

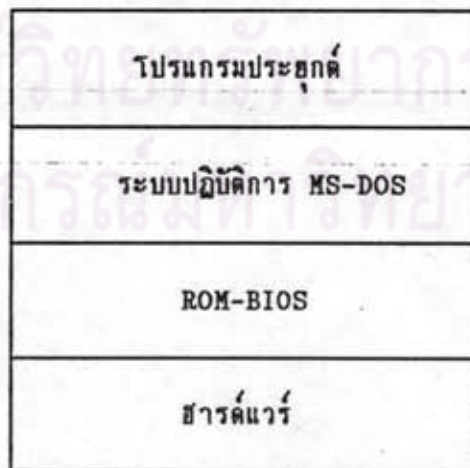


การสร้างบัพเฟอร์แคช

สำหรับเครื่องไมโครคอมพิวเตอร์ไอบีเอ็ม พีซี ใช้ระบบปฏิบัติการ MS-DOS เป็นระบบปฏิบัติการ โปรแกรมควบคุมคิสต์ใน ROM-BIOS หรือ Disk-BIOS ทำหน้าที่ควบคุมการอ่าน/เขียนข้อมูลในคิสต์ให้ระบบปฏิบัติการหรือโปรแกรมประยุกต์อื่น ๆ ที่ต้องการอ่าน/เขียนข้อมูลในคิสต์โดยตรง ร้องขอรับบริการผ่านซอฟต์แวร์อินเตอร์พรีต 13H (ฮัน กูวาร์เรน, ซึยฮงค์ วงศ์ชัยวัฒน์, และไพศาล สงวนหมู่, 2530, 176-184) Disk-BIOS ไม่มีการทำงานแบบบัพเฟอร์แคช เราทดลองสร้างบัพเฟอร์แคชประกอบด้วยโปรแกรมที่ควบคุมบัพเฟอร์แคชที่ใช้วิธีการเขียนข้อมูลแบบ Write-Through พร้อมทั้งจัดพื้นที่หน่วยความจำส่วนหนึ่งเป็นบัพเฟอร์บล็อก เพื่อขยายความสามารถในการโอนย้ายข้อมูลระหว่างคิสต์ซึ่งเป็นหน่วยความจำรองกับหน่วยความจำหลัก

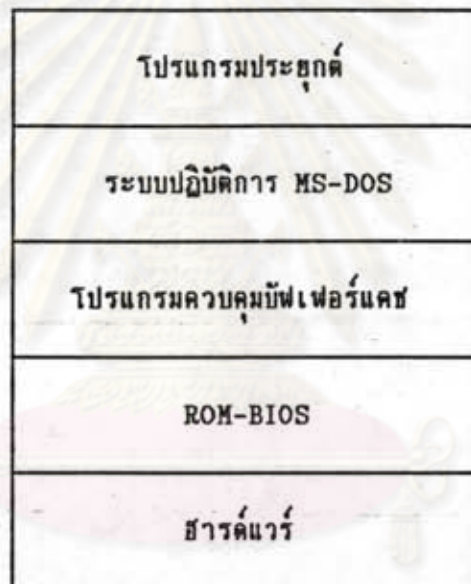
4.1 ลักษณะของโปรแกรมควบคุมบัพเฟอร์แคช

เมื่อโปรแกรมประยุกต์ต้องการอ่าน/เขียนข้อมูลในคิสต์ โปรแกรมประยุกต์จะร้องขอรับบริการดังกล่าวจากระบบปฏิบัติการ ระบบปฏิบัติการจะแปลความต้องการของโปรแกรมประยุกต์มาเป็นตำแหน่งของบล็อกหรือเซกเตอร์ในคิสต์ ความต้องการดังกล่าวจะถูกส่งต่อไปให้ Disk-BIOS ดำเนินการต่อไป ระบบปฏิบัติการ MS-DOS ทำงานร่วมกับ Disk-BIOS ตามลำดับชั้นดังรูปที่ 4.1



รูปที่ 4.1 แสดงลำดับชั้นการทำงานของระบบปกติ

โปรแกรมควบคุมบัพเฟอร์แชนทำงานร่วมกับ Disk-BIOS โดยเข้าไปทำงานอยู่ระหว่าง Disk-BIOS กับระบบปฏิบัติการ MS-DOS ระบบปฏิบัติการ MS-DOS ร้องขอรับบริการอ่าน/เขียนบล็อกข้อมูลกับโปรแกรมควบคุมบัพเฟอร์แชนแทน Disk-BIOS และเมื่อโปรแกรมควบคุมบัพเฟอร์แชนจำเป็นต้องอ่าน/เขียนบล็อกข้อมูลในดิสค์จริงๆ โปรแกรมควบคุมบัพเฟอร์แชนจะร้องขอรับบริการอ่านเขียนบล็อกข้อมูลกับ Disk-BIOS อีกทีหนึ่ง การเพิ่มโปรแกรมควบคุมบัพเฟอร์แชนทำได้โดยการย้ายอินเตอร์พดท์เวกเตอร์ 13H จาก Disk-BIOS ไปที่โปรแกรมควบคุมบัพเฟอร์แชน ระบบปฏิบัติการ MS-DOS, โปรแกรมควบคุมบัพเฟอร์แชน, และ Disk-BIOS ทำงานร่วมกันเป็นลำดับขั้นดังรูปที่ 4.2

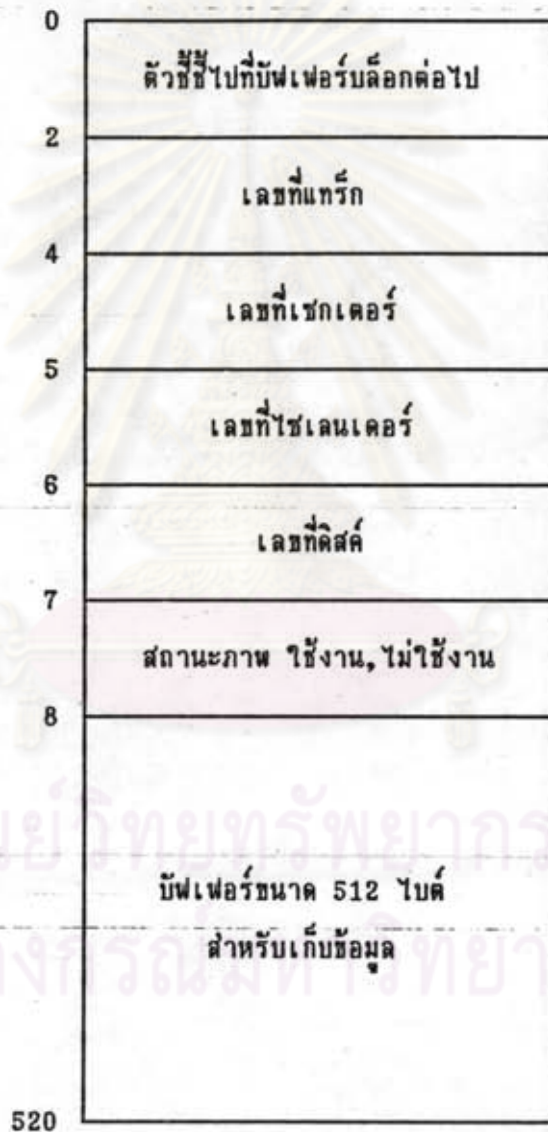


รูปที่ 4.2 แสดงลำดับขั้นการทำงานของระบบปกติที่เพิ่มบัพเฟอร์แชน

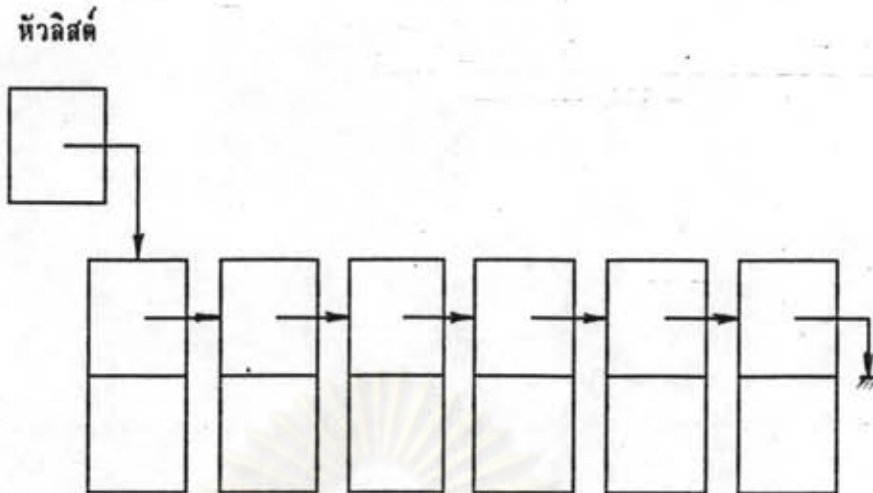
บัพเฟอร์แชนจะมีบัพเฟอร์บล็อกขนาด 520 ไบต์ บัพเฟอร์บล็อกแต่ละบล็อกมีโครงสร้างข้อมูลแบ่งออกเป็น 2 ส่วนคือมีบัพเฟอร์เฮดขนาด 8 ไบต์ไว้จัดการการใช้บัพเฟอร์และบัพเฟอร์ไว้เก็บข้อมูลขนาด 512 ไบต์ บัพเฟอร์บล็อกมีโครงสร้างดังรูปที่ 4.3

บัพเฟอร์บล็อกทั้งหมด จะมาเชื่อมโงกันเป็นลิสต์เชื่อมโง (Linked List) จำนวน 2 ชุด ลิสต์เชื่อมโงชุดหนึ่งประกอบด้วยบัพเฟอร์บล็อกที่ใช้เก็บข้อมูล อีกชุดหนึ่งประกอบด้วยบัพเฟอร์บล็อกที่ไม่ใช้เก็บข้อมูล (ลิสต์เชื่อมโงชุดหลังนี้จะเรียกว่า Storage Pools)

การควบคุมการใช้บัฟเฟอร์บล็อกร ของโปรแกรมควบคุมบัฟเฟอร์แชนในการเก็บข้อมูลชุดใหม่เป็นการคัดเลือกแบบ LRU(Least Recently Used) คือโปรแกรมควบคุมบัฟเฟอร์แชนจะเลือกบัฟเฟอร์บล็อกรที่เก็บข้อมูลที่ไม่ได้ใช้มานานที่สุด มาเก็บข้อมูลชุดใหม่(Wilkinson, 1991, 45-47) โปรแกรมควบคุมบัฟเฟอร์แชนสามารถเลือกบัฟเฟอร์บล็อกรแบบ LRU ได้ด้วยโครงสร้างข้อมูลแบบลิสต์เชื่อมโยงช่วยในการคัดเลือก



รูปที่ 4.3 แสดงโครงสร้างของบัฟเฟอร์บล็อกร

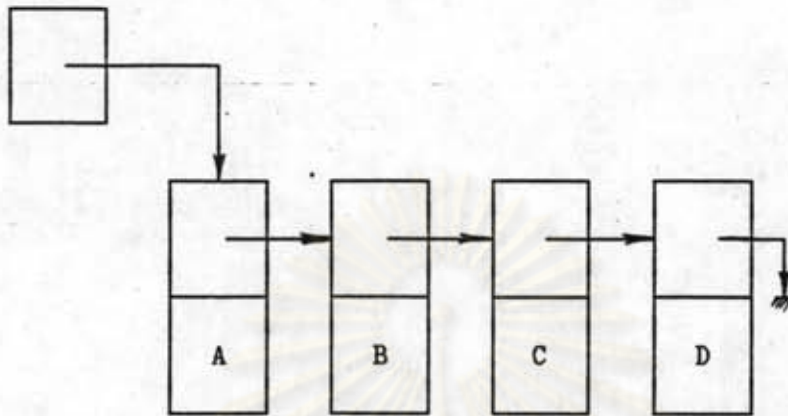


รูปที่ 4.4 แสดงโครงสร้างของบัฟเฟอร์บล็อกจัดเป็นลิสต์เชื่อมโยง

การทำงานของโปรแกรมควบคุมบัฟเฟอร์แชนแนลเมื่อระบบปฏิบัติการ MS-DOS ต้องการอ่านข้อมูล โปรแกรมควบคุมบัฟเฟอร์แชนแนลจะค้นหาข้อมูลในบัฟเฟอร์บล็อกในลิสต์เชื่อมโยง ถ้าพบข้อมูล โปรแกรมควบคุมบัฟเฟอร์แชนแนลจะย้ายบัฟเฟอร์บล็อกที่มีข้อมูลจากระบบปฏิบัติการ MS-DOS ต้องการมาที่หัวลิสต์เชื่อมโยง แต่ถ้าไม่พบข้อมูลที่ต้องการ โปรแกรมควบคุมบัฟเฟอร์แชนแนลจะขอบัฟเฟอร์บล็อกใหม่จาก Storage Pools มาที่หัวลิสต์เชื่อมโยงแทน ในกรณีที่ไม่มีบัฟเฟอร์บล็อกเหลือใน Storage Pools โปรแกรมควบคุมบัฟเฟอร์แชนแนลจะคืนบัฟเฟอร์บล็อกที่ทางลิสต์เชื่อมโยงซึ่งเป็นบัฟเฟอร์บล็อกที่ไม่ได้ถูกเรียกใช้มานานที่สุดให้ Storage Pools ก่อน หลังจากที่ได้บัฟเฟอร์บล็อกใหม่มาไว้ที่หัวลิสต์ โปรแกรมควบคุมบัฟเฟอร์แชนแนลจะร้องขอรับบริการจาก Disk-BIOS ให้อ่านข้อมูลบล็อกที่ต้องการในลิสต์บรรจลงในบัฟเฟอร์บล็อก และเมื่อได้บัฟเฟอร์บล็อกที่มีข้อมูลจากระบบปฏิบัติการ MS-DOS ต้องการที่หัวลิสต์เชื่อมโยง โปรแกรมควบคุมบัฟเฟอร์แชนแนลจะโอนย้ายข้อมูลในบัฟเฟอร์บล็อกลงหน่วยความจำจากระบบปฏิบัติการ MS-DOS กำหนดให้

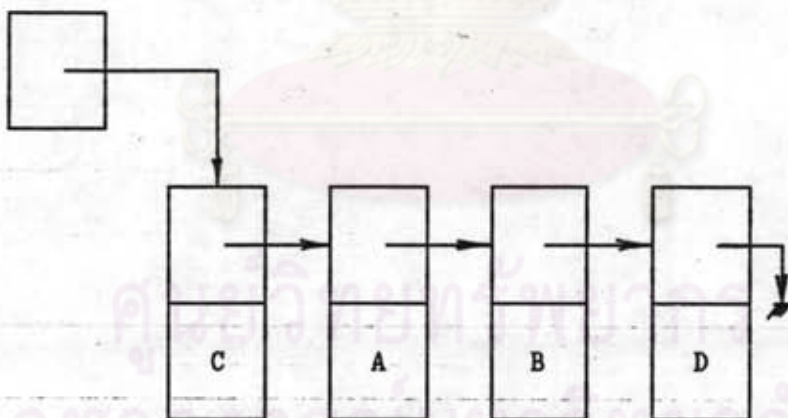
เมื่อระบบปฏิบัติการ MS-DOS ต้องการเขียนข้อมูล โปรแกรมควบคุมบัฟเฟอร์แชนแนลจะค้นหาบัฟเฟอร์บล็อกที่เก็บข้อมูลจากระบบปฏิบัติการ MS-DOS ต้องการเขียนเข้าไปใหม่ในลิสต์เชื่อมโยงและ ย้ายบัฟเฟอร์บล็อกมาที่หัวลิสต์เชื่อมโยงเช่นเดียวกับการค้นหาบล็อกข้อมูลเมื่อบัฟเฟอร์แชนแนลให้บริการระบบปฏิบัติการ MS-DOS ในการอ่านข้อมูล หลังจากย้ายบัฟเฟอร์บล็อกมาที่หัวลิสต์เชื่อมโยงแล้ว โปรแกรมควบคุมบัฟเฟอร์แชนแนลจะโอนย้ายข้อมูลจากหน่วยความจำลงในบัฟเฟอร์บล็อกและร้องขอรับบริการเขียนข้อมูลในบัฟเฟอร์บล็อกจาก Disk-BIOS

หัวลิสต์



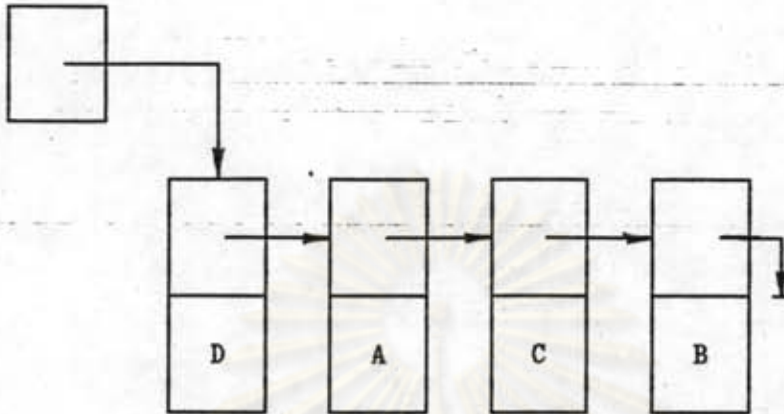
รูปที่ 4.5 บัฟเฟอร์แคชมีบัฟเฟอร์บล็อกจำนวน 4 บล็อก

หัวลิสต์



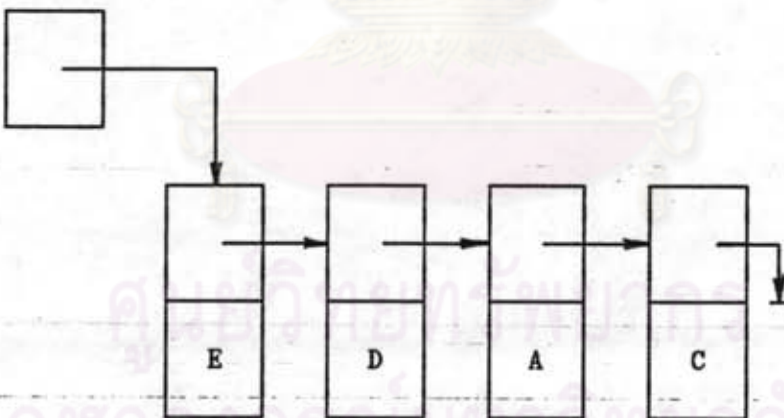
รูปที่ 4.6 เมื่อมีการเรียกข้อมูลบล็อก C
บล็อก C จะถูกย้ายมาที่หัวลิสต์

หัวลิ้น



รูปที่ 4.7 เมื่อมีการเรียกข้อมูลบล็อก A และบล็อก D ตามลำดับ
บล็อก A และบล็อก D จะถูกย้ายมาที่หัวลิ้นตามลำดับ

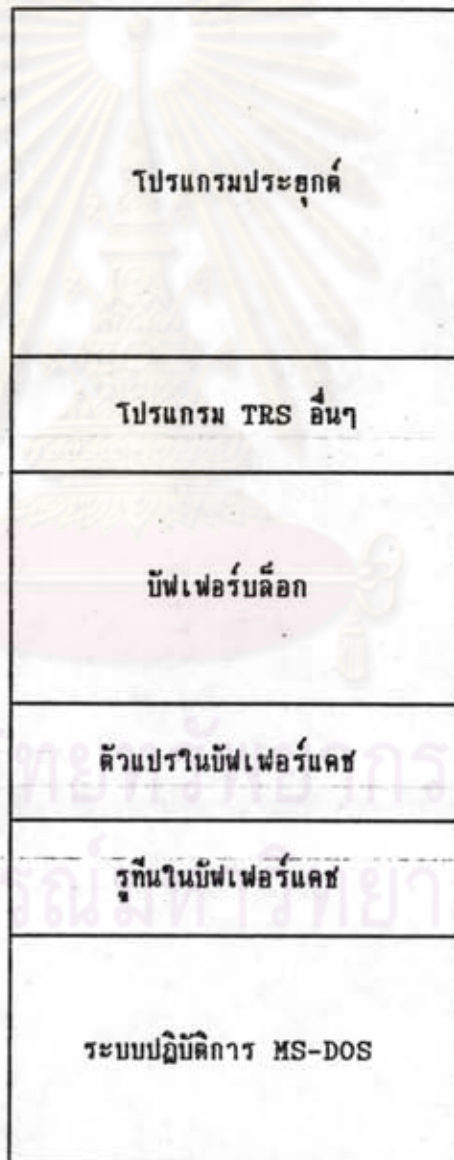
หัวลิ้น



รูปที่ 4.8 เมื่อมีการเรียกข้อมูลบล็อก E
บล็อก B เป็นบล็อกที่ไม่ได้ถูกเรียกนานที่สุดจะถูกย้ายมาที่หัวลิ้น
และโอนข้อมูลบล็อก E จากคิส์มาที่ข้อมูลบล็อก B

เนื่องจากดิสก์แผ่น (Floppy Disk) เป็นดิสก์ที่สามารถเปลี่ยนแผ่นดิสก์ได้ การที่เก็บข้อมูลจากดิสก์แผ่นไว้ในบัพเฟอร์ อาจมีผลให้ข้อมูลที่มืออยู่ในบัพเฟอร์บล็อกกับข้อมูลที่มีอยู่จริงในดิสก์ไม่ตรงกันได้เนื่องจากการเปลี่ยนแผ่นดิสก์ โปรแกรมควบคุมบัพเฟอร์แคชได้แก้ไขปัญหาดังกล่าวโดยการจับเวลาหลังการร้องขอรับบริการจาก DOS ครั้งสุดท้าย ถ้านานเกินกว่าที่กำหนดไว้โปรแกรมควบคุมบัพเฟอร์แคชจะทิ้งข้อมูลจากดิสก์แผ่นที่มีอยู่ในบัพเฟอร์บล็อกทั้งหมด เพราะอาจมีการเปลี่ยนแผ่นดิสก์ในช่วงเวลาดังกล่าวได้

9FFFF



00000

รูปที่ 4.9 แสดงการใช้พื้นที่หน่วยความจำเมื่อติดตั้งบัพเฟอร์แคช

โปรแกรมควบคุมบู๊ตเฟิร์มแวร์ที่สร้างขึ้นมา เป็นโปรแกรมประเภท Terminate and Stay Resident เมื่อมีการติดตั้งโปรแกรมควบคุมบู๊ตเฟิร์มแวร์เข้ากับระบบปฏิบัติการ MS-DOS โปรแกรมควบคุมบู๊ตเฟิร์มแวร์จะเข้าแทนที่ Disk-BIOS ได้โดยเก็บอินเทอร์พรีตเตอร์ 13H ไว้ และย้ายอินเทอร์พรีตเตอร์ 13H ที่โปรแกรมควบคุมบู๊ตเฟิร์มแวร์ เมื่อระบบปฏิบัติการ MS-DOS ต้องการอ่าน/เขียนข้อมูลในดิสก์ ระบบปฏิบัติการ MS-DOS จะติดต่อกับโปรแกรมควบคุมบู๊ตเฟิร์มแวร์โดยผ่านซอฟต์แวร์อินเทอร์พรีตเตอร์ 13H

4.2 รูทีนต่างๆในโปรแกรมควบคุมบู๊ตเฟิร์มแวร์

โปรแกรมควบคุมบู๊ตเฟิร์มแวร์ที่สร้างขึ้นเพื่อทดสอบประกอบด้วยรูทีนต่างๆทำงานควบคุมการใช้บู๊ตเฟิร์มแวร์บล็อก รูทีนต่างๆได้แก่

1. รูทีน _Initialize
2. รูทีน _Cache
3. รูทีน _Read_Function
4. รูทีน _Write_Function
5. รูทีน _To_BIOS
6. รูทีน _To_DOS
7. รูทีน _Time_Routine

รูทีน _Initialize เป็นรูทีนที่ทำหน้าที่ติดตั้งโปรแกรมควบคุมบู๊ตเฟิร์มแวร์ และจัดพื้นที่หน่วยความจำเพื่อใช้เป็นบู๊ตเฟิร์มแวร์บล็อก รูทีน _Initialize จะจัดเตรียมบู๊ตเฟิร์มแวร์บล็อกให้จำนวน 32 บล็อก หรือตามที่กำหนดไว้ในพารามิเตอร์คำสั่งงาน(Command Line Parameter) รูทีน _Initialize เริ่มทำงานด้วยการตั้งค่าเริ่มต้น ตรวจสอบว่ามีตัวอักษรในพารามิเตอร์คำสั่งหรือไม่ ถ้าไม่มีรูทีน _Initialize จะข้ามไปทำการติดตั้งโปรแกรมควบคุมบู๊ตเฟิร์มแวร์ต่อไป

ถ้ามีการกำหนดจำนวนบู๊ตเฟิร์มแวร์บล็อกไว้ในพารามิเตอร์คำสั่ง รูทีน _Initialize จะนำตัวอักษรในพารามิเตอร์คำสั่งมาตรวจสอบหาตัวอักษร "/" ซึ่งเป็นตัวอักษรแรกของพารามิเตอร์

เมื่อพบตัวอักษร"/" รูทีน `_Initialize` จะตรวจสอบว่าตัวอักษรต่อไปเป็นตัวอักษร"B"หรือตัวอักษร"R"ถ้าเป็นตัวอักษร"B"แสดงว่าเป็นการกำหนดจำนวนบัฟเฟอร์บล็อกไว้ในพารามิเตอร์คำสั่ง จากตัวอักษร"B"จะเป็นตัวอักษร"="และตัวเลขบอกจำนวนบัฟเฟอร์ รูทีน `_Initialize` จะแปลงตัวเลขบอกจำนวนบัฟเฟอร์ให้เป็นค่าจำนวนบัฟเฟอร์เพื่อใช้ในการจัดเตรียมบัฟเฟอร์บล็อกต่อไป ถ้าตัวอักษรที่ตามมาเป็นตัวอักษร"R"แสดงว่าเป็นการกำหนดให้ให้รูทีน `_Initialize` ดำเนินการปลดโปรแกรมควบคุมบัฟเฟอร์แชนด์ที่ติดตั้งไว้ก่อนแล้วออกจากระบบปฏิบัติการ

หลังจากรูทีน `_Initialize` รับค่าพารามิเตอร์คำสั่งแล้ว รูทีน `_Initialize` จะดำเนินการติดตั้งโปรแกรมควบคุมบัฟเฟอร์แชนด์ต่อไป โดยรูทีน `_Initialize` จะตรวจสอบว่าการติดตั้งโปรแกรมควบคุมบัฟเฟอร์แชนด์ไว้ก่อนแล้วหรือไม่ ต่อจากนั้นรูทีน `_Initialize` จะขอพารามิเตอร์ต่างๆเพื่อใช้ควบคุมฮาร์ดดิสก์พารามิเตอร์เหล่านี้ได้แก่ จำนวนไซลินเดอร์, จำนวนแทร์ก, และจำนวนเซกเตอร์ในหนึ่งแทร์ก ต่อมารูทีน `_Initialize` ขออินเตอร์พดเวกเตอร์ที่ 13H และอินเตอร์พดเวกเตอร์ที่ 1CH เปลี่ยนค่าอินเตอร์พดเวกเตอร์ที่ 13H ให้ชี้มาที่รูทีน `_Cache` และอินเตอร์พดเวกเตอร์ที่ 1CH ให้ชี้มาที่รูทีน `_Time_Routine` จัดเตรียมพื้นที่หน่วยความจำเพื่อใช้เป็น บัฟเฟอร์บล็อก และสิ้นสุดการทำงานโดยคงโปรแกรมบัฟเฟอร์แชนด์ไว้ในหน่วยความจำ

ในการปลดโปรแกรมควบคุมบัฟเฟอร์แชนด์ที่ติดตั้งไว้ก่อนแล้วออกจาก Disk-BIOS รูทีน `_Initialize` จะตรวจสอบว่าการติดตั้งโปรแกรมควบคุมบัฟเฟอร์แชนด์ไว้ก่อนแล้วหรือไม่ ถ้ามีการติดตั้งโปรแกรมควบคุมบัฟเฟอร์แชนด์ไว้ก่อนแล้ว รูทีน `_Initialize` จะแจ้งโปรแกรมควบคุมบัฟเฟอร์แชนด์ที่ติดตั้งไว้ก่อนให้คืนอินเตอร์พดเวกเตอร์ที่ 13H และอินเตอร์พดเวกเตอร์ที่ 1CH และให้โปรแกรมควบคุมบัฟเฟอร์แชนด์ส่งค่า PSP Segment คืนให้รูทีน `_Initialize` ใช้คืนพื้นที่หน่วยความจำที่เป็นบัฟเฟอร์บล็อก และสิ้นสุดการทำงาน

รูทีน `_Cache` เป็นรูทีนที่รูทีน `_Initialize` ติดตั้งกับ Disk-BIOS โดยการเปลี่ยนอินเตอร์พดเวกเตอร์ที่ 13H ให้ชี้มาที่รูทีน `_Cache` แทน Disk-BIOS ดังนั้น ระบบปฏิบัติการ MS-DOS จะติดต่อกับรูทีน `_Cache` ด้วยซอฟต์แวร์อินเตอร์พดคำสั่ง INT 13H แทน Disk-BIOS เมื่อระบบปฏิบัติการ MS-DOS ร้องขอรับบริการอ่าน/เขียนข้อมูล รูทีน `_Cache` จะเก็บค่าพารา



มีเคอร์ที่ระบบปฏิบัติการ MS-DOS ส่งผ่านวีจีส์เตอร์มาให้, ตรวจสอบความต้องการขอรับบริการของระบบปฏิบัติการ และโอนการทำงานให้รูทีนอื่นๆดำเนินการต่อตามความต้องการขอรับบริการของระบบปฏิบัติการต่อไป

รูทีน `_Read_Function` เป็นรูทีนที่ดำเนินการต่อจากรูทีน `_Cache` เมื่อระบบปฏิบัติการ MS-DOS ร้องขอรับบริการอ่านข้อมูล รูทีน `_Read_Function` จะค้นหาข้อมูลในระบบปฏิบัติการ MS-DOS ต้องการในบัฟเฟอร์บล็อกตามลิสต์ เมื่อรูทีน `_Read_Function` ค้นพบบัฟเฟอร์บล็อกที่เก็บข้อมูลในระบบปฏิบัติการ MS-DOS ต้องการ รูทีน `_Read_Function` จะย้ายบัฟเฟอร์บล็อกนั้นมาที่หัวลิสต์ แต่ถ้าไม่พบบัฟเฟอร์บล็อกที่เก็บข้อมูลในระบบปฏิบัติการ MS-DOS ต้องการในลิสต์ รูทีน `_Read_Function` จะตรวจสอบว่าใน Storage Pools มีบัฟเฟอร์บล็อกเหลืออยู่หรือไม่ ถ้ามีบัฟเฟอร์บล็อกเหลืออยู่ รูทีน `_Read_Function` จะขอบัฟเฟอร์บล็อกจาก Storage Pools มาไว้ที่ส่วนหัวของลิสต์เพื่อเก็บข้อมูล ในกรณีที่ไม่มีบัฟเฟอร์บล็อกเหลืออยู่ใน Storage Pools รูทีน `_Read_Function` จะคืนบัฟเฟอร์บล็อกที่หางลิสต์ให้ Storage Pools ก่อนขอบัฟเฟอร์บล็อกจาก Storage Pools ไปใช้ หลังจากรูทีน `_Read_Function` ขอบัฟเฟอร์บล็อกจาก Storage Pools รูทีน `_Read_Function` จะติดต่อกับ Disk-BIOS ให้อ่านข้อมูลในระบบปฏิบัติการ MS-DOS ต้องการไว้ในบัฟเฟอร์บล็อกนั้น

ต่อจากการค้นหาข้อมูลในลิสต์ รูทีน `_Read_Function` จะย้ายข้อมูลที่มีอยู่ในบัฟเฟอร์บล็อกที่หัวลิสต์ลงหน่วยความจำที่ระบบปฏิบัติการ MS-DOS กำหนดให้ ถ้าการโอนย้ายข้อมูลยังไม่ครบตามจำนวนเซกเตอร์ที่ระบบปฏิบัติการ MS-DOS ต้องการ รูทีน `_Read_Function` จะคำนวณหาไซลินเดอร์, แทร็ก, และเซกเตอร์ของเซกเตอร์ต่อไป จากค่าพารามิเตอร์ที่ได้ขอไว้จาก Disk-BIOS และรูทีน `_Read_Function` จะวนกลับไปดำเนินการค้นหาข้อมูลเซกเตอร์ต่อไปจนครบตามที่ระบบปฏิบัติการ MS-DOS ต้องการ

รูทีน `_Write_Function` เป็นรูทีนที่ต้องดำเนินการต่อจากรูทีน `_Cache` เมื่อระบบปฏิบัติการ MS-DOS ต้องการเขียนข้อมูล รูทีน `_Write_Function` จะดำเนินการค้นหาบัฟเฟอร์บล็อกที่เก็บข้อมูลในระบบปฏิบัติการ MS-DOS ต้องการเขียนใหม่ในลิสต์ เมื่อพบบัฟเฟอร์บล็อกที่เก็บข้อมูลในระบบปฏิบัติการ MS-DOS ต้องการเขียนข้อมูลใหม่ รูทีน `_Write_Function` จะย้ายบัฟ

เพอร์บล็อคนั้นมาที่หัวลิสต์ แต่ถ้าไม่พบ รูทีน `_Write_Function` จะตรวจสอบว่ามีบัพเพอร์บล็อกว่างเหลือใน Storage Pools หรือไม่ ถ้าไม่มีบัพเพอร์บล็อกว่างเหลืออยู่ใน Storage pools รูทีน `_Write_Function` จะคืนบัพเพอร์บล็อกที่หางลิสต์ให้ Storage Pools ก่อน และขอบัพเพอร์บล็อกจาก Storage Pools มาไว้ที่หัวลิสต์

ต่อจากการค้นหาข้อมูลในลิสต์ รูทีน `_Write_Function` ย้ายข้อมูลในหน่วยความจำที่ระบบปฏิบัติการ MS-DOS กำหนดให้ไว้ในบัพเพอร์บล็อกที่หัวลิสต์ และติดต่อกับ Disk-BIOS ให้เขียนข้อมูลในบัพเพอร์บล็อกนั้นลงดิสก์ ถ้าการโอนย้ายและเขียนข้อมูลยังไม่ครบตามจำนวนเซกเตอร์ที่ระบบปฏิบัติการ MS-DOS ต้องการ รูทีน `_Write_Function` จะคำนวณหาไซลินเดอร์, แทร็ก, และเซกเตอร์ของเซกเตอร์ต่อไปและวนไปดำเนินการค้นหาข้อมูลเซกเตอร์ต่อไปจนครบตามที่ระบบปฏิบัติการ MS-DOS ต้องการ

รูทีน `_To_BIOS` เป็นรูทีนที่จะดำเนินการต่อจากรูทีน `_Cache` เมื่อระบบปฏิบัติการ MS-DOS ร้องขอรับบริการอื่นนอกจากอ่านข้อมูลหรือเขียนข้อมูล เมื่อรูทีน `_Cache` โอนการทำงานให้รูทีน `_To_BIOS` รูทีน `_To_BIOS` จะติดต่อกับ Disk-BIOS ให้ทำงานตามที่ระบบปฏิบัติการ MS-DOS ร้องขอรับบริการ หลังจากนั้นรูทีน `_To_BIOS` จะโอนการทำงานให้รูทีน `_To_DOS` ต่อไป

สำหรับบริการร้องขอรับบริการฟอร์แมทแทร็ก จะมีผลทำให้ข้อมูลในดิสก์ไม่ตรงกับข้อมูลในบัพเพอร์บล็อกได้ ดังนั้นในกรณีที่เป็นการร้องขอรับบริการฟอร์แมทแทร็ก จำเป็นต้องทิ้งข้อมูลในบัพเพอร์บล็อก ซึ่งรูทีน `_To_BIOS` จะคืนบัพเพอร์บล็อกที่เก็บข้อมูลของดิสก์ที่ถูกฟอร์แมทคืนให้ Storage Pools ก่อนดำเนินการติดต่อกับ Disk-BIOS ทำการฟอร์แมทแทร็กต่อไป

รูทีน `_To_DOS` เป็นรูทีนที่จะทำงานต่อจากรูทีนอื่นๆ เช่น รูทีน `_Read_Function`, รูทีน `_Write_Function`, รูทีน `_To_BIOS` เมื่อรูทีนต่างๆได้ดำเนินการตามที่ระบบปฏิบัติการร้องขอรับบริการแล้ว รูทีน `_To_DOS` จะดำเนินการจัดคำรีจิสเตอร์ก่อนส่งคืนการทำงานให้ระบบปฏิบัติการ MS-DOS ต่อไป

รoutines `_Check_Cache` และ routines `_Get_PSP` ทั้งสอง routines เป็น routines ที่ช่วยในการทำงานของบัพเฟอร์แคช routines `_Check_Cache` จะให้ค่า Carry Flag เพื่อบอกให้รู้ว่ามีการควบคุมบัพเฟอร์แคชติดต่อกับ Disk-BIOS ส่วน routines `_Get_PSP` เป็น routines ที่ทำหน้าที่คืนค่าอินเตอร์พด์เวกเตอร์เพื่อก่อนโปรแกรมควบคุมบัพเฟอร์แคชออกจาก Disk-BIOS และคืนค่า PSP Segment ให้ผ่านรีจิสเตอร์ AX เพื่อดำเนินการคืนพื้นที่หน่วยความจำที่ใช้เป็นบัพเฟอร์บล็อกให้ระบบปฏิบัติการ MS-DOS

เมื่อโปรแกรมควบคุมบัพเฟอร์แคชทำงานกับข้อมูลในดิสก์แผ่น ข้อมูลในบัพเฟอร์บล็อกยังคงเดิมถึงจะมีการเปลี่ยนแผ่นดิสก์ ทำให้ข้อมูลที่มีอยู่ในบัพเฟอร์บล็อกไม่ตรงกับข้อมูลที่มีอยู่ในแผ่นดิสก์ซึ่งถูกเปลี่ยนไปแล้ว ข้อมูลที่อ่านได้จะไม่ถูกต้อง ดังนั้นหลังจากที่ระบบปฏิบัติการ MS-DOS ร้องขอรับบริการอ่าน/เขียนข้อมูลไปแล้วช่วงเวลาหนึ่ง โปรแกรมควบคุมบัพเฟอร์แคชไม่ควรใช้ข้อมูลในบัพเฟอร์บล็อกที่มีอยู่เดิมแต่เรียกข้อมูลจากดิสก์แผ่นขึ้นมาใหม่

วิธีการแก้ปัญหาที่เกิดจากการเปลี่ยนแผ่นดิสก์ดังกล่าว จำเป็นต้องมี routines ที่มีหน้าที่จัดการเกี่ยวกับเวลาที่ได้กล่าวมาแล้วคือ routines `_Time_Routine` routines `_Time_Routine` นี้จะถูกติดตั้งกับ ROM-BIOS ที่อินเตอร์พด์เวกเตอร์ที่ 1CH เนื่องจาก DOS จะเรียกซอฟต์แวร์อินเตอร์พด์ที่ 1CH ประมาณ 18 ครั้งต่อ 1 วินาที

เนื่องจาก routines `_Time_Routine` เป็น routines อินเตอร์พด์ มีโอกาสที่จะเข้าไปขัดจังหวะการทำงานของ routines อื่นๆ ในโปรแกรมควบคุมบัพเฟอร์แคชและยังเข้าไปจัดการกับบัพเฟอร์บล็อกด้วยทำให้เกิดปัญหา Concurrency (Deitel, 1984, 75-95) ขึ้น จำเป็นต้องทำการ Synchronize การทำงานของ routines `_Time_Routine` กับ routines อื่นๆ การ Synchronize การทำงานทำได้โดยใช้ตัวแปร `_Lock` เป็นตัวแปรควบคุม ก่อนที่ routines `_Time_Routine` เข้าไปจัดการบัพเฟอร์บล็อก routines `_Time_Routine` ต้องตรวจค่าในตัวแปร `_Lock` ถ้าค่าในตัวแปร `_Lock` มีค่าเป็น 1 คือมี routines อื่นจัดการกับข้อมูลในบัพเฟอร์บล็อกอยู่ก่อน routines `_Time_Routine` จะคืนการทำงานให้ระบบปฏิบัติการ MS-DOS ทันที แต่ถ้าตัวแปร `_Lock` มีค่าเป็น 0 หรือไม่ มี routines ใดกำลังทำงานกับข้อมูลในบัพเฟอร์แคช routines `_Time_Routine` จะเข้าไปจัดการข้อมูลในบัพเฟอร์แคช

ทุกทีน Time_Routine จะตรวจสอบตัวแปร _Timer มีค่าเป็น 0 หรือไม่เนื่องจากตัวแปร _Timer เป็นตัวแปรที่เก็บค่าเวลาที่ยังเหลืออยู่ เมื่อระบบปฏิบัติการ MS-DOS ร้องขอรับบริการแต่ละครั้งตัวแปร _Timer จะถูกตั้งให้มีค่าเป็น 18 เสมอและเมื่อทุกทีน _Time_Routine ถูกเรียกให้ทำงานแต่ละครั้งถ้าตัวแปร _Timer มีค่ามากกว่า 0 ค่าตัวแปร _Timer จะลดลง 1 แต่ถ้าตัวแปร _Timer มีค่าเท่ากับ 0 ทุกทีน _Time_Routine จะคืนบัพเฟอร์บล็อกในลิสต์ที่เก็บข้อมูลจากดิสค์แผ่นให้ Storage Pools ก่อนคืนการทำงานให้ระบบปฏิบัติการ MS-DOS

4.3 การใช้งานบัพเฟอร์แคช

โปรแกรมควบคุมบัพเฟอร์แคชที่ทดลองสร้างขึ้นเป็นโปรแกรมแบบ TSR (Terminate and Stay Resident) โปรแกรมควบคุมบัพเฟอร์แคชจะคงตัวโปรแกรมไว้ในหน่วยความจำหลังจากเรียกให้โปรแกรมทำงานโดยใช้คำสั่งที่ DOS Prompt ดังนี้

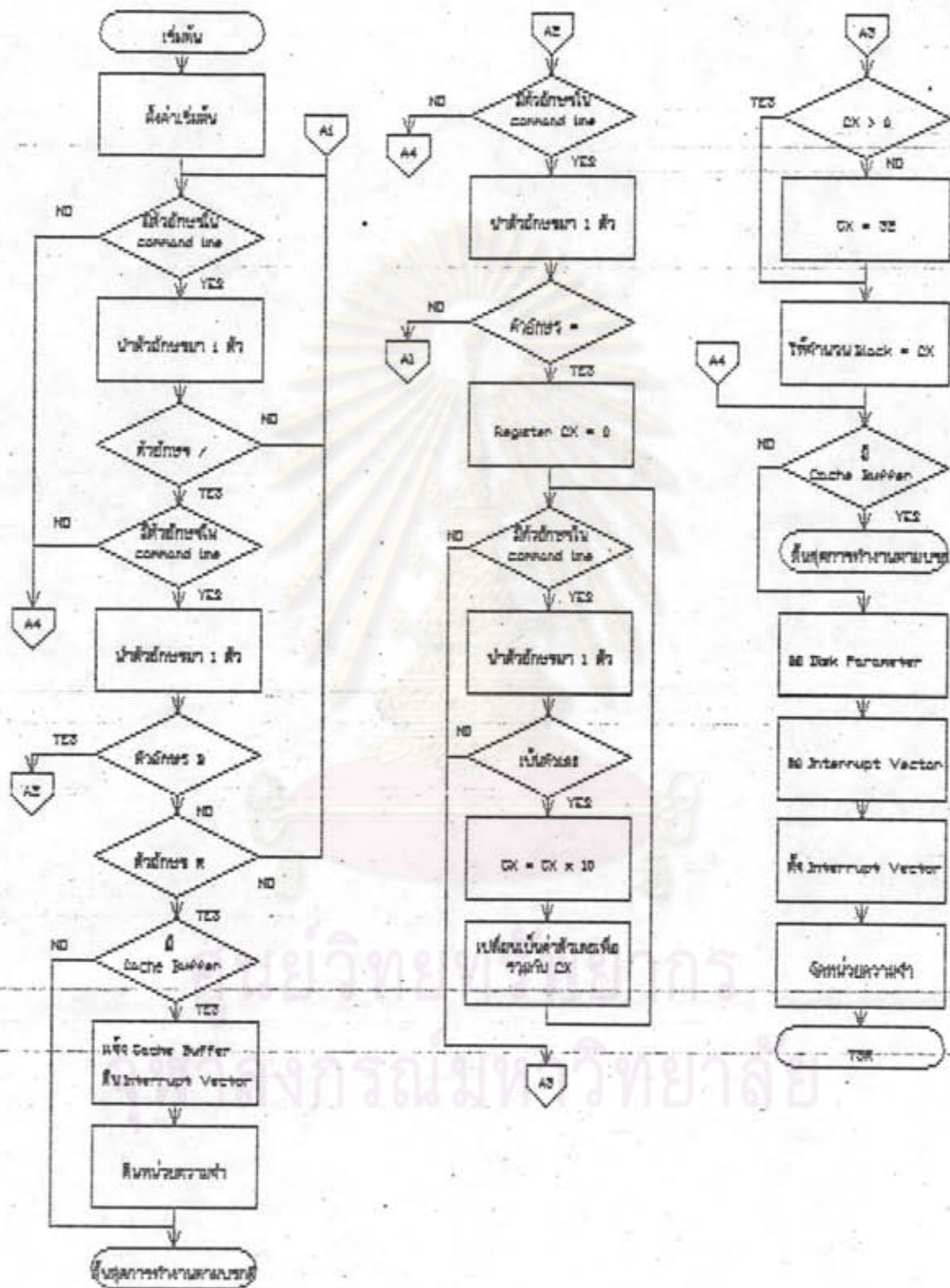
```
cache <enter>
```

โปรแกรมควบคุมบัพเฟอร์แคชจะเริ่มทำงานโดยใช้บัพเฟอร์บล็อกจำนวน 32 บล็อกขนาด 512 ไบต์ ในกรณีที่ต้องการให้โปรแกรมควบคุมบัพเฟอร์แคชทำงานโดยกำหนดจำนวนบัพเฟอร์บล็อกให้จะใช้คำสั่งเรียกให้โปรแกรมควบคุมบัพเฟอร์แคชทำงานที่ DOS Prompt ดังนี้

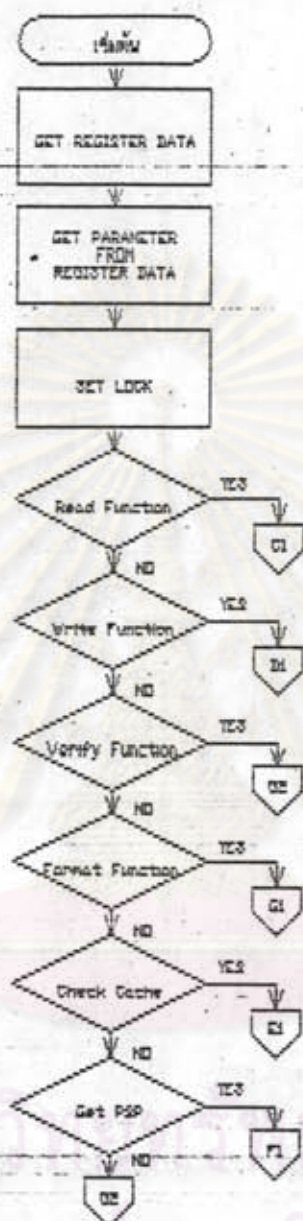
```
cache /b=xx <enter>
```

โดยที่ xx คือจำนวนบัพเฟอร์บล็อกที่ต้องการ บัพเฟอร์แคชที่ทดลองสร้างขึ้นสามารถกำหนดให้ใช้บัพเฟอร์บล็อกได้ถึงประมาณ 120 บล็อก ในกรณีที่ต้องการปลดโปรแกรมควบคุมบัพเฟอร์แคชออกจากดิสค์ไบออส จะใช้คำสั่งเรียกให้โปรแกรมควบคุมบัพเฟอร์แคชทำงานที่ DOS Prompt ดังนี้

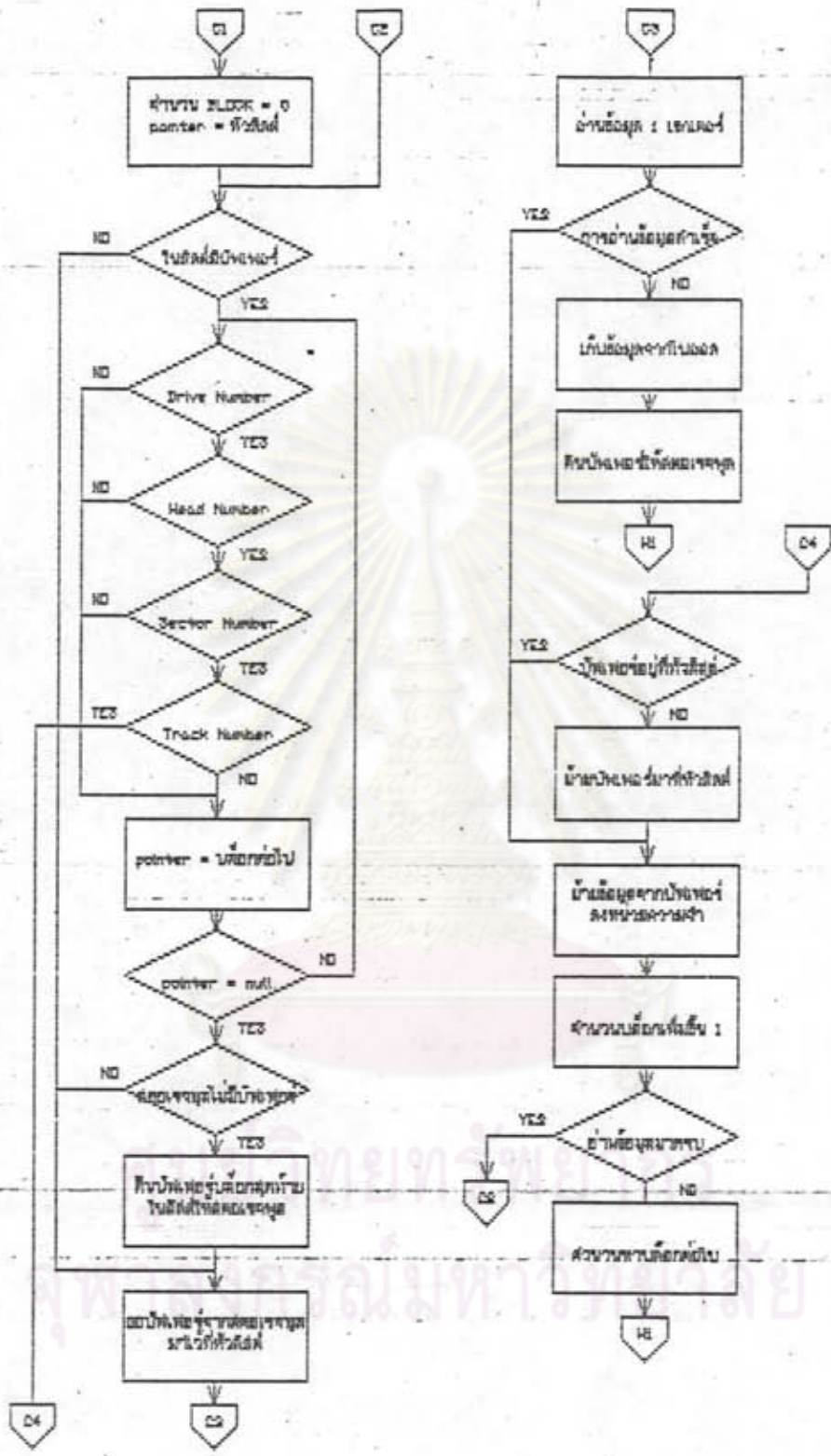
```
cache /r <enter>
```

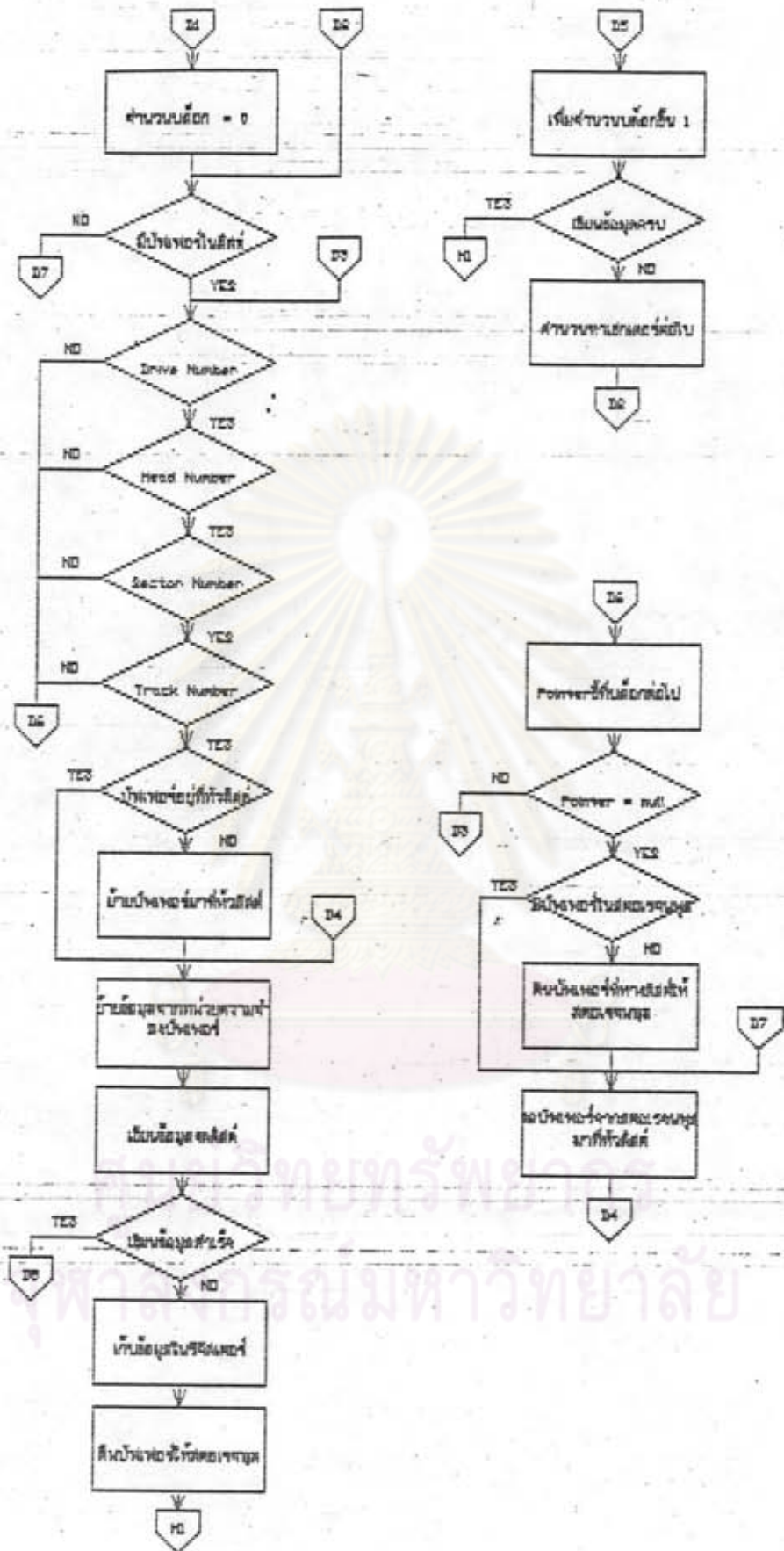
รูปที่ 4.10 Flow Chart routine _initialize



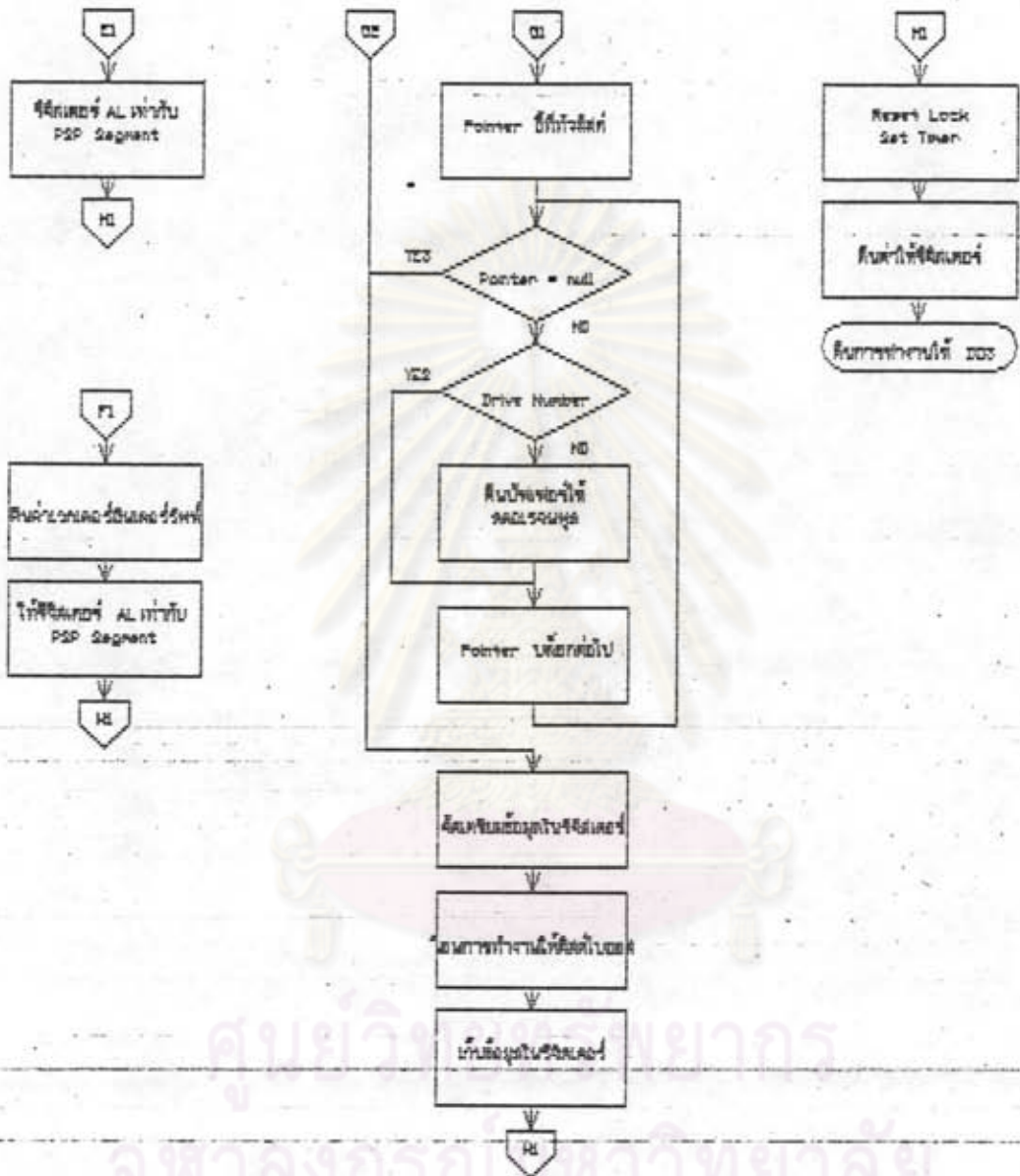
รูปที่ 4.11 Flow Chart รูปที่ _Cache



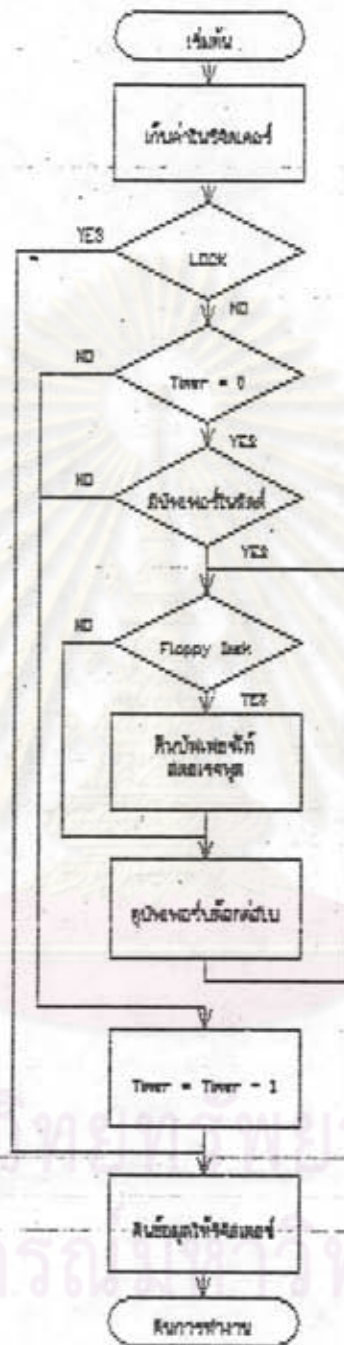
รูปที่ 4.12 Flow Chart ฟังก์ชัน Read Function



รูปที่ 4.13 Flow Chart หน้าที่ Write_Function



รูปที่ 4.14 Flow Chart หน้าที่ To_BIOS, หน้าที่ To_DOS



รูปที่ 4.15 Flow Chart วัฏจักร Time Routine