บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

ไบออส ดอส และอินเทอร์รัพท์

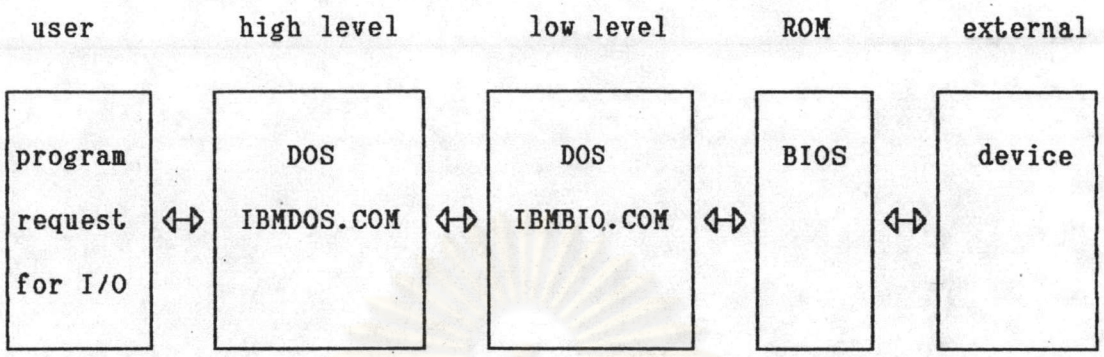
ไบออสและดอสกำหนดตารางอินเทอร์รัพท์ที่ตำแหน่ง 0 ถึง 1,023 (0000H-03FFH) ในหน่วยความจำ ตารางอินเทอร์รัพท์สามารถรองรับอินเทอร์รัพท์ได้ 256 อินเทอร์รัพท์โดยแต่ละช่องมีขนาด 4 ไบต์สำหรับเก็บตำแหน่งของชุดคำสั่งอินเทอร์รัพท์ในรูป CS:IP โดยไบออสจัดการอินเทอร์รัพท์ 0 ถึง 31 (0000H-001FH) และดอสจัดการอินเทอร์รัพท์ 32 ถึง 63 (0020H-003FH) ฮาร์ดแวร์อินเทอร์รัพท์ (hardware interrupt) เกิดจากอุปกรณ์ภายนอกที่ต่อกับโปรเซสเซอร์ จุดกำเนิดของฮาร์ดแวร์อินเทอร์รัพท์มีได้ 2 ประเภท ประเภทแรกคืออินเทอร์รัพท์ที่ไม่สามารถระงับได้ (nonmaskable interrupt-NMI) เกิดจากความผิดพลาดพาริตี้ของไอโอและหน่วยความจำ (memory and I/O parity error) โปรเซสเซอร์สนองตอบต่ออินเทอร์รัพท์นี้เสมอ อีกประเภทคือเกิดจากการร้องขออินเทอร์รัพท์ (interrupt request-INTR) เช่น การร้องขอจากอุปกรณ์ภายนอกคือ อินเทอร์รัพท์ 5 ถึงอินเทอร์รัพท์ 15 (0005H-000FH) สำหรับแป้นพิมพ์ พอร์ตอนุกรม จานบันทึก และพอร์ตขนาน ส่วนซอฟต์แวร์อินเทอร์รัพท์ (software interrupt) เกิดจากการทำคำสั่งอินเทอร์รัพท์ (INT instruction) การทำคำสั่งหารที่ทำให้เกิดการล้น (overflow) การทำงานในโหมดขั้นตอนเดียว (single-step mode) หรือการร้องขออินเทอร์รัพท์ภายนอก (request for external interrupt) เช่น การรับและส่งข้อมูลกับจานบันทึก โปรแกรมใช้ซอฟต์แวร์อินเทอร์รัพท์เพื่อติดต่อกับชุดคำสั่งไบออสและดอส ซึ่งเป็นอินเทอร์รัพท์ที่ไม่สามารถระงับได้ (1)

ส่วนข้อมูลของไบออส (BIOS Data) ซึ่งมีตำแหน่งอยู่ที่ส่วนล่างของหน่วยความจำ โดยมีตำแหน่งเริ่มต้นที่ตำแหน่ง 40:00H ไบออสจะทำการดูแลข้อมูลในส่วนนี้ ซึ่งเป็นข้อมูลที่ใช้

อ้างอิงในการทำงานของระบบ ครอบคลุมพอร์ตอนุกรม พอร์ตขนาน แป้นพิมพ์ หน่วยขับแผ่นบันทึก ส่วนควบคุมการแสดงภาพ (Video Control data area) ตลอดจนสัญญาณนาฬิกาของระบบ (Real-time Clock data area)

ไบออสอินเทอร์รัพท์เริ่มจากอินเทอร์รัพท์ 0 ถึงอินเทอร์รัพท์ 31 (0000H-001FH) ซึ่งครอบคลุมอินเทอร์รัพท์ที่เกี่ยวกับการหารด้วยศูนย์ (divide by zero) การพิมพ์ข้อมูลจากจอภาพออกเครื่องพิมพ์ (print screen) สัญญาณนาฬิกา (timer) การควบคุมการแสดงผล (video control) การสื่อสาร (communication input/output) การรับข้อมูลจากแป้นพิมพ์ (keyboard input) การส่งข้อมูลออกเครื่องพิมพ์ (printer output) และโปรแกรมบูทสแตปโหลดเดอร์ (bootstap loader) อินเทอร์รัพท์ 32 ถึง 63 (0020H-003FH) สงวนไว้ให้ดอส อินเทอร์รัพท์ที่สำคัญที่สุดของดอสคืออินเทอร์รัพท์ที่ 33 (INT 0021H) ซึ่งให้บริการเกี่ยวกับการรับข้อมูลจากแป้นพิมพ์ การส่งข้อมูลออกจอภาพ การส่งข้อมูลออกเครื่องพิมพ์ เปิดและปิดแฟ้มข้อมูล ลบแฟ้มข้อมูล อ่านและเขียนข้อมูลแบบสุ่มและแบบเรียงลำดับ การสร้างไดเรกทอรี โปรแกรมฝังตัว และการจบการทำงานของโปรแกรม โดยที่โปรแกรมไอบีเอ็มบีไอโอและไอบีเอ็มดอสจะช่วยให้การใช้งานไบออสทำได้ง่ายและเป็นอิสระจากตัวเครื่อง (machine independent) มากขึ้น โปรแกรมไอบีเอ็มบีไอโอจะทำหน้าที่ติดต่อกับไบออสในระดับล่าง (low-level interface) โปรแกรมไอบีเอ็มดอสจะทำหน้าที่จัดการแฟ้มข้อมูลและฟังก์ชันอื่น ๆ เมื่อโปรแกรมของผู้ใช้เรียกใช้งานอินเทอร์รัพท์ 33 โปรแกรมไอบีเอ็มดอสจะแปลคำร้องขอเป็นหนึ่งหรือหลาย ๆ คำสั่ง ส่งไปยังโปรแกรมไอบีเอ็มบีไอโอ ซึ่งจะส่งคำสั่งนั้นลงไปยังไบออสเพื่อติดต่อกับอุปกรณ์ภายนอกในท้ายที่สุด ความสัมพันธ์ตามแผนภาพที่ 1 (1)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



แผนภาพที่ 1 แสดงความสัมพันธ์ของไบออสและดอส

ที่มา Peter, A. IBM PC Assembly language and programming. (New Jersey: Prentice-Hall,Inc. , 1991), p. 421.

ส่วนที่เป็นตัวเชื่อมโยงโปรเซสเซอร์ (processor) กับอุปกรณ์ภายนอก (external device) เรียกว่าพอร์ต (port) โปรเซสเซอร์จะได้รับสัญญาณ (signal) จากอุปกรณ์รับเข้า (input device) และส่งสัญญาณออกไปยังอุปกรณ์แสดงผล (outputt device) โดยผ่านพอร์ต และตำแหน่งของพอร์ตจะถูกระบุระหว่าง 0 ถึง 1,023 รวม 1,024 พอร์ต

โปรแกรมฝังตัว

โปรแกรมฝังตัว (Terminate but Stay Resident program - TSR) คือ โปรแกรมใดๆ ที่มีการเรียกใช้อินเทอร์รัพท์ 33 ฟังก์ชัน 49 (INT 0021H function 0031H) หรือ อินเทอร์รัพท์ที่ 39 (INT 0027H) ซึ่งอินเทอร์รัพท์นี้จะจัดเตรียมเนื้อที่ในหน่วยความจำตามขนาดที่ได้รับที่ร้องขอเป็นพารากราฟหลังจากจบการทำงานของโปรแกรม ซึ่งแตกต่างจากการเรียกอินเทอร์รัพท์ 33 ฟังก์ชัน 76 (INT 0021H function 004CH) เพื่อจบการทำงานของโปรแกรมโดยทั่วไป ตรงที่เนื้อที่ในหน่วยความจำซึ่งโปรแกรมเคยครอบครองอยู่จะไม่ถูกปลดปล่อยไปทั้งหมด แต่บางส่วนหรือทั้งหมดของเนื้อที่ในหน่วยความจำนั้นจะถูกกันไว้มิให้โปรแกรมที่ถูกบรรจุเข้ามาทำงานใหม่นำไปใช้ได้ (2)

เครื่องคอมพิวเตอร์ที่มีพื้นฐานทางสถาปัตยกรรมเป็นอินเทล 80X86 อนุญาตให้โปรแกรมใด ๆ สามารถติดตั้งชุดคำสั่ง เพื่อถูกเรียกให้ทำงานเมื่อเกิดอินเทอร์รัพท์ทางซอฟต์แวร์หรือฮาร์ดแวร์ได้ ตัวอย่างในที่นี้ก็คือดอส ซึ่งอาศัยอินเทอร์รัพท์ที่ 33 (INT 0021H) เป็นประตูไปสู่การเรียกใช้งานดอส โดยที่อินเทอร์รัพท์ 33 นี้ชี้ไปยังตำแหน่งของชุดคำสั่งของดอสที่ทำหน้าที่ให้บริการต่าง ๆ จากคุณสมบัติที่สามารถติดตั้งชุดคำสั่งเพื่อให้ถูกเรียกทำงานเมื่อเกิดอินเทอร์รัพท์นั้น ทำให้โปรแกรมฝังตัวเป็นสิ่งที่มีประโยชน์มากกว่าจะเป็นเพียงโปรแกรมที่เบียดบังเนื้อที่ในหน่วยความจำแต่เพียงอย่างเดียว การติดตั้งชุดคำสั่งเพื่อให้ถูกเรียกใช้เมื่อเกิดอินเทอร์รัพท์นี้ดอสได้เตรียมบริการไว้ให้ใช้คือ อินเทอร์รัพท์เพื่อตรวจสอบตำแหน่งที่อินเทอร์รัพท์นั้น ๆ ชี้อยู่ (DOS GET Interrupt Vector function) คืออินเทอร์รัพท์ 33 ฟังก์ชัน 53 (INT 0021H function 0035H) และอินเทอร์รัพท์เพื่อติดตั้งให้อินเทอร์รัพท์นั้น ๆ ชี้ไปยังชุดคำสั่งที่ต้องการ (DOS SET Interrupt Vector function) คืออินเทอร์รัพท์ 33 ฟังก์ชัน 37 (INT 0021H function 0025H) เมื่อได้ติดตั้งให้อินเทอร์รัพท์ที่ต้องการชี้ไปยังชุดคำสั่งของโปรแกรมฝังตัวภายหลังจากที่โปรแกรมได้จบสิ้นการทำงานในลักษณะของโปรแกรมฝังตัวแล้ว เมื่อใดก็ตามที่เกิดอินเทอร์รัพท์ดังกล่าวขึ้นโปรแกรมฝังตัวจะตื่นขึ้นมาทำงาน โปรแกรมฝังตัวมีส่วนประกอบดังนี้ (1)

1) ส่วนที่ทำหน้าที่ตรวจสอบเงื่อนไขของการเริ่มทำงานของโปรแกรมฝังตัว ซึ่งจะทำงานเมื่อมีการเรียกใช้บริการจากชุดคำสั่งในตารางอินเทอร์รัพท์ที่ได้กำหนดไว้

2) ส่วนที่ทำหน้าที่ติดตั้งโปรแกรมฝังตัว ซึ่งจะเปลี่ยนแปลงตำแหน่งที่เก็บอยู่ในตารางอินเทอร์รัพท์ให้เป็นตำแหน่งของชุดคำสั่งของโปรแกรมฝังตัว กำหนดขนาดของชุดคำสั่งที่จะยังคงค้างอยู่ในหน่วยความจำ และเรียกใช้บริการจากอินเทอร์รัพท์เพื่อสั่งให้ดอสจบการทำงานของโปรแกรมในลักษณะโปรแกรมฝังตัว

3) ส่วนที่คงค้างอยู่ในหน่วยความจำ และตื่นขึ้นมาทำงานเมื่อมีเหตุการณ์ที่กำหนดไว้เกิดขึ้นเพื่อทำงานต่าง ๆ ตามที่ได้กำหนดไว้

เมื่อทำการติดตั้งโปรแกรมฝังตัวแล้วหน่วยความจำจะมีลักษณะดังต่อไปนี้ (1)

Interrupt service table

IO.SYS and MSDOS.SYS

COMMAND.COM

Resident portion of program (stay in memory)

Initialization portion of program (overlaid by next program)

Rest of available memory

โดยทั่วไปโปรแกรมฝังตัวที่ต้องการให้ทำงานตามความต้องการของผู้ใช้ จะออกแบบให้เริ่มต้นการทำงานโดยผู้ใช้กดโปรแกรมฮอทคีย์ (program hot key) ซึ่งเป็นแป้นพิมพ์ที่ได้กำหนดการทำงานไว้ โดยอาศัยอินเทอร์รัพท์ 9 (INT 0009H) ซึ่งเกิดขึ้นเมื่อมีการกดแป้นพิมพ์กำหนดการเริ่มทำงานของโปรแกรมฝังตัว ในกรณีที่ต้องการให้โปรแกรมฝังตัวมีการทำงานอยู่ตลอดเวลาโดยทั่วไปมักออกแบบให้อาศัยการเกิดขึ้นของอินเทอร์รัพท์ 8 (INT 0008H - Timer Interrupt) เนื่องจากไทม์เมอร์อินเทอร์รัพท์นี้จะเกิดขึ้น 18.2 ครั้งทุกวินาที (3) จึงสามารถนำเอาอินเทอร์รัพท์นี้มาปลุกให้โปรแกรมฝังตัวทำงานเป็นระยะได้ตลอด นอกจากไทม์เมอร์อินเทอร์รัพท์แล้ว ดอสไอเดิลอินเทอร์รัพท์ (DOS Idle Interrupt - Keyboard Busy Loop) คือ อินเทอร์รัพท์ 40 (INT 0028H) ซึ่งถูกสร้างเป็นระยะ ๆ โดย 12 ฟังก์ชันแรกของดอสเป็นอินเทอร์รัพท์ที่นำมาใช้ประกอบกับไทม์เมอร์อินเทอร์รัพท์ได้ เนื่องจากการถูกเรียกให้ทำงานของโปรแกรมฝังตัวนั้นมิได้เกิดภายในโปรแกรมที่กำลังทำงานอยู่ แต่อยู่นอกเหนือการควบคุมของโปรแกรมที่กำลังทำงานอยู่ขณะนั้น ทำให้เกิดความสับสนระหว่างสถานะภายใน (internal state) ของโปรแกรมที่กำลังทำงานกับสถานะภายในของดอสขึ้น เช่น ขณะที่โปรแกรมกำลังส่งแฟ้มข้อมูลออกไปยังเครื่องพิมพ์โดยการเรียกใช้อินเทอร์รัพท์ที่ 33 ทันใดนั้นถูกขัดจังหวะโดยการตื่นขึ้นมาทำงานของโปรแกรมฝังตัว เมื่อโปรแกรมฝังตัวทำการเรียกใช้อินเทอร์รัพท์ 33 เพื่อทำงานของตนบ้าง ก็มักจะเกิดความผิดพลาดขึ้น เนื่องจากดอสนั้นถูกออกแบบมาเพื่อเป็น ระบบปฏิบัติการแบบงานเดียว (single-task operating system) การออกแบบชุดคำสั่งของอินเทอร์รัพท์ที่ 33 มิได้ถูกออกแบบให้สามารถรองรับการถูกขัดจังหวะขณะที่กำลังทำงานอย่างหนึ่งและไปทำงานอีกอย่างหนึ่ง เสร็จแล้วจึงกลับมาทำงานเดิมต่ออย่างถูกต้องได้ (2) ซึ่งเรียกว่า รีเอนทรานซี (reentrancy) สาเหตุที่ส่วนใหญ่แล้วดอสมีลักษณะ

นอนรีเอนทรานซี (nonreentrancy) เกิดจากเมื่อมีการเรียกใช้อินเทอร์รัพท์ 33 จากโปรแกรมฝังตัวในขณะที่ดอสกำลังทำงานซึ่งถูกเรียกผ่านอินเทอร์รัพท์ 33 ค้างอยู่ ข้อมูลต่างๆของโปรแกรมฝังตัวจะถูกบรรจุเข้าหน่วยความจำแบบสแตกภายในของดอสทับข้อมูลเดิมของโปรแกรมที่กำลังทำงานอยู่ทำให้เกิดความผิดพลาดขึ้น นอกเสียจากว่าโปรแกรมฝังตัวและโปรแกรมเดิมที่กำลังทำงานอยู่จะเรียกอินเทอร์รัพท์ของดอสฟังก์ชัน ที่มีการใช้งานหน่วยความจำแบบสแตกคนละส่วนกัน นอกจากนี้ในอินเทอร์รัพท์ 33 ฟังก์ชันใหม่ ๆ เช่น ฟังก์ชัน 51 ฟังก์ชัน 80 ฟังก์ชัน 81 ฟังก์ชัน 98 และฟังก์ชัน 100 เป็นต้น ใช้หน่วยความจำแบบสแตกของโปรแกรมที่เรียกแทนหน่วยความจำแบบสแตกภายในของดอส ทำให้การเรียกใช้ฟังก์ชันเหล่านี้ไม่เกิดปัญหาดังกล่าว (2)

การแก้ปัญหาดังกล่าวสามารถทำได้โดยการตรวจสอบว่า ขณะนั้นดอสอยู่ในภาวะที่ปลอดภัยต่อการขัดจังหวะหรือไม่ ซึ่งสามารถตรวจสอบได้จากสถานะของ อินดอสแฟลก (InDOS flag) เมื่อดอสกำลังทำงานฟังก์ชันต่าง ๆ อยู่นั้นแฟลกนี้จะมีสถานะเป็นจริง และถูกเปลี่ยนเป็นเท็จเมื่อดอสทำงานฟังก์ชันเสร็จสิ้น สามารถหาตำแหน่งของอินดอสแฟลกนี้ได้โดยการเรียกใช้อินเทอร์รัพท์ 33 ฟังก์ชัน 52 (INT 0021H function 0034H) แต่การตรวจสอบสถานะของอินดอสแฟลกอย่างเดียวไม่เพียงพอ เนื่องจากขณะที่ดอสอยู่ในสภาวะดอสพรอมท์นั้นดอสจะทำฟังก์ชัน 10 (function 000AH) อยู่ทำให้เสมือนกับว่าดอสกำลังทำงานตลอดเวลาไม่สามารถขัดจังหวะได้ กรณีนี้จะต้องตรวจสอบอินเทอร์รัพท์ที่ 40 ซึ่งถูกสร้างตลอดเวลาที่ดอสทำฟังก์ชัน 10 โปรแกรมฝังตัวจึงจะทำงานได้ในภาวะดอสพรอมท์ซึ่งปลอดภัยต่อการถูกขัดจังหวะ

สิ่งหนึ่งที่การออกแบบโปรแกรมฝังตัวต้องคำนึงถึงคือ สารสนเทศของความผิดพลาดส่วนขยาย (Extended Error Information) ซึ่งเป็นความผิดพลาดที่เกิดขึ้นขณะที่ดอสทำฟังก์ชันในภาวะปกติโปรแกรมจะสามารถตรวจสอบรับรู้ได้ แต่ถ้าโปรแกรมฝังตัวเข้าทำงานแทนโปรแกรมนั้น และเรียกให้ดอสทำฟังก์ชันของตนแล้วเกิดความผิดพลาดขึ้น สารสนเทศของความผิดพลาดนั้นจะทับสารสนเทศเดิมของโปรแกรมทำให้เกิดความผิดพลาด ทางแก้ไขคือโปรแกรมฝังตัวจะต้องเก็บสารสนเทศของความผิดพลาดส่วนขยายของโปรแกรมเดิมไว้ ก่อนที่จะทำงานของตน และคืนให้ก่อนที่โปรแกรมฝังตัวจะกลับไปสู่ภาวะรอคอย (wait state) การดึงเอาสารสนเทศของความผิดพลาดส่วนขยาย ทำได้โดยใช้บริการจากอินเทอร์รัพท์ 33 ฟังก์ชัน 89 (INT 0021H function 0059H) และใช้อินเทอร์รัพท์ 33 ฟังก์ชัน 23,818 (INT 0021H function 5D0AH) เพื่อกำหนดค่าให้สารสนเทศดังกล่าว (2)

การรับข้อมูลจากแป้นพิมพ์

การรับข้อมูลจากแป้นพิมพ์นั้น โปรแกรมประยุกต์สามารถทำได้หลายทาง ทั้งอาศัยอินเทอร์รัพท์ของดอสและของไบออส นอกจากนั้นส่วนข้อมูลของไบออสยังเก็บข้อมูลที่มีประโยชน์ในการตรวจสอบการกดแป้นพิมพ์ของผู้ใช้ เช่น ชิฟท์สเตตัส (Shift status) เป็นต้น ระบบสามารถแยกแยะตำแหน่งของแป้นพิมพ์ที่ถูกกดได้โดยอาศัยสแกนโค้ด (scan code) ซึ่งเป็นจำนวนเลขที่เฉพาะตัวของแต่ละแป้นพิมพ์

ในตำแหน่งที่ 40:1EH ของส่วนข้อมูลของไบออส เป็นตำแหน่งบัฟเฟอร์ของแป้นพิมพ์ (Keyboard buffer) ซึ่งทำให้ผู้ใช้สามารถกดแป้นพิมพ์ล่วงหน้าก่อนที่โปรแกรมจะรับข้อมูลจากแป้นพิมพ์ได้ถึง 15 ตัวอักษร เมื่อผู้ใช้กดแป้นพิมพ์คีย์บอร์ดโปรเซสเซอร์ (Keyboard Processor) จะสร้างสแกนโค้ดและเกิดอินเทอร์รัพท์ 9 ขึ้น ไบออสอินเทอร์รัพท์จะเปลี่ยนสแกนโค้ดเป็นแอสกีโค้ด (ASCII code) แล้วนำไปใส่ในคีย์บอร์ดบัฟเฟอร์ ยกเว้นการกด แป้นยกแคร่ แป้น Atl แป้น Ctrl ซึ่งชุดคำสั่งอินเทอร์รัพท์ 9 จะไม่นำไปใส่ในคีย์บอร์ดบัฟเฟอร์แต่จะทำการเปลี่ยนแปลงค่าของชิฟท์สเตตัส นอกจากนั้นอินเทอร์รัพท์ 9 จะไม่สนใจการกดแป้นพิมพ์มากกว่า 1 แป้นพร้อมกันอย่างไม่มีความหมาย หลังจากที่ปล่อยแป้นพิมพ์ภายใน 0.5 วินาทีคีย์บอร์ดโปรเซสเซอร์จะสร้างสแกนโค้ดที่ 2 ซึ่งเป็นสแกนโค้ดแรกบวกด้วย 128 แสดงว่าแป้นพิมพ์ได้ถูกปล่อยแล้ว ถ้าแป้นพิมพ์ถูกกดนานกว่า 0.5 วินาที คีย์บอร์ดโปรเซสเซอร์จะทำงานซ้ำไปเรื่อยๆ หลังจากนั้นสามารถเรียกอินเทอร์รัพท์ 22 (INT 0016H) เพื่ออ่านแอสกีโค้ดจากคีย์บอร์ดบัฟเฟอร์ นอกจากอินเทอร์รัพท์ 22 แล้วอินเทอร์รัพท์ 33 ของดอสยังมีฟังก์ชันที่สามารถใช้ในการรับค่าจากแป้นพิมพ์ได้

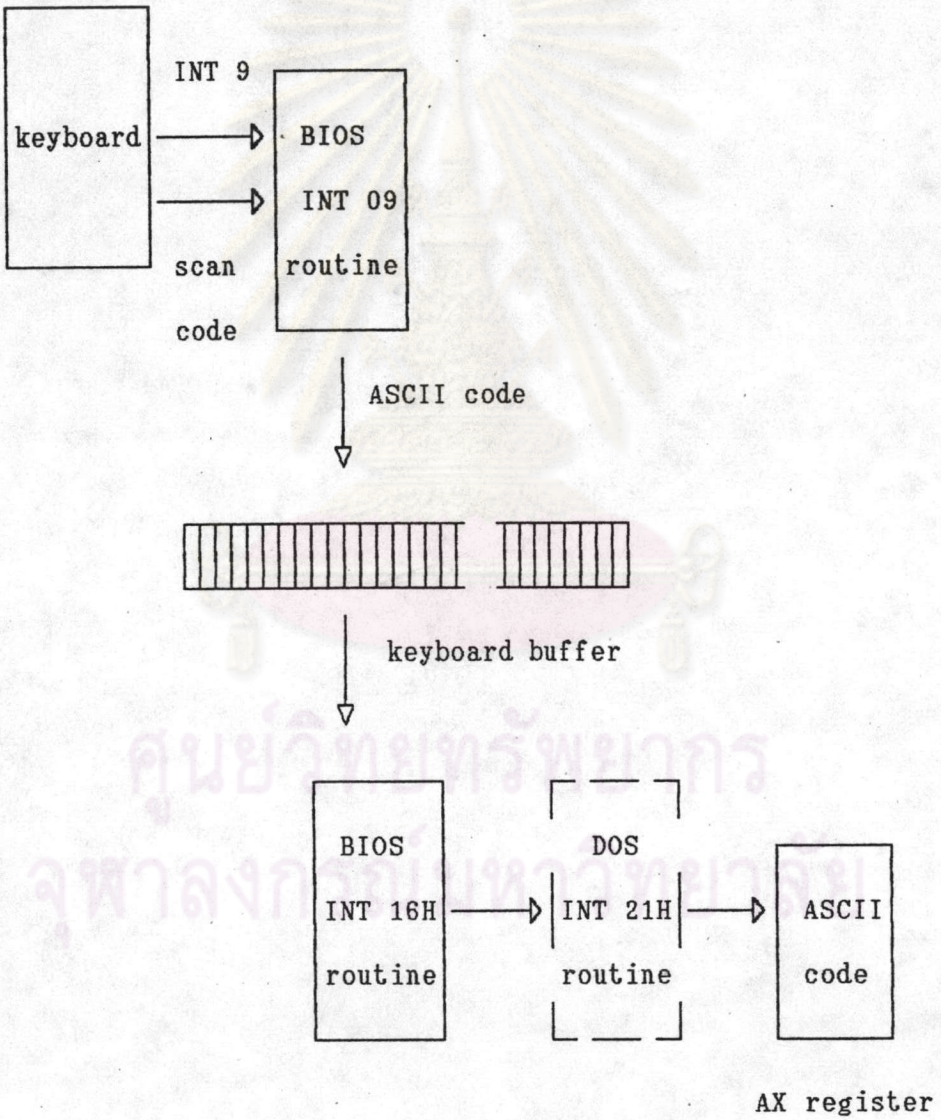
คีย์บอร์ดบัฟเฟอร์มีลักษณะเป็นแถวเรียงลำดับหรือคิว (Queue) มีรายละเอียดดังนี้

1050 คือตำแหน่งส่วนต้นของบัฟเฟอร์ปัจจุบัน (Address of current head of buffer) ซึ่งเป็นตำแหน่งของตัวอักษรที่จะถูกนำไปจากบัฟเฟอร์เป็นตัวต่อไป

1052 คือตำแหน่งส่วนท้ายของบัฟเฟอร์ปัจจุบัน(Address of current tail of buffer) เป็นตำแหน่งต่อไปที่จะถูกใส่ตัวอักษร

1054 คือคีย์บอร์ดบัฟเฟอร์ มีความยาว 32 ไบต์ซึ่งเก็บแอสกีโค้ดและสแกนโค้ดเพื่อรอการอ่านไปใช้งานภายหลัง โดยใช้ 2 ไบต์สำหรับรหัสอักษรและสแกนโค้ดของมัน

เมื่อมีการกดแป้นพิมพ์อินเทอร์รัพท์ 9 จะเพิ่มค่าตำแหน่งท้ายบัฟเฟอร์ เมื่ออินเทอร์รัพท์ 22 อ่านค่าจากบัฟเฟอร์ จะเพิ่มค่าตำแหน่งหัวบัฟเฟอร์ โดยที่บัฟเฟอร์มีลักษณะเป็นคิววงกลม (Circulary Queue) เมื่อบัฟเฟอร์ว่างหัวบัฟเฟอร์และท้ายบัฟเฟอร์จะมีค่าเดียวกัน ในขณะที่เมื่อบัฟเฟอร์เต็ม ท้ายของบัฟเฟอร์จะเก็บค่าตำแหน่งที่ต่อจากหัวของบัฟเฟอร์ การรับข้อมูลจากแป้นพิมพ์ของโปรแกรมประยุกต์แสดงไว้ในแผนภาพที่ 2



แผนภาพที่ 2 แสดงการรับข้อมูลจากแป้นพิมพ์

ที่มา Peter, A. IBM PC Assembly language and programming. (New Jersey: Prentice-Hall,Inc. , 1991), p. 169.

การพิมพ์ข้อมูลออกเครื่องพิมพ์

ในระบบปฏิบัติการดอส การพิมพ์สามารถทำได้โดยขอบริการจากอินเทอร์รัพท์ 33 หรือ อินเทอร์รัพท์ 23 (INT 0017H) คำสั่งที่ส่งไปยังเครื่องพิมพ์ประกอบด้วย คำสั่งให้ขึ้นหน้าใหม่ (Form feed) เลื่อนบรรทัด (Line feed) และกลับไปต้นบรรทัด (Carriage return) เครื่องพิมพ์สามารถแบ่งตามลักษณะการต่อเชื่อมกับคอมพิวเตอร์ได้เป็น 2 ชนิด คือ เครื่องพิมพ์แบบขนาน (parallel printer) ซึ่งรับข้อมูลครั้งละ 8 บิต และเครื่องพิมพ์แบบอนุกรม (serial printer) ซึ่งรับข้อมูลครั้งละ 1 บิต เครื่องพิมพ์จะต้องเข้าใจในสัญญาณที่ส่งจากคอมพิวเตอร์ ขณะเดียวกันคอมพิวเตอร์ก็ต้องเข้าใจสัญญาณที่ตอบกลับมาจากเครื่องพิมพ์ เช่น กระดาษหมด เป็นต้น เครื่องพิมพ์แต่ละชนิดตอบสนองต่อสัญญาณจากคอมพิวเตอร์แตกต่างกัน ตัวอย่างรหัสควบคุมการพิมพ์การพิมพ์มาตรฐานมีดังนี้

Decimal	Hex	function
08	08	ถอยหลัง
09	09	แทบตามขวาง
10	0A	เลื่อนบรรทัด
11	0B	แทบตามแนวตั้ง
12	0C	ขึ้นหน้าใหม่
13	0D	กลับไปต้นบรรทัด

การเรียกใช้อินเทอร์รัพท์ 23 เพื่อทำการพิมพ์นั้นมี 3 ฟังก์ชันในการทำงานคือ

1) อินเทอร์รัพท์ 23 ฟังก์ชัน 0 สำหรับพิมพ์ตัวอักษรออกเครื่องพิมพ์

2) อินเทอร์รัพท์ 23 ฟังก์ชัน 1 สำหรับเตรียมเครื่องพิมพ์ให้พร้อมทำการพิมพ์และตั้งกระดาษ (Initialize the Printer Port) สถานะของเครื่องพิมพ์ถูกส่งกลับมาผ่านเอเอชรีจิสเตอร์ (AH register) สามารถแปลความหมายดังนี้

019187

บิต	สาเหตุ
0	หมดเวลา (Time out)
3	การรับส่งข้อมูลผิดพลาด (Input/Output error)
4	ถูกเลือก (Selected)
5	กระดาษหมด (Out of paper)
6	รับรู้ (Acknowledged from printer)
7	ว่าง (Not busy)

ถ้าเครื่องพิมพ์อยู่ในสภาวะเปิดเครื่องพร้อม สถานะจะเป็น 10010000 หมายความว่าเครื่องพิมพ์ถูกเลือกและว่าง ซึ่งเป็นสภาวะที่สามารถส่งข้อมูลพิมพ์ได้ ในกรณีที่เครื่องพิมพ์ไม่พร้อมด้วยสาเหตุใดก็ตามบิตนั้น ๆ จะมีค่าเป็น 1 ยกเว้นบิตที่ 4 6 และ 7

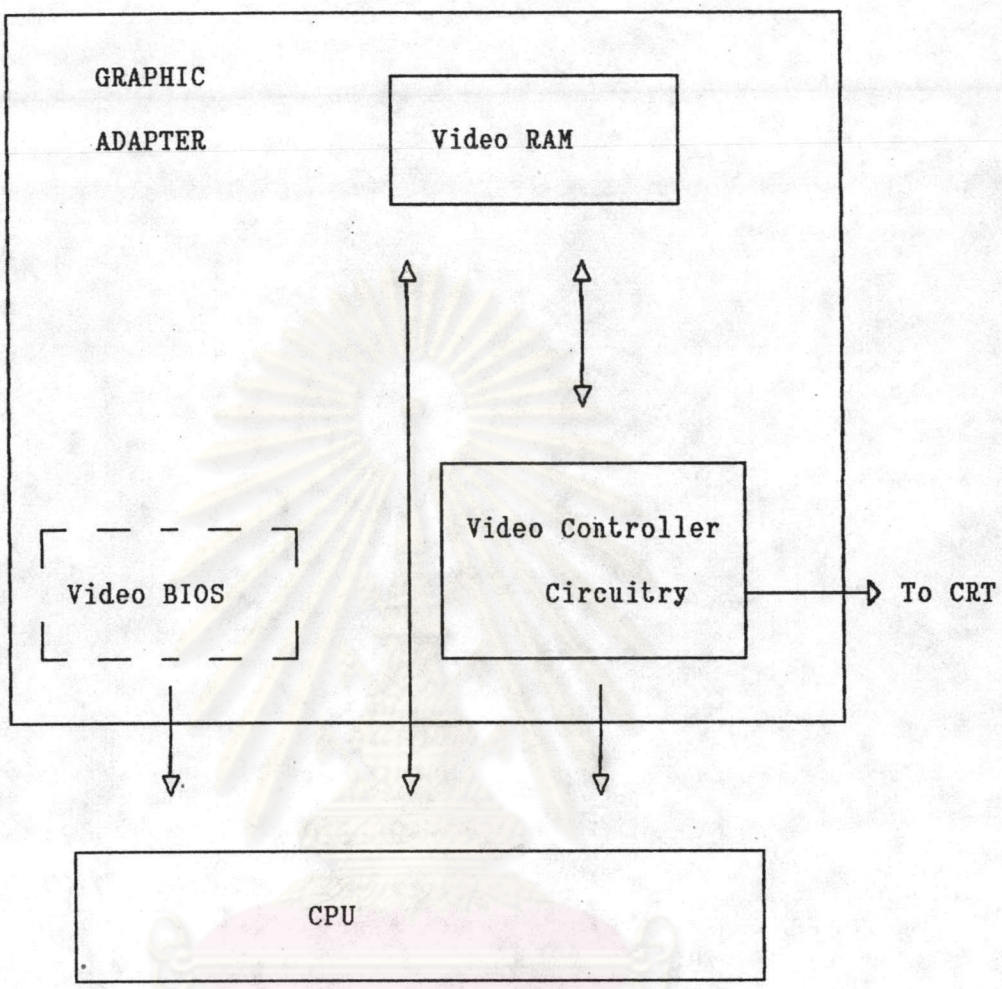
3) อินเทอร์รัพท์ 23 ฟังก์ชัน 2 สำหรับตรวจสอบสถานะของเครื่องพิมพ์ (Get Printer Port status) การส่งค่าและความหมายของสถานะเช่นเดียวกับฟังก์ชัน 1

ในการพิมพ์เมื่อมีความผิดพลาดเกิดขึ้น ไบออสไม่สามารถที่จะแจ้งให้ทราบได้โดยอัตโนมัติ โปรแกรมจะต้องทำการตรวจสอบสถานะเองขณะเริ่มทำการพิมพ์ในแต่ละครั้ง

<u>การจัดการจอภาพ</u>

การแสดงภาพบนเครื่องคอมพิวเตอร์ส่วนบุคคล ถูกควบคุมโดยวงจรแสดงผล (video adapter) วงจรแสดงผลที่สามารถแสดงได้ทั้งภาพและตัวอักษรเรียกว่า กราฟฟิก-อแดปเตอร์ (graphics adapter) ซึ่งมีส่วนประกอบที่สำคัญดังนี้ (4)

- ซีอาร์ทีคอนโทรลเลอร์ (CRT-controller - CRTC)
- วิดีโอรอมไบออส (video ROM BIOS)
- วงจรติดต่อกับโปรเซสเซอร์ (CPU interface circuits)
- วิดีโอบัฟเฟอร์ (video buffer)



แผนภาพที่ 3 แสดงส่วนประกอบของวงจรแสดงผล

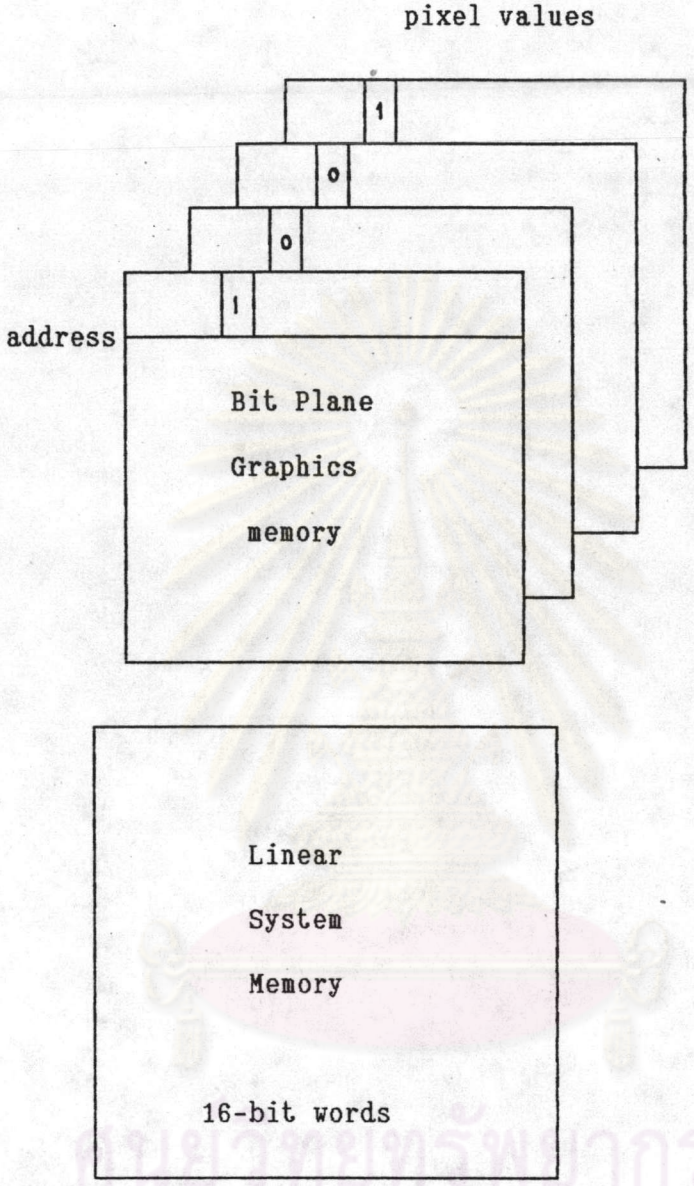
ที่มา Ed, T., and Laura, J. IBM PC and PS/2 graphic handbook. (Dubai: Micro-Tech Publication, 1990.), p. 4.

ซีอาร์ทีคอนโทรลเลอร์ ทำหน้าที่ควบคุมการทำงานของจอภาพ วิดีโอรอมไบออส เป็นที่อยู่ของโปรแกรมที่ทำหน้าที่ควบคุมการทำงานของวงจรแสดงผล ซึ่งจะแตกต่างกันในแต่ละประเภทของวงจรแสดงผล เมื่อโปรแกรมประยุกต์ต้องการแสดงผลการทำงานออกจอภาพ ก็จะเรียกชุดคำสั่งในวิดีโอรอมไบออสนี้ วงจรติดต่อกับโปรเซสเซอร์ทำหน้าที่ติดต่อโปรเซสเซอร์และวงจรแสดงผล ทำให้โปรเซสเซอร์สามารถติดต่อกับวิดีโอบัฟเฟอร์ และรีจิสเตอร์ของซีอาร์ที-คอนโทรลเลอร์ได้

วิดีโอบัฟเฟอร์เป็นหน่วยความจำที่อยู่บนวงจรแสดงผล มีตำแหน่งอยู่ในช่วงที่โปรเซสเซอร์สามารถเข้าถึงได้ ข้อมูลภาพที่ถูกแสดงออกจอภาพจะอยู่ในบัฟเฟอร์นี้ ขนาดของบัฟเฟอร์จะขึ้นกับความสามารถของวงจรแสดงผล โครงสร้างโดยทั่วไปของวิดีโอบัฟเฟอร์และหน่วยความจำปกติจะเหมือนกัน แต่วิดีโอบัฟเฟอร์มีวิธีการจัดเป็นแบบบิตเดียวที่เรียกว่า บิตเพลน (bit planes) แตกต่างจากหน่วยความจำทั่วไปที่มีการจัดเป็น 8 16 หรือ 32 บิต ตามแต่ขนาดของบัส (bus) เรียกโครงสร้างการจัดหน่วยความจำแบบนี้ว่า โครงสร้างหน่วยความจำแบบพลานาร์ (planar memory structure) (4)

เมื่อแต่ละพิกเซลของภาพต้องการเพียง 1 บิต ในการแสดงภาพ เช่น จอภาพสีเดียว (monochrome monitor) เป็นต้น เพียง 1 บิตเพลนก็เพียงพอแล้ว สำหรับจอภาพสี (color monitor) แต่ละพิกเซลจะต้องการมากกว่า 1 บิตในการแสดงภาพ บิตเพลนจึงต้องมีมากกว่า 1 ตามไปด้วย สำหรับวงจรแสดงผลที่ออกแบบให้มี 4 บิตเพลนแต่ละพิกเซลก็จะประกอบไปด้วย 4 บิตที่มีความหมายที่แตกต่างกัน ซึ่งอาจจะเป็นสีของพิกเซล แต่ละบิตถูกเก็บแยกบนแต่ละเพลน เมื่อต้องการใช้ข้อมูลเหล่านี้ในการแสดงภาพทั้ง 4 เพลนจะต้องถูกอ่านประกอบกัน ตามแผนภาพที่ 4

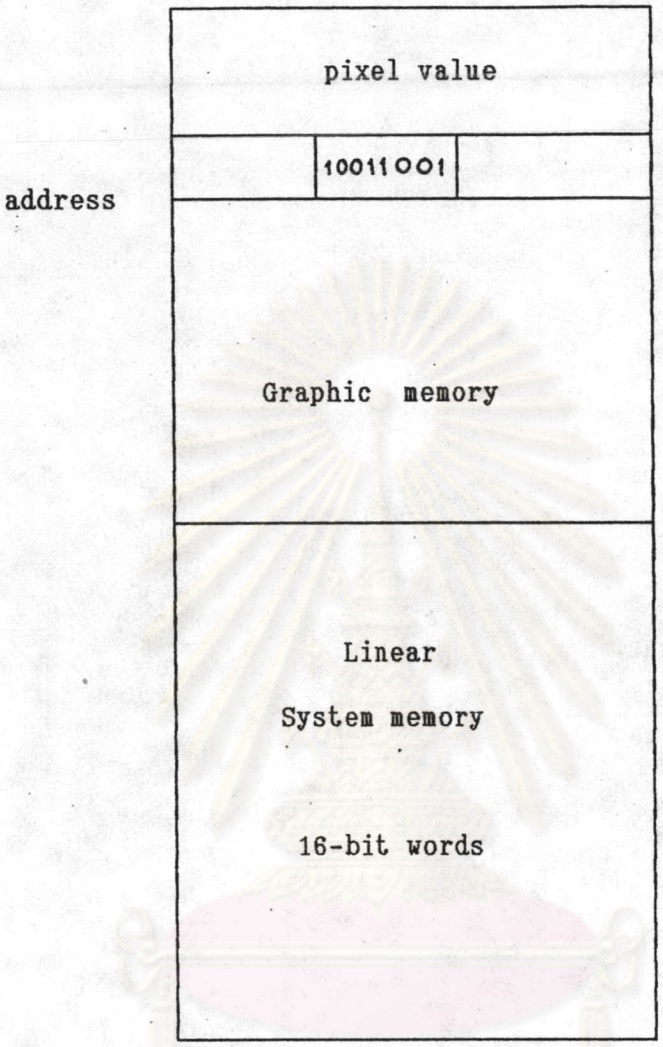
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



แผนภาพที่ 4 แสดงโครงสร้างวิดีโอบัฟเฟอร์แบบพลานาร์

ที่มา Ed, T., and Laura, J. IBM PC and PS/2 graphic handbook. (Dubai: Micro-Tech Publication, 1990.), p. 7.

อีกรูปแบบหนึ่งของการจัดวิดีโอบัฟเฟอร์มีการจัดเรียงต่อกันไปเรียกว่า โครงสร้างแบบแพกค์พิกเซล (packed pixel architecture) ซึ่งมีการจัดวิดีโอบัฟเฟอร์คล้ายกับหน่วยความจำทั่วไป โดยเก็บทุก ๆ พิกเซลเข้าเป็น 1 ไบต์คล้ายแอสกีโค้ด ตามแผนภาพที่ 5



หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

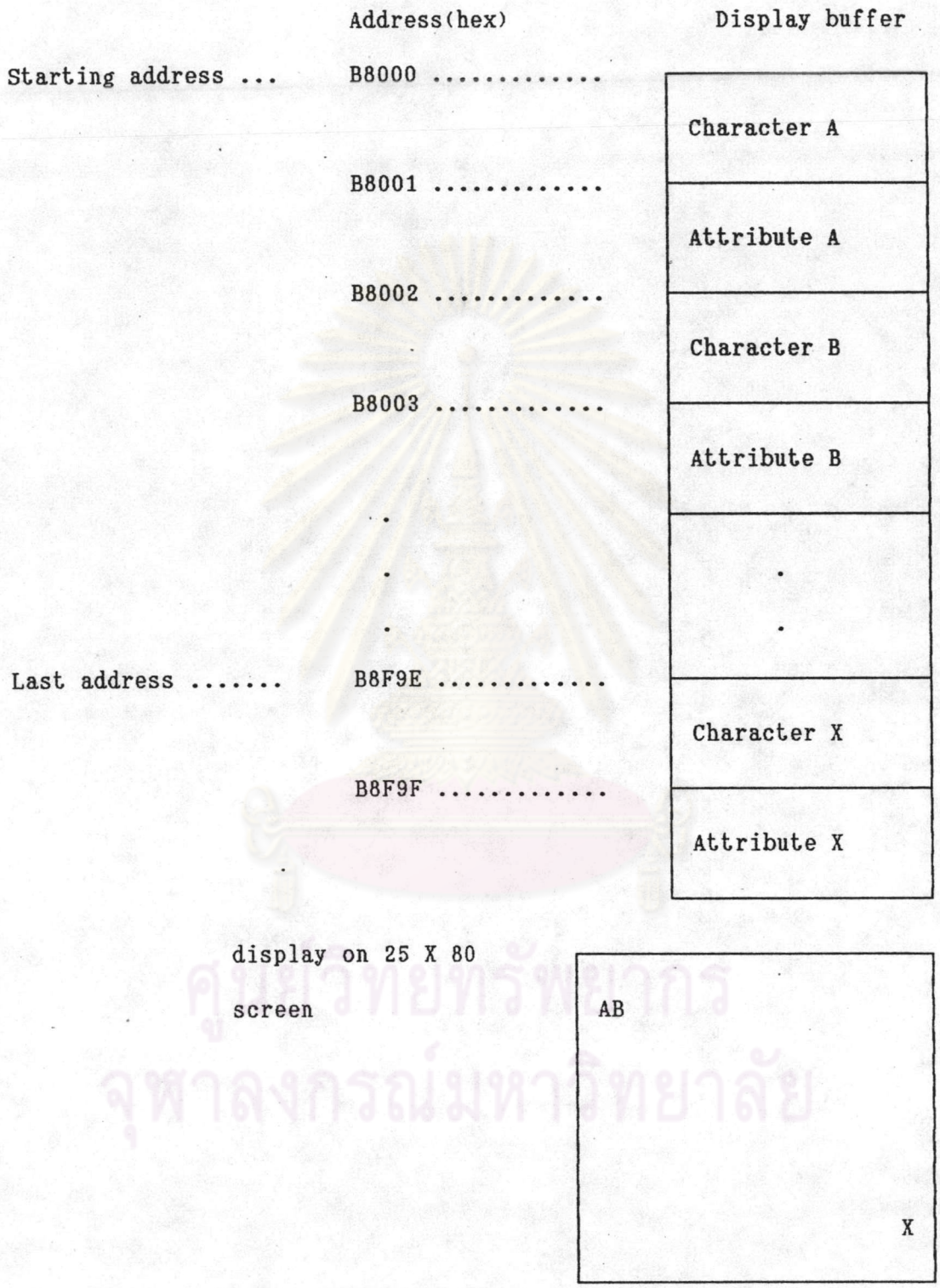
แผนภาพที่ 5 แสดงโครงสร้างวิดีโอบัฟเฟอร์แบบแพกค์พิกเซล

ที่มา Ed, T., and Laura, J. IBM PC and PS/2 graphic handbook. (Dubai: Micro-Tech Publication, 1990.), p. 7.

วงจรแสดงผลมีวิธีการในการจัดการจอภาพที่เรียกว่าโหมดการแสดงผล (display mode) ได้หลากหลาย ซึ่งจะเป็นตัวกำหนดจำนวนแถวและสดมภ์ของพิกเซล และจำนวนสีที่สามารถแสดงได้ โปรแกรมประยุกต์สามารถเปลี่ยนแปลงโหมดการทำงานผ่านรีจิสเตอร์ของวงจรแสดงผล (4)

โหมดข้อความ (text mode) เป็นโหมดการทำงานที่โปรแกรมประยุกต์ส่วนใหญ่ใช้ในการทำงาน วงจรแสดงผลสีจะต้องมีอย่างน้อย 1 โหมดข้อความซึ่งเทียบได้กับวงจรแสดงผลแบบเอ็มดีเอ (Monochrome Display Adapter - MDA) เนื่องจากการแสดงข้อความใช้ตัวอักษรไม่มากนัก ดังนั้นการบรรจุรูปแบบการเรียงจุดเป็นตัวอักษรแต่ละตัวลงในรอมบนวงจรแสดงผลจะเพิ่มประสิทธิภาพการทำงานได้ดีกว่าการสร้างตัวอักษรทุกครั้งที่ต้องการใช้ โดย 128 ตัวแรกจะเป็นตัวอักษร ส่วน 128 ตัวหลังจะเป็นอักษรภาพซึ่งมักจะถูกดัดแปลงให้เป็นตัวอักษรภาษาอื่น ในโหมดข้อความนี้วงจรแสดงผลจะกำหนดให้จอภาพสามารถแสดงตัวอักษรได้ 2,000 ตัวอักษร คือมีขนาด 80 สดมภ์และ 25 แถว ในโหมดข้อความมาตรฐานแต่ละตัวอักษรที่แสดงบนจอภาพถูกกำหนดโดยข้อมูล 2 ไบต์ ไบต์หนึ่งเก็บแอสกีโค้ด อีกไบต์เก็บลักษณะของตัวอักษร (attribute) ตัวอักษรตัวแรกจะแสดงที่มุมบนซ้ายของจอภาพ ดังแผนภาพที่ 6

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



แผนภาพที่ 6 แสดงตำแหน่งในวิดีโอบัฟเฟอร์ส่วนข้อความ

ที่มา Ed, T., and Laura, J. IBM PC and PS/2 graphic handbook. (Dubai: Micro-Tech Publication, 1990.), p. 15.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ในโหมดการแสดงผลแบบข้อความมีลักษณะของตัวอักษรที่แตกต่างกันดังนี้

- ตัวอักษรสว่างบนพื้นหลังมืด (normal)
- ตัวอักษรมืดบนพื้นสว่าง (inverse video)
- ไม่แสดง (blank)
- ขีดเส้นใต้ (underline)
- สว่าง (high intensity)
- กระพริบ (blinking)

Attribute	Bit Number						
Function	6 red	5 green	4 blue	3 highlight	2 red	1 green	0 blue
	(Background)			(Foreground)			
Normal	0	0	0	1	1	1	1
Reverse	1	1	1	1	0	0	0
nondisplay black	0	0	0	1	0	0	0
nondisplay white	1	1	1	1	1	1	1
bit 7 turns blinking on (1) or off (0)							

ตารางที่ 1 แสดงส่วนต่าง ๆ ของไบต์แสดงลักษณะตัวอักษร

ที่มา Ed, T., and Laura, J. IBM PC and PS/2 graphic handbook. (Dubai: Micro-Tech Publication, 1990.), p. 16.

นอกจากการแสดงข้อความแล้ว บางครั้งโปรแกรมประยุกต์ยังต้องการแสดงภาพ ซึ่งต้องการการควบคุมมากกว่าเดิม ซึ่งไอบีเอ็มเรียกว่าการกำหนดตำแหน่งทุกจุด (All Point Addressable mode - APA) ในโหมดแสดงภาพวงจรแสดงผลอนุญาตให้โปรแกรมจัดการทุก ๆ จุดบนจอภาพอย่างเป็นอิสระต่อกัน โหมดแสดงภาพแต่ละโหมดจะแตกต่างกันตามความละเอียดและจำนวนสี ค่าที่เก็บอยู่ในรีจิสเตอร์ของวงจรแสดงผลเป็นตัวกำหนดโหมดการทำงาน ลักษณะของพิกเซล (pixel) และลักษณะการทำงานอื่น ๆ ทุก ๆ สิ่งที่แสดงบนจอภาพในโหมดแสดงภาพรวมทั้งตัวอักษรจะถูกกำหนดลงบนวิดีโอบัฟเฟอร์ ซึ่งจะแปลความหมายภาพออกเป็นพิกเซลซึ่งอาจเป็นสว่างหรือมืด แต่ละตำแหน่งบนวิดีโอบัฟเฟอร์ควบคุมแต่ละพิกเซล เนื่องจากเป็นการแปลจากภาพเป็นบิตและจากบิตเป็นพิกเซล ดังนั้นจึงเรียกว่าการทำงานแบบบิตแมป (bit-map)

<u>วงจรแสดงผลแบบวีจีเอ</u>

ไอบีเอ็มเสนอวงจรแสดงผลแบบวีจีเอ เพื่อใช้กับเครื่องคอมพิวเตอร์แบบพีเอสทู (PS/2) และต่อมาได้ผลิตขึ้นเพื่อใช้กับคอมพิวเตอร์แบบ พีซีเอ็กซ์ที (PC/XT) และพีซีเอที (PC/AT) โดยวงจรแสดงผลแบบวีจีเอสำหรับพีซีมีคุณสมบัติหลายเพิ่มเติมขึ้นมาเช่นการขยายภาพ (Zoom) เป็นต้น ภาวะที่ไม่มีการเปลี่ยนแปลงจากโปรแกรมประยุกต์วงจรแสดงผลแบบวีจีเอจะทำงานในโหมด 18 (0012H) ซึ่งเหมาะสมกับจอภาพไอบีเอ็มพีเอสทู ที่ใช้อัตราการสแกน ที่ 31.5 กิโลเฮิร์ทซ์ ไอบีเอ็มรอมไบออสมีชุดคำสั่งที่สามารถเลือกโหมดการทำงานที่แตกต่างกัน 19 โหมดการทำงาน ซึ่งแตกต่างกันใน

- จำนวนจุดตามขวาง
- จำนวนจุดตามแนวตั้ง
- ลักษณะของข้อมูลในวิดีโอบัฟเฟอร์
- การแปลความหมายของลักษณะการแสดงภาพ

Mode	Type	Graphic Dimension	Text Dimension	Adapter
0	text,b/w	n/a	40*25	CGA/EGA/VGA
1	text,16 color	n/a	40*25	CGA/EGA/VGA
2	text,b/w	n/a	80*25	CGA/EGA/VGA
3	text,16 color	n/a	80*25	CGA/EGA/VGA
4	graphics,4 color	320*200	40*25	CGA/EGA/VGA
5	graphics,4 gray	320*200	40*25	CGA/EGA/VGA
6	graphics,2 color	640*200	80*25	CGA/EGA/VGA
7	text,b/w	n/a	80*25	monochrome
8	PCjr mode,not supported			
9	PCjr mode,not supported			
10	Pcjr mode,not supported			
11	reserved			
12	reserved			
13	graphics,16 color	320*200	40*25	EGA/VGA
14	graphics,16 color	640*200	80*25	EGA/VGA
15	graphics,2 color	640*350	80*25	EGA/VGA
16	graphics,16 color	640*350	80*25	EGA/VGA
17	graphics,2 color	640*480	80*25	VGA
18	graphics,16 color	640*480	80*25	VGA
19	graphics,256 color	640*200	40*25	VGA

ตารางที่ 2 แสดงโหมดการทำงานของวงจรแสดงผลแบบวีจีเอ

ที่มา Herbert, S. The art of C.

(California: Osborn McGraw-Hill, 1991.), p. 154.

เนื่องจากวงจรแสดงผลแบบวีจีเอใช้กับจอภาพความละเอียดสูง มีสัญญาณนาฬิกาในการทำงานความถี่สูง เมื่อทำงานในโหมด 200 เส้นจึงต้องทำการกวาด (scan) แต่ละเส้น 2 ครั้งซึ่งทำให้ภาพที่ได้มีความคมชัดขึ้นแต่ทำให้อัตราส่วนของภาพ (ratio) ผิดไป เมื่อทำงานในโหมดข้อความวงจรแสดงผลแบบวีจีเอจะทำการติดตั้งรูปแบบตัวอักษร (font) จากรอมซึ่งมี 3 ชุด 2 ชุดแรกเหมือนรูปแบบตัวอักษรที่มีอยู่ในวงจรแสดงผลแบบเอ็มดีเอ ซีจีเอ (CGA) และอีจีเอ (EGA) ชุดที่ 3 เป็นรูปแบบตัวอักษรขนาด 9 X 16 จุด จำนวนของเส้น (line) ที่กำหนดให้กับวงจรแสดงผลแบบวีจีเอมีผลต่อโหมดการทำงานและรูปแบบตัวอักษรที่สามารถใช้ได้

เพื่อให้สามารถแสดงสีได้มากกว่าเดิมวงจรแสดงผลแบบวีจีเอถูกออกแบบให้มีการติดต่อกับจอภาพเป็นแบบอนาลอก (analog connection)

สำหรับวงจรแสดงผลแบบซีจีเอและอีจีเอ ข้อมูลพิกเซลที่อยู่ในวิดีโอบัฟเฟอร์เก็บข้อมูลสี แต่ในวงจรแสดงผลแบบวีจีเอจะเก็บตำแหน่งที่ไปค้นหาข้อมูลพาเลทสี (color palette) ซึ่งอาศัยตัวเปลี่ยนดิจิตอลเป็นอนาลอก (digital-to-analog converters - DAC) เพื่อเปลี่ยนข้อมูลรหัสสีเป็นสัญญาณอนาลอกสำหรับจอภาพ ซึ่งไอบีเอ็มออกแบบให้มีตัวเปลี่ยนดิจิตอลเป็นอนาลอกขนาด 6 บิต 3 ชุดในการควบคุมสัญญาณสีแดง สีน้ำเงิน และสีเขียว ข้อมูลในตารางพาเลทเป็นค่าแอมปลิจูด (amplitudes) สำหรับตัวเปลี่ยนดิจิตอลเป็นอนาลอกทั้ง 3 ชุด ในการผสมสัญญาณสี ซึ่งสามารถผสมเป็นสีที่แตกต่างกันได้ 262,144 สี เนื่องจากที่เก็บข้อมูลสีของแต่ละพิกเซลมีขนาดจำกัดทำให้ไม่สามารถอ้างถึงสีทุก ๆ สีได้ ในโหมดความละเอียด 640 X 480 จุด แต่ละพิกเซลใช้ 4 บิตในการเก็บข้อมูลสีทำให้สามารถอ้างสีที่แตกต่างกันได้ 16 สี ในโหมดความละเอียด 320 X 480 จุด แต่ละพิกเซลใช้ 8 บิตในการเก็บข้อมูลสี สามารถอ้างสีที่แตกต่างกันได้ 256 สี (4)

ขนาดของวิดีโอบัฟเฟอร์ในวงจรแสดงผลแบบวีจีเอจะมีอย่างน้อยที่สุด 256 กิโลไบต์ซึ่งถูกจัดเป็นบิตเพลนขนาด 64 กิโลไบต์จำนวน 4 ชุด ในโหมดการแสดงผลสี (โหมดที่ 13 โหมดที่ 14 โหมดที่ 16 และโหมดที่ 18) บิตเพลนเหล่านี้ประกอบกันเป็น 16 สี ตามตารางที่ 3

- C0 พิกเซลสีน้ำเงิน
- C1 พิกเซลสีเขียว
- C2 พิกเซลสีแดง
- C3 พิกเซลความสว่าง (intensified pixel)

C3	C2	C1	C0	Color
0	0	0	0	Black
0	0	0	1	Blue
0	0	1	0	Green
0	0	1	1	Cyan (Blue Green)
0	1	0	0	Red
0	1	0	1	Magenta (Pinkish Purple)
0	1	1	0	Brown
0	1	1	1	White
1	0	0	0	Dark Gray
1	0	0	1	Light Blue
1	0	1	0	Light Green
1	0	1	1	Light Cyan
1	1	0	0	Light Red
1	1	0	1	Light Magenta
1	1	1	0	Yellow
1	1	1	1	Intensified White

ตารางที่ 3 แสดงพาเลทสี

ที่มา Ed, T., and Laura, J. IBM PC and PS/2 graphic handbook.

(Dubai: Micro-Tech Publication, 1990.), p. 25.

วงจรแสดงผลแบบวีจีเอมีรีจิสเตอร์ที่ควบคุมการทำงานอยู่จำนวนหนึ่ง ดังแสดงในตารางที่ 4

Register		Write Address	Read Address
General			
Miscellaneous Output		3C2	3CC
Feature Control		3DA/3BA	3CA
Input Status 0			3C2
Input Status 1			3DA/3BA
Sequencer			
Address		3C4	3C4
Reset	0	3C5	3C5
Clocking Mode	1	3C5	3C5
Map Mask	2	3C5	3C5
Character Map Select	3	3C5	3C5
Memory Mode	4	3C5	3C5

ตารางที่ 4 แสดงตำแหน่งพอร์ตของรีจิสเตอร์บนวงจรแสดงผลแบบวีจีเอ

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

Register		Write Address	Read Address
CRTC			
Address		3D4/3B4	3D4/3B4
Horizontal Total	0	3D5/3B5	3D5/3B5
Horizontal Display End	1	3D5/3B5	3D5/3B5
Start Horizontal Blank	2	3D5/3B5	3D5/3B5
End Horizontal Blank	3	3D5/3B5	3D5/3B5
Start Horizontal Retrace	4	3D5/3B5	3D5/3B5
End Horizontal Retrace	5	3D5/3B5	3D5/3B5
Vertical Total	6	3D5/3B5	3D5/3B5
Overflow	7		3D5/3B5
Preset Row Scan	8	3D5/3B5	3D5/3B5
Maximum Scan Line	9	3D5/3B5	3D5/3B5
Cursor Start	A	3D5/3B5	3D5/3B5
Cursor End	B	3D5/3B5	3D5/3B5
Start Address High	C	3D5/3B5	3D5/3B5
Start Address Low	D	3D5/3B5	3D5/3B5
Cursor Location High	E	3D5/3B5	3D5/3B5
Cursor Location Low	F	3D5/3B5	3D5/3B5

ตารางที่ 4 แสดงตำแหน่งพอร์ตของรีจิสเตอร์บนวงจรแสดงผลแบบวีจีเอ (ต่อ)

Register		Write Address	Read Address
CRTC			
Vertical Retrace Start	10	3D5/3B5	3D5/3B5
Vertical Retrace Low	11	3D5/3B5	3D5/3B5
Vertical Dispay End	12	3D5/3B5	3D5/3B5
Offset	13	3D5/3B5	3D5/3B5
Underline Location	14	3D5/3B5	3D5/3B5
Start Vertical Blank	15	3D5/3B5	3D5/3B5
End Vertical Blank	16	3D5/3B5	3D5/3B5
Mode Control	17	3D5/3B5	3D5/3B5
Line Compare	18	3D5/3B5	3D5/3B5
Graphic Controller			
Graphic Address		3CE	3CE
Set/Reset	0	3CF	3CF
Enable Set/Reset	1	3CF	3CF
Color Compare	2	3CF	3CF
Data Rotate	3	3CF	3CF

ตารางที่ 4 แสดงตำแหน่งพอร์ตของรีจิสเตอร์บนวงจรแสดงผลแบบวีจีเอ (ต่อ)

Register		Write Address	Read Address
Graphic Controller			
Read Map Select	4	3CF	3CF
Mode	5	3CF	3CF
Miscellaneous	6	3CF	3CF
Color Don't Care	7	3CF	3CF
Bit Mask	8	3CF	3CF
Attribute			
Address		3C0	3C1
Palette	0-F	3C0	3C1
Mode Control	10	3C0	3C1
Overcsan Color	11	3C0	3C1
Color Plane Enable	12	3C0	3C1
Horizontal Pixel Planning	13	3C0	3C1
Color Select	14	3C0	3C1

ตารางที่ 4 แสดงตำแหน่งพอร์ตของรีจิสเตอร์บนวงจรแสดงผลแบบวีจีเอ (ต่อ)

ที่มา Ed, T., and Laura, J. IBM PC and PS/2 graphic handbook. (Dubai: Micro-Tech Publication, 1990.), p. 38.

รูปแบบการจัดเก็บข้อมูลในแฟ้มข้อมูลรูปภาพแบบพีซีเอ็กซ์

รูปแบบการจัดเก็บข้อมูลในแฟ้มข้อมูลรูปภาพที่ใช้กันแพร่หลายมีอยู่มากมายหลายแบบ เช่น

CGM (Computer Graphic Metafile)

CUT (Cut File)

IMG (Bit Image)

PIC (Picture File)

PCX (PC Paintbrush File)

SCR (Screen File)

TIFF (Tag Image File Format)

ซึ่งแต่ละแบบเป็นรูปแบบการจัดเก็บข้อมูลภาพของโปรแกรมกราฟฟิกแต่ละโปรแกรม ซึ่งมีความสามารถและความเหมาะสมกับลักษณะงานที่แตกต่างกัน ในที่นี้จะขอกล่าวถึงรายละเอียดของรูปแบบการเก็บแบบพีซีเอ็กซ์

รูปแบบการจัดเก็บแบบพีซีเพนท์บรัช (PC Paintbrush - PCX) ออกแบบเพื่อใช้กับโปรแกรมพีซีเพนท์บรัชซึ่งเป็นโปรแกรมวาดภาพระบายสีของบริษัทแซดซอฟต์ (Zsoft) ภายหลังจากการพัฒนาต่อเนื่องกัน รูปแบบการเก็บแบบพีซีเอ็กซ์ในปัจจุบันสามารถจัดเก็บได้หลายแบบทั้งแบบสีเดียว 16 สีจนถึง 256 สี ทุกรูปแบบจะมีส่วนหัวของแฟ้มข้อมูล (PCX file header) ความยาว 128 ไบต์เก็บรายละเอียดของภาพภายในแฟ้มข้อมูล รายละเอียดของส่วนหัวของแฟ้มข้อมูลแสดงในตารางที่ 5

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

BYTE	ITEM	SIZE	COMMENTS
0	Manufacturer	1	10 = Zsoft.PCX
1	Version	1	0 = Version 2.5 2 = Version 2.8 with palette 3 =Version 2.8 without palette 5 = Version 3.0
2	Encoding	1	1 = PCX run length encoding
3	Bits per pixel	1	Number of bits/pixel/plane
4	Window	8	Picture dimensions (Xmin,Ymin)-(Xmax,Ymax)
12	HRes	2	Horizontal resolution of creating device
14	VRes	2	Vertical resolution of creating device

ตารางที่ 5 แสดงโครงสร้างส่วนหัวของแฟ้มข้อมูลแบบพีซีเอ็กซ์

BYTE	ITEM	SIZE	COMMENTS
16	ColorMap	48	Color palette setting
64	Reserved	1	
65	NPlanes	1	Number of color planes
66	Bytes per Line	2	Number of bytes per scan line per color plane (always even)
68	Palette Information	2	1 = Color or black and white 2 = Gray scale
70	Filler	58	Pad out header

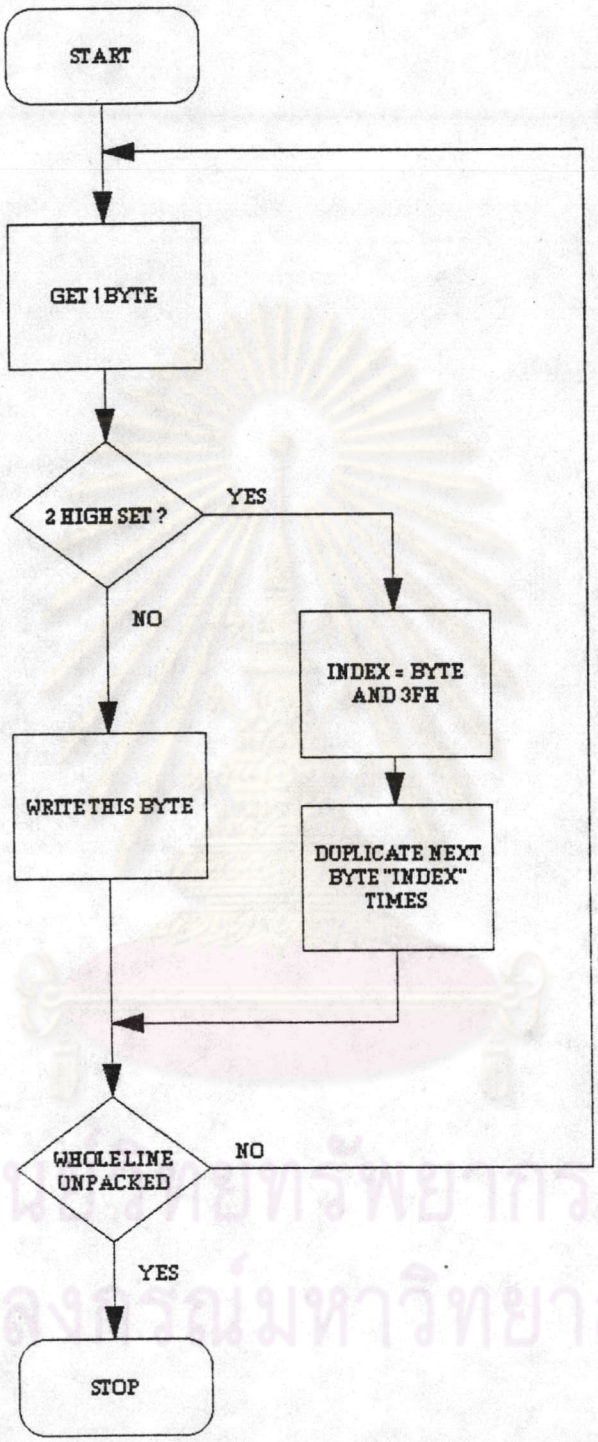
ตารางที่ 5 แสดงโครงสร้างส่วนหัวของแฟ้มข้อมูลแบบพีซีเอ็กซ์ (ต่อ)

ที่มา Ed, T., and Laura, J. IBM PC and PS/2 graphic handbook. (Dubai: Micro-Tech Publication, 1990.), p. 182.

ส่วนที่เหลือของแฟ้มข้อมูลจะเก็บข้อมูลรูปภาพที่ผ่านการเข้ารหัสแบบที่เรียกว่า การเข้ารหัสแบบพีซีเอ็กซ์รันเลนจ์ (PCX run-length limited coding algorithm) ซึ่งประกอบด้วยชุดของข้อมูลเป็นชุด ๆ ต่อกันไป ในแต่ละชุดประกอบด้วย 2 ส่วนคือ คีย์ไบต์ (key byte) และดาต้าไบต์ (data byte) ในกรณีที่ 2 บิตบนของคีย์ไบต์มีค่าเป็นจริงอีก 6 บิตที่เหลือของคีย์ไบต์จะแสดงจำนวนไบต์ที่ซ้ำกันของไบต์ข้อมูลที่ตามมา เนื่องจำนวนไบต์ที่ซ้ำกันถูกระบุด้วย 6 บิตเท่านั้น จำนวนสูงสุดที่จะซ้ำกันได้คือ 63 ไบต์ กรณีที่ชุดข้อมูลดังกล่าวมีซ้ำกันมากกว่า

63 ไบต์จะต้องใช้มากกว่า 1 ชุดข้อมูลในการเก็บ กรณีที่ 2 บิตบนของคีย์ไบต์ไม่มีค่าเป็นจริง แสดงว่าข้อมูลชุดนี้ไม่มีจำนวนซ้ำ คีย์ไบต์จะเก็บข้อมูลภาพและไม่มีดาต้าไบต์สำหรับข้อมูลชุดนี้ ข้อมูลรูปภาพจะถูกเข้ารหัสด้วยวิธีการนี้ไปจนหมดแต่ละแถว ความยาวของแต่ละแถวถูกระบุอยู่ในส่วนหัวของแฟ้มข้อมูล (5) แผนภาพที่ 7 แสดงการถอดรหัสแฟ้มข้อมูลรูปแบบการเก็บแบบพีซีเอ็กซ์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



แผนภาพที่ 7 แสดงวิธีการถอดรหัสแฟ้มข้อมูลภาพรูปแบบการเก็บแบบพีซีเอ็กซ์ ที่มา Steve, R. Bit-Mapped Graphics. (California : McGraw-Hill, 1990.), p. 95.

รูปแบบการเก็บแบบพีซีเอ็กซ์สามารถแยกได้เป็น 3 กลุ่มตามลักษณะสี กลุ่มแรกคือแฟ้มข้อมูลภาพสีเดียว (Monochrome PCX files) ซึ่งวิธีการถอดรหัสแฟ้มข้อมูลที่กล่าวมาเพียงพอแล้วสำหรับการแสดงภาพชนิดนี้ กลุ่มที่ 2 คือแฟ้มข้อมูลที่เก็บภาพ 16 สีหรือน้อยกว่านั้น กลุ่มสุดท้ายคือแฟ้มข้อมูลที่เก็บภาพ 256 สี เนื่องจากข้อจำกัดของพาเลทสี แฟ้มข้อมูลแบบที่ 2 สามารถเก็บสีได้ 2 สี 4 สี 8 สี หรือ 16 สี กรณีที่ต้องการสีมากกว่านี้จะต้องใช้รูปแบบการเก็บแบบพีซีเอ็กซ์ 256 สี ในการทำความเข้าใจการเก็บสีของรูปแบบการเก็บแบบพีซีเอ็กซ์จะต้องอ้างถึงการแสดงสีของวงจรแสดงผลแบบ อีจีเอ หรือ วีจีเอ ซึ่งเป็นดังนี้

การเกิดสีบนจอภาพมีลักษณะเป็นการสังเคราะห์สีแบบเสริมกัน (additive color synthesis) บนจอภาพสีดำ มีสีหลัก 3 สีคือ สีแดง สีเขียว และสีน้ำเงิน ทำให้มีอีกชื่อหนึ่งว่า อาร์จีบี (RGB) สีที่เกิดขึ้นบนจอภาพเกิดจากหลักการสำคัญที่เราสามารถกำหนดให้แต่ละพิกเซลสามารถแสดงความเข้มของแสงแดง แสงเขียว และแสงน้ำเงินที่แตกต่างกันได้โดยอาศัยพาเลทสีทำหน้าที่เป็นตารางเทียบสี แต่ละช่องของพาเลทสีจะประกอบด้วย 3 ไบต์ โดยแต่ละไบต์เก็บค่าความเข้มของสีหลัก (percentage of color) แต่ละสี และแต่ละพิกเซลก็จะมีข้อมูลที่แสดงตำแหน่งช่องของพาเลทสีที่เก็บข้อมูลสีของพิกเซลนั้น สำหรับวงจรแสดงผลแบบอีจีเอจะแสดงสีได้พร้อมกัน 16 สีจากจำนวนสีทั้งหมด 64 สี วิดีโอบัฟเฟอร์สำหรับวงจรแสดงผลแบบอีจีเอจะประกอบไปด้วยเพจ (page) ขนาด 32 กิโลไบต์จำนวน 4 เพจ แต่ละเพจคือเพลน (plane) ที่เมื่ออาศัยข้อมูลแต่ละบิตของทุกเพลนร่วมกันแล้วจะสามารถระบุสีของพิกเซลนั้นได้ โดยบิตบนเพลนที่ 1 สัมพันธ์กับบิตที่ 4 ของดัชนีที่ระบุตำแหน่งในพาเลทสี ตัวอย่างเช่น พิกเซลหนึ่งบนจอภาพมีค่าสีบนเพลนที่ 1 และ 2 เป็น 1 พิกเซลนี้จะมีสีเป็น 12 (จากการรวมกันของ 0008H กับ 0004H) โดยจะแสดงสีออกมาเป็นเช่นใดขึ้นกับว่า พาเลทสีตำแหน่งที่ 12 ได้กำหนดความเข้มของสีหลัก ให้ผสมกันออกมาเป็นสีใด

การเก็บภาพสีในรูปแบบการเก็บแบบพีซีเอ็กซ์จะแยกแต่ละเพลนออกจากกัน โดยเก็บข้อมูลในเพลนแรกจนหมดความยาวของรูปแล้ว จึงเริ่มเก็บความข้อมูลของเพลนที่ 2 เมื่อครบ 4 เพลนแล้วจึงเริ่มเก็บข้อมูลเพลนแรกของแถวต่อมา ดังนั้นการถอดรหัสแฟ้มข้อมูลก็ทำโดยหลักการเดียวกัน