิตข้อมูลออกแบบอนุกรมเพื่อให้ได้บิตข้อมูลที่ต้องการ) เทมนี้อาจสร้างความสับสนให้กับหลายคนเพราะหน่วยความจำแบบรวมที่ได้ศึกษาไปนั้นก็มีลักษณะเช่นนี้คือเป็นแบบ “Random Access” เหมือนกัน แต่โดยทั่วไปแล้วชื่อ Random Access Memory จะหมายถึง หน่วยความจำแบบแรมที่สามารถอ่านและเขียนได้หลาย ๆ ครั้งตามที่เราต้องการหลายท่านอาจจะมีความสงสัยต่อว่าอีอีพร้อมที่ได้ศึกษาไปก็สามารถนำมาใช้อ่านและเขียนข้อมูลได้ แล้วมันมีความแตกต่างอย่างไรกับหน่วยความจำแบบแรม คำตอบก็คือ หน่วยความจำแบบแรมเก็บบิตข้อมูลใน ฟลิปฟล็อปหรือในรูปประจุที่เก็บอยู่บนตัวเก็บประจุ เพื่อว่าบิตข้อมูลใหม่สามารถเขียนลงในตำแหน่งที่ตั้งอยู่ที่ต้องการได้ภายในเวลาเร็วมาก (นาโนวินาที) และการอ่านข้อมูลก็ทำได้ในเวลารวดเร็วได้เช่นกัน แต่อีอีพร้อมต้องใช้เวลาในการเขียนหรือโปรแกรมข้อมูลนานกว่า (เป็นไมโครวินาทีหรือมิลลิวินาที)

โครงสร้างพื้นฐานของหน่วยความจำแบบแรมก็คล้าย ๆ กับหน่วยความจำแบบรวม คือ มีแอดเดรสอินพุต คอนโทรลอินพุต ดาต้าเอาต์พุต แต่ที่เพิ่มมาอีกอันคือ ดาต้าอินพุต หน่วยความจำแบบแรมขนาด $2^n \times b$ บิต สามารถแสดงได้ดังรูปที่ 8.5 โดย n เป็นจำนวนสายของแอดเดรสอินพุต b เป็นจำนวนสายของดาต้าอินพุต และดาต้าเอาต์พุต



รูปที่ 8.5 โครงสร้างพื้นฐานของหน่วยความจำแบบแรมขนาด $2^n \times b$ บิต

จากรูปที่ 8.5 พบว่ามันสามารถใช้เก็บหรือจดจำข้อมูลจำนวน 2^n คำ โดยที่แต่ละคำมีความยาว b บิต ถ้าพิจารณาหน่วยความจำแบบแรมในรูปไอซีจากกล่าวได้ว่า ไอซีชิปนี้จะต้องมีรีจิสเตอร์อยู่ 2^n ตัวคือ R_0, R_1, R_2 จนถึง R_{n-1} และแต่ละรีจิสเตอร์จะประกอบด้วย ฟลิปฟล็อปจำนวน b ตัว รีจิสเตอร์เหล่านี้จะถูกระบุโดยระดับลอจิกของ A_0, A_1, A_2 จนถึง A_{n-1} ซึ่งเราจะเรียกมันว่า แอดเดรสบิตหรือแอดเดรสอินพุต หรืออาจเรียกว่า แอดเดรสเฉย ๆ ก็ได้ เพื่อความง่ายในการอธิบายรองมาพิจารณาหน่วยความจำแบบแรมขนาด 8 คำ $\times 4$ บิต ($n=3$ และ $b=4$) จะพบว่าเมื่อ $A_2, A_1, A_0 = 000$ นั้นเป็นการระบุถึงรีจิสเตอร์ R_0 เมื่อ $A_2, A_1, A_0 = 001$ เท่ากับระบุถึงรีจิสเตอร์ R_1 ไปเรื่อย ๆ จนถึงรีจิสเตอร์ R_7 (เนื่องจากมีแอดเดรสบิต = 3 ดังนั้นจึงมีรีจิสเตอร์ทั้งหมด $2^3 = 8$) ตอนนี้รีจิสเตอร์แต่ละตัวมีแอดเดรสหรือตำแหน่งที่อยู่แล้ว การที่เราจะเข้าถึงรีจิสเตอร์นั้นการกระทำทำได้โดยการระบุแอดเดรสของมันนั่นเอง

ถ้าเราต้องการเขียนหรือเก็บคำขนาด 4 บิต ลงในรีจิสเตอร์ที่มีแอดเดรส (Addressed Registers) นั้นข้อมูลที่ต้องการเขียนจะถูกป้อนเข้าที่ดาต้าอินพุต D_3, D_2, D_1, D_0 และแอดเดรสของรีจิสเตอร์ต้องถูกป้อนเข้าที่ A_2, A_1, A_0 รวมทั้งคอนโทรลอินพุตนั้นก็คือ $\overline{WE}$ (Write Enable) และ $\overline{CS}$ (Chip Select) ต้องถูกเอนาเบิลหรือแอกทีฟในรูปจะเห็นได้ว่าทั้ง $\overline{WE}$ และ $\overline{CS}$ เป็นคอนโทรลอินพุตแบบแอกทีฟโลว์

ตรงกันข้ามถ้าเราต้องการอ่านคำขนาด 4 บิต จากหน่วยความจำแบบแรมนั้นเราต้องระบุแอดเดรสโดยการป้อนค่าแอดเดรสเข้าที่ A_2, A_1, A_0 เอนาเบิลอินพุต $\overline{CS}$ และอัสเอนาเบิลอินพุต $\overline{WE}$

หน่วยความจำก็จะนำค่าที่ต้องการออกมาที่ดาต้าเอาต์พุต O_3, O_2, O_1, O_0

ข้อสังเกต

1. ถ้า $\overline{WE} = \text{ไฮ}$ จะทำให้หน่วยความจำอ่านข้อมูลออกมา แต่ถ้า $\overline{WE} = \text{โลว์}$ เขียนหรือเก็บข้อมูลลงในหน่วยความจำ
2. ทั้งการเขียนและการอ่านนั้นอินพุต $\overline{CS}$ จะต้องถูกเอนาเบิลหรือแอกทีฟ ถ้าอินพุต $\overline{CS}$ ไม่อยู่ในสถานะแอกทีฟแล้ว จะเท่ากับเป็นการตัดชิปออกจากโลกภายนอก ดังนั้นจึงไม่สามารถกระทำการเขียนและอ่านได้
3. จำนวนบิตที่หน่วยความจำแบบแรมที่ยกตัวอย่างไปมีค่าเท่ากับ $2^3 \times 4 = 32$ บิต

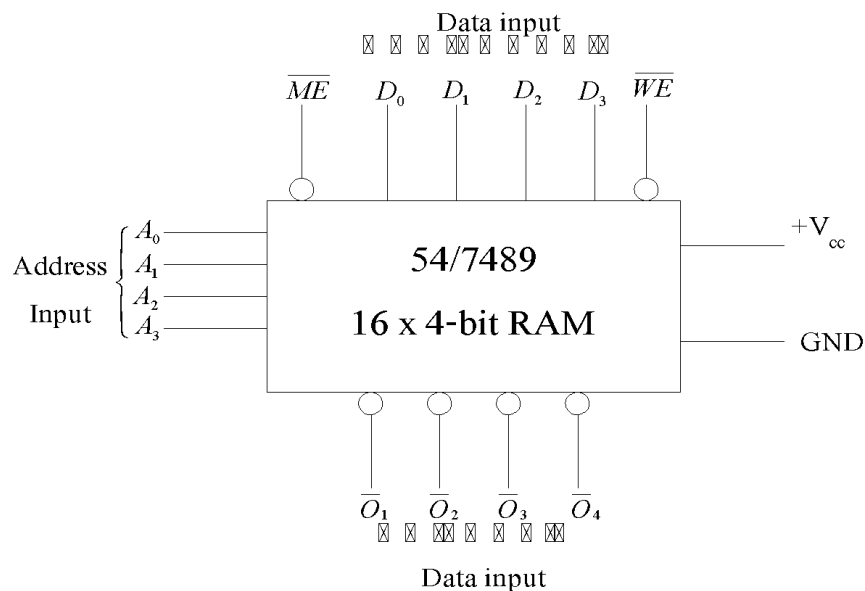
ตัวอย่างที่ 8.2 พิจารณาหน่วยความจำแบบแรมที่มีอยู่ในรูปไอซีชิป เช่น 54/7489 จัดเป็นหน่วยความจำแบบแรมขนาด 64 บิตชนิดทีทีแอล (TTL) และถูกจัดเป็นวงจรรวมขนาดกลาง (MSI) เซมิคอนดักเตอร์ชิป สามารถนำมาใช้เก็บ

ข้อมูลขนาด 4 บิต (1 คำ = 4 บิต) ได้ถึง 16 คำ ตำแหน่งที่อยู่ของคำทั้ง 16 คำ จะถูกเลือกได้โดยใช้แอดเดรสอินพุตที่เหมาะสม ดังนั้นเราจึงสามารถอ่านเขียน คำขนาด 4 บิต ในตำแหน่งที่เลือกไว้ได้

สัญลักษณ์ลอจิกของ 54/7489 สามารถแสดงได้ดังรูปที่ 8.6 เราสามารถ ป้อนคำขนาด 4 บิต เข้าสู่หน่วยความจำที่ดาต้าอินพุต D_4, D_3, D_2, D_1 และสามารถ อ่านคำขนาด 4 บิตจากหน่วยความจำได้ที่ดาต้าเอาต์พุต O_4, O_3, O_2, O_1 ตำแหน่งที่ อยู่ที่เราต้องการนำข้อมูลไปเก็บไว้สามารถเลือกได้โดยแอดเดรสอินพุต

A_3, A_2, A_1, A_0 การควบคุมชิปจะกระทำโดยอินพุต $\overline{ME}$ และอินพุต $\overline{WE}$ (ME ย่อมา จากคำว่า Memory Enable หรือมีอีกชื่อดังที่ได้ศึกษามาแล้วคือ Chip Select ($\overline{CS}$)

ส่วน $\overline{WE}$ ย่อมาจากคำว่า Write Enable)



รูปที่ 8.6 สัญลักษณ์ลอจิกของ 54/489 หน่วยความจำแบบแรม 64 บิตชนิดที่ ที่แอส

การกำหนดแอดเดรส

การกำหนดตำแหน่งที่อยู่หรือแอดเดรสของคำทั้งหมด 16 คำนี้ต้องอาศัย อินพุตแอดเดรส A_3, A_2, A_1, A_0 โดยที่อินพุตแอดเดรสแต่ละคำที่เป็นไปได้จะ ประกอบด้วยเลขฐานสอง 4 หลักเช่น

$$A_3, A_2, A_1, A_0 = 0000 \text{ จะเป็นการกำหนดตำแหน่งที่อยู่ } 00_{10}$$

$$A_3, A_2, A_1, A_0 = 0001 \text{ จะเป็นการกำหนดตำแหน่งที่อยู่ } 01_{10}$$

$$A_3, A_2, A_1, A_0 = 1111 \text{ จะเป็นการกำหนดตำแหน่งที่อยู่ } 15_{10}$$

ข้อสังเกต ตำแหน่งที่อยู่ของหน่วยความจำเป็นเลขฐานสิบจาก 0_{10} (ศูนย์) ถึง 15_{10} ซึ่งมีทั้งหมด

การควบคุมการทำงาน

โหมดการทำงานของ 54/7489 นั้นถูกกำหนดโดยระดับลิจิกของคอนโทรลอินพุตซึ่งก็คือชิปเซเลกต์อินพุต ($\overline{ME}$) กับไรต์เอนาเบิลอินพุต ($\overline{WE}$) ดังแสดงในตารางข้างล่าง

อินพุต		การดำเนินการ	สถานะของเอาต์พุต
$\overline{ME}$	$\overline{WE}$		
โลว์	โลว์	เขียนข้อมูล	เป็นคอมพลีเมนต์ของดาต้าอินพุต
โลว์	ไฮ	อ่านข้อมูล	เป็นคอมพลีเมนต์ของค่าที่ถูกเลือก
ไฮ	โลว์	ห้ามการเก็บข้อมูล	กำหนดแน่นอนไม่ได้
ไฮ	ไฮ	จะไม่ทำอะไร	เป็นไฮ

ตารางแสดงการทำงานของ 54/7489

การทำงานดังตารางสามารถอธิบายได้ดังเป็นกรณี ๆ ดังนี้

1. เมื่อ $\overline{ME} = \overline{WE} = \text{โลว์}$ ระดับของข้อมูล (โลว์หรือไฮ) ที่ปรากฏที่ดาต้าอินพุต D_4, D_3, D_2, D_1 , จะถูกเก็บไว้ที่ตำแหน่งที่อยู่ที่เหลือโดยแอดเดรสอินพุต ในขณะเดียวกันก็จะมีคอมพลีเมนต์ของระดับข้อมูลไปปรากฏที่ดาต้าเอาต์พุต O_4, O_3, O_2, O_1 ตัวอย่างเช่น ถ้า $A_3, A_2, A_1, A_0 = 1010$ แล้ว ค่าขนาด 4 บิตนี้จะถูกเก็บไว้ที่ตำแหน่ง $0110_2 = 6_{10}$ ในขณะเดียวกันดาต้าเอาต์พุตจะเป็น 0101 (หรือเป็นคอมพลีเมนต์ของดาต้าเอาต์พุต) ซึ่งการดำเนินการนี้ก็คือ **“การเขียนข้อมูล”**

2. เมื่อ $\overline{ME} = \text{โลว์}$ และ $\overline{WE} = \text{ไฮ}$ ค่าที่ถูกเลือกโดยแอดเดรสอินพุตจะไปปรากฏที่เอาต์พุต (เป็นคอมพลีเมนต์) เช่น ถ้าค่า 1101 ได้ถูกเก็บไว้ในตำแหน่งที่อยู่ $A_3, A_2, A_1, A_0 = 1101 = 12_{10}$ แล้วดาต้าเอาต์พุตจะปรากฏเป็น $O_4, O_3, O_2, O_1 = 0010$ (หรือคอมพลีเมนต์ของค่าที่อยู่ตำแหน่ง 12_{10}) ซึ่งการดำเนินการนี้ก็คือ **“การอ่านข้อมูล”**

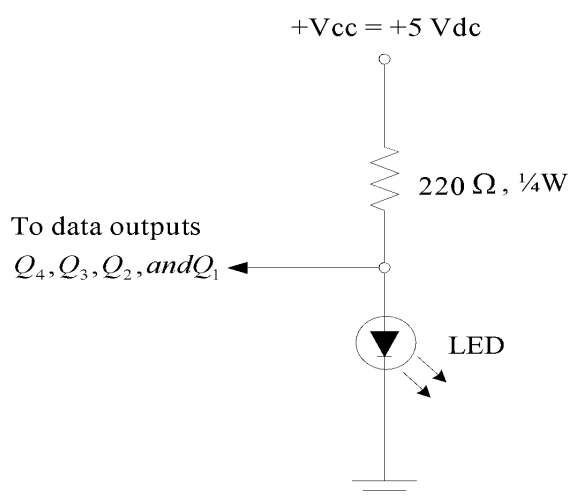
3. เมื่อ $\overline{ME} = \text{ไฮ}$ และ $\overline{WE} = \text{โลว์}$ สิ่งนี้เป็นการข้ามไม่ให้เก็บข้อมูล ในหน่วยความจำดาต้าเอาต์พุตก็จะมีสถานะที่กำหนดแน่นอนไม่ได้

4. เมื่อ $\overline{ME} = \overline{WE} = \text{ไฮ}$ จะเป็นการดิสเอเบิลและชิปจะไม่ทำอะไร เอาต์พุตทั้งหมดจะมีระดับเป็นไฮ หรือเอาต์พุตทรานซิสเตอร์เป็นออฟ (OFF) ทั้งหมดนั่นเอง

การแสดงผลของเอาต์พุต

เนื่องจากเอาต์พุต O_4, O_3, O_2, O_1 เป็นทรานซิสเตอร์แบบคอลเลคเตอร์เปิด (Open-Collector Transistors) ซึ่งหมายความว่าเราต้องมีโหลดข้างนอกมาต่อกับดาต้าเอาต์พุต เพื่อว่าทรานซิสเตอร์จะทำงานได้ถูกต้อง ซึ่งในทาง

ปฏิบัติก็อาจใช้แอลอีดี (LED) และตัวต้านทานขนาด $220\Omega, \frac{1}{4}W$ (ดังแสดงในรูปที่ 8.7) มาต่อกับดาต้าเอาต์พุตแต่ละตัวโดยถ้ามีบิต 1 ที่เอาต์พุตจะทำให้แอลอีดีเปล่งแสงและถ้ามีบิต 0 ที่เอาต์พุตจะทำให้แอลอีดีดับ



รูปที่ 8.7 การแสดงผลของเอาต์พุต

ชนิดของหน่วยความจำแบบแรม

หน่วยความจำแบบแรมนั้นมีอยู่หลายประเภท เช่น ประเภทที่สร้างจากแกนแม่เหล็กที่ทำให้มีอำนาจแม่เหล็กได้ หรือประเภทที่สร้างจากสารกึ่งตัวนำ เช่น ทรานซิสเตอร์ ซึ่งเราอาจแยกเป็นประเภทย่อย ๆ ได้ ดังนี้

1. เอสแรม (“S” RAM หรือ Static RAM) ในหน่วยความจำแบบนี้จะสร้างขึ้นโดยใช้ไบโพลาร์หรือมอสเฟต โดยที่ข้อมูลจะถูกเก็บในแถวลำดับ (Array) ของฟลิปฟล็อป ดังนั้นเราจึงสามารถเขียน อ่าน และเขียนข้อมูลซ้ำลงไปและข้อมูลจะยังคงค่าเดิมอยู่ตลอดตราบเท่าที่มีไฟเลี้ยงป้อนเข้าไป จัดเป็นหน่วยความจำประเภทโวลตาไทล์ (Volatile) คือข้อมูลจะสูญหายถ้าไม่มีไฟเลี้ยงป้อนเข้าไป

2. ดีแรม (“D” RAM หรือ Dynamic Ram) ในหน่วยความจำแบบนี้จะมีตัวเก็บประจุขนาดเล็กและมอสเฟตเป็นตัวเก็บข้อมูล ดังนั้นข้อมูลที่ถูกเก็บไว้

จะต้องถูกรีเฟรช(Refresh)อยู่บ่อยๆในทุกๆ 2-3 มิลลิวินาที โดยอาศัย สัญญาณนาฬิกาต่างหากเข้าไปกระตุ้น(อุปมาเหมือนนักวิ่งมาราธอนที่ต้อง ให้น้ำเป็นช่วงๆไม่เช่นนั้นนักวิ่งจะหมดแรง)เพราะไม่เช่นนั้นข้อมูลที่เก็บไว้ จะหายไป จัดเป็นหน่วยความจำแบบโวลาทิล

3. ชูโดสแตติกแรม(Pseudostatic RAM) จัดเป็นดีแรมที่มีวงจรรีเฟรช อยู่ภายในชิปดังนั้นถ้าเรามองดูเผินๆ จากภายนอกก็เหมือนกับว่ามันเป็น เอสแรม (คำว่า Pseudostaticแปลว่า สแตติกแบบเทียมนั่นเอง) และจัดเป็น หน่วยความจำแบบโวลาทิล

4. เอ็นวีแรม (“NV” RAM หรือNonvolatile RAM)ภายในจะประกอบไป ด้วยเอสแรมกับอีอีพรีมที่ขนานกันอยู่ ดังนั้นเมื่อมีการตัดไฟเลี้ยงเกิดขึ้น ข้อมูลแทนที่จะหายไปแบบแรมชนิดอื่นๆ ข้อมูลที่เก็บไว้ในเอสแรม สามารถถูกถ่ายโอนอย่างรวดเร็วเพื่อไปเก็บไว้ในอีอีพรีมก่อนและเมื่อมี ไฟเลี้ยงกลับคืนมาข้อมูลก็สามารถถูกถ่ายโอนกลับเข้าสู่เอสแรมดังนั้นมัน จึงจัดเป็นหน่วยความจำแบบนอนโวลาทิล(ข้อมูลไม่สูญหาย)

5. เอฟแรม (“F” RAM หรือ Ferroelectric RAM) มีโครงสร้างคล้ายกับ เอสแรมหรือดีแรมแต่ละข้อมูลจะถูกเก็บไว้บนตัวเก็บประจุไฟฟ้าเฟอร์โร เมื่อมีการตัดไฟเลี้ยงเกิดขึ้น ดังนั้นมันจึงจัดเป็นหน่วยความจำแบบนอนโวลาทิล

6. แบบแกนแม่เหล็ก (Magnetic Core) ในหน่วยความจำแบบนี้ใช้มากใน คอมพิวเตอร์ยุค

แรก ๆ ข้อมูลถูกเก็บในการทำแม่เหล็กกับเฟอร์ไรต์รูปโดนัทอันเล็ก ๆ ใน ทิศทางหนึ่งหรือในทิศทางตรงข้าม ข้อมูลสามารถถูกอ่านและเขียนเข้าไป ใหม่ได้ จัดเป็นหน่วยความจำแบบนอนโวลาทิล