

องค์ประกอบของเครื่องคอมพิวเตอร์
และภาษาแอสเซมบลี:
ARM และ RaspberryPi3

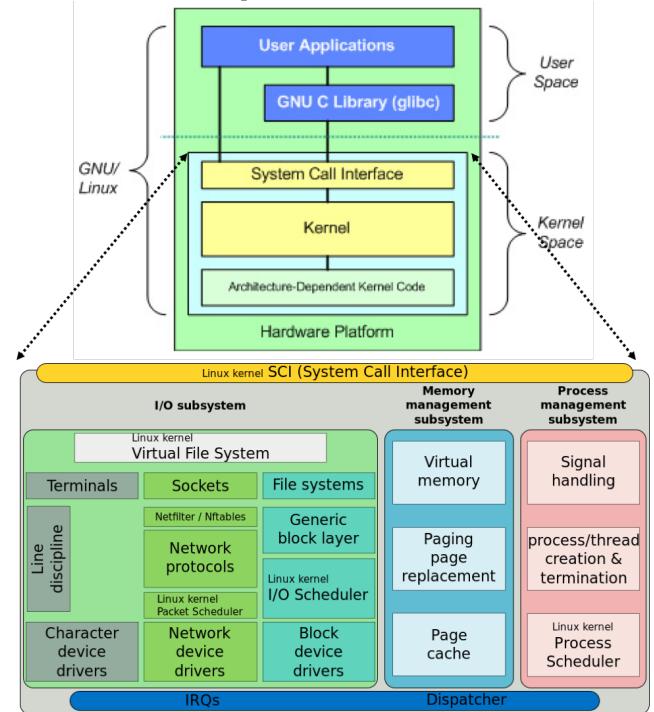
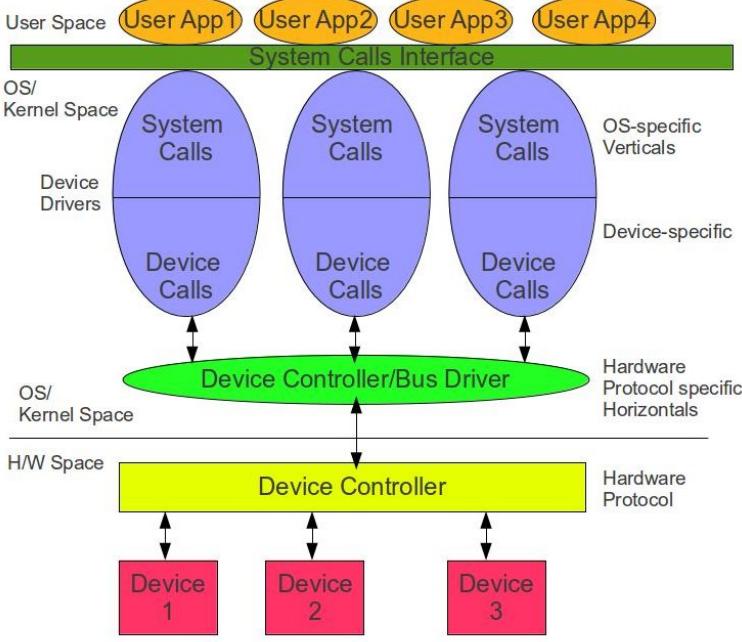
บทที่ 7 อุปกรณ์เก็บรักษาข้อมูลและระบบไฟล์

ผศ.ดร.สุรินทร์ กิตติธรกุล
ภาควิชาจิตวิทยาและนิเทศศาสตร์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สารบัญ

- บทที่ 1 บทนำ
- บทที่ 2 ข้อมูลและคณิตศาสตร์ในคอมพิวเตอร์
- บทที่ 3 ชาร์ดแวร์และซอฟท์แวร์ของคอมพิวเตอร์
- บทที่ 4 ภาษาแอสเซมบลีของ ARM เวอร์ชัน 32 บิต
- บทที่ 5 ลำดับขั้นของหน่วยความจำ
- บทที่ 6 อุปกรณ์/วงจรอินพุตและเอาท์พุต
- บทที่ 7 อุปกรณ์เก็บรักษาข้อมูลและระบบไฟล์

Input/Output: Software Perspective



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

3

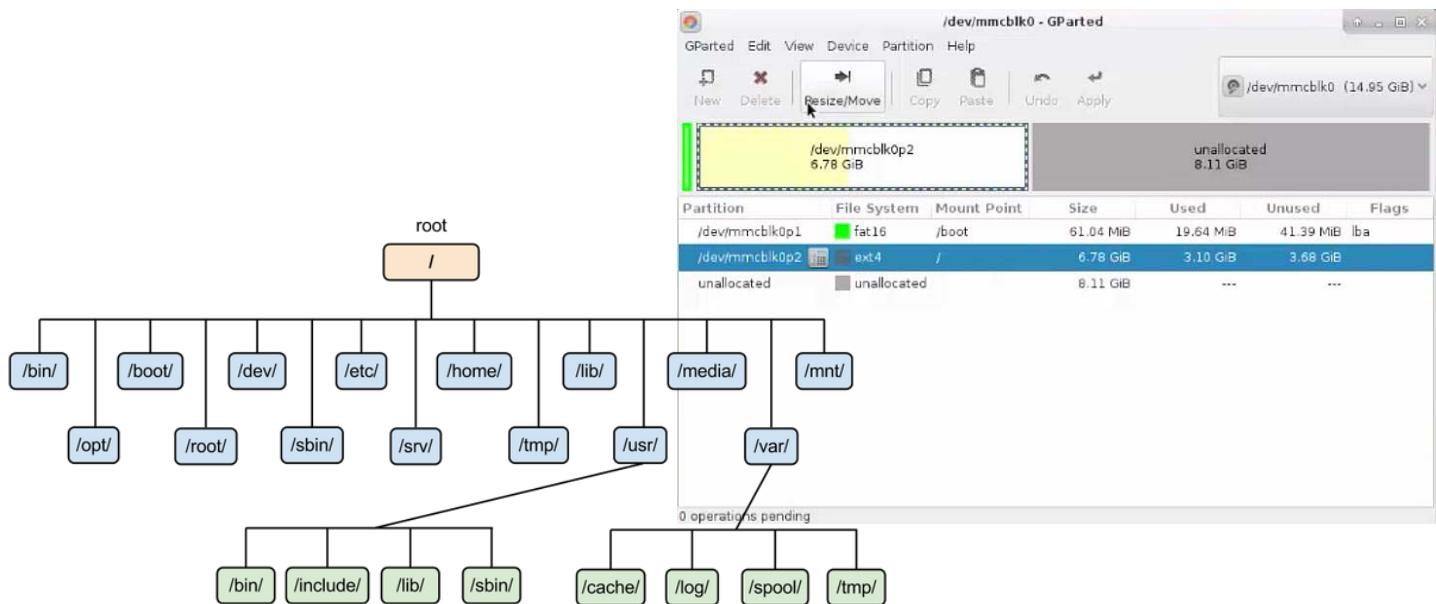
7.1 ระบบไฟล์ (File System)

- อุปกรณ์เก็บรักษาข้อมูล ทำหน้าที่บันทึกและเก็บไฟล์ทั้งหมดของระบบปฏิบัติการ ไฟล์โปรแกรม ไฟล์ข้อมูล และไฟล์อื่นๆ เพื่อให้เครื่องคอมพิวเตอร์สามารถทำงานได้ต่อเนื่องเมื่อเปิดเครื่องใหม่อีกรอบหลังจากการชั่งดาวน์
- การปิดเครื่องคอมพิวเตอร์เป็นการช่วยประหยัดพลังงาน หรือกรณีฉุกเฉินไฟฟ้าดับ หรือแบตเตอรี่หมด ซึ่งอาจทำให้ไฟล์ข้อมูลเสียหายหรือถึงขั้นสูญหาย
- ดังนั้น ผู้ใช้และซอฟต์แวร์ต้องมีหน้าที่บันทึก (Save) ข้อมูลลงในไฟล์ข้อมูลระหว่างการปฏิบัติงานด้วยเช่นกัน
- กรณีผู้ใช้ไม่สามารถชั่งดาวน์ระบบอาจทำให้ระบบไฟล์และไฟล์ข้อมูลเสียหายได้ทั้งนี้ขึ้นอยู่กับชนิดของระบบไฟล์

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

4

7.1 ระบบไฟล์ (File System)

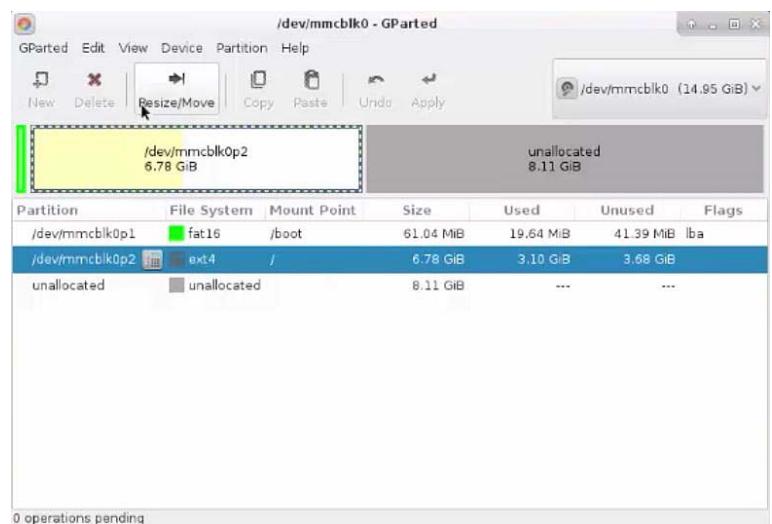
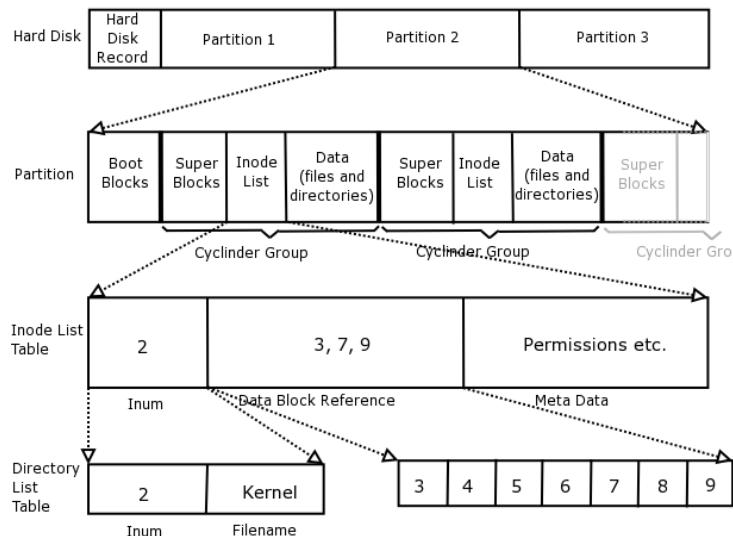


7.1 ระบบไฟล์ (File System)

- การแบ่งพาร์ติชัน คือ การแบ่งพื้นที่แบบข้อมูลออกเป็นส่วนต่างๆตามต้องการ
- การแบ่งพาร์ติชันสามารถตอบสนองความต้องการที่หลากหลาย ได้แก่ การบูทระบบปฏิบัติการได้หลากหลาย การจัดเก็บข้อมูลในบางพาร์ติชัน เป็นต้น
- ผู้ใช้สามารถติดตั้งระบบปฏิบัติการในแต่ละพาร์ติชันได้โดยอิสระจากกัน โดยตำแหน่งเริ่มต้นจะเป็นพื้นที่สำหรับเก็บ ตารางพาร์ติชัน (Partition Table) โดยรายชื่อและข้อมูลประกอบของแต่ละพาร์ติชันจะถูกบันทึกลงในตาราง

7.1 ระบบไฟล์ (File System)

UNIX File System Layout

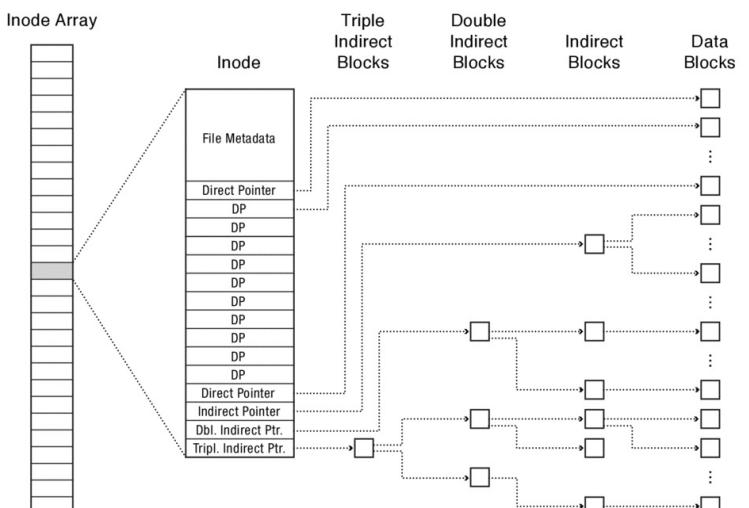
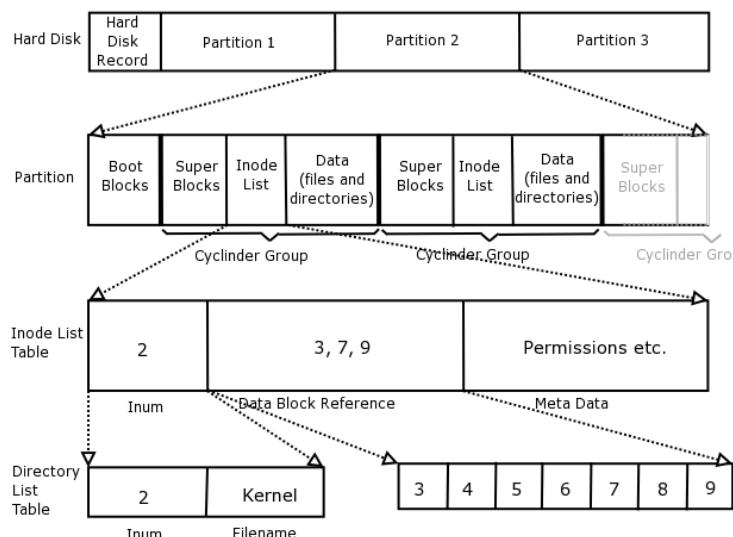


Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

7

7.1 ระบบไฟล์ (File System)

UNIX File System Layout



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

8

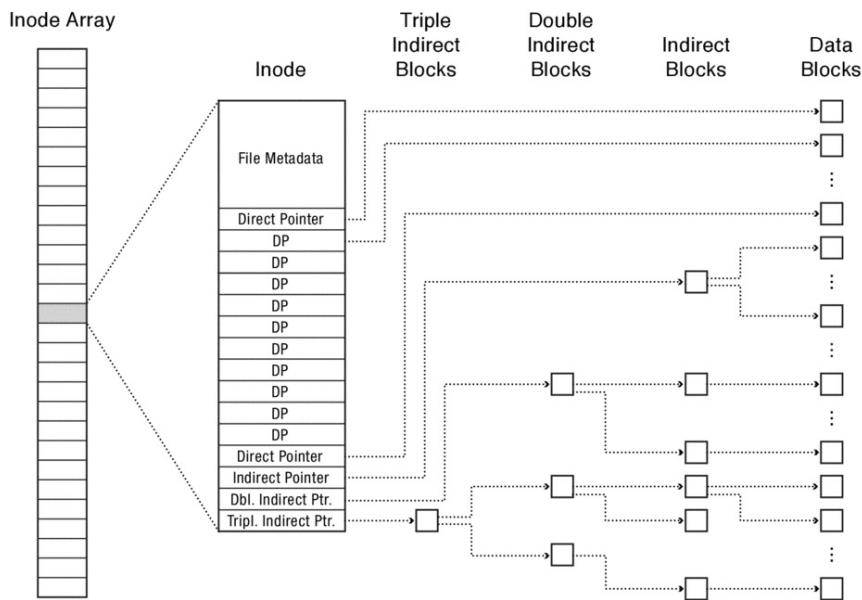
7.1 ระบบไฟล์ (File System)

- หน้าที่ของระบบไฟล์ คือ รองรับการจัดเก็บไฟล์ต่างๆ บันทึกประวัติการเข้าถึง (Access) ไฟล์ การบริหารสิทธิ์ (Permission) การเข้าถึงไฟล์ต่างๆ จากผู้ใช้งานจำนวนมาก การคุ้มครองไฟล์คืน การลบไฟล์ การใช้งานไฟล์ เป็นต้น ตามที่ผู้ใช้และซอฟต์แวร์ประยุกต์ต้องขอ
- โครงสร้างของระบบไฟล์ในระบบปฏิบัติการตระกูล Unix จะมีโครงสร้างและหน้าที่ของระบบไฟล์เป็นชั้นการจัดการ 3 ชั้น โดยเรียงลำดับ ดังนี้
 - ชั้น Logical เป็นชั้นบนสุดทำหน้าที่เชื่อมต่อ กับซอฟต์แวร์ประยุกต์ โดยใช้คำสั่งเปิดไฟล์ ปิดไฟล์ อ่านไฟล์ เขียนไฟล์ เป็นต้น ซึ่งไฟล์ที่ซอฟต์แวร์ประยุกต์ต้องการใช้งาน โดยเนื้อหาในบทนี้จะเน้นที่การทำงานในชั้น Logical
 - ชั้น Virtual ทำหน้าที่เชื่อมต่อระหว่างชั้น Logical และชั้น Physical รายละเอียดข้อมูลที่เกี่ยวข้องกับวิธีการพัฒนาโปรแกรมของแต่ละระบบปฏิบัติการ
 - ชั้น Physical เป็นชั้นล่างสุด ทำหน้าที่จัดการบล็อกข้อมูลบนอุปกรณ์ เพื่อตอบสนองต่อการร้องขอจากชั้น Virtual ซึ่งกลไกสำคัญคือ การบริหารบัฟเฟอร์ (Buffer) และหน่วยความจำหลัก การจัดวางตำแหน่งบล็อกข้อมูลให้ประสิทธิภาพการอ่านหรือเขียนได้ดีขึ้น การเขื่อมต่อ กับไดรเวอร์ของอุปกรณ์เก็บรักษาข้อมูล

7.1 ระบบไฟล์ (File System)

- การฟอร์แมท (Format) คือ การจัดแบ่งพื้นที่ออกเป็นพื้นที่ต่างๆ เป็นพื้นที่ส่วนย่อยๆ ดังนี้
 - บล็อกพิเศษ (Superblock) ทำหน้าที่เก็บรายละเอียดต่างๆ ของระบบไฟล์ ขนาดของบล็อกข้อมูล รายละเอียดการใช้งานบล็อกข้อมูลต่างๆ เช่น สถานะว่างหรือใช้งานอยู่ เป็นต้น
 - ตาราง Inode (Inode Table) หรืออาจเรียกว่า ไอโหนดอะเรย์ (Inode Array) หรือ ไอโหนดลิสต์ (Inode List) ทำหน้าที่เก็บโครงสร้างข้อมูล Inode ภายใต้ Cylinder Group นี้
 - บล็อกข้อมูล (Data Block) จำนวนหนึ่งแพร์พันตามขนาดที่ผู้ติดตั้งระบบกำหนดระหว่างที่ทำการฟอร์แมท

7.1 ระบบไฟล์ (File System)



7.1 ระบบไฟล์ (File System)

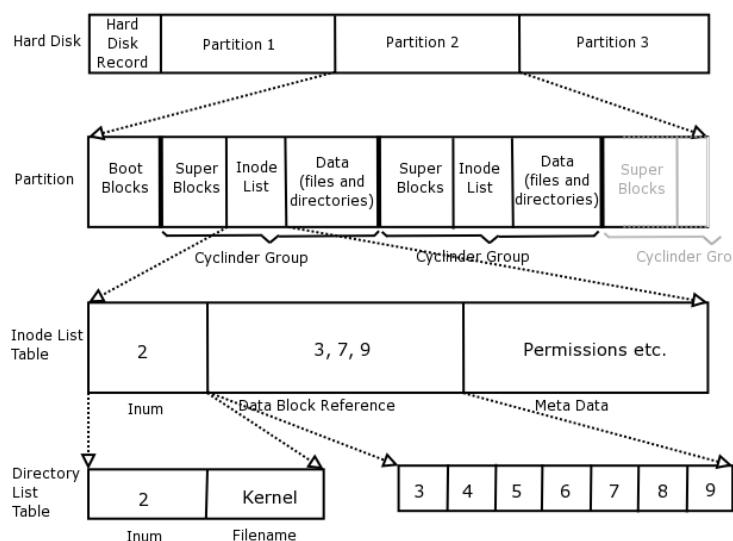
- ไอโหนด (Inode) คือโครงสร้างข้อมูล (Data Structure) ชนิดหนึ่งมีรายละเอียดและหมายเลขกำกับประจำตัว (Inode Number: Inum)
- ไอโหนดหนึ่งตัวจะทำหน้าที่แทนไฟล์ หรือ ไดเรคทอรี อย่างได้อย่างหนึ่ง
- โครงสร้างไอโหนด 1 ตัวต้องการพื้นที่ประมาณ 128 Bytes ขึ้นกับรายละเอียดของแต่ละระบบ ตาราง Inode Table หรือ Inode List จะจัดเรียง Inode ทั้งหมด โดยเรียงตามหมายเลข 0 เป็นต้นไป

7.1 ระบบไฟล์ (File System)

- ไอโหนดหนึ่งตัวทำหน้าที่จัดเก็บรายละเอียด ดังนี้
 - หมายเลขไอโหนด (Inode Number: Inum): หมายเลขไอโหนดประจำตัวของไฟล์หรือไดเรคทอรีนั้นๆ เป็นเลขจำนวนเต็มชนิดไม่มีเครื่องหมาย
 - การเขียนโยงไปยังบล็อกข้อมูล (Links to Data Blocks) ทำหน้าที่เก็บหมายเลขบล็อกข้อมูล (บล็อกสีเหลืองจตุรัส) ซึ่งทำหน้าที่เก็บข้อมูลจริงๆ การเขียนโยงแบ่งเป็น 3 ชนิด
 - หมายเลขบล็อกข้อมูลทางตรง (Direct Blocks หรือ Direct Pointers: DP)
 - หมายเลขบล็อกข้อมูลทางอ้อม (Indirect Blocks หรือ Indirect Pointers : Indirect Pointer, Double Indirect Pointer, Triple Indirect Pointer)

7.1 ระบบไฟล์ (File System)

UNIX File System Layout



- ไฟล์ชื่อ Kernel ในไดเรคทอรีราก (/ หรือ root directory)
- มีไอโหนดหมายเลข (Inum = 2) ซึ่งข้อมูลจะบันทึกอยู่ในบล็อกข้อมูลหมายเลข 3, 7, 9
- มีรายละเอียดสิทธิ์ต่างๆ ตามมา เรียกรวมๆ ว่า MetaData
- ส่วนชื่อไฟล์ (Filename) Kernel จะเก็บบันทึกในตารางไดเรคทอรีอ้างอิงโดยใช้ Inum = 2 เป็นหมายเลขอ้างอิง

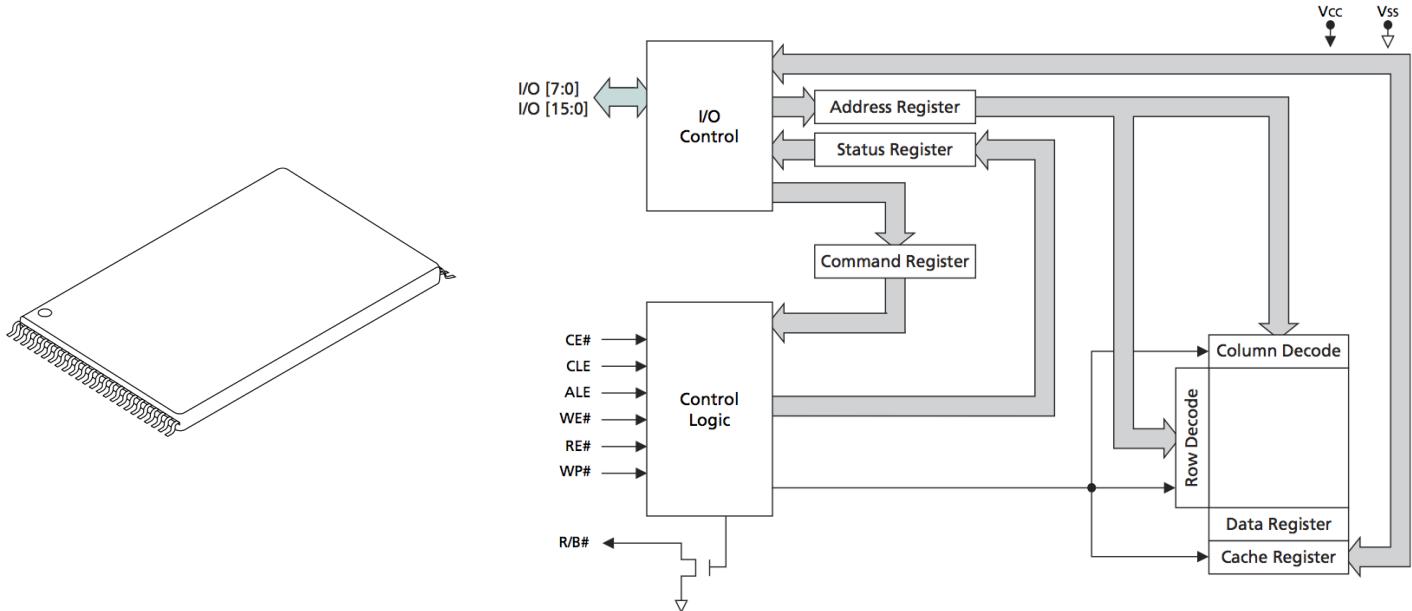
7.1 ระบบไฟล์ (File System)

- ข้อมูลอื่นเกี่ยวกับไฟล์ (File Metadata) แบ่งเป็น
- ชนิดไฟล์: ไฟล์ธรรมดา (regular file), ไดเรกทอรี (directory), ไป (pipe) เป็นต้น
 - ไฟล์ เกิดจากการเรียกของ บล็อกข้อมูล อย่างน้อย 1 บล็อกขึ้นไป เพื่อจัดเก็บข้อมูลหรือคำสั่งต่างๆ ลงในอุปกรณ์เก็บรักษาข้อมูล
 - ไดเรกทอรี หรือ โฟลเดอร์ คือ ไฟล์ชนิดหนึ่งที่เก็บหมายเลขอ่อนดที่อยู่ภายใต้โครงสร้างนี้ โดยไดเรกทอรีสามารถซ้อนกันได้ เพื่อความสะดวกในการจัดหมวดหมู่ของไฟล์ข้อมูล
- pipe
- หมายเลขผู้ใช้ (User ID) ซึ่งเป็นเจ้าของ
- หมายเลขกลุ่ม (Group ID) ซึ่งเจ้าของเป็นสมาชิกอยู่
- บิทข้อมูลกำหนดการเข้าถึง (Access control list) เพื่อกำหนดสิทธิ์ของเจ้าของ กลุ่ม และผู้ใช้คนอื่นๆ ทุกคน
- ขนาดไฟล์ที่แท้จริง (หน่วยเป็นไบต์) ...

7.2 ชิพหน่วยความจำแฟลช (Flash Memory Chip)

- หน่วยความจำแฟลช เดิมเรียกว่า แฟลชร้อม (Flash ROM)
- คำว่า ROM ย่อมาจากคำว่า Read-Only Memory
- วัตถุประสงค์เดิมของแฟลชร้อม ทำหน้าที่เป็นหน่วยความจำที่สร้างจากเทคโนโลยีสารภิคตัวนำ โดยมีคุณสมบัติใช้เก็บคำสั่งและข้อมูลเพื่อการอ่านเป็นหลัก
- นักพัฒนาประยุกต์ใช้เก็บคำสั่งประจำเครื่อง หรือ เฟิร์มแวร์ (Firmware) เป็นหลัก
- ด้วยความนิยมในเทคโนโลยีชิโนดี้ หน่วยความจำแฟลชจึงถูกพัฒนาอย่างต่อเนื่องจนสามารถเขียนหรือแก้ไขข้อมูลภายในแฟลชร้อมได้รวดเร็วขึ้นจนกลายเป็นอุปกรณ์เก็บรักษาข้อมูล

7.2 ชิพหน่วยความจำแฟลช (Flash Memory Chip)



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

17

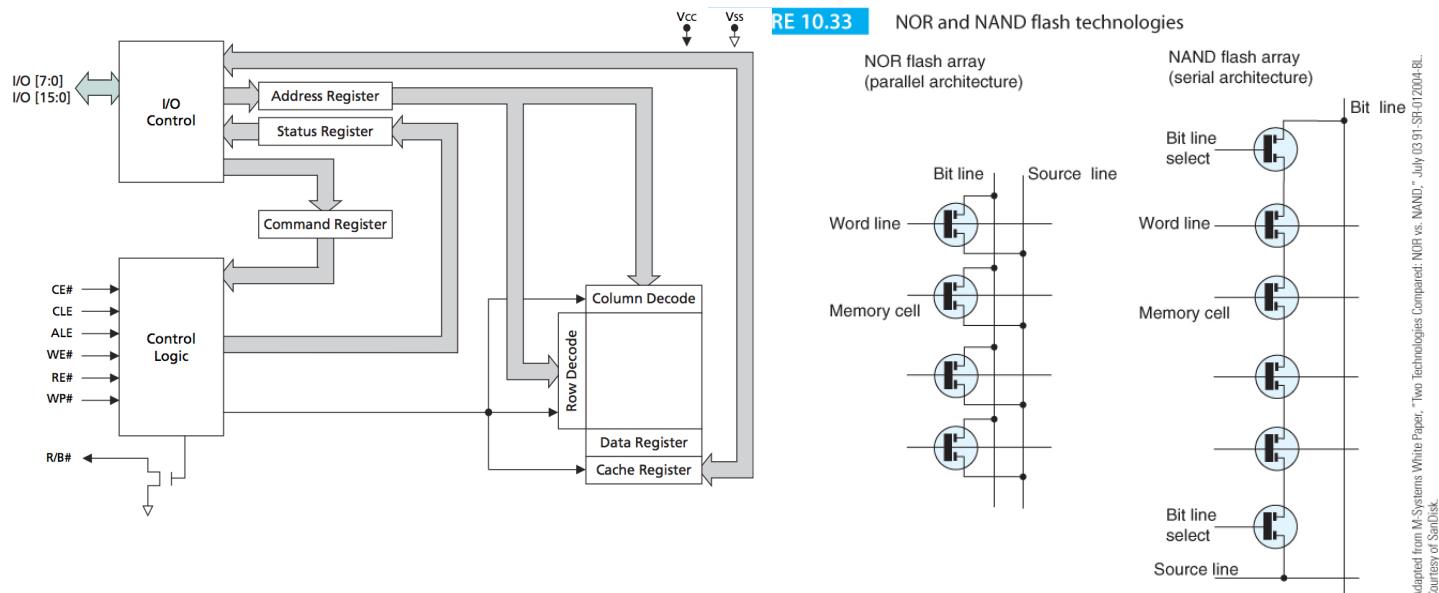
7.2 ชิพหน่วยความจำแฟลช (Flash Memory Chip)

- ประสิทธิภาพการอ่านข้อมูล ขึ้นอยู่กับตำแหน่งและรูปแบบการอ่านและรูปแบบการเรียงตัวของข้อมูล เช่น การอ่านข้อมูลที่เรียงตัว กันอย่างต่อเนื่อง (Sequential Address) จะใช้เวลาอคตอย (Latency) สั้นมากเพียง 30 นาโนวินาที
- การอ่านข้อมูลที่ไม่เรียงตัวต่อเนื่องและสุ่มตำแหน่ง (Random Address) จะใช้เวลาอ่อนานขึ้นเป็น 25 มิลลิวินาที จะเห็นได้ว่าใช้เวลาเพิ่มขึ้นเกือบ 1000 เท่า เทียบกับการอ่านข้อมูลแบบเรียงตัวต่อเนื่อง
- ประสิทธิภาพการเขียนข้อมูล มีลักษณะเช่นเดียวกับการอ่านข้อมูล โดยการเขียนข้อมูลต้องเขียนครั้งละเพจ (Page Program) หรือ 2048 ไบต์ ซึ่งจะใช้เวลาอคตอยนานกว่าการอ่านข้อมูลเป็น 300 มิลลิวินาที ใช้เวลาเพิ่มขึ้นเป็น 10 เท่า เทียบกับการอ่านข้อมูลแบบสุ่ม
- ประสิทธิภาพการลบข้อมูล (Erase) จะใช้เวลาเพิ่มขึ้นสูงมากถึง 2 มิลลิวินาที จะเห็นได้ว่าใช้เวลาเพิ่มขึ้นเกือบ 100 เท่าเทียบกับการเขียนข้อมูลจำนวนหนึ่งเพจ
- อายุการใช้งานของชิพจะนับจากจำนวนครั้งที่เขียนข้อมูล หรือ จำนวนครั้งที่ลบข้อมูล ประมาณ 100,000 ครั้งเท่านั้น แต่ชิพสามารถรักษาข้อมูลได้เป็นระยะเวลานานถึง 10 ปีตามที่แจ้งไว้ในเอกสารคุณสมบัติ

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

18

7.2 ชิพหน่วยความจำแฟลช (Flash Memory Chip)



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

19

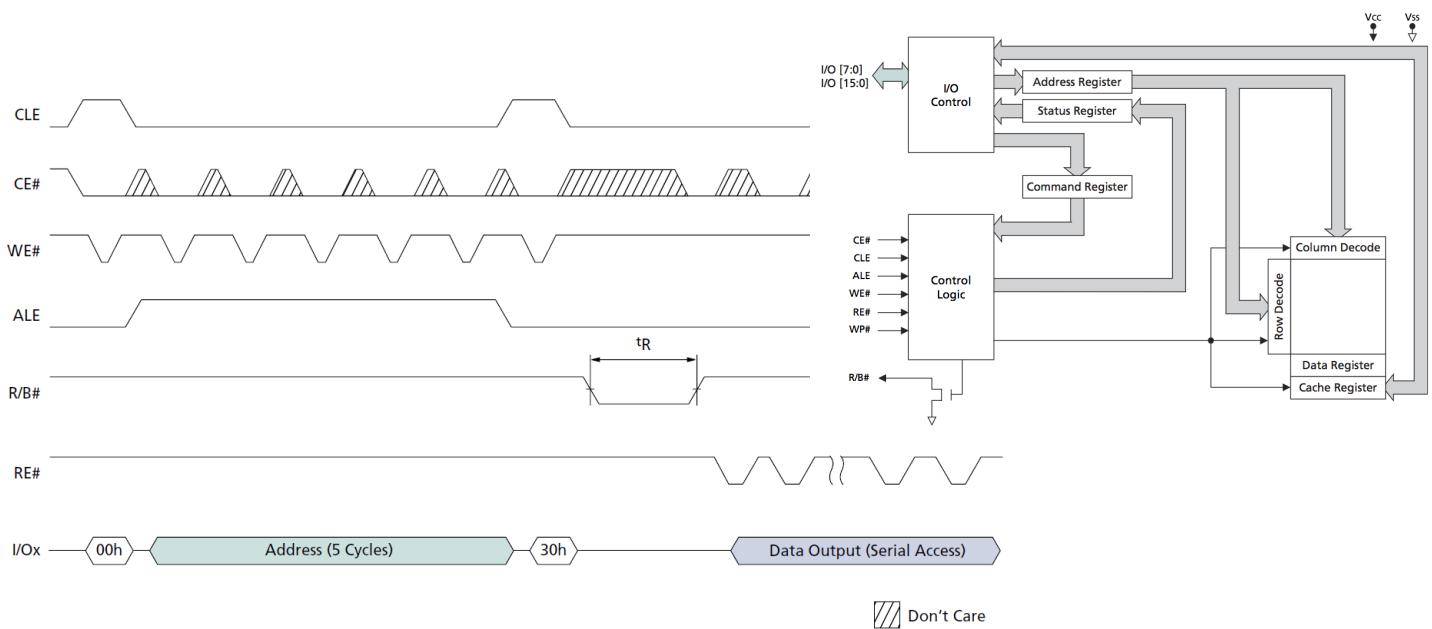
7.2 ชิพหน่วยความจำแฟลช (Flash Memory Chip)

- โครงสร้างภายในชิพแฟลช NAND มีลักษณะคล้ายกับหน่วยความจำ SRAM และ DRAM คือมี อะเรย์ของเซลล์หน่วยความจำเป็นหลัก และ วงจรควบคุมการอ่านหรือเขียนข้อมูลลงไปในอะเรย์นี้ จะเข้าถึงโดยใช้ แอดเดรสແຕວ (เลขແຕວ) และ แอดเดรஸຄອລິມນໍ (เลขຄອລິມນໍ)
- เพื่อป้องชี้ตำแหน่งเซลล์น້ຳ ในอะเรย์ แต่การทำงานของหน่วยความจำแฟลชมีความซับซ้อนกว่า

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

20

7.2 ชิพหน่วยความจำแฟลช (Flash Memory Chip): Read



Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

21

7.2 ชิพหน่วยความจำแฟลช (Flash Memory Chip): Read

- สัญญาณ CLE เปลี่ยนจาก 0 เป็น 1 เพื่ออ่านค่าคำสั่งที่ปรากฏบนบัส I/O
- สัญญาณ CE# เปลี่ยนจาก 1 เป็น 0 เพื่อสั่งชิพให้เริ่มต้นทำงาน
- สัญญาณ WE# จะมีการเปลี่ยนแปลงจาก 1 เป็น 0 สลับไปมา เพื่อเขียนคำสั่งและออดเดรสทางขา I/Ox ไปเก็บพักในรีจิสเตอร์ต่างๆ
- สัญญาณ ALE จะเปลี่ยนแปลงจาก 0 เป็น 1 ในระหว่างที่บัส I/O รับออดเดรสเป็นระยะเวลา 5 คากา (Cycles)
- สัญญาณ R/B# จะเปลี่ยนแปลงจาก 1 เป็น 0 ระยะเวลาหนึ่ง t_R เพื่อบ่งบอกว่าชิพ มีสถานะยุ่ง (Busy) แล้วกลับไปเป็น 1 เพื่อบ่งบอกว่าชิพร้อม

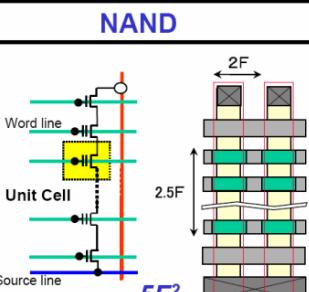
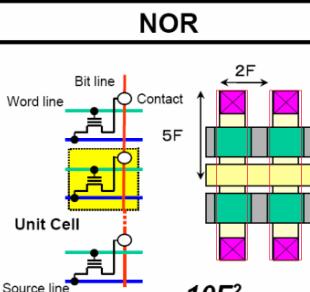
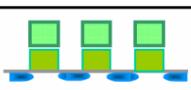
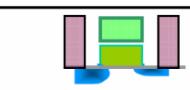
Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

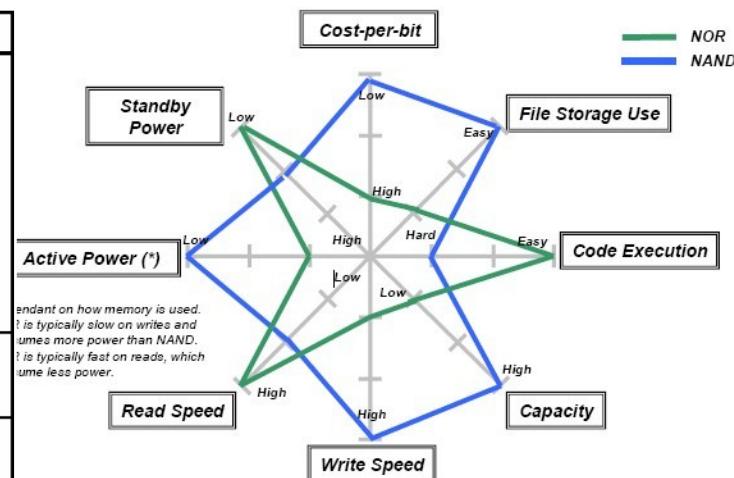
22

7.2 ชิพหน่วยความจำแฟลช (Flash Memory Chip): Read

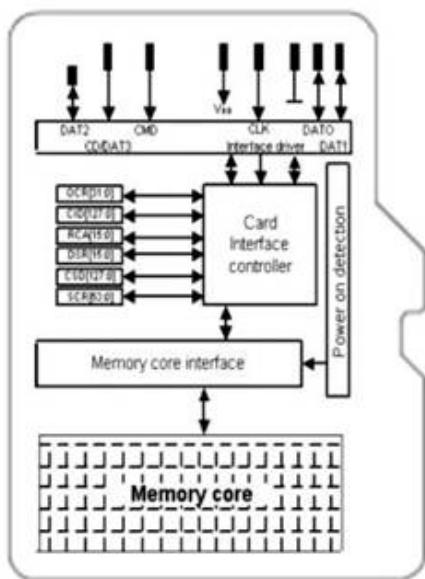
- สัญญาณ RE# จะเปลี่ยนแปลงจาก 1 เป็น 0 สถาปัตย์มา เมื่อตรวจจับว่าสัญญาณ R/B# เปลี่ยนจาก 0 เป็น 1 เพื่ออ่านข้อมูลทางขา I/Ox โดยใช้ขอบขาขึ้นหรือขอบขาลงของสัญญาณ RE#
- สัญญาณ I/O[0:15] ทั้งหมด 16 ขา จะเปลี่ยนแปลงตามลำดับตั้งแต่
 - คำสั่ง 00h จำนวน 1 คatabเวลา - แอ็ดเดรส จำนวน 5 คatabเวลา - คำสั่ง 30h จำนวน 1 คatabเวลา
- ข้อมูล จำนวนหลายคatabเวลาตามขนาดของเพจข้อมูลโดยอ่านข้อมูลเรียงตามลำดับหมายเลขแอ็ดเดรส (Serial Access)

7.2 ชิพหน่วยความจำแฟลช (Flash Memory Chip)

	NAND	NOR
Cell Array & Size	 <p>Word line Unit Cell Source line</p> <p>$5F^2$</p>	 <p>Bit line Word line Unit Cell Source line</p> <p>$10F^2$</p>
Cross-section		
Features	<p>Small Cell Size, High Density Low Power & Good Endurance → Mass Storage</p>	<p>Large Cell Current, Fast Random Access → Code Storage</p>



7.3 การ์ดหน่วยความจำ SD (Secure Digital)

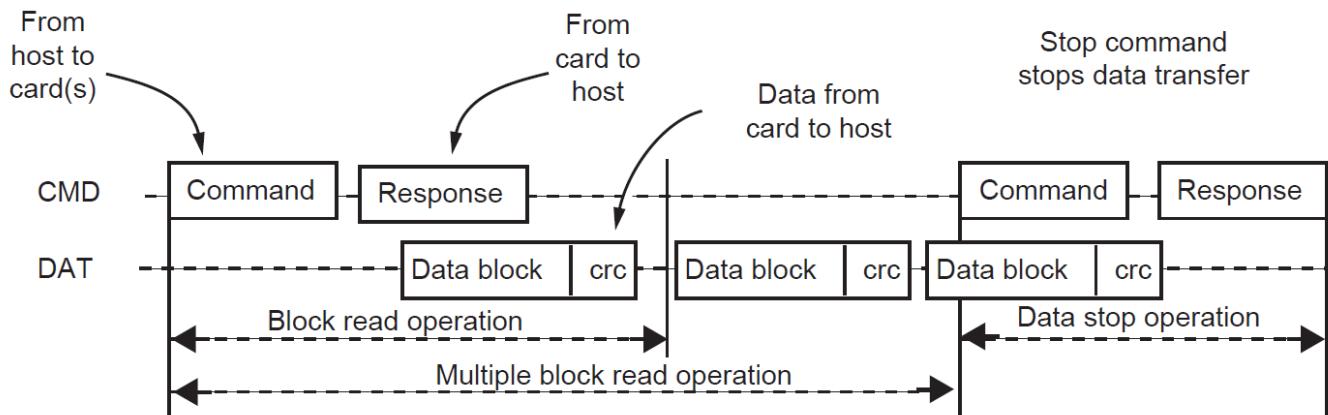


ขา	ชื่อ	วัตถุประสงค์
1	DAT2	Data บิตที่ 2
2	DAT3/CD	Data บิตที่ 3 หรือ Card Detect
3	CMD	ขารับคำสั่ง (Command)
4	VDD	ขั้วบวกแหล่งจ่ายไฟ
5	CLK	สัญญาณคล็อก
6	VSS	ขาร้าวต์แหล่งจ่ายไฟ
7	DAT0	Data บิตที่ 0
8	DAT1	Data บิตที่ 1
		<ul style="list-style-type: none"> • CMD17: Read Single Block • CMD18: Read Multiple Block • CMD24: Write Single Block • CMD25: Write Multiple Block • CMD32: Erase Block Start • CMD33: Erase Block End • CMD38: Erase

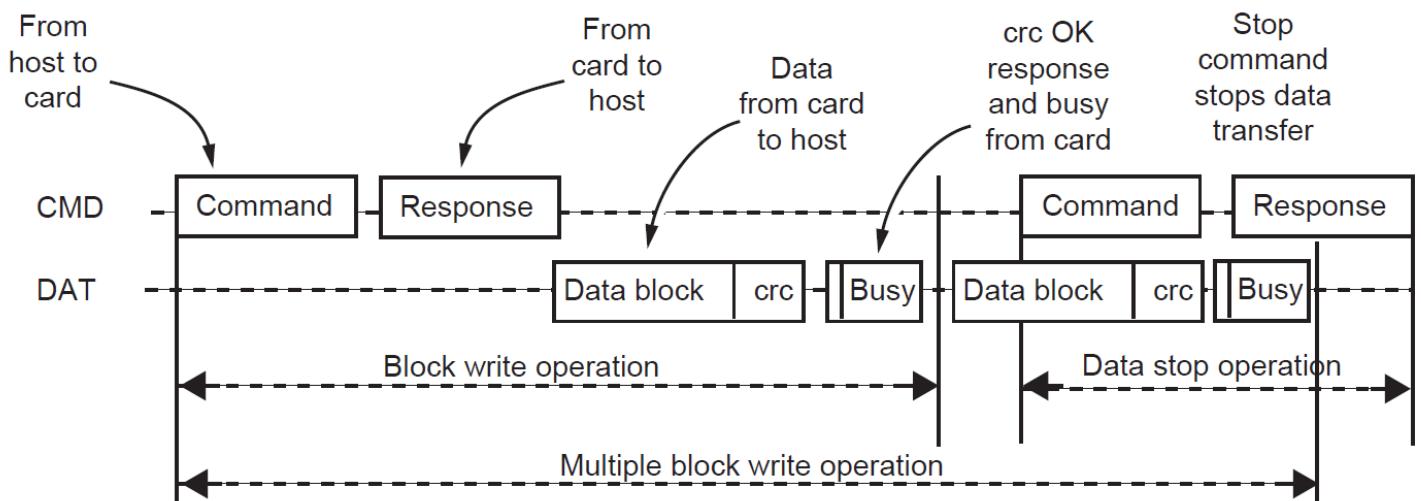
7.3 การ์ดหน่วยความจำ SD (Secure Digital)

- อะเรย์เซลล์หน่วยความจำชนิดแฟลช ตามขนาดของความจำที่ต้องการใช้
- Memory Core Interface เพื่อเชื่อมต่อกับหน่วยความจำ
- วงจรควบคุม Card Interface Controller เพื่อเชื่อมต่อกับโซลิฟ์ ทำงานร่วมกับรีจิสเตอร์ต่างๆ เหล่านี้
 - รีจิสเตอร์ CID[27:0] (Card Identification) ขนาด 28 บิต เพื่อเก็บหมายเลขประจำตัวการ์ด
 - รีจิสเตอร์ OCR[31:0] (Operation Condition Register) ขนาด 32 บิต เพื่อเก็บสถานะการทำงานของการ์ด
 - รีจิสเตอร์ RCA[15:0] (Relative Card Address) ขนาด 32 บิต เพื่อเก็บหมายเลขเดรรี่ของการ์ดซึ่งจะเปลี่ยนแปลงระหว่างที่ถอดเข้าออกจากระบบ
 - รีจิสเตอร์ CSD[27:0] (Card Specific Data) ขนาด 28 บิต เพื่อเก็บสถานะจำเพาะของการ์ด
 - รีจิสเตอร์ SCR[63:0] (SD Configuration Register) ขนาด 64 บิต เพื่อเก็บการตั้งค่าพิเศษประจำตัวการ์ด

7.3 การ์ดหน่วยความจำ SD (Secure Digital): Read



7.3 การ์ดหน่วยความจำ SD (Secure Digital): Write

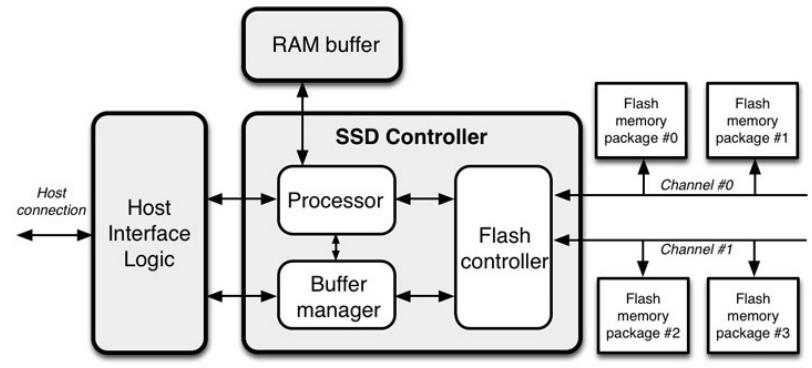
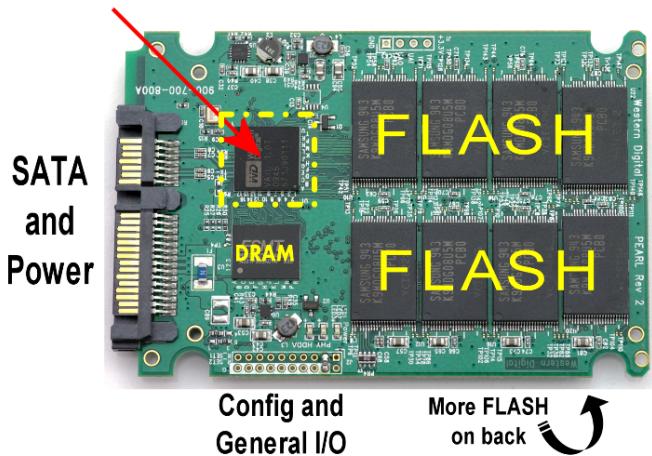


7.4 ໂໜລິດສເຕທດີສກໍ (Solid-State Disk: SSD)

- ຄວາມຈຸຂອງ SSD ມີແນວໂນັມເພີ່ມສູງຂຶ້ນ ທຳໄໝ້ SSD ມີຂັາດເຮີ່ມຕົ້ນຕັ້ງແຕ່ 120-128 ຈິກະໄບທີ່ຂຶ້ນໄປ
- ແນວໂນົມຂອງ SSD ຕັ້ນຖຸນ/ຄວາມຈຸຈະຄຸກລົງເຮື່ອຍໆ ຈຳກັດເຄີຍຫາຣັດດີສຄົມໃນອາຄາຕ ໂດຍອົງປະປະກອບຫລັກທີ່ສຳຄັນ ອື່ນ ມີຄວາມຈຸຕ່ອງຊີບເພີ່ມສູງຂຶ້ນໄປເອີກ
- ໃຊ້ໜ່ວຍຄວາມຈຳ DRAM ເພື່ອທຳນ້າທີ່ເປັນແຄ່ ອີ່ອ ບັຟເຟອ່ວ ເພື່ອເພີ່ມປະສິທິພາກການທໍາງານໃຫ້ຮັດເວົ້ນ
- SSD ໃຊ້ໜ້ອທີ່ໄກລ້າເຄີຍກັບຫາຣັດດີສຄົມໄດ້ ເພື່ອຄວາມໝາຍທີ່ໄກລ້າເຄີຍກັນ
- ຊີບໜ່ວຍຄວາມຈຳແພລ໌ NAND ຄວາມຈຸສູງເຮີ່ມຕົ້ນຕັ້ງຕ່າງໆ ເພື່ອໄດ້ຄວາມຈຸມາກພອ
- ການຈັດເຮີ່ມຕົ້ນຕັ້ງດ້ານບັນ (Channel 0) ແລະ ດ້ານລ່າງ (Channel 1) ຂອງແຜ່ນວົງຈົບປັດຂອງໜ່ວຍຄວາມຈຳແພລ໌ ແລະ ອາສັຍຊີບໜ່ວຍຄວາມຈຳໄດ້ນາມືກແຮມທຳນ້າທີ່ເປັນແຄ່

7.4 ໂໜລິດສເຕທດີສກໍ (Solid-State Disk: SSD)

SSD Controller



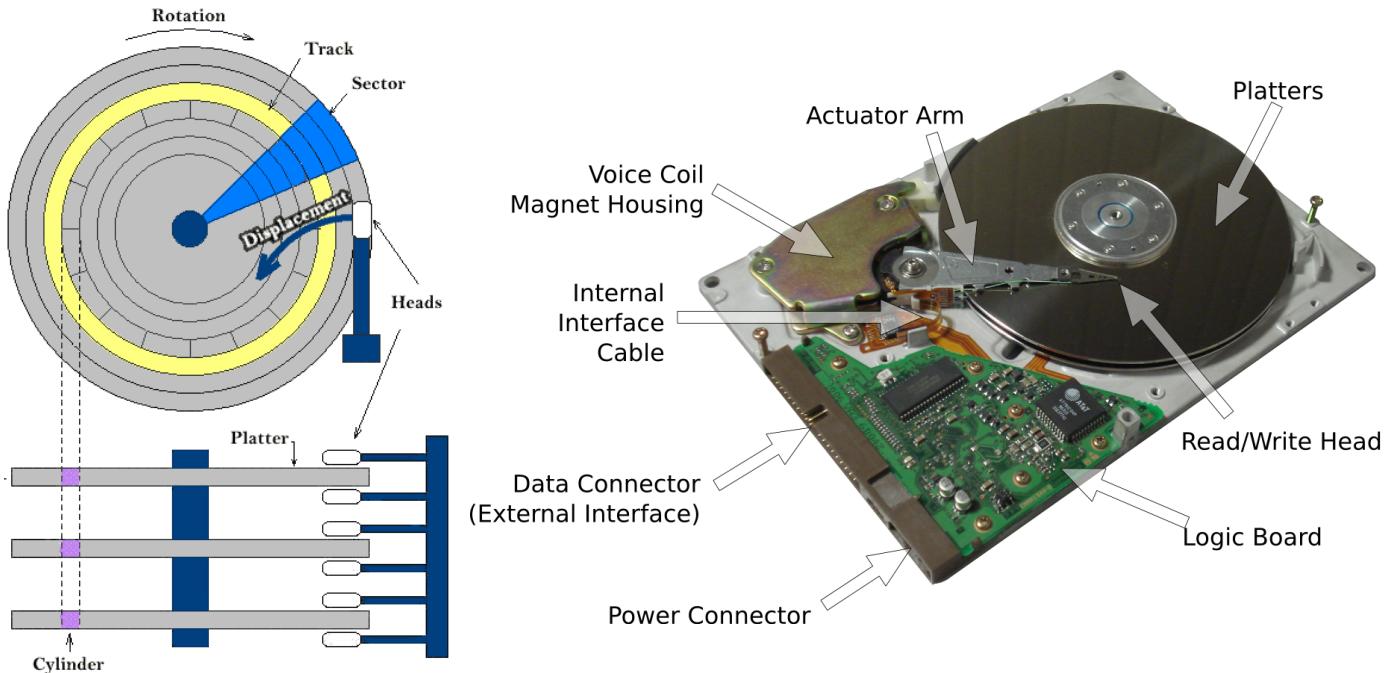
7.5 ฮาร์ดดิสก์ไดร์ฟ (Hard Disk Drive: HDD)

- อุปกรณ์เก็บรักษาข้อมูลที่ได้รับความนิยมจากในอดีต และพัฒนามาเป็นเวลาระยะนาน
- แผ่นจานแม่เหล็กหมุนมีเส้นผ่าศูนย์กลางสองขนาดที่นิยมผลิต คือ 2.5 นิ้ว และ 3.5 นิ้ว หมุนด้วยความเร็วสูงตั้งแต่ 5400 ถึง 10000 รอบต่อนาที
- จุดเด่นของหน่วยความจำชนิดสารแม่เหล็ก คือ ความจุข้อมูลที่มากกว่า เริ่มต้นที่หลายร้อย吉字节 (Tera Byte) โดย 1 เทอร่าไบท์ ประมาณเท่ากับ 1000 กิกะไบท์
- ราคายังคงต่ำลง ต้นทุนโดยรวมจึงถูกลง มีอายุการใช้งานที่ยาวนานพอสมควร

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

31

7.5 ฮาร์ดดิสก์ไดร์ฟ (Hard Disk Drive: HDD)



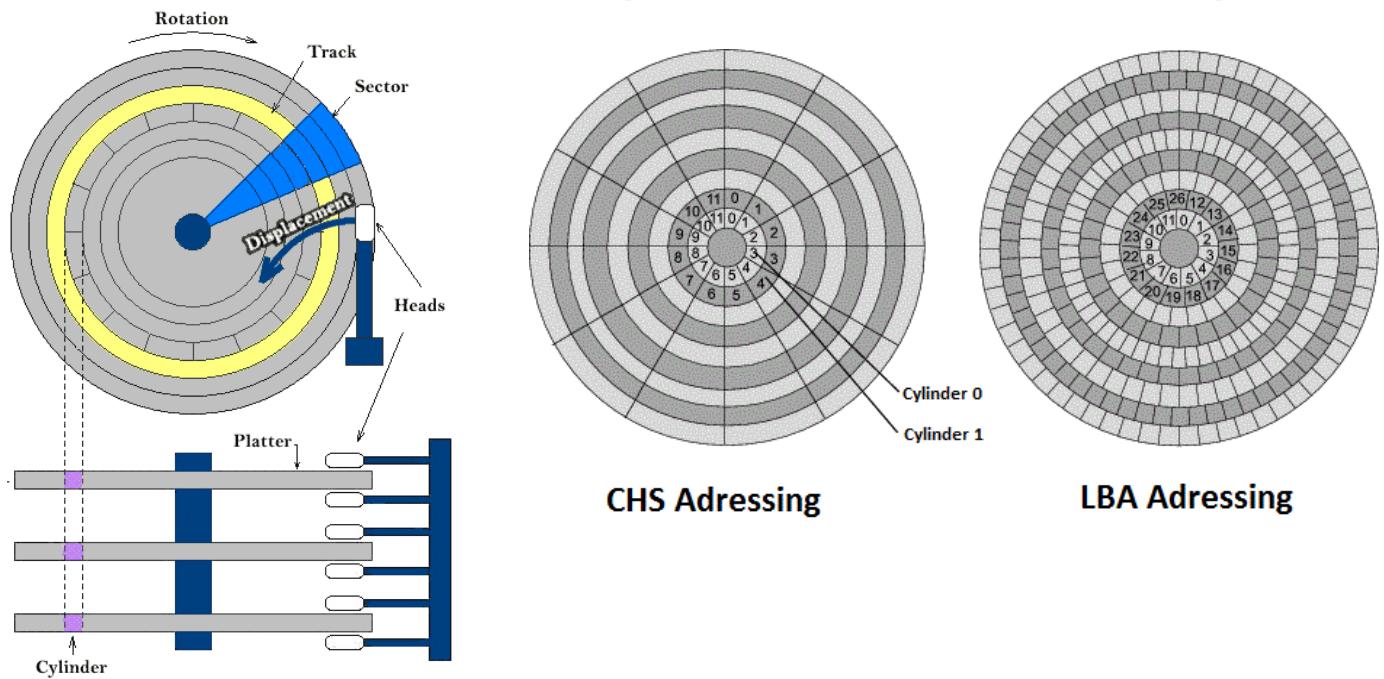
Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

32

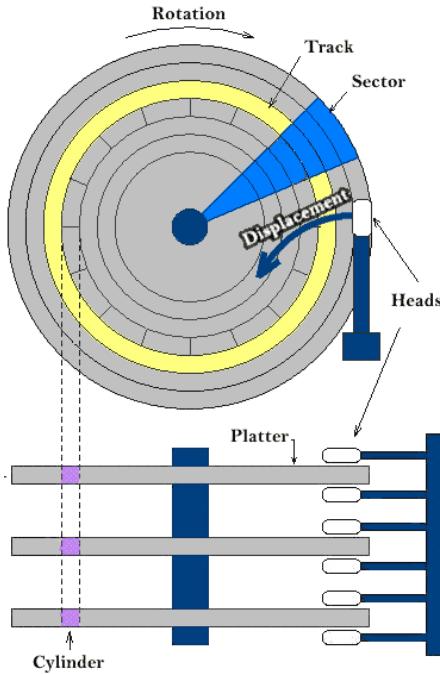
7.5 ฮาร์ดดิสก์ไดร์ฟ (Hard Disk Drive: HDD)

- แผ่นจานแม่เหล็กแบ่งเป็นหลายๆ แทร็ค หรือ วงรอบ แต่ละแทร็คจะมีจำนวนเซ็คเตอร์เท่าๆ กัน
- พื้นที่หนึ่งเซ็คเตอร์ มีความจุ 512 ไบต์เสมอ โดยจะแบ่งพื้นที่ในแนวศูนย์กลางมาบังขอบจาน
- ไซลินเดอร์ (Cylinder) คือ แทร็คที่อยู่บนจานแม่เหล็กต่างๆ และอยู่ห่างจากจุดศูนย์กลางเท่ากัน เรียกว่าเป็นทรงกระบอก โดยจะนับจากไซลินเดอร์หรือแทร็คหมายเลข 0 ซึ่งอยู่ขอบนอกสุดของจานแม่เหล็ก โดยจะนับจากขอบนอกสุดเข้าสู่จุดศูนย์กลาง
- บล็อก หรือ คลัสเตอร์ประกอบด้วย เซ็คเตอร์ที่ต่อเนื่องกัน จำนวน 2^n เซ็คเตอร์เสมอ เช่น 2 4 8 .. เซ็คเตอร์ ในแทร็คเดียวกัน โดยระบบปฏิบัติการจะกำหนดจำนวนเซ็คเตอร์ที่ต้องการ ยกตัวอย่างเช่น คลัสเตอร์ขนาด 4096 ไบต์ต้องการพื้นที่จำนวน 8 เซ็คเตอร์
- หมายเลขบล็อก LBA: Logical Block Addressing คือ การตั้งหมายเลขบล็อกที่กล่าวมาก่อนหน้านี้ให้เรียงตัวตามหมายเลขไซลินเดอร์ หมายเลขหัวอ่าน และหมายเลขแทร็ค ทำให้ง่ายต่อการบริหารจัดการ และเป็นพื้นฐานของระบบบริหารจัดการไฟล์

7.5 ฮาร์ดดิสก์ไดร์ฟ (Hard Disk Drive: HDD)



7.5 ฮาร์ดดิสก์ไดร์ (Hard Disk Drive: HDD)



การวัดประสิทธิภาพของฮาร์ดดิสก์ อาศัยการวัดเวลาการเข้าถึง (T_{acc} , Access Time/Latency) หน่วยเป็น มิลลิวินาที ประกอบด้วย ช่วงเวลาการอ่านเขียนให้ตำแหน่งที่ต้องการอ่านหมุนมาอยู่หัวอ่าน เรียกว่า **Rotation Latency** T_{rotate} ช่วงเวลาการขยับหัวอ่านมาอยู่แทร็คที่ต้องการ T_{head} และช่วงเวลาการถ่ายโอนข้อมูล (T_{trans} , Transfer Time)

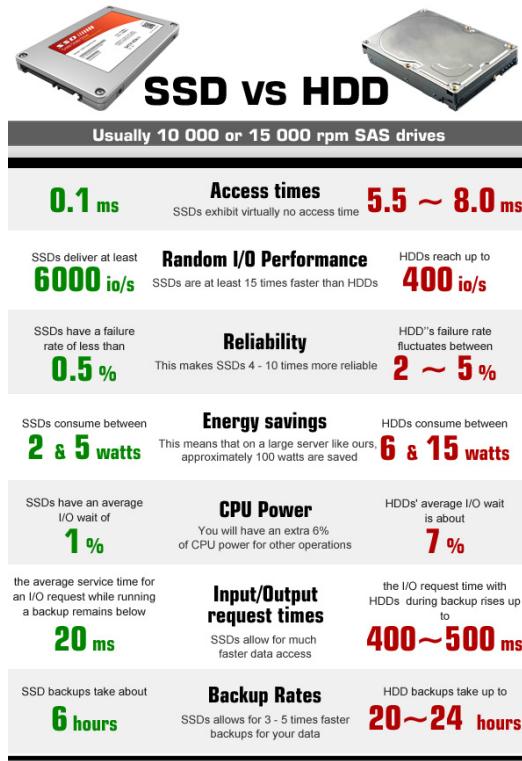
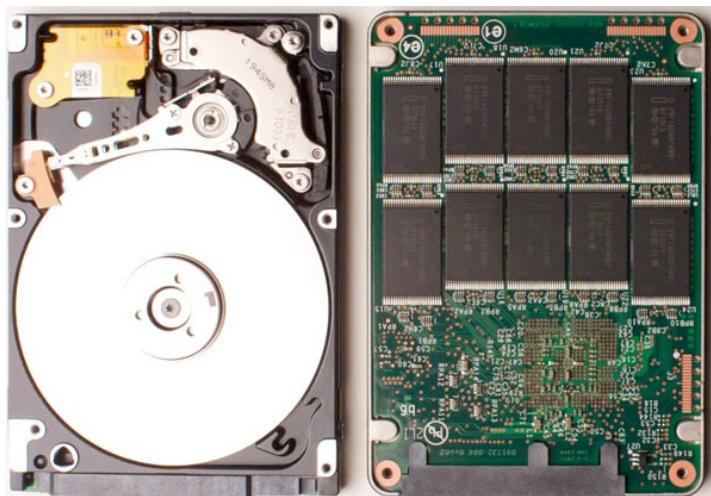
$$T_{acc} = T_{rotate} + T_{head} + T_{transf} \quad (7.1)$$

ดังนั้น เวลาการเข้าถึงจึงมีความสัมพันธ์กับความเร็worobในการหมุนของจานแม่เหล็ก

$$T_{rotate} = \frac{0.5}{V_{rotate}} \quad (7.2)$$

โดยแบ่งผันกับ V_{rotate} หรือ ความเร็worobในการหมุนของจาน หน่วยเป็นรอบต่อวินาที (Round per Minute: RPM) และตำแหน่งของแทร็คที่ต้องขยับหัวอ่านไป ประสิทธิภาพการอ่านและการเขียน จะขึ้นกับตำแหน่งของข้อมูล เช่นเดียวกับการอ่านและเขียนข้อมูลของหน่วยความจำแฟลช ประสิทธิภาพการ

7.5 ฮาร์ดดิสก์ไดร์ (Hard Disk Drive: HDD)



สรุปท้ายบท

อุปกรณ์	แฟลช NAND	SSD	HDD
ผู้ผลิต หมายเลขโมเดล ที่มา: ปี ค.ศ. ความจุ (จิกะไบต์)	Micron MT29F2G08AABWP micron.com 2004 0.25	Micron MTFDDAK120MAV micron.com 2013 120	Western Digital 5K1000 hgst.com 2016 1000
การอ่าน			
- ค่าเวลาเฉลี่ย	30 ns	160 μ s	5.5 ms
- ค่าเวลาสูงสุด	25 μ s	5 ms	-
การเขียน			
- ค่าเวลาเฉลี่ย	300 μ s	40 μ s	5.5 ms
- ค่าเวลาสูงสุด	2 ms	25 ms	-
โวลเตจสูงสุด (โวลท์) กำลังไฟสูงสุด	4.6 23 มิลลิวัตต์	5.0 150 มิลลิวัตต์	5.0 1.6 วัตต์

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

37

สรุปท้ายบท

- ระบบไฟล์และอุปกรณ์เก็บรักษาข้อมูลจะต้องประสานงานกัน เพื่อให้เครื่องเนล ผู้ใช้งานและแอพพลิเคชันอื่นๆ สามารถบริหารจัดการไฟล์โปรแกรม ไฟล์ข้อมูลต่างๆ ได้อย่างถูกต้อง และมีประสิทธิภาพสอดคล้องกับภารกิจของระบบคอมพิวเตอร์
- ขนาดหรือความจุของบล็อกข้อมูลในอุปกรณ์เก็บรักษาข้อมูล เช่น หน่วยความจำแฟลชรีอม จะมีขนาดเท่ากับ 2^{11} หรือ 2048 ไบต์ ฮาร์ดดิสก์ จะมีขนาดเท่ากับ 2^9 หรือ 512 ไบต์ สอดคล้องกับ ขนาดของบล็อกข้อมูลในระบบบริหารจัดการไฟล์ และ
- ขนาดของเพจข้อมูลในหน่วยความจำสมเมื่อน ซึ่งจะมีขนาดเป็นจำนวนเท่าของสองเสมอ และสำหรับลินก์มีขนาดเท่ากับ 4096 หรือ 2^{12} ไบต์
- อุปกรณ์เก็บรักษาข้อมูลจะเชื่อมต่อกับหน่วยความจำผ่านวงจรด้านอินพุท/เอาท์พุท โดยใช้กลไก Direct Memory Access และ กลไกการทำ Interrupt ในชั้น Physical ร่วมกับกลไก Memory Mapped File ในระดับสูงขึ้น

Computer Organization & ARM Assembly Language: RaspberryPi3, Surin K., KMITL

38