บทที่ 3

ระบบปฏิบัติการเอ็มเอสดอส (MS-DOS Operating System)

ระบบปฏิบัติการเอ็มเอสดอสเป็นระบบปฏิบัติการที่มีผู้นิยมใช้มากมีโปรแกรมสำเร็จประยุกต์ต่าง ๆ ที่ทำงานบนระบบปฏิบัติการนี้มากเช่นกัน

3.1 องค์ประกอบของระบบปฏิบัติการเอ็มเอสดอส[1]

ระบบปฏิบัติการเอ็มเอสดอสประกอบไปด้วยส่วนสำคัญต่าง ๆ คือ

3.1.1 โปรแกรมไบออส (BIOS or Basic Input/ Output System)

เป็นโปรแกรมย่อยที่ใช้ควบคุมส่วนของฮาร์ดแวร์ต่าง ๆ โดยตรง เพื่อช่วยให้การเรียกใช้อุปกรณ์ทางฮาร์ดแวร์เป็นไปโดยสะดวก โปรแกรมไบออสเป็นส่วนมาตรฐานที่ผู้ผลิตเครื่องคอมพิวเตอร์ส่วนบุคคลจัดทำและให้มาพร้อมกับเครื่อง โดยบรรจุอยู่ในหน่วยความจำถาวร (Read Only Memory)

3.1.2 โปรแกรมติดต่อหน่วยอินพุตเอาต์พุต

ใช้ชื่อแฟ้มว่า *IO.SYS* (ในบางระบบอาจจะใช้ชื่อที่แตกต่างกันไปจากนี้แต่การทำงานจะมีลักษณะเช่นเดียวกัน) เป็นโปรแกรมที่ทำหน้าที่เชื่อมต่อระหว่างโปรแกรมไบออสกับแก่นของระบบปฏิบัติการเพื่อให้ระบบปฏิบัติการแยกออกจากส่วนฮาร์ดแวร์โดยเด็ดขาด ดังนั้น ในกรณีที่ต้องการให้ระบบปฏิบัติการเอ็มเอสดอสใช้กับเครื่องแบบไหนก็สามารถทำได้โดยการแก้ไขเฉพาะโปรแกรมส่วนนี้เท่านั้น

3.1.3 แก่นของระบบปฏิบัติการเอ็มเอสดอส

ใช้ชื่อแฟ้มว่า *MSDOS.SYS* ซึ่งเป็นส่วนของระบบปฏิบัติการจริง ๆ โดยจะมีโปรแกรม

---

[1] Ray Duncan, ADVANCED MS-DOS (Washington : Micrsoft press, A devision of Microsoft Corp., 1986), pp. 11-19.

ระบบมาตรฐานที่เรียกใช้โดยผ่านสัญญาณขัดจังหวะหมายเลขต่าง ๆ สำหรับบริการให้กับโปรแกรมที่พัฒนาขึ้น โดยปกติระบบปฏิบัติการเอ็มเอสดอสจะใช้สัญญาณหมายเลข 21H เป็นมาตรฐานในการเรียกใช้

3.1.4 โปรแกรมเชลล์

เป็นโปรแกรมที่ทำหน้าที่ในการติดต่อกับผู้ใช้ โดยทำการรับคำสั่งจากผู้ใช้แล้วแปลความหมาย จากนั้นจึงส่งคำสั่งและการควบคุมระบบให้ระบบปฏิบัติการดำเนินการตามคำสั่งนั้นต่อไป เมื่อกระบวนการสิ้นสุดระบบจะส่งการควบคุมกลับมาที่โปรแกรมเชลล์นี้อีกครั้ง นอกจากนี้โปรแกรมเชลล์ยังบรรจุคำสั่งบางส่วนสำหรับช่วยการทำงานเบื้องต้นไว้ให้ผู้ใช้อีกด้วย เช่น COPY, EDIT หรือ FORMAT เป็นต้น โปรแกรมเชลล์ของระบบปฏิบัติการเอ็มเอสดอสใช้ชื่อแฟ้มว่า *COMMAND.COM*

3.2 การทำงานของระบบเชลล์ของระบบปฏิบัติการเอ็มเอสดอส

เมื่อทำการเปิดเครื่องคอมพิวเตอร์ส่วนบุคคล การทำงานจะเริ่มจากโปรแกรมไบออสจะตรวจสอบระบบฮาร์ดแวร์ โดยการทำงานนี้จะขึ้นอยู่กับการออกแบบของผู้ผลิตและจะให้รหัสบอกความผิดพลาด (Error codes) ในกรณีที่เครื่องมีปัญหา จากนั้นโปรแกรมไบออสจะอ่านโปรแกรม *IO.SYS* และ *MS-DOS.SYS* จากระบบจานบันทึกข้อมูล เช่น จานบันทึกข้อมูลชนิดอ่อน หรือ จานบันทึกข้อมูลชนิดแข็ง ลงสู่หน่วยความจำและส่งการควบคุมระบบให้กับระบบปฏิบัติการเอ็มเอสดอสต่อไป ระบบปฏิบัติการจะทำการตรวจสอบแฟ้มที่ใช้เริ่มการทำงานชื่อ CONFIG.SYS[2] และ AUTOEXEC.BAT[3] ตามลำดับ ถ้ามีก็จะดำเนินการตามค่าที่กำหนดไว้ จากนั้นจะทำอ่านโปรแกรมเชลล์และวิ่งโปรแกรมเชลล์โดยแสดงสัญญลักษณ์บอกบท (Prompt) มาตรฐานเพื่อรอรับคำสั่งจากผู้ใช้ต่อไป รูปที่ 3.2 แสดงให้เห็นถึงลำดับการทำงานของระบบเชลล์ของเอ็มเอสดอส และรูปที่ 3.1 แสดงให้เห็นตำแหน่งของโปรแกรมต่าง ๆ ในหน่วยความจำของระบบเชลล์

3.3 ระบบจานบันทึกข้อมูลของระบบปฏิบัติการเอ็มเอสดอส (MS-DOS Disk Internal)[4]

ในระบบปฏิบัติการเอ็มเอสดอส จะมีการจัดการกับระบบจานบันทึกข้อมูลที่มีรูปแบบค่อน

---

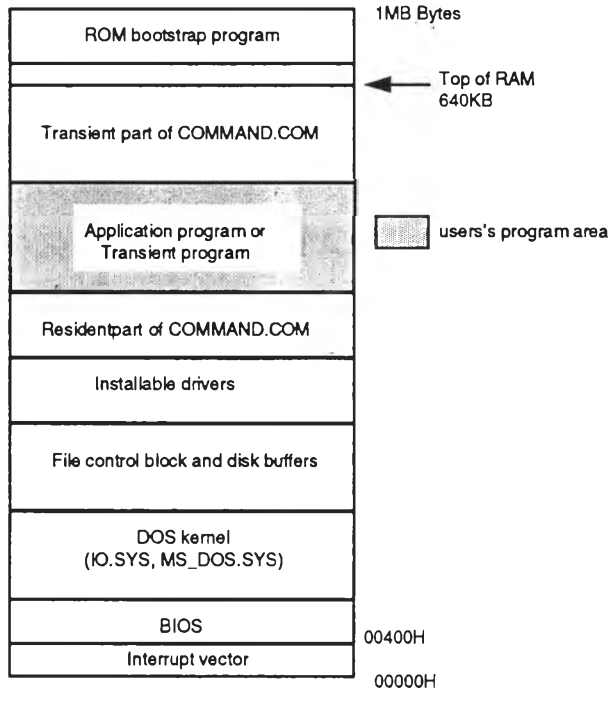
[2] John Angermeyer and Kevin Jaeger, MS-DOS DEVELOPER'S GUIDE (Indianapolis : Howard W. Sams & Co., 1986), p. 6,12.

[3] Ray Duncan. Op. cit. pp. 6,12.

[4] Ray Duncan. Op. cit. pp. 162-173.

ข้างแน่นอนสามารถทำความเข้าใจได้ง่ายและการนำไปใช้งานก็ไม่ยุ่งยาก ระบบปฏิบัติการเอ็มเอสดอสจะแสดงลักษณะของระบบจานบันทึกข้อมูลเป็นหน่วยตรรก (Logical Unit) หน่วยหนึ่ง การเรียกใช้จะใช้วิธีการอ้างชื่อตามลำดับของตัวอักษร เช่น A, B, C เป็นต้นและแต่ละหน่วยจะมีชื่อเฉพาะ มีส่วนประกอบเป็นไดเรกทอรีราก (Root Directory)[5] และไดเรกทอรีย่อย (Subdirectory)[6] หรือแฟ้มตั้งแต่ 0 แฟ้มขึ้นไป ในกรณีการเรียกใช้จากโปรแกรมที่พัฒนาขึ้นระบบปฏิบัติการเอ็มเอสดอสจะมีโปรแกรมระบบที่เรียกว่ารูทีนบริการขัดจังหวะ (Interrupt Services Routines) ช่วยในการทำงาน เพื่อลดความยุ่งยากในการเรียกใช้และทำให้ผู้ใช้ไม่จำเป็นต้องรู้จักคุณสมบัติทางกายภาพของอุปกรณ์นั้น ๆ

การขอใช้บริการแฟ้มในระบบปฏิบัติการเอ็มเอสดอส มีการทำงานแบ่งออกเป็น 2 ระดับ ก่อนจะมีการถ่ายโอนข้อมูลระหว่างอุปกรณ์ เช่น หน่วยขับจานบันทึกข้อมูลกับหน่วยความจำหลัก ซึ่งได้แก่

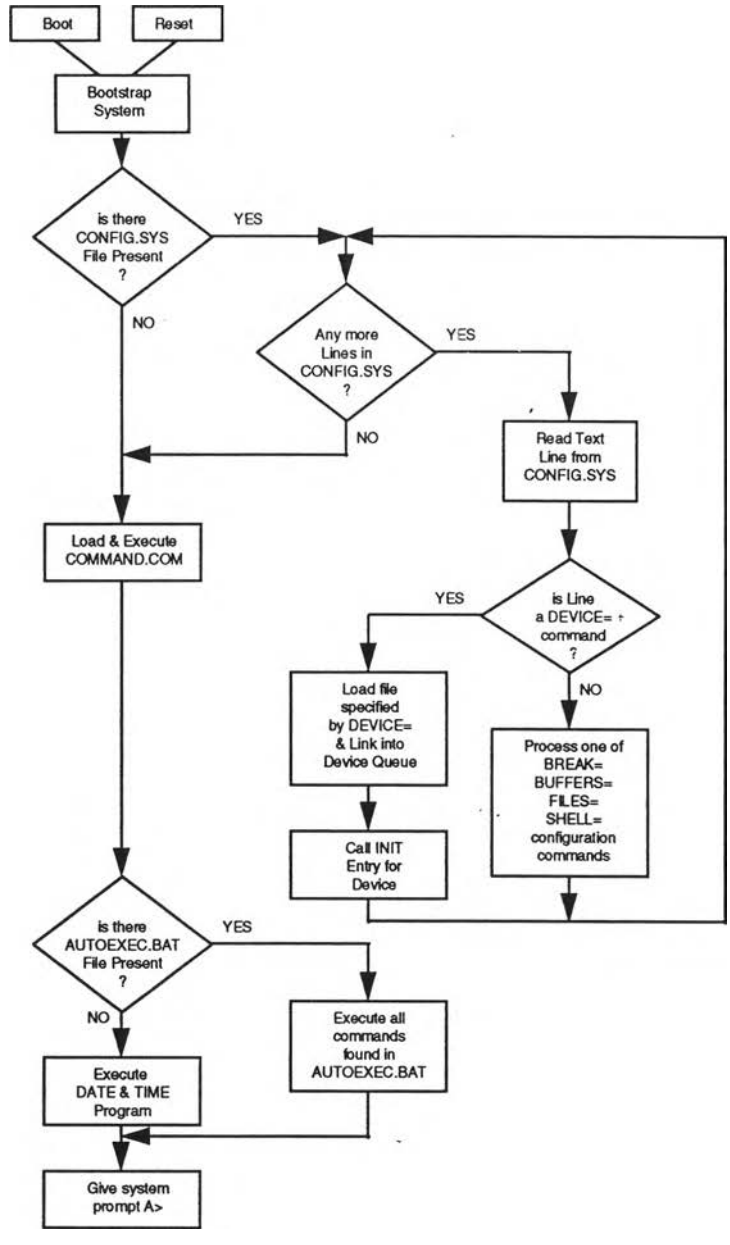


รูปที่ 3. 1 แสดงตำแหน่งของโปรแกรมต่าง ๆ ในหน่วยความจำ

---

[5] Ray Duncan, ADVANCED MS-DOS (Washington : Micrsoft press, A devision of Microsoft Corp., 1986), p. 149.

[6] Ibid. p. 150.



รูปที่ 3.2 แสดงขั้นตอนการทำงานของระบบปฏิบัติการเอ็มเอสดอสตั้งแต่เริ่มเปิดเครื่อง

3.3.1 การทำงานระดับผิวนอกระบบปฏิบัติการเอ็มเอสดอส

การทำงานจะเป็นในลักษณะที่อุปกรณ์แต่ละหน่วยจะถูกมองว่าประกอบด้วยหน่วยเก็บข้อมูลหรือเซกเตอร์ (Sector)[7] ที่ต่อเนื่องเรียงลำดับกันไปเป็นลูกโซ่ โดยมีค่าเริ่มต้นตั้งแต่เซกเตอร์ที่ 0 เป็นต้นไป

[7] H.M. Deitel. OPERATING SYSTEMS. 2nd ed. (Singapore : Addison-Wesley Publishing Company, 1990), p. 396.

แฟ้มหรือระเบียนที่ถูกใช้โดยโปรแกรมจะถูกแปลความหมายเป็นการเรียกใช้ข้อมูลจากเซกเตอร์แทน โดยอาศัยข้อมูลที่เก็บไว้ในไดเรกทอรีและตารางการจัดสรรแฟ้ม (File Allocation table) หรือเรียกกันทั่ว ๆ ไปว่า *แฟต (Fat)* ในการค้นหาและการเรียกใช้เซกเตอร์โดยตรงแบบนี้จะเรียกใช้โดยผ่านสัญญาณขัดจังหวะหมายเลข 25H และ 26H

3.3.2 การทำงานระดับภายใน

โดยการจัดผังของเซกเตอร์ทางตรรก ให้ตรงกับตำแหน่งที่แท้จริงกับลักษณะทางกายภาพ เช่น หัวอ่าน, แทร็ก หรือเซกเตอร์ เป็นต้น ในส่วนนี้จะทำงานโดยผ่านส่วนที่เรียกว่า *โปรแกรมย่อยอุปกรณ์ (Device Driver)*[8] โปรแกรมย่อยนี้จะขึ้นอยู่กับลักษณะของฮาร์ดแวร์โดยเฉพาะ และทำงานโดยการติดต่อโดยตรงกับส่วนแผงควบคุมของหน่วยขับจานบันทึกข้อมูล โปรแกรมย่อยส่วนใหญ่เกือบทั้งหมดจะถูกเขียนด้วยโปรแกรมภาษาแอสเซมบลี และต้องระมัดระวังในการเขียนเพื่อให้ได้ประสิทธิภาพที่ดีที่สุด ปกติผู้ผลิตอุปกรณ์หรือระบบจานบันทึกข้อมูลจะเขียนโปรแกรมย่อยส่วนนี้มาให้มากกว่าที่จะจัดทำโดยผู้ผลิตระบบปฏิบัติการ โปรแกรมย่อยจะต้องบรรจุไว้เป็นส่วนหนึ่งของระบบปฏิบัติการเสมอโดยจะเชื่อมต่อโดยตรงเข้ากับส่วนของโปรแกรมไบออสหรือโดยการติดตั้งไว้ในแฟ้ม CONFIG.SYS เพื่อติดตั้งเข้ากับระบบภายหลังก็ได้

Sector 0	OEM identification, BIOS parameter table, Boottrap routine Reserved area
	File Allocation Table (FAT) #1
	Possible additional of FAT
	Root disk directory
	File Area ..... File Area
Sector N	

รูปที่ 3.3 แสดงโครงสร้างของหน่วยตรรกหรือระบบจานบันทึกข้อมูลในระบบปฏิบัติการเอ็มเอสดอส

[8] Ray Duncan, ADVANCED MS-DOS (Washington : Micrsoft press, A devision of Microsoft Corp., 1986), pp. 222-225.

3.3.3 ส่วนประกอบของระบบจานบันทึกข้อมูล

ในแต่ละหน่วยตรรกของระบบปฏิบัติการเอ็มเอสดอส จะถูกแบ่งออกเป็น 2 ส่วนใหญ่ ๆ ที่มีขนาดแน่นอน โดยส่วนหนึ่งจะเป็นส่วนของข้อมูลที่ใช้ควบคุมการทำงานและอีกส่วนหนึ่งจะใช้ในการเก็บข้อมูลซึ่งแสดงรายละเอียดให้เห็นในรูปที่ 3.3 ขนาดของส่วนควบคุมนั้นขึ้นอยู่กับผู้ผลิตอุปกรณ์ซึ่งข้อมูลที่ถูกบรรจุไว้ต้องสามารถแปลความหมายเพื่อบอกโครงสร้างของอุปกรณ์หรือหน่วยขับจานบันทึกข้อมูลได้ ส่วนประกอบของส่วนควบคุมนี้ได้แก่

3.3.3.1 บูตเซกเตอร์ (Boot Sector)

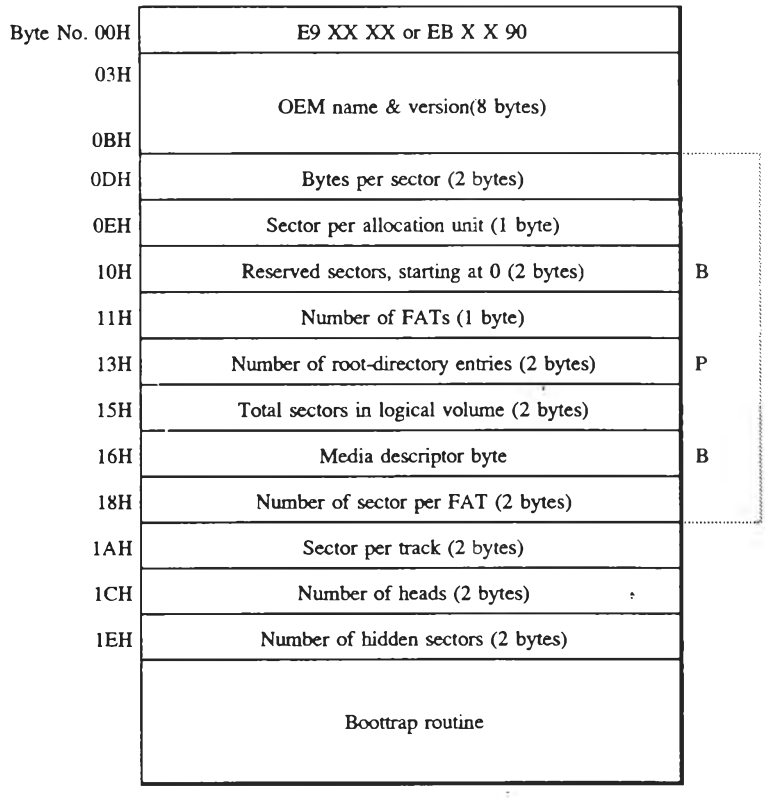
โดยปกติเซกเตอร์ที่ 0 จะถูกกำหนดให้เป็นบูตเซกเตอร์เพื่อบรรจุข้อมูลที่เกี่ยวข้องกับคุณสมบัติของอุปกรณ์นั้น ๆ ดังแสดงในรูปที่ 3.4 ข้อมูลชุดแรกหรือไบต์แรกจะเป็นคำสั่งกระโดดและตามด้วยตำแหน่งในไบต์ถัดมา ในกรณีที่จานบันทึกข้อมูลยังไม่ได้กำหนดรูปแบบ (format) มาก่อนจะไม่ปรากฏข้อมูลนี้อยู่ หรือในกรณีที่การกำหนดรูปแบบไม่ได้กระทำโดยระบบปฏิบัติการเอ็มเอสดอสก็จะไม่ปรากฏข้อมูลนี้เช่นกัน

ข้อมูลถัดมาจะเป็นข้อมูลที่แสดงชื่อผู้ผลิต เพื่อบอกถึงว่าอุปกรณ์นี้ใครเป็นผู้จัดทำขึ้นหรือรุ่นที่ทำขึ้นเพื่อใช้ในการอ้างอิงต่อไป ข้อมูลถัดไปเป็นส่วนที่สำคัญที่สุดของข้อมูลในบูตเซกเตอร์เรียกว่า *บีพีบี* (BPB มาจากคำย่อว่า BIOS Parameter Block)[9] ซึ่งจะเริ่มตั้งแต่ตำแหน่งไบต์ที่ 0BH ไปจนถึงไบต์ที่ 17H ของบูตเซกเตอร์ ข้อมูลชุดนี้จะอธิบายถึงโครงสร้างและคุณสมบัติทางกายภาพของอุปกรณ์ไว้อย่างละเอียดเพื่อให้โปรแกรมย่อยอุปกรณ์สามารถคำนวณค่าตำแหน่งทางกายภาพ ให้สอดคล้องกับค่าทางตรรกที่ถูกกำหนดไว้ได้อย่างเหมาะสม และนอกจากนี้ส่วนนี้ยังบรรจุข้อมูลสำหรับระบบปฏิบัติการเอ็มเอสดอสและโปรแกรมอรรถประโยชน์ต่าง ๆ ในการคำนวณตำแหน่งและขนาดของเนื้อที่ที่ใช้สำหรับควบคุมอุปกรณ์ด้วย เช่น ตารางการจัดสรรแฟ้มและไดเรกทอรีราก เป็นต้น

ข้อมูลชุดสุดท้าย ได้แก่ โปรแกรมการฉุดเริ่มเอง (Disk bootstep Program) ซึ่งปกติจะถูกอ่านลงสู่หน่วยความจำหลัก โดยโปรแกรมไบออสที่สั่งให้หน่วยขับจานบันทึกข้อมูลจัดตำแหน่งไปอยู่ที่ตำแหน่งเริ่มต้นและอ่านข้อมูลของโปรแกรมนี้ลงสู่หน่วยความจำที่กำหนดไว้แล้วทำการกระโดดไปดำเนินการ โปรแกรมการฉุดเริ่มเองจะอ่านและคำนวณหาค่าต่าง ๆ จากตำแหน่งทางกายภาพของจานบันทึกข้อมูลทำการอ่านโปรแกรมระบบปฏิบัติการทั้งหมดลงสู่หน่วยความจำ แล้วส่งการควบคุมการทำงานไปให้โปรแกรมระบบปฏิบัติการดำเนินการต่อไป

---

[9] Ray Duncan, ADVANCED MS-DOS (Washington : Micrsoft press, A devision of Microsoft Corp., 1986), p. 164.



รูปที่ 3.4 แสดงข้อมูลและองค์ประกอบภายในบูตเซกเตอร์

3.3.3.2 *เนื้อที่สำรอง (Reserved Area)*

จริง ๆ แล้วบูตเซกเตอร์ก็เป็นส่วนหนึ่งของเนื้อที่ที่สำรองไว้ ซึ่งสามารถนำไปใช้ได้ตั้งแต่ 1 จนถึงหลาย ๆ เซกเตอร์ ขนาดของเนื้อที่ที่สำรองไว้นี้กำหนดไว้ที่ตัวแปรชื่อ เซกเตอร์สำรอง (Reserved Sector) ในส่วนของบีพีบีในบูตเซกเตอร์

3.3.3.3 *ตารางการจัดสรรเนื้อที่แฟ้ม*

เป็นส่วนที่สัมพันธ์โดยตรงกับการกำหนดค่าของเกาะกลุ่ม(Cluster)[10] ในระบบจานบันทึกข้อมูลโดยมีการแบ่งออกเป็นเขต (field) โดยแต่ละเขตจะมีขนาดความยาว 12 บิตสำหรับในระบบ

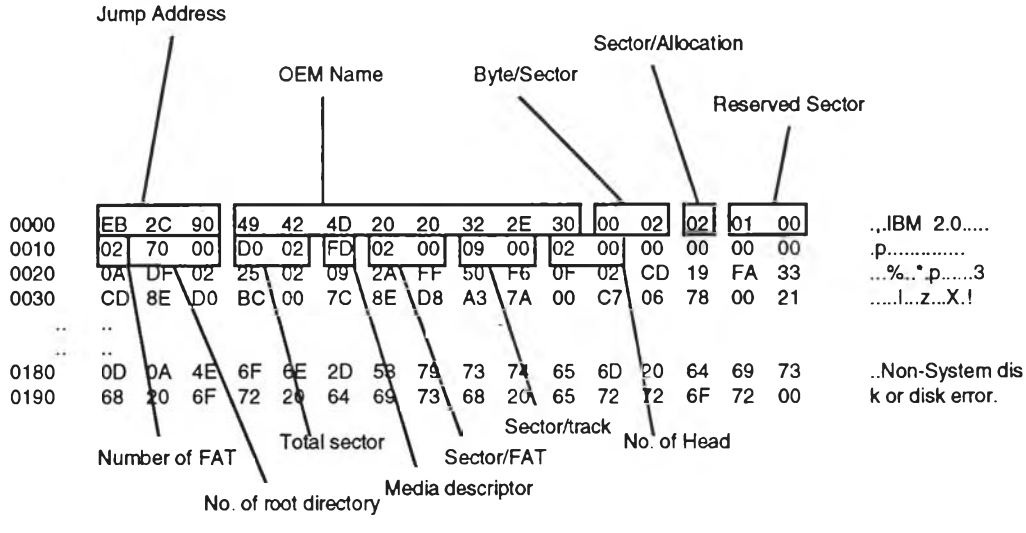
---

[10] Ray Duncan, ADVANCED MS-DOS (Washington : Micrsoft press, A devision of Microsoft Corp., 1986), pp. 170-173.

ปฏิบัติการเอ็มเอสดอสรุ่นแรก ๆ (รุ่นที่ 1 และ 2) และ 12 หรือ 16 บิต ในรุ่นที่ 3 หรือสูงกว่าซึ่งขึ้นอยู่กับขนาดของสื่อที่ใช้ สองเซตแรกของตารางการจัดสรรแฟ้มจะถูกสำรองไว้ ส่วนที่เหลือจะถูกนำไปใช้เก็บส่วนข้อมูลของแฟ้มซึ่งแฟ้มที่ถูกเก็บไว้ในไดเรกทอรีจะมีการบรรจุหมายเลขของเกาะกลุ่มแรกของแฟ้มไว้ เพื่อใช้เป็นจุดเริ่มต้นในการหาเกาะกลุ่มอื่น ๆ ของแฟ้มจากตารางการจัดสรรแฟ้มต่อไป จากจุดเริ่มต้นนี้แต่ละชุดจะมีการชี้ไปยังตำแหน่งของเกาะกลุ่มถัด ๆ ไปจนถึงเกาะกลุ่มสุดท้าย ก็จะเป็นการสิ้นสุดของข้อมูลตารางการจัดสรรแฟ้มของแต่ละอุปกรณ์ ตารางการจัดสรรแฟ้มอาจจะมีหนึ่งหรือสองชุดก็ได้

3.3.3.4 *ไดเรกทอรี*

ประกอบด้วยพื้นที่ที่เรียกว่าไดเรกทอรีราก ซึ่งมีรูปแบบที่กำหนดไว้แน่นอน คือ มีขนาด 32 ไบต์ เพื่อเก็บรายละเอียดของแฟ้ม ไดเรกทอรีย่อย และชื่ออุปกรณ์ ขนาดของไดเรกทอรีรากจะถูกกำหนดไว้ตั้งแต่แรกเมื่อมีตั้งค่าเริ่มต้น ซึ่งค่านี้จะถูกเก็บไว้ในบีพีบีที่ตำแหน่ง 0011H ของบูตเซกเตอร์



รูปที่ 3.5 แสดงตัวอย่างข้อมูลในบูตเซกเตอร์ของระบบจานบันทึกข้อมูล

3.3.3.5 *เนื้อที่เก็บแฟ้ม*

ส่วนที่เหลือของระบบจานบันทึกข้อมูลถัดจากส่วนของไดเรกทอรี จะเป็นส่วนที่ใช้เก็บแฟ้มหรือข้อมูลจริง ๆ จำนวนเซกเตอร์ของจานบันทึกข้อมูลในส่วนนี้จะมองเสมือนเป็นก้อนของ

เกาะกลุ่มที่บรรจุจำนวนเซกเตอร์ต่าง ๆ เอาไว้ ในแต่ละเกาะกลุ่มจะสัมพันธ์กับข้อมูลในตารางการจัดสรรแฟ้มเพื่อให้รู้ว่าเกาะกลุ่มนั้นถูกใช้ สำรองไว้ ยังว่างอยู่ หรือใช้ไม่ได้ เมื่อแฟ้มมีการเพิ่มขนาดตารางการจัดสรรแฟ้มจะถูกใช้ในการค้นหาจนกระทั่งเกาะกลุ่มที่ว่าง ค่าของสถานะของตารางการจัดสรรแฟ้มในตำแหน่งที่พบจะถูกเปลี่ยนให้เป็นสถานะเกาะกลุ่มสุดท้าย และเกาะกลุ่มก่อนหน้าจะถูกเปลี่ยนจากสถานะสุดท้ายมาเป็นสถานะที่ชี้ไปยังเกาะกลุ่มใหม่แทน เมื่อมีการเพิ่มขนาดอีกก็ทำในทำนองเดียวกัน ในกรณีของไดเรกทอรีย่อยจะเห็นว่าการที่สามารถเพิ่มจำนวนเกาะกลุ่มต่อแฟ้มได้ ทำให้จำนวนแฟ้มในไดเรกทอรีย่อยจึงไม่ถูกจำกัดเหมือนไดเรกทอรีราก

3.4 โปรแกรมย่อยอุปกรณ์

3.4.1 ประเภทของโปรแกรมย่อยอุปกรณ์

โปรแกรมย่อยอุปกรณ์ที่ใช้ในระบบปฏิบัติการเอ็มเอสดอสจะแบ่งออกเป็น 2 ประเภท ตามลักษณะการทำงาน คือ โปรแกรมย่อยอุปกรณ์สำหรับอุปกรณ์ที่รับส่งข้อมูลเป็นตัวอักษร (Character Device Driver) และ สำหรับอุปกรณ์ที่รับส่งข้อมูลเป็นบล็อก (Block Device Driver) การที่ต้องแบ่งประเภทก็เพื่อทำให้รู้ว่าโปรแกรมย่อยอุปกรณ์นั้น ๆ ทำหน้าที่อะไร หรือในมุมมองของระบบปฏิบัติการเอ็มเอสดอสจะเห็นลักษณะเป็นอย่างไรและการทำงานมีส่วนสัมพันธ์กับลักษณะทางกายภาพที่ถูกเรียกใช้งานตามปกติอย่างไร ดังนั้น จึงต้องมีการระบุไว้เสมอว่าโปรแกรมย่อยอุปกรณ์นั้นเป็นประเภทไหน

3.4.1.1 *โปรแกรมย่อยอุปกรณ์ประเภทตัวอักษร*

เรียกกันอีกอย่างหนึ่งว่า *โปรแกรมย่อยอุปกรณ์แบบมีชื่อ (Named Driver)* เป็นโปรแกรมย่อยอุปกรณ์ที่ใช้ควบคุมอุปกรณ์ที่มีการทำงานกับข้อมูลในลักษณะเป็นไบต์ เช่น โมเดม หรือเครื่องพิมพ์ เป็นต้น ซึ่งในระบบปฏิบัติการเอ็มเอสดอสจะใช้โปรแกรมย่อยประเภทนี้ 1 ตัวกับอุปกรณ์ 1 ชิ้นเท่านั้น โปรแกรมย่อยอุปกรณ์ประเภทตัวอักษรสามารถเปิดใช้งานเป็นได้ทั้งอินพุตหรือเอาต์พุตเช่นเดียวกับแฟ้ม

3.4.1.2 *โปรแกรมย่อยอุปกรณ์ประเภทบล็อก*

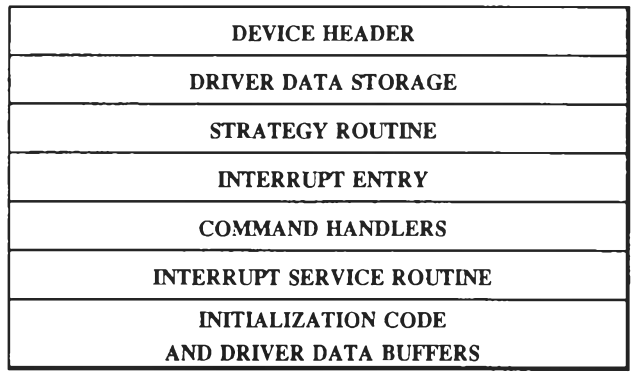
หรือเรียกว่า *โปรแกรมย่อยอุปกรณ์แบบไม่มีชื่อ (Unnamed Driver)* ปกติระบบปฏิบัติการเอ็มเอสดอสจะใช้โปรแกรมย่อยอุปกรณ์ประเภทนี้กับอุปกรณ์ที่ใช้เก็บข้อมูลจำนวนมาก และมีการใช้งานในลักษณะการอ่านหรือบันทึกแบบสุ่ม (Random Access) เช่น จานบันทึกชนิดแข็งหรือจานบันทึก

ชนิดอ่อน เป็นต้น โปรแกรมย่อยอุปกรณ์แบบนี้จะทำการรับส่งข้อมูลทีละชุดหรือบล็อกในแต่ละครั้ง แทนที่จะเป็นทีละไบต์เหมือนกับโปรแกรมย่อยอุปกรณ์ประเภทตัวอักษร โดยที่แต่ละชุดอาจจะเป็นแบบที่มีการกำหนดขนาดของข้อมูลที่แน่นอน (Fixed Size) หรือแบบไม่แน่นอน (Variabled Size) ก็ได้ ขึ้นอยู่กับลักษณะการใช้งานเป็นหลัก

โปรแกรมย่อยอุปกรณ์ประเภทบล็อกสามารถใช้ควบคุมอุปกรณ์ได้มากกว่า 1 ชุด หรือ จำลองอุปกรณ์ตรรกได้ถึง 2 ชุดหรือมากกว่าต่ออุปกรณ์จริง (Physical Unit) 1 ชุด หรือทั้ง 2 ประเภท ที่แตกต่างไปจากประเภทตัวอักษร ก็คือ ไม่สามารถใช้การอ้างถึงด้วยชื่อในลักษณะเช่นเดียวกับแฟ้มได้ การอ้างถึงจะกำหนดในลักษณะลำดับของตัวอักษร A, B หรือ C ไปตามลำดับ นอกจากประเภทบล็อกจะมีการอ่านและเขียนข้อมูลในลักษณะเป็นชุด ๆ แบบเซกเตอร์ที่มีขนาดแน่นอนและจะไม่มีการแก้ไข หรือเปลี่ยนแปลงขนาดแต่อย่างใด

3.4.2 โครงสร้างของโปรแกรมย่อยอุปกรณ์ (Device Driver Structure)[11]

ส่วนประกอบพื้นฐานของโปรแกรมย่อยอุปกรณ์จะประกอบไปด้วยส่วนต่าง ๆ ดังแสดงให้เห็นได้ดังในรูปที่ 3.6 ซึ่งส่วนประกอบหลักที่สำคัญ ได้แก่



รูปที่ 3.6 ส่วนประกอบต่าง ๆ ของโปรแกรมย่อยอุปกรณ์

---

[11] John Angermeyer and Kevin Jaeger, MS-DOS Developer's Guide (Indianapolis : Howard W. Sams & Co., 1986), pp. 192-202.

3.4.2.1 *ส่วนหัวของอุปกรณ์ (Device Header)* เป็นส่วนเริ่มต้นที่จะบอกรายละเอียดของโปรแกรมย่อยอุปกรณ์ทั้งหมดมีขนาด 18 ไบต์ ซึ่งประกอบไปด้วยข้อมูลสำหรับเชื่อมต่อกับอุปกรณ์ถัดไปในลูกโซ่เดียวกัน ข้อมูลที่กำหนดถึงคุณลักษณะของอุปกรณ์ และข้อมูลตำแหน่งของโปรแกรมต่าง ๆ ในโปรแกรมย่อยอุปกรณ์ ตลอดจนชื่ออ้างอิงถึงอุปกรณ์นั้น ๆ

3.4.2.2 *รูทีนสตราทีจี (Strategy Routine)* เป็นส่วนของโปรแกรมย่อยอุปกรณ์ที่ถูกเรียกใช้ในครั้งแรกที่เริ่มติดตั้ง และอีกครั้งเมื่อมีการเรียกใช้โปรแกรมย่อยอุปกรณ์จากโปรแกรมใช้งานปกติ เมื่อมีเรียกใช้หน่วยอินพุตและเอาต์พุตระบบปฏิบัติการจะส่งค่าตำแหน่งขนาด 4 ไบต์ของตำแหน่งโปรแกรมโดยผ่านโครงสร้างข้อมูลที่เรียกว่า *รีเควสเฮดเดอร์ (Request Header)* ซึ่งโครงสร้างนี้จะบรรจุข้อมูลเกี่ยวกับชนิดของการทำงานที่ต้องการให้ดำเนินการ

3.4.2.3 *รูทีนขัดจังหวะ (Interrupt Routine)* เป็นส่วนที่สำคัญและซับซ้อนที่สุดของโปรแกรมย่อย ซึ่งจะถูกเรียกใช้ทันทีหลังจากมีการเรียกรูทีนสตราทีจี ส่วนประกอบที่สำคัญ คือ โปรแกรมย่อยที่จัดทำตามลักษณะการทำงานต่าง ๆ ที่ต้องมีในโปรแกรมย่อยอุปกรณ์ ตำแหน่งของจุดกลางที่เข้าถึงโปรแกรมและตำแหน่งจุดกลางที่ออกจากโปรแกรม โปรแกรมส่วนนี้จะมีการพัฒนาให้เหมาะสมกับการทำงานของอุปกรณ์ซึ่งอาจจะจัดทำขึ้นใหม่หรือเรียกใช้จากโปรแกรมอื่น ๆ ที่ฝังตัวอยู่ในหน่วยความจำก็ได้

3.4.2.4 *คำสั่งในโปรแกรมย่อยอุปกรณ์* การทำงานต่าง ๆ ของโปรแกรมย่อยอุปกรณ์แสดงให้เห็นในตารางที่ 3.1 ซึ่งในเอ็มเอสดอสรุ่นที่ 2 และ รุ่นที่ 3 จะใช้เพียง 13 คำสั่ง ส่วนที่เหลือได้มีการเพิ่มเติมในภายหลัง

3.4.3 การใช้งานโปรแกรมย่อยอุปกรณ์โดยผ่านระบบเอ็มเอสดอส

ระบบปฏิบัติการเอ็มเอสดอสมีการเรียกใช้งานอยู่ 4 แบบ ดังในตารางที่ 3.2 ซึ่งได้แก่

3.4.3.1 *การใช้งานในแบบระบบปฏิบัติการซีพีเอ็ม*[12] *(CP/M Style Character Device I/O)* เป็นการทำงานแบบอิสระที่คล้ายกับที่ใช้ในระบบปฏิบัติการซีพีเอ็ม โดยกำหนดให้อุปกรณ์มาตรฐาน เช่น คอนโซล (CON) มีการทำงานในลักษณะของเทอร์มินอล คือ มีส่วนที่เป็นบัฟเฟอร์สำหรับ การโต้ตอบ การรอรับตัวอักษร ที่เข้ามาและการตรวจสถานะอุปกรณ์อื่น ๆ ที่ไม่อยู่ในมาตรฐานต้องมีแฟ้มควบคุมเพื่อควบคุมการใช้งาน

---

[12] William H. Gates, Jr. MS-DOS Technical Reference Encyclopedia (Washington : Microsoft press, Microsoft Corp., 1986), pp.4-41.

ตารางที่ 3.1 แสดงคำสั่งต่าง ๆ ในโปรแกรมย่อยอุปกรณ์

Command	Block	Char	Function Meaning
0:	X	X	INIT
1:	X		MEDIA CHECK
2:	X	X	BUILD BIOS BPB
3:	X	X	IOCTL INPUT
4:	X	X	INPUT (read)
5:		X	nondestructive INPUT no-wait
6:		X	INPUT STATUS
7:		X	INPUT FLUSH
8:	X	X	OUTPUT(write)
9:	X	X	OUTPUT(write) with VERIFY
10:		X	OUTPUT STATUS
11:		X	OUTPUT FLUSH
12:	X	X	IOCTL OUTPUT
13:	X	X	DEVICE OPEN
14:	X	X	DEVICE CLOSE
15:	X		REMOVABLE MEDIA

3.4.3.2 *การใช้งานผ่านโครงสร้างควบคุมอุปกรณ์ (Using the File Control Block)* เป็นวิธีการเรียกใช้อุปกรณ์ที่นิยมใช้กันมากแม้ว่าการใช้วิธีการอื่นจะง่ายกว่า ซึ่งวิธีนี้อนุญาตให้ผู้เขียนโปรแกรมสามารถเข้าถึงระเบียนในแฟ้มได้โดยตรง ทำให้สามารถใช้การทำงานในแบบสุ่มได้

3.4.3.3 *การใช้งานผ่านแฟ้มควบคุม (Using File Handles For Device I/O)* แม้ว่าการใช้โครงสร้างข้อมูลควบคุมจะดีสำหรับการจัดการแฟ้มแต่ในกรณีที่อุปกรณ์นั้น ๆ ไม่ใช่อุปกรณ์ประเภทหน่วยบันทึกข้อมูล (Non-disk) โครงสร้างควบคุมจึงไม่จำเป็น จึงสามารถใช้งานโดยผ่านฟังก์ชันที่ชื่อ Ioctl หรือ ฟังก์ชันหมายเลข 44H ในระบบปฏิบัติการเอ็มเอสดอสได้

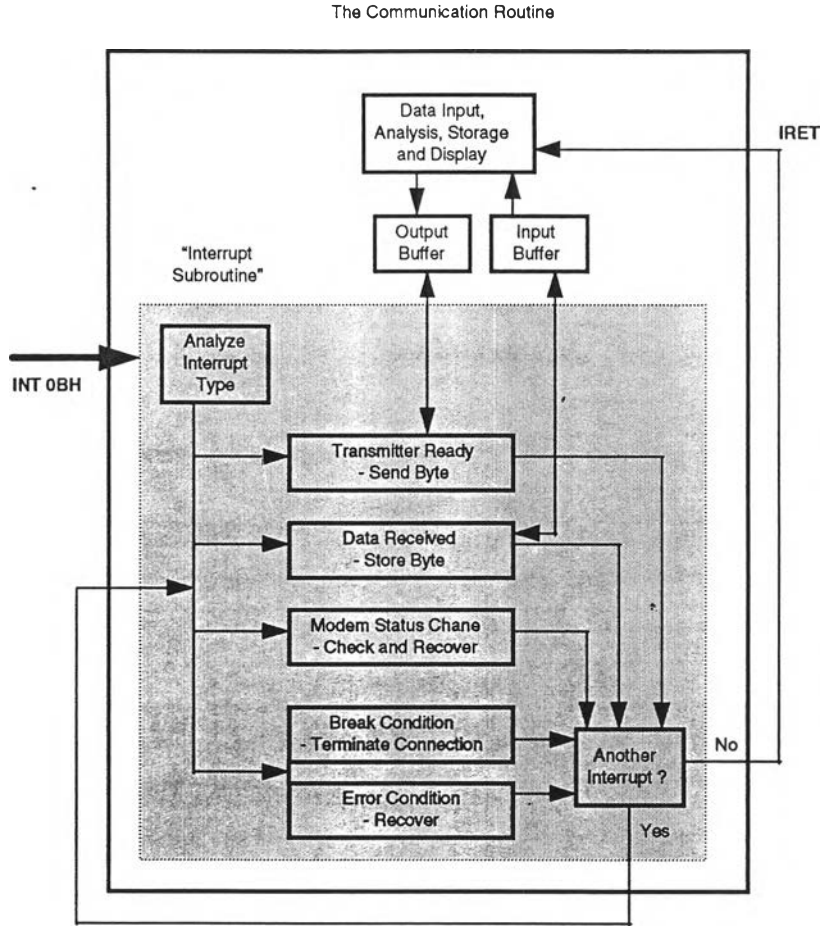
ตารางที่ 3.2 แสดงประเภทของการเรียกใช้งานโปรแกรมย่อยอุปกรณ์

CP/M-style dedicated I/O functions for devices such as console, printer	CON: PRN: AUX:	Function 01, 02, 06 - 0Ch Function 05 Function 03 and 04
CP/M-style file access using the FCB(file control block)	Open/Close: Device/File Read/Write: File Only Read/Write:	Functions 0Fh and 10h Functions 14h and 15h Functions 21h, 22h, 27h and 28h
MS-DOS style file access using FCB	Open/Close: Device/File Read/Write: I/O Control for Devices:	Functions 3Dh and 3Eh Functions 3Fh and 40h Functions 44h
Direct disk I/O functions using absolute disk reads and writes	Absolute read Absolute write	interrupt 25H interrupt 26H

3.4.3.4 *การเรียกใช้งานโดยตรง (Direct Disk Access)* เป็นการจัดการแฟ้มข้อมูลสำหรับอุปกรณ์แบบสุดท้าย โดยการเรียกใช้ *สัญญาณขัดจังหวะหมายเลข 25H (Absolute Disk Read)* และ *หมายเลข 26H (Absolute Disk Write)* ซึ่งจะเป็นการเรียกใช้หน่วยบันทึกข้อมูลโดยตรง ซึ่งเป็นอุปกรณ์แบบบล็อกเกือบทั้งหมด วัตถุประสงค์ของการเรียกใช้แบบนี้ก็เพื่อให้การใช้งานหน่วยบันทึกข้อมูลแบบจานบันทึกข้อมูลสามารถกระทำได้โดยตรง ไม่จำเป็นต้องผ่านโครงสร้างแฟ้มข้อมูลของระบบปฏิบัติการเอ็มเอสดอส

3.4.4 การติดตั้งโปรแกรมย่อยอุปกรณ์

สามารถทำการติดตั้งได้โดยกำหนดชื่อโปรแกรมย่อยอุปกรณ์ไว้ในคำสั่ง DEVICE = ในแฟ้มชื่อ *CONFIG.SYS* ที่ใช้กำหนดการติดตั้งอุปกรณ์พิเศษอื่น ๆ นอกเหนือไปจากปกติ ซึ่งได้กล่าวมาแล้วในตอนต้น



รูปที่ 3.7 แสดงขั้นตอนการทำงานของทางเข้าออกสื่อสารโดยวิธีขัดจังหวะ

3.5 ระบบการสื่อสารของระบบปฏิบัติการเอ็มเอสดอส[13]

เครื่องคอมพิวเตอร์ส่วนบุคคลจะติดต่อโดยผ่านทางเข้าออกอสมวาร RS-232C โดยมีหน่วยควบคุมเป็นไอซีหมายเลข INS8250 ของบริษัท เนชันแนลเซมิคอนดักเตอร์ จำกัด สามารถรับส่งข้อมูลเป็นไบต์และมีความเร็วในการรับส่ง สูงถึง 115,200 โบด รีจิสเตอร์ที่ใช้อยู่มีอยู่ 10 ตัว ขนาด 1 ไบต์ โดยในการใช้งานสามารถตั้งค่าให้กับรีจิสเตอร์เหล่านี้ให้สอดคล้องกับการทำงานที่ต้องการได้

ในจำนวนรีจิสเตอร์ทั้ง 10 ตัวนั้น จะมีเพียง 6 ตัวเท่านั้นที่จำเป็นสำหรับการใช้งาน ได้แก่ *รีจิสเตอร์ทรานสมิตเตอร์โฮลดิง (Transmitter holding register)* ทำหน้าที่เก็บข้อมูลไว้เพื่อรอส่ง *รีจิสเตอร์*

---

[13] ไพศาล สงวนหมู่ และ ยืน ภู่วรวรรณ, การสื่อสารข้อมูลและไมโครคอมพิวเตอร์เนตเวอร์ค (กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชัน จำกัด, 2528), หน้า 102-111.

รีซีพเคดา (Received data register) ทำหน้าที่รับข้อมูลที่เข้ามาที่ทางเข้าออกสื่อสาร *รีจิสเตอร์ไลน์สเตตัส (Line status register)* ทำหน้าที่กำหนดคุณลักษณะของข้อมูลและสถานะของการรับส่งข้อมูลในสายสัญญาณและ *รีจิสเตอร์โบดเรตดีไวเซอร์ (Baud-rate divisor register)* อีก 2 ตัวสำหรับเก็บค่าตัวแปรในการกำหนดความเร็ว ส่วนที่เหลืออีก 4 ตัวซึ่งได้แก่ *รีจิสเตอร์โมเดมคอนโทรล* และ *สเตตัส (Modem Control* และ *status register)* และ*รีจิสแตอร์อินเตอร์รัพต์อีนาเบิล* และ *ไอเดนติฟิเคชัน (Interrupt enable* และ *Identification register)* จะทำหน้าที่ในการกำหนดคุณลักษณะการทำงานในการใช้งานร่วมกับอุปกรณ์โมเดม และการใช้งานในลักษณะของการเรียกใช้โดยวิธีขัดจังหวะ (Interrupt driven)[14] รูปที่ 3.7 แสดงให้เห็นหลักการการใช้งานทางเข้าออกสื่อสารโดยวิธีนี้

---

[14] ไพศาล สงวนหมู่ และ ยืน ภู่วรวรรณ, การสื่อสารข้อมูลและไมโครคอมพิวเตอร์เนตเวอร์ค (กรุงเทพมหานคร : บริษัท ซีเอ็ดยูเคชัน จำกัด, 2528), หน้า 88-90.