



ปีการศึกษา 2530

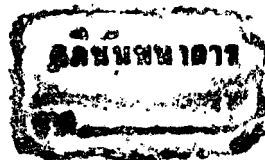
GRAPHIC DISPLAY SYSTEM

โดย

นายเสน่ห์ พุ่มเอี่ยม
นายพัฒนพงศ์ พันพรม

อาจารย์ที่ปรึกษา

อ. สุธชินันท์ ปริมาธิกุล



ปริญญาโทปีการศึกษา 2530

เรื่อง GRAPHICS DISPLAY SYSTEM

ผู้จัดทำ

1. นายเสน่ห์ พุ่มไฉยม

2. นายพัฒนพงศ์ พันพรม



อาจารย์ที่ปรึกษา

(อ. สุธินันท์ ประมาธิกุล)

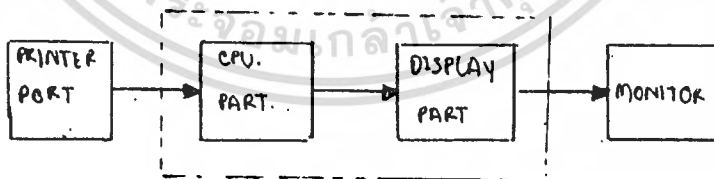
บทคัดย่อ

โครงการนี้มีชื่อว่า "GRAPHIC DISPLAY SYSTEM" ซึ่งเป็นหนึ่งในสี่โครงการดังนี้

1. AUTOMOMILE MEASURING SYSTEM
2. SPEECH SYNTHESIZER
3. CENTRAL PROCESSING UNIT
4. GRAPHIC DISPLAY SYSTEM

โครงการนี้เป็นส่วนสุดท้ายของทั้งหมด คือ จะรับข้อมูลจากโครงการที่ 3 มาแสดงผลทางจอภาพ ซึ่งลักษณะการ INTERFACE ระหว่างโครงการนี้กับโครงการที่ 3 เป็นแบบ PRINTER PORT ซึ่งเป็น PORT มาตรฐาน หลักการทำงานคือโครงการที่ 3 จะมองโครงการนี้เสมือนเป็น PRINTER ตัวหนึ่ง ดังนั้นโครงการนี้จึงสามารถนำไปประยุกต์ใช้ในงานอื่นๆ ได้อีกด้วย

โครงการนี้จะถูกแบ่งออกเป็น 2 ส่วน คือ ส่วน CPU (ส่วนควบคุม) และส่วนแสดงผล ดัง BLOCK DIAGRAM ข้างล่างนี้



เราสามารถนำโครงการนี้ไปต่อกับ PRINTER PORT ของ COMPUTER IBM PC ได้ เมื่อเราต้องการส่งข้อมูลออกจาก PRINTER ก็จะมาปรากฏทางจอภาพใหม่แทน และเมื่อเรากดปุ่ม CTRL-P (สั่งให้ PRINTER พิมพ์ข้อมูลทุกอย่างที่ปรากฏบนจอภาพ) เราก็จะได้จอภาพ 2 เครื่องบน COMPUTER เพียงเครื่องเดียว และยกเลิกโดยกดปุ่ม CTRL-P อีกครั้งหนึ่ง (TOGGLE)

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 การทำงานของจอภาพ	2
บทที่ 3 การทำงานของ CRTC 6845	11
บทที่ 4 การแสดงภาพอักษร	26
บทที่ 5 หลักการออกแบบ	35
บทที่ 6 สรุปผลและวิจารณ์	48
กิตติกรรมประกาศ	59
หนังสืออ้างอิง	60



บทที่ 1

บทนำ

ปริญญานิพนธ์นี้จะกล่าวถึงทฤษฎีและการทำงานของวงจรในโครงงานนี้โดยจะแบ่งเนื้อหาออกเป็นส่วน ๆ ดังนี้

- หลักการทำงานของจอภาพ จะกล่าวถึงแบบของจอภาพ, หลักการทำงาน, การแลกนในแบบต่าง ๆ, การเชื่อมต่อระหว่าง CRT CONTROLLER และ CRT MONITOR
- หลักการทำงานของ CRTIC 6845 จะกล่าวถึงส่วนประกอบภายใน, ขาสัญญาณ, การเชื่อมต่อกับคาแรคเตอร์เจนเนอเรเตอร์, การโปรแกรมรีเจสเตอร์ภายใน
- การแสดงภาพอักขระ จะกล่าวถึงวิธีสร้างภาพบนจอภาพ, การออกแบบ-PATTERN ของอักขระต่าง ๆ, การจัดการหน่วยความจำจอภาพ
- หลักการออกแบบวงจร จะกล่าวถึงวิธีการและหลักการออกแบบในทุกขั้นตอน พร้อมทั้งวิธีการนำไปประยุกต์ใช้งานอื่น ๆ

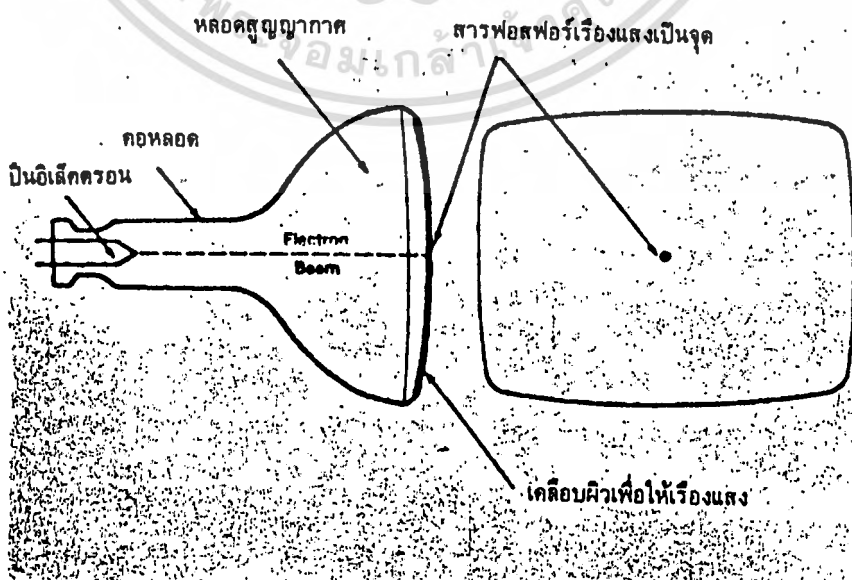
บทที่ 2

หลักการทํางานของ CRT

2.1 การทํางานของ CRT

CRT [CATHOD RAY TUBE] เป็นอุปกรณ์พื้นฐานที่ใช้แสดงผล ซึ่งปกติเราจะรู้จักกันดีในนามว่าโทรทัศน์ แต่สิ่งที่แตกต่างระหว่าง CRT MONITOR ที่ใช้ในงานทางคอมพิวเตอร์กับโทรทัศน์คือ สำหรับจอทีวี นั้นมิได้ถูกออกแบบใช้งานเฉพาะทางด้านคอมพิวเตอร์ดังนั้นคุณภาพของตัวอักษรที่แสดงจึงไม่ดีนัก แต่ก็ราคาถูก

อุปกรณ์ทางจอภาพประกอบขึ้นด้วย หลอดภาพ ขดลวดหลอดภาพ (YOKE) เครื่องขยายสัญญาณภาพ ซึ่งหลอดภาพนี้จะเคลือบไว้ด้วยสารเรืองแสงซึ่งนิยมใช้สาร PHOSPHER แบบ P31 (Non-Glare) ที่คอหลอดมีปืนอิเล็กตรอนเพื่อสร้างลำอิเล็กตรอนอิสระ ลำอิเล็กตรอนนี้จะถูกเร่งความเร็วโดย VOLTAGE ANODE ไปยังจอภาพเมื่ออนุภาคอิเล็กตรอนชนกับสารเรืองแสงที่จอภาพจะเกิดจุดสว่างขึ้น และจะคงความสว่างอยู่ชั่วขณะหนึ่ง ซึ่งเป็นคุณสมบัติของสารเรืองแสง เรียกว่า PERSISTANCE โดยจะคงอยู่ประมาณ 20-50 มิลลิวินาที



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้เผยแพร่ลงสื่อใดๆ ทั้งสิ้น รูปที่ 2.1 แสดงถึงโครงสร้างภายใน CRT ทุกครั้งที่มีการนำไปใช้

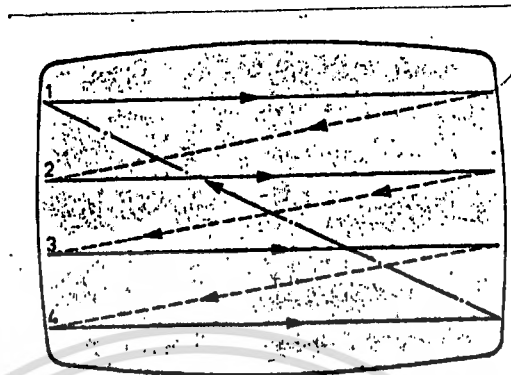
VOLTAGE ANODE ที่ป้อนให้จะมีค่าอยู่ระหว่าง 12-15 กิโลโวลต์ ลำโวลตรอนที่ได้มานี้จะถูกปรับให้แคบลงโดย ELECTRONIC LENS SYSTEM และเราสามารถปรับการเบี่ยงเบนของลำโวลตรอนได้ที่ YOKE โดยเมื่อยังไม่มีการป้อนกระแสไฟฟ้าเข้าเส้นลวดวงกลม (YOKE) ลำโวลตรอนที่ออกมาจะเป็นเส้นตรง ถ้าต้องการให้จุดที่สว่างบนจอภาพเคลื่อนที่ได้ ก็สามารถทำได้โดยป้อนกระแสไฟฟ้าเข้าที่ขดลวดหลอดภาพทำให้เกิดสนามแม่เหล็กขึ้น ซึ่งจะมีผลทำให้ลำโวลตรอนเบี่ยงเบนหรือเคลื่อนที่ได้

ความเข้มของจุดสว่างนี้ขึ้นอยู่กับจำนวนอนุภาคโวลตรอนอิสระที่ยิงมา ซึ่งก็ขึ้นอยู่กับระดับแรงดันของสัญญาณภาพ ดังนั้นเราสามารถควบคุมความสว่างได้โดยสัญญาณภาพ ถ้าจำนวนอนุภาพมากก็จะสว่างมาก ถ้าน้อยก็สว่างน้อย แต่นิยมแบ่งเป็น 3 ระดับ คือ สว่างมากเรียกว่าขาว เมื่อมืดที่สุดเรียกว่าดำ และปานกลางเรียกว่าเทา โดยปกติจะแบ่ง CRT ออกเป็น 2 ประเภท คือ

1. REFRESH CRT ภาพที่ปรากฏบนจอจะถูก Refresh ตลอดเวลา เพื่อภาพ Stable นิยมใช้ในทาง Text Processing
2. STORAGE CRT มีความสามารถที่จะรักษาภาพที่ปรากฏบนจอให้คงอยู่โดยไม่ต้องมีการ Refresh ตลอดเวลา ใช้ในการแสดงผลทาง Graphics เช่น Charts , Line pictures และอื่น ๆ แต่มีข้อเสียตรงที่ว่ามรความสว่างน้อยกว่าและกินไฟมากกว่า ซึ่งในที่นี้จะกล่าวถึงแบบ Refresh CRT ที่ใช้ในการทดลอง

การเคลื่อนที่ของลำโวลตรอนจะถูกควบคุมให้เคลื่อนที่ไปในจังหวะที่ถูกต้องทั้งในแนวนอนและแนวตั้งของภาพที่เรียกว่าการสแกน (BEAM SCANNING) โดยการสแกนนี้ควบคุมโดยอินพุต 2 อัน ขดลวดหักเหแนวนอน (HORIZONTAL YOKE) จะมีค่าเหนี่ยวนำ 13 มิลลิเฮนรี่ ใช้กระแส 1000 มิลลิแอมป์ และขดลวดหักเหแนวตั้ง (VERTICAL YOKE) จะมีค่าเหนี่ยวนำประมาณ 40 มิลลิเฮนรี่ ใช้กระแส 30 มิลลิแอมป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



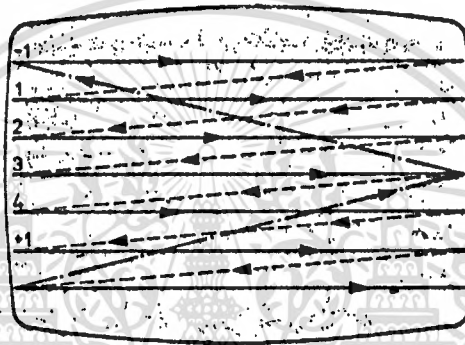
รูปที่ 2.2 แสดงเส้นสแกนไลน์ของจอภาพ

การสแกนจะเริ่มต้นขึ้นโดยลำโพงทำให้จุดสว่างเคลื่อนที่จากซ้ายมือด้านบนของจอภาพไปทางขวามือในทางแนวนอน ซึ่งเมื่อไปถึงตำแหน่งขวามือสุดก็จะถูกเบนต่ำลงเล็กน้อย แล้วก็กลับไปตั้งต้นใหม่ทางซ้ายมือเพื่อเคลื่อนที่มาจากขวามือในทางแนวนอนอีก เป็นเช่นนี้เรื่อย ๆ จนกระทั่งจุดสว่างไปถึงตำแหน่งขวามือล่างสุดของจอภาพก็เป็นอันเสร็จสิ้นการสแกนภาพหนึ่งภาพซึ่งเรียกว่าหนึ่งเฟรม หลังจากนั้นลำโพงอิเล็กตรอนก็จะกลับไปตั้งต้นทางซ้ายมือบนสุดของจอภาพอีก ระยะเวลาที่แสดงภาพ (แสดงด้วยเส้นทึบ) และระยะที่เรียกว่า RETRACE (แสดงด้วยเส้นประ) เรียกว่า 1 SCAN LINE TIME RETRACE TIMES นั้นเป็นช่วงที่ปืนอิเล็กตรอนหยุดยิง จึงเป็นช่วงที่ให้ภาพจุดดำ เรียกว่า BLANKING หรือ HORIZONTAL BLANKING

โดยที่การสแกนและสัญญาณต่าง ๆ ทางด้านแนวนอนหรือ HORIZONTAL นี้ ปกติจะให้ยาวกว่าขนาดของจอภาพ และจำนวนของจุด เลเยอร์นอกจอภาพเรียกว่า สกรีนโอเวอร์สแกน [SCREEN OVER SCAN] ดังนั้น ส่วนที่ต้องการจะแสดงบนจอจะต้องให้มีขนาดสั้นกว่าความกว้างของจอภาพ ความยาวของเวลาในการแสดงภาพจะน้อยกว่าความกว้างของลำโพงอิเล็กตรอนทั้งหมดที่สแกนทางแนวนอน เพราะว่าจะรวมส่วนที่เป็นจุดมืดไว้ด้วยนั่นเอง ส่วนช่วงเวลาที่ลำโพงอิเล็กตรอนสลับกลับไปตั้งต้น SCAN ใหม่ที่ด้านบนทางซ้ายมือเรียกว่า VERTICAL RETRACE -

TIME หรือ VERTICAL BLANKING

ในรูป 2.2 นั้นได้แสดงเส้นสแกนไลน์ไว้เพียง 4 เส้น ซึ่งจริง ๆ แล้วจะต้องมีจำนวนที่มากกว่านี้มาก (312เส้น) หากให้เวลาของ VBLNK เป็น 2 สแกนไลน์ ระบบจอภาพนี้ก็จะเป็ระบบ 6 เส้น (สมมุติ)



รูป 2.3 แสดงเส้นสแกนไลน์ในช่วง VBLANK

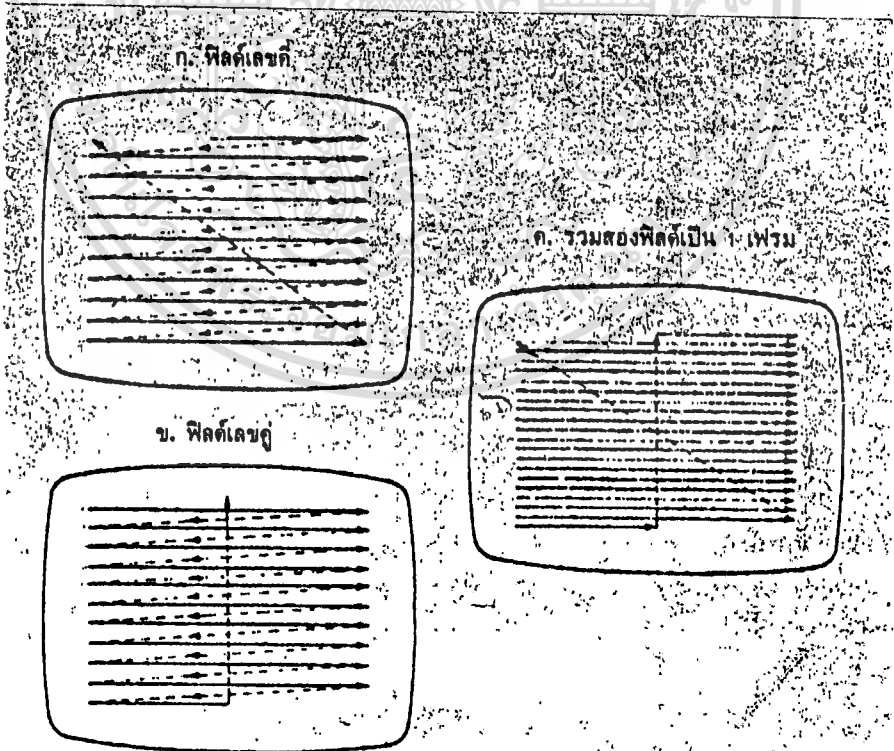
รูปที่ 2.3 แสดงให้เห็นว่าการกลับมายังจุดเริ่มต้นของเส้นที่ 1 ความจริงไม่ได้กลับมาตรง ๆ เหมือนกับรูปที่ 1-2 หากยังมีเส้นที่ +1 และเส้นที่ -1 อีก ซึ่งต้องบอกก่อนว่าเส้นที่ +1 และ -1 ที่ยกตัวอย่างมานี้เพียง 2 เส้นนั้นยังน้อยไป ความจริงแล้วยังมีมากกว่านี้มาก ทั้งนี้เพราะในระยะเวลาที่ยังประกอบด้วยเส้นสแกนไลน์ ตามปกติการสแกนไลน์ด้วยจุดภาพดำและเป็นเวลาคาบเพื่อการซิงค์สัญญาณ (คิดเป็นสแกนไลน์)

การสแกนดังที่กล่าวมาแล้วเราเรียกว่าเป็นแบบ NON-INTERLACE เพื่อลดอาการกระพริบของภาพเราจะสแกนภาพแบบ INTERLACE MODE ซึ่งวิธีนี้ใช้ในการส่งโทรทัศน์ โดยกำหนดให้ภาพหนึ่งเฟรม (Frame) ประกอบด้วยภาพหนึ่ง 2 Fields คือ Even Fields และ Odd Fields และเริ่มด้วยการสแกนภาพหนึ่ง

ฟิล์มเส้นคี่ก่อนเมื่อเสร็จสิ้นถึงตำแหน่งขวามือล่างสุดของจอภาพ แล้วจึงกลับไปตั้งต้นไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหม่ทางซ้ายมือบนสุดของจอ แล้วเริ่มต้นสแกนภาพนิ่งอันดับอื่นถัดไปใหม่ และในการสแกนภาพนิ่งหนึ่งภาพหรือหนึ่งเฟรมจึงประกอบด้วย การสแกนภาพนิ่งด้วยฟิลด์เส้นคี่ และการสแกนภาพนิ่งด้วยฟิลด์เส้นคู่

ดังนั้นในกรณีของการสแกนแบบ INTERLACE MODE ถ้าหากว่า VERTICAL SWEEP FREQUENCY เท่ากับ 60 เฮิรตซ์ จะต้องทำการ Refresh ที่ความถี่ 30 เฮิรตซ์ ซึ่งการ refresh ที่ 30 เฮิรตซ์ จะมีผลทำให้เกิดอาการกระพริบของจอภาพ ซึ่งแก้ไขได้โดยใช้สาร PHOSPHOR ที่มีค่า PERSISTANCE สูง และถ้าภาพนิ่งหนึ่งเฟรมใช้ HORIZONTAL LINE เท่ากับ 525 เส้นแล้ว (สำหรับมาตรฐานอเมริกา ส่วนมาตรฐานยุโรปใช้ 625 เส้น) ในหนึ่งฟิลด์ก็จะใช้ 262.5 เส้นต่อหนึ่งฟิลด์



แต่ในกรณีของจอภาพ ซีอาร์ที CRT ที่ใช้แสดงผลในระบบไมโครคอมพิวเตอร์นั้นไม่ใช้วิธีการสแกนแบบอินเทอร์ลैทซ์ แต่จะใช้การสแกนแบบตรงซึ่งจะสแกนทีละแถว 312 เส้น หรือมากกว่านี้ได้ แต่ละเฟรมหรือภาพ จะใช้อัตราความเร็ว 50 เฮิร์ต และใช้สัญญาณซิงค์ 2 สัญญาณ คือ สัญญาณซิงค์ทางแนวนอน และ สัญญาณซิงค์ทางแนวตั้ง เพื่อควบคุมการเคลื่อนไหวของลำอิเล็กตรอนบนจอภาพ

สัญญาณ SYNC จะมี 2 แบบคือ

1. HORIZONTAL SYNCHRONOUS จะเป็นตัวกำหนดว่าจะเริ่มจุดภาพของแต่ละเส้นเมื่อไร

2. VERTICAL SYNCHRONOUS จะเป็นตัวกำหนดว่าจะมีการเริ่มเส้นที่หนึ่งเมื่อไร

ซึ่งโดยปกติสัญญาณทั้ง 2 นี้จะต้อง Locked กันและ Stable การผลิตสัญญาณภาพของ ซี.อาร์.ที. จะต้องให้สัญญาณ VSYNC ทุก ๆ 20 มิลลิวินาทีนานประมาณ 16 เส้น SCAN LINE คือประมาณ 1 วินาที และสัญญาณ HSYNC ทุก ๆ 64 ไมโครวินาที นานประมาณ 5 ไมโครวินาที นอกจากนี้ตามมาตรฐานของ NORTHBROADCAST STANDARD ได้กำหนดให้ HSYNC PULSE FREQUENCY มีค่าเท่ากับ 15.75 KHz และส่วนเบี่ยงเบนของ CRT Monitor ทำงานที่ความถี่ $\pm 5-10\%$ และ VERTICAL SYNC FREQUENCY เท่ากับ 60 Hz และสำหรับการส่งระบบ NTSC เท่ากับ 50 Hz

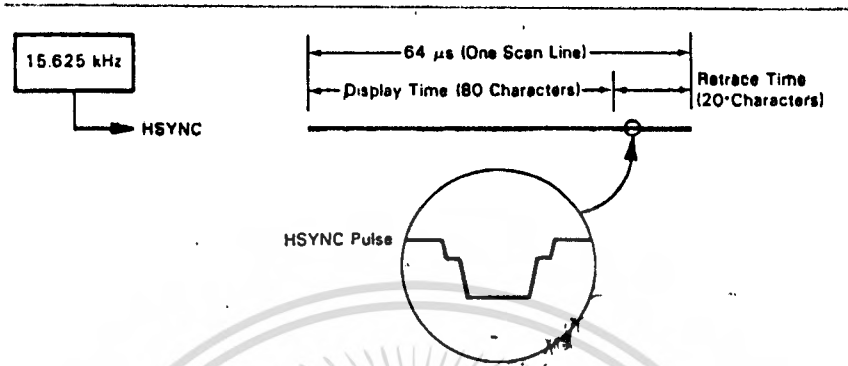
ถ้าเราใช้ HSYNC FREQUENCY = 15.625 KHz เพราะฉะนั้นคาบเวลาของ HORIZONTAL TIME จะเป็น 64 ไมโครวินาที ($1/15625 = 64\mu s$) HSYNC PULSE นี้จะบ่อนให้ CRT MONITOR ที่เกือบสุดเส้น HORIZONTAL SCAN LINE . แต่ละเส้นในช่วงเวลา 64 ไมโครวินาทีจะใช้เป็น HSYNC PULSE TIME และ HORIZONTAL RETRACE TIME ซึ่งจะใช้เวลาประมาณ 20% ของ

HORIZONTAL SCAN TIME ทั้งหมด รูป 2.5 แสดงความสัมพันธ์ระหว่าง HSYNC-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

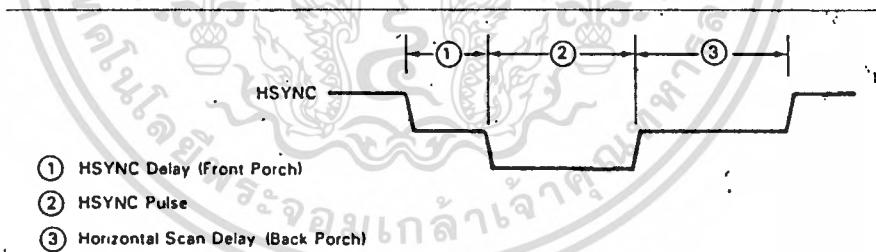
PULSE TIME และ RETRACE TIME

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.5 สัญญาณ HSYNC PULSE

ซึ่งจะเห็นว่า HORIZONTAL RETRACE TIME จะแบ่งออกเป็น 3 ช่วงเวลาดังรูปที่ 2.6 คือ HSYNC DELAY, HSYNC PULSE, HORIZONTAL SCAN DELAY



- ① HSYNC Delay (Front Porch)
- ② HSYNC Pulse
- ③ Horizontal Scan Delay (Back Porch)

รูปที่ 2.6 ส่วนของสัญญาณ HSYNC PULSE

สำหรับ CRT MONITOR นั้น RETRACE TIME ทั้งหมดจะมีค่าอยู่ระหว่าง 10-12 มิลลิวินาทีและอัตราส่วนระหว่าง HSYNC DELAY ต่อ HSYNC PULSE ต่อ SCAN DELAY เท่ากับ 1:2:2

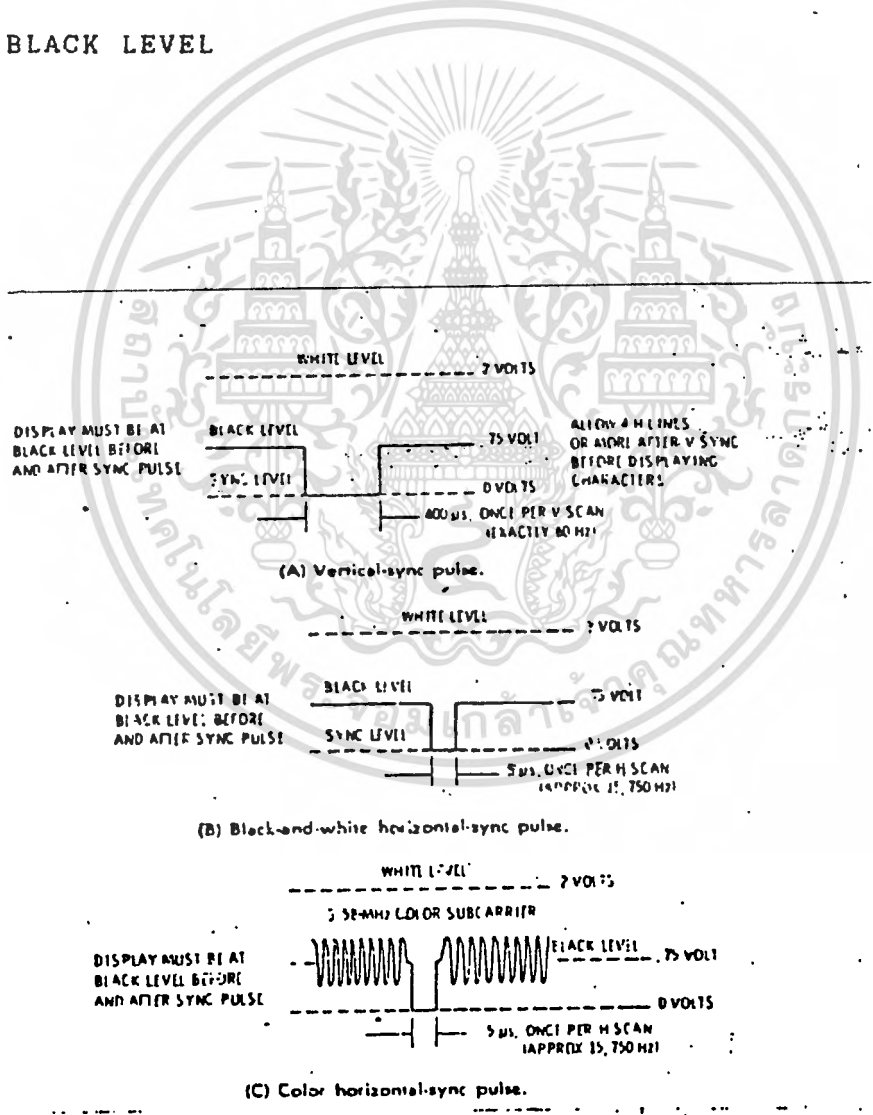
ส่วนสำคัญอีกส่วนหนึ่งของ CRT MONITOR ก็คือคุณสมบัติของ VIDEO -

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า RESPONSE หรือ BANDWIDTH ซึ่งจะเป็นตัวกำหนดจำนวนจุดที่เกิดภาพบนจอ ซึ่งไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CRT สำหรับโทรทัศน์ขาวดำจะมี BANDWIDTH เท่ากับ 3.5 MHZ และโทรทัศน์สี เท่ากับ 3 KHZ ส่วน CRT MONITOR สำหรับงานทางด้านคอมพิวเตอร์จะมี BANDWIDTH อยู่ในช่วง 15-20 MHKZ ซึ่งขึ้นอยู่กับว่า RESOLUTION มากน้อย เพียงไร

ระดับสัญญาณเชิงค้จะแตกต่างกับสัญญาณภาพตัวเลขขึ้นอยู่กับความต้องการของ- ภาคผสมสัญญาณคลื่นพาหะหรือมอดิเตอร์ที่ใช้ ส่วนสัญญาณ BLANK จะเท่ากับระดับ สัญญาณ BLACK LEVEL



2.2 การเชื่อมต่อระหว่าง CRT CONTROLLER และ CRT MONITOR (THE CRT CONTROLLER AND CRT MONITOR INTERFACEING)

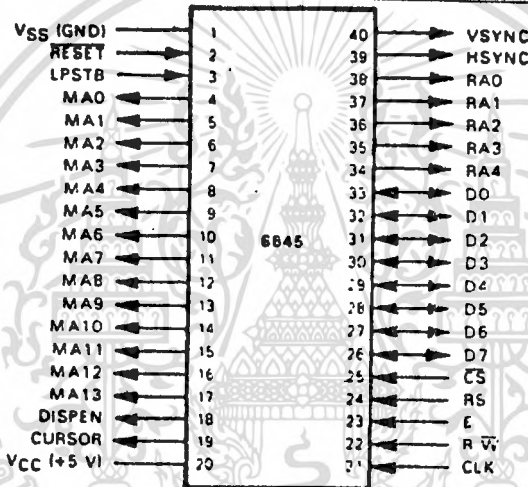
ในระบบของนอนคอมโพสิทวิดีโอ (NONCOMPOSITE VIDEO) นั้น CRT-มอนิเตอร์ (CRT MONITOR) จะประกอบด้วยสัญญาณที่เป็นอินพุทของมอนิเตอร์ 3 สัญญาณที่มีส่วนต่อการสแกนของลำอิเล็กตรอน คือ สัญญาณวิดีโอสัญญาณทางแนวนอน และสัญญาณทางแนวตั้ง สัญญาณเหล่านี้จะเข้าไปในวงจรอิเล็กทรอนิกส์ของมอนิเตอร์ และการสแกนของ ลำอิเล็กตรอนทางแนวนอน และแนวตั้งจะถูกกำหนดโดยอินพุทของการป้อนเข้าสู่มอนิเตอร์อย่างต่อเนื่องกัน นอกจากนี้แล้วอาจมีสัญญาณความเข้มของลำอิเล็กตรอนซึ่งจะทำให้ภาพมีความสว่างกว่าปกติด้วย

สำหรับคอมโพสิทวิดีโอ (COMPOSITE VIDEO) นั้น จะใช้กับมอนิเตอร์ที่มีอินพุทเดียว โดยจะรวมสัญญาณวิดีโอ สัญญาณซิงค์ทางแนวนอน และ สัญญาณซิงค์ทางแนวตั้งเข้าด้วยกัน ทำให้ใช้สัญญาณได้เพียงอินพุทเดียว จากนั้นมอนิเตอร์จะมีการแยกสัญญาณทั้ง 3 ออกจากกัน โดยวงจรภายในของมอนิเตอร์ จะเห็นได้ว่าข้อได้เปรียบของคอมโพสิทวิดีโอ ก็คือ สามารถใช้สายโคแอกเซียลเพียงเส้นเดียว สะดวกและต่อได้ยาวกว่าแบบนอนคอมโพสิทวิดีโอ

บทที่ 3

CRT CONTROLLER 6845

6845 เป็น IC ซึ่งจะจัดการสร้างสัญญาณต่าง ๆ ที่จำเป็นในการแสดงภาพ โดยที่สามารถจะโปรแกรมได้ รูปที่ 3.1 เป็นขาต่าง ๆ ของ CRT 6845



	Pin Name	Description	Type
Microprocessor System Interface	D0-D7	Data Bus	Directional Tristate
	CS	Chip Select	Input
	RS	Register Select	Input
	R W	Read Write Select	Input
	E	Enable-Synchronization Signal	Input
	CLK	Character Rate Clock	Input
	RESET	Reset	Input
	VCC VSS	Power (+5 V) and Ground	Input
Screen Memory and Character Generator Signals	MA0-MA13	Screen Memory Address Bus	Output
	RA0-RA4	Raster (Scan Line) Address Signals	Output
CRT Monitor Interface Signals	HSYNC	Horizontal Synchronization	Output
	VSYNC	Vertical Synchronization	Output
	DISPEN	Display (Video) Enable	Output
	CURSOR	Cursor enable	Output
	LPSTB	Light Pen Strobe	Input

เป็น LOW ระหว่าง HORIZONTAL และ VERTICAL RETRACE หรืออาจจะเรียก
ว่า VIDEO BLANKING CURSOR เป็นสัญญาณ CURSOR ซึ่งสามารถ PROGRAM
ได้

LPSTB [LIGH PEN STROBE INPUT SIGNAL] จะใช้ในการ DETECT
ตำแหน่งของ LIGH PEN

หลักการของ CRT CONTROLLER 6845

IC 6845 เป็นวงจรรวมที่ผลิตโดยบริษัทโมโตโรล่า (MOTOROLA) ทำหน้าที่
เป็นตัวควบคุมการแสดงผลบนจอภาพ ซึ่งมีสัญญาณซิงค์ (SYNCHRONIZATION) อัน
ประกอบด้วย สัญญาณซิงค์ทางด้านแนวนอน และแนวตั้ง สัญญาณนี้จะ ออกมาจากขาของ
ของ 6845 โดยการโปรแกรมรีจิสเตอร์ภายใน 6845 อย่างเหมาะสม

การเชื่อมต่อกันของ IC6845, สกรีนเมมโมรี และคาแรคเตอร์เจนเนอเรเตอร์
(THE 6845-SCREEN MEMORY AND CHARACTER GENERATOR INTERFACING)

สัญญาณที่ติดต่อกับสกรีนเมมโมรีโดยตรงของ IC 6845 มีอยู่ 14 เส้น
(MA0-MA13) หรือ 16 กิโลไบต์ ตัวนับแอดเดรสภายในจะถูกตั้งไว้ที่ 12 และ 13
ซึ่งเป็นจุดเริ่มต้นของแอดเดรส (TOP OF PAGE REGISTER; R12 : R13)

หลังจากนั้นแอดเดรสของ IC 6845 จะเพิ่มขึ้นไปหนึ่งตาม CHARACTER CLOCK
ที่จุดปลายของแต่ละเส้น จะทำให้แอดเดรสรีจิสเตอร์จะถูกตั้งใหม่อีกครั้งที่จุดต้นของ
แอดเดรส และจะหมุนเวียนเช่นนี้ต่อไปทุก ๆ เส้นจนกว่าจะประกอบกันเป็นแถวของ
อักษรอย่างสมบูรณ์ หลังจากเสร็จสิ้นของแถว อักษรแรกจะทำให้ตัวนับแอดเดรสถูก
ไหลลด้วยอักษรของแถวต่อไปตัวแรก ข้อมูลจะถูกไหลลไปภายในส่วนของสกรีนเมม
โมรี ภายใต้การควบคุมของไมโครโปรเซสเซอร์ เมื่อมีการอ่านหรือเขียนข้อมูลลง

ในสกรีนเมมโมรี ในช่วงนี้แอดเดรสของ IC 6845 จะยังไม่ติดต่อกับสกรีนเมมโมรี

CRT 6845 ได้แบ่งสัญญาณเป็นกลุ่ม 3 กลุ่มใหญ่ ๆ คือ

1. กลุ่ม INTERFACE กับ MICROPROCESSOR

DO-D7 เป็น BIRECTIONAL LINES ใช้ TRANSFER DATA ระหว่าง MICROPROCESSOR และ 6845 INTERNAL REGISTERS

\overline{CS} [CHIP SELECT INPUT SIGNAL] เป็น LOW เมื่อ READ หรือ WRITE ข้อมูลใน INTERNAL REGISTER OF 6845

RS [REGISTER SELECT INPUT SIGNAL] เมื่อเป็น LOW, DATA จะแสดง ADDRESS ของ INTERNAL REGISTER ซึ่งจะมี 19 ตัว เมื่อเป็น HIGH จะเป็นการใส่ DATA ลงใน INTERNAL REGISTER ADDRESS ที่เราเลือกก่อนแล้ว

$\overline{R/W}$ [READ/WRITE SIGNAL] ใช้ READ หรือ WRITE INTERNAL REGISTERS, '1' FOR READ, '0' FOR WRITE

E [ENABLE SIGNAL] ใช้ในการ ENABLE INTERNAL I/O BUFFER และ CLOCK DATA เข้าหรือออกจาก INTERNAL REGISTERS ผ่านทาง DATA BUFFER

CLK [CLOCK INPUT] ใช้สร้าง CONTROL SIGNAL ต่างๆ จะเป็น CHARACTER RATE CLOCK

\overline{RESET} ใช้สำหรับ INITIAL 6845 ถ้าเป็น LOW, INTERNAL COUNTERS จะถูก CLEAR และ 6845 OUTPUT จะ SET LOW แต่จะไม่มีผลต่อ PROGRAM ใน REGISTERS

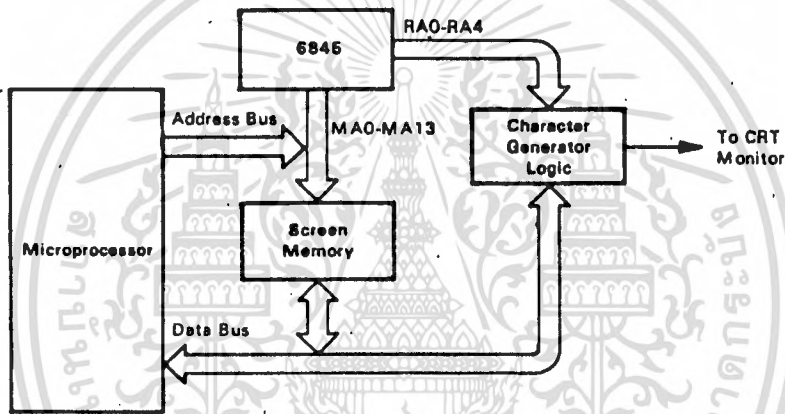
2. SYSTEM BUS ของ SCREEN MEMORY

แบ่งเป็น MA0-MA13 SCREEN MEMORY ADDRESS จะ COUNT ได้ถึง 16 K และ RA0-RA4 เป็น RASTER ADDRESS ทำให้สามารถ SCAN ได้ถึง 32 SCAN LINES

3. CRT MONITOR INTERFACE SIGNALS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆก็ตาม บริษัทฯ ขอสงวนสิทธิ์ในสิ่งที่ปรากฏ และขอสงวนสิทธิ์ในข้อมูลที่มีอยู่ ณ วันที่พิมพ์

เรียบร้อยแล้ว IC 6845 ก็จะส่ง แอดเดรสไปที่สกรีนเมมโมรี่ เพื่อส่งข้อมูลไปยังคาแรคเตอร์เอนเนอเรเตอร์ลอจิก ในขณะเดียวกัน IC 6845 ก็จะส่งสัญญาณเส้นคือ (RA0-RA4) ไปยังส่วนคาแรคเตอร์เอนเนอเรเตอร์โดยปกติ มักเป็นหน่วยความจำแบบ ROM เพื่อกำหนดเส้นแต่ละเส้นในแถวอักษรนั้น ๆ



รูปที่ 3.2 การเชื่อมต่อแบบพื้นฐานของ IC 6845

รีจิสเตอร์ที่สามารถโปรแกรมได้ของ CRT คอนโทรลเลอร์ 6845

รีจิสเตอร์ภายใน 6845 มีทั้งหมด 19 ตัว แต่ที่ใช้ในการควบคุมการทำงานของ IC 6845 มีอยู่ 18 ตัว อีก 1 ตัวเป็นรีจิสเตอร์ที่ควบคุมการโปรแกรมรีจิสเตอร์ คือ แอดเดรสรีจิสเตอร์ (ADDRESS REGISTER) มีอยู่เพียง 5 บิตเท่านั้น

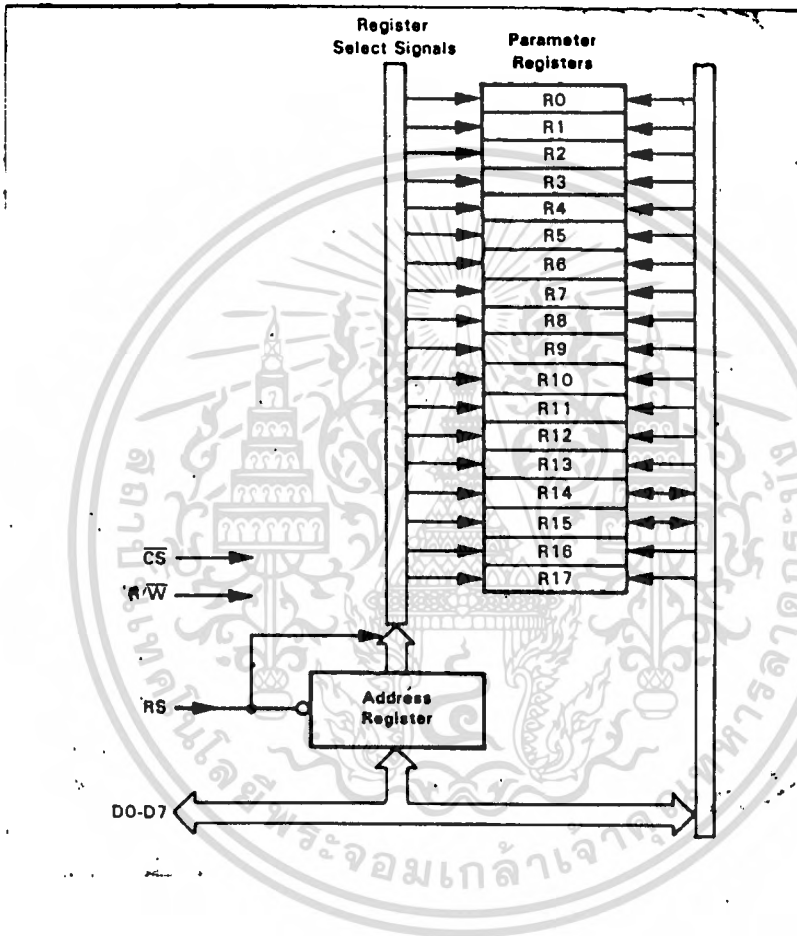
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับรวมให้นำไปใช้ประโยชน์ด้านการค้า (2⁵ = 32) แอดเดรส ตามรูปที่ เมื่อตัวเลือก รีจิสเตอร์ RS (REGISTER) ไม่วากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SELECT) เป็น 0 (LOW) จะเป็นค่าของแอดเดรสรีจิสเตอร์และเมื่อ RS เป็น-1 (HIGH) จะเป็นค่าของข้อมูลของรีจิสเตอร์นั้น ๆ โดยปกติแล้ว RS มักต่อกับ A0

การโปรแกรมรีจิสเตอร์ของ 6845 แบ่งออกเป็น 3 ส่วนใหญ่ ๆ คือ ส่วนกำหนดทางด้านแนวนอน ส่วนกำหนดทางด้านแนวตั้ง และรีจิสเตอร์ที่ทำงานในช่วงเริ่มแรก ดังตารางที่ 3.1

	Register		Read (R) Write (W)	Bits	Range - Units	
	No.	Name/Function				
Horizontal Format and Timing	0100 ₁₆	Horizontal Total	W	8	1 - 256 (0-FF) ₁₆ CLKs	
	1101 ₁₆	Chars/sectors Row	W	8	1 - 256 (0-FF) ₁₆ CLKs	
	2102 ₁₆	H SYNC Position	W	8	1 - 256 (0-FF) ₁₆ CLKs	
	3103 ₁₆	H SYNC Width	W	4	1 - 16 (0-F) ₁₆ CLKs	
Vertical Format and Timing	4104 ₁₆	Vertical Total	W	7	1 - 128 (0-7F) ₁₆ Character Rows	
	5105 ₁₆	V SYNC Adjust	W	5	1 - 32 (0-1F) ₁₆ Scan Lines	
	6106 ₁₆	Character Rows Frame	W	7	1 - 128 (0-7F) ₁₆ Character Rows	
	7107 ₁₆	V SYNC Position	W	7	1 - 128 (0-7F) ₁₆ Character Rows	
	8108 ₁₆	Interface Mode	W	2	0-3	
	9109 ₁₆	Scan Lines Row	W	5	1 - 32 (0-1F) ₁₆ Scan Lines	
Primary Operating Registers	1010A ₁₆	Cursor Start Scan Line	R	7*	1 - 32 (0-1F) ₁₆ Scan Lines	
	1110B ₁₆	Cursor Stop Scan Line	R	5	1 - 32 (0-1F) ₁₆ CLKs	
	1210C ₁₆	(MSB)	Start Address (Top of Page)	R	6	1 - 16.384 (0000-4FFF) ₁₆
	1310D ₁₆	(LSB)		R	7	
	1410E ₁₆	(MSB)	Cursor Position	R W	5	0 - 16.384 (0000-4FFF) ₁₆
	1510F ₁₆	(LSB)		R W	6	
	16110 ₁₆	(MSB)	Light Pen Position	R	6	0 - 16.384 (0000-4FFF) ₁₆
	17111 ₁₆	(LSB)		R	8	

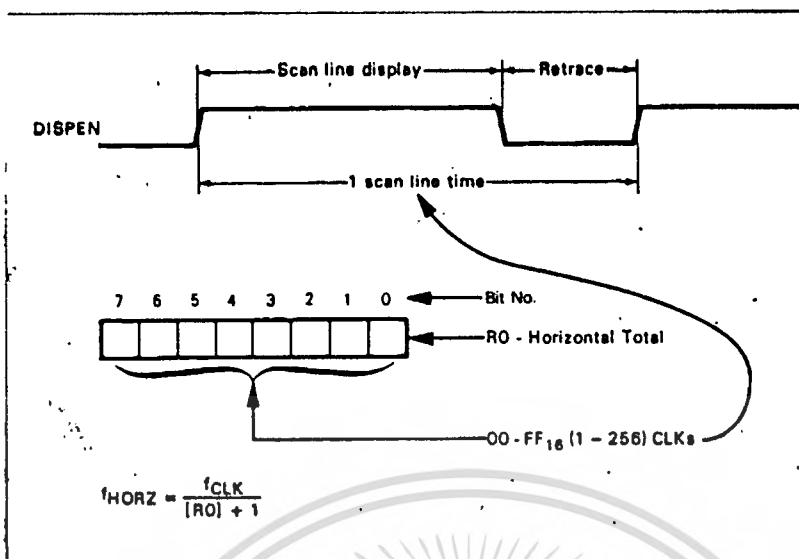
* Two bits used to specify cursor blink characteristics



รูป 3.3 ACCESSING THE 6845 CRT CONTROLLER REGISTERS

6845 PROGRAMABLE REGISTER

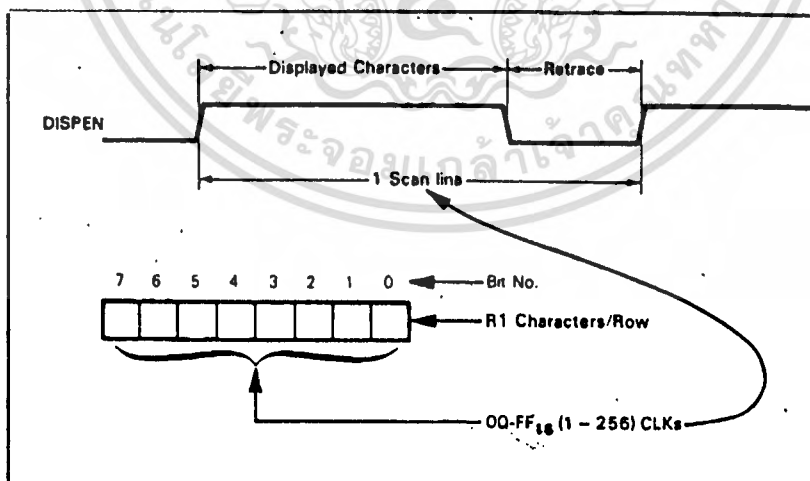
6845 มี INTERNAL REGISTER อยู่ 19 ตัว หน้าที่ของ REGISTER ต่างๆ มีดังนี้



รูป 3.4 6845 CRT CONTROLLER HORIZONTAL REGISTERS (RO)

จากรูป 3.4

จำนวน CHARACTERS ทั้งหมดทางแนวนอนรวมทั้งที่ DISPLAY และ UNDISPLAY ลงด้วย 1 จะเป็นค่าใน REGISTER RO

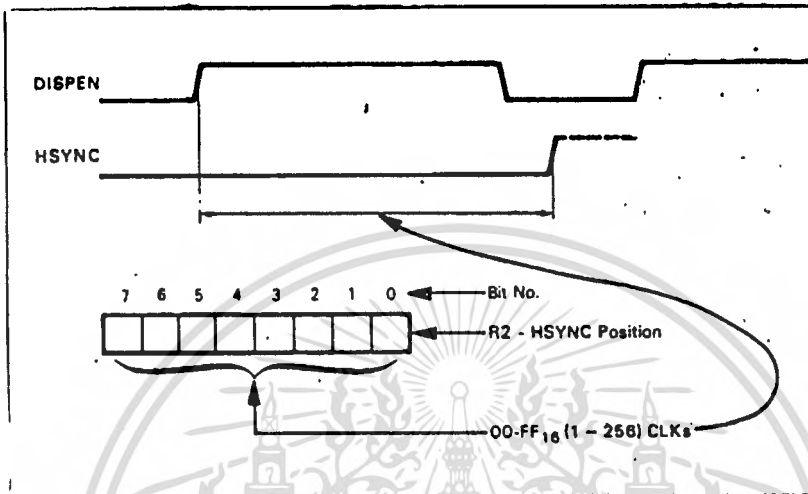


รูป 3.5 6845 CRT CONTROLLER CHARACTERS/ROW (R1)

จากรูป 3.5

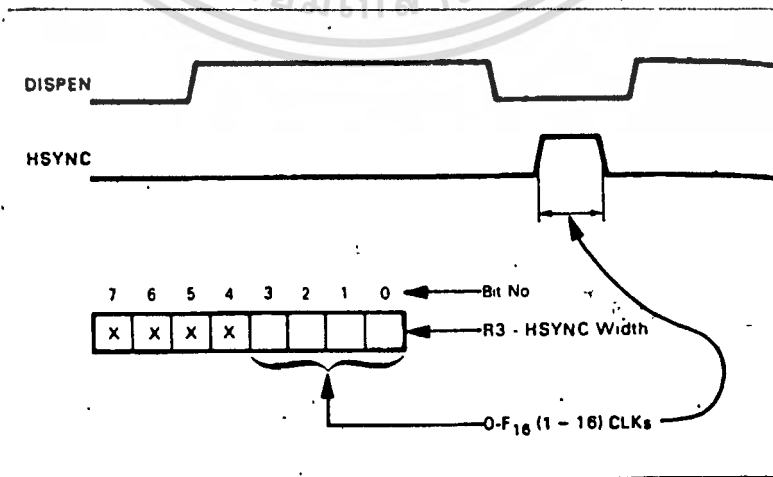
R1 เป็นจำนวน CHARACTERS ที่จะ DISPLAY ในแต่ละ HORIZONTAL LINE เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

“ไม่มีกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้”



รูป 3.6 6845 CRT CONTROLLER HSYNC WIDTH REGISTER (R2)

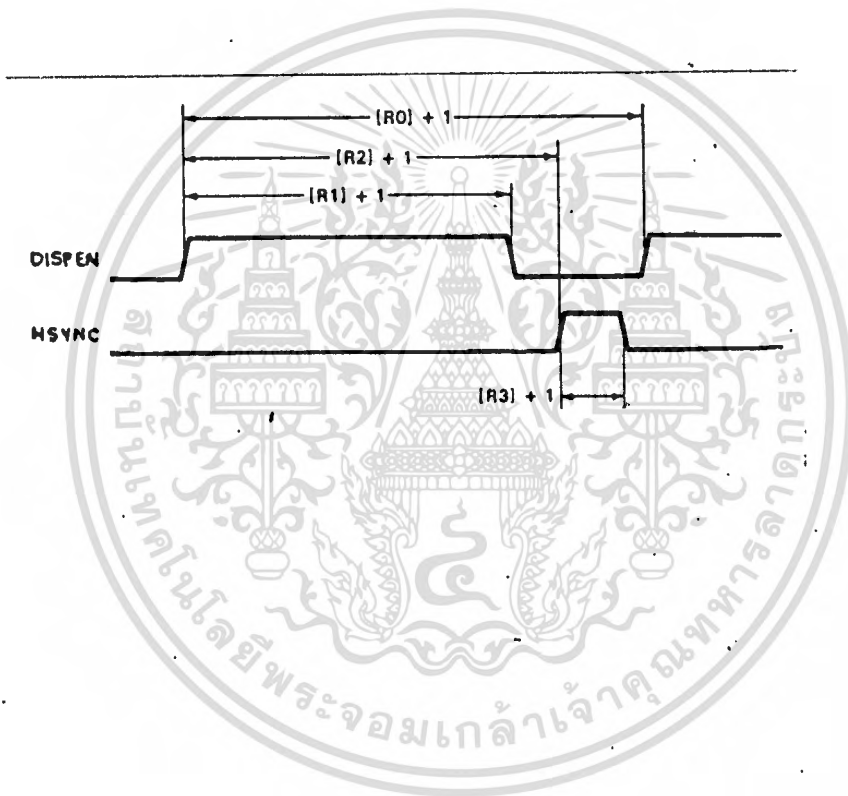
จากรูป R2 เป็นตำแหน่งของ HSYNC ที่จะเปลี่ยนสถานะจาก LOW เป็น HIGH
R2 จะเป็นตัวปรับให้ภาพอยู่กลางจอ



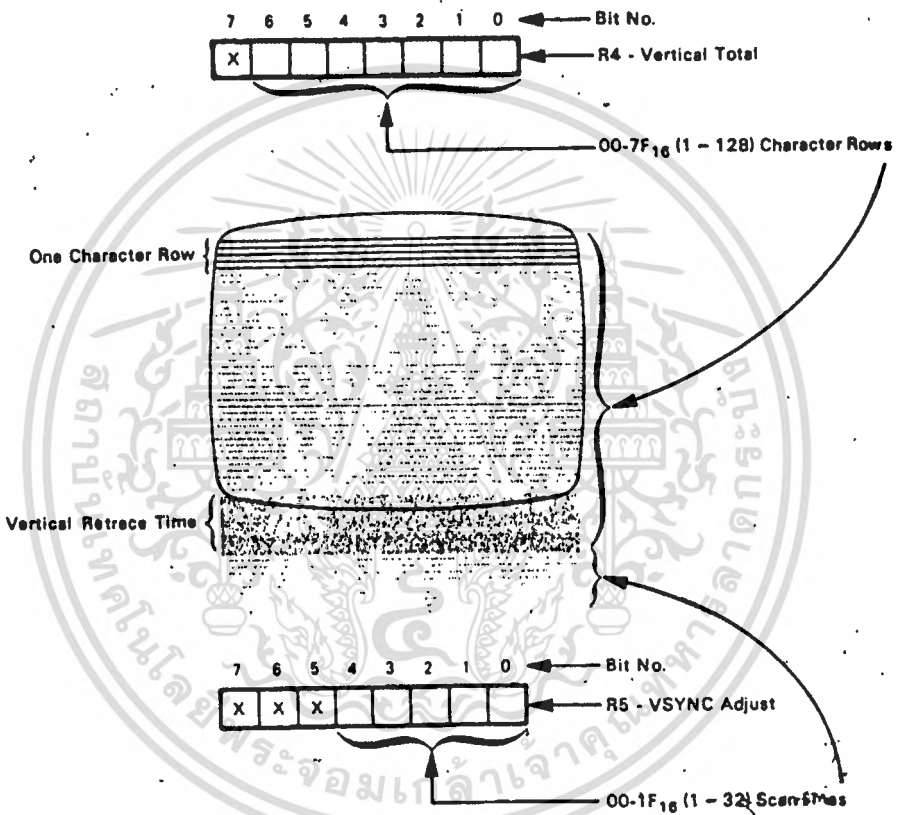
รูป 3.7 6845 CRT CONTROLLER YSYNC WIDTH REGISTER (R3)

จากรูป 3.7

R3 เป็นขนาดของ HSYNC ปกติจะใช้ประมาณ 1-19 CHARACTER CLOCK
ตาม SPECIFICATION ของ CRT MONITOR



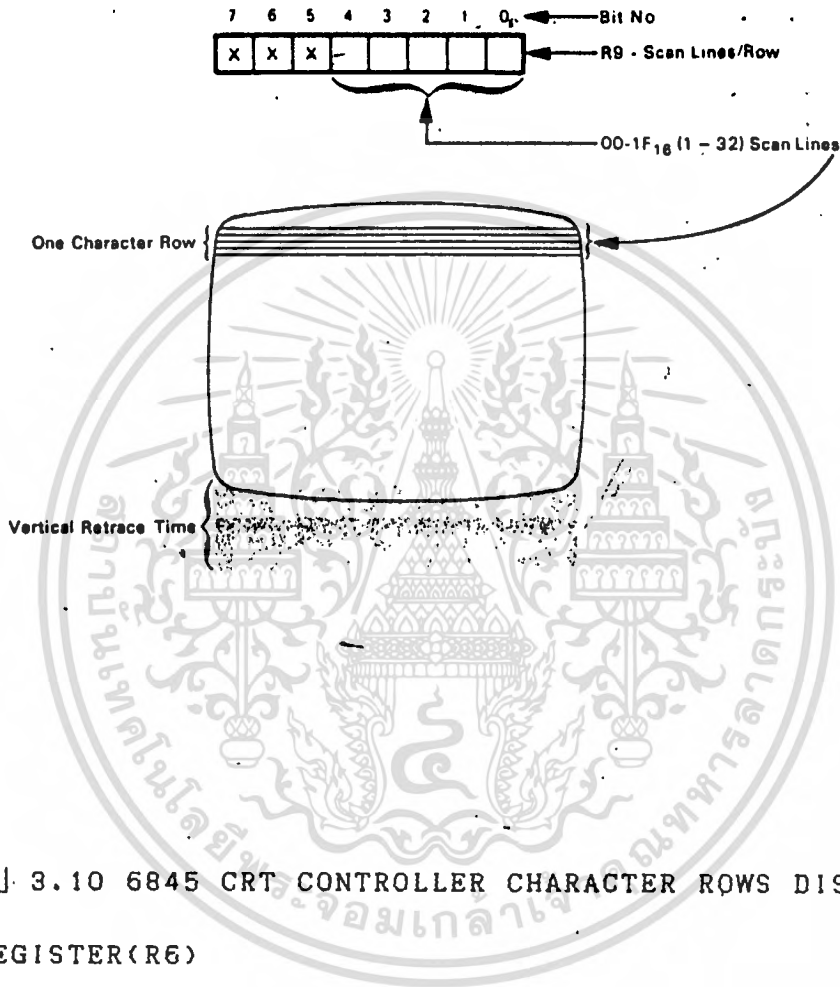
รูป 3.8 แสดงถึงความสัมพันธ์ระหว่าง R0-R3



รูป 3.9 6845 CRT CONTROLLER VERTICAL TOTAL (R4) AND VSYNC ADJUST (R5) REGISTERS

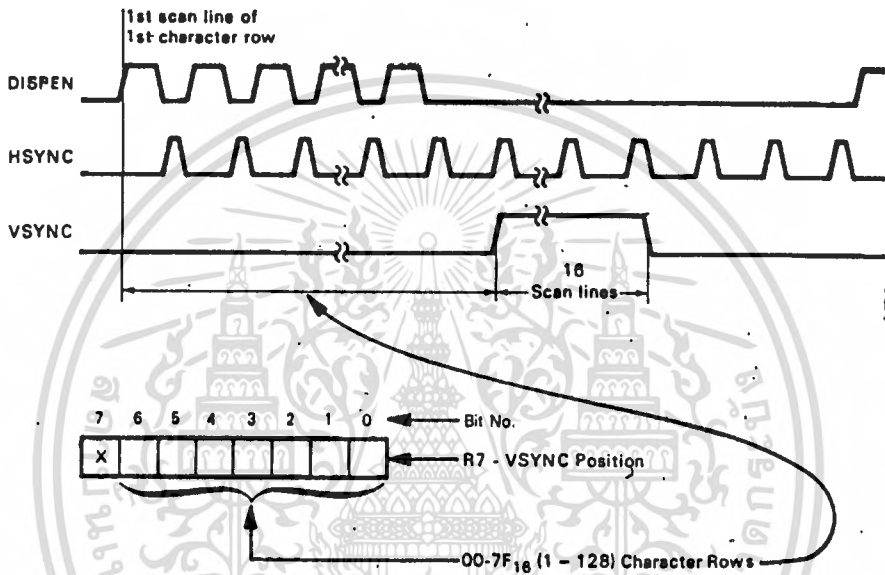
จากรูป 3.9 R4 เป็นจำนวน CHARACTER ROWS ทั้งหมดโดยมี R5 เป็น LINE ADJUST ซึ่งจะปรับละเอียดให้ได้ VSYNC ที่เราต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
 ไม่ควรเผยแพร่ทางอินเทอร์เน็ตหรือสื่อสังคมออนไลน์ และต้องขออนุญาตทุกครั้งที่มีการนำไปใช้



รูป 3.10 6845 CRT CONTROLLER CHARACTER ROWS DISPLAYED-REGISTER(R6)

จากรูป 3.10 R6 เป็นจำนวน CHARACTER ที่จะ DISPLAYED ใน 1 LINE

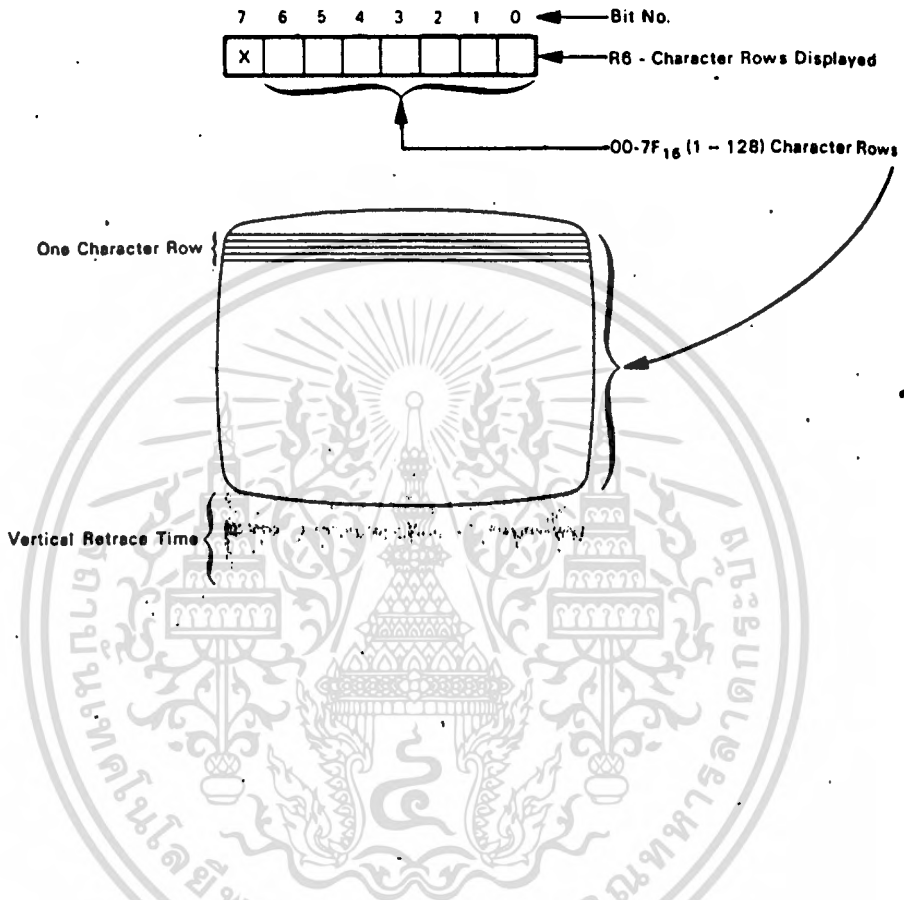


รูป 3.11 6845 CRT CONTROLLER VSYNC POSITION REGISTER(R7)

จากรูป 3.11

R7 เป็นตำแหน่งของ VSYNC ขนาดของ VSYNC PULSE จะประมาณ 16 SCAN LINES ซึ่งไม่สามารถปรับค่านี้ได้

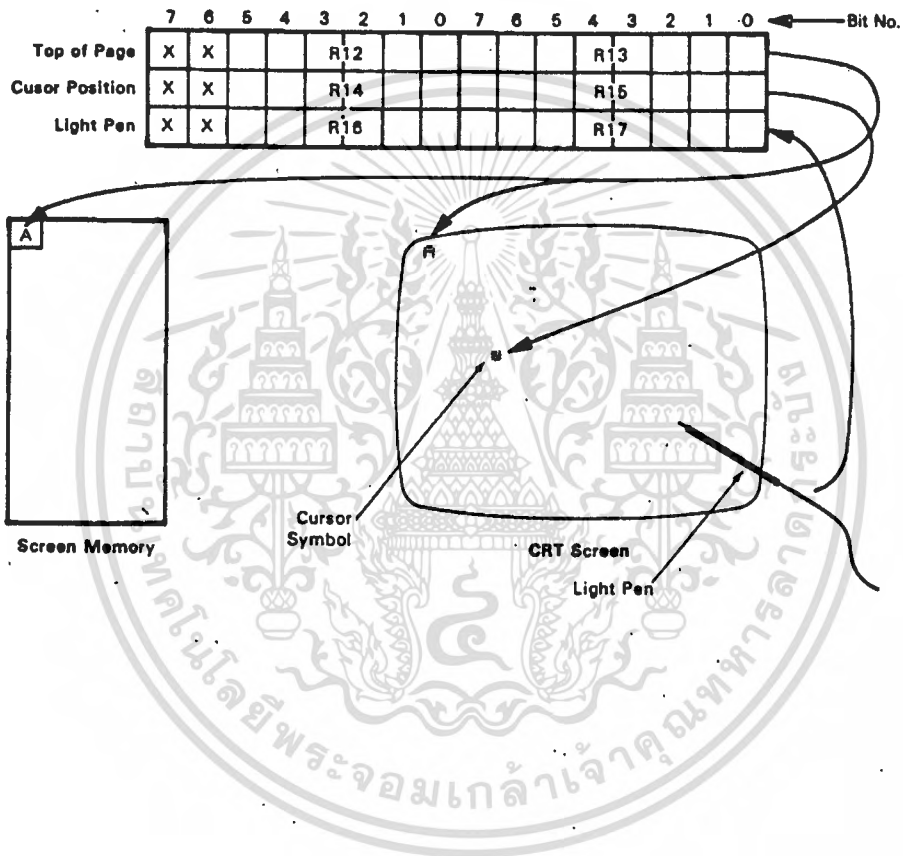
R8 เป็นการเลือก MODE โดย 00,10 เป็น NONINTERLACE MODE
 01 เป็น INTERLACE SYNC, 11 เป็น INTERLACE SYNC AND VIDEO



รูป 3.12 6845 CRT CONTROLLER SCAN LINE/ROW REGISTER (R9)

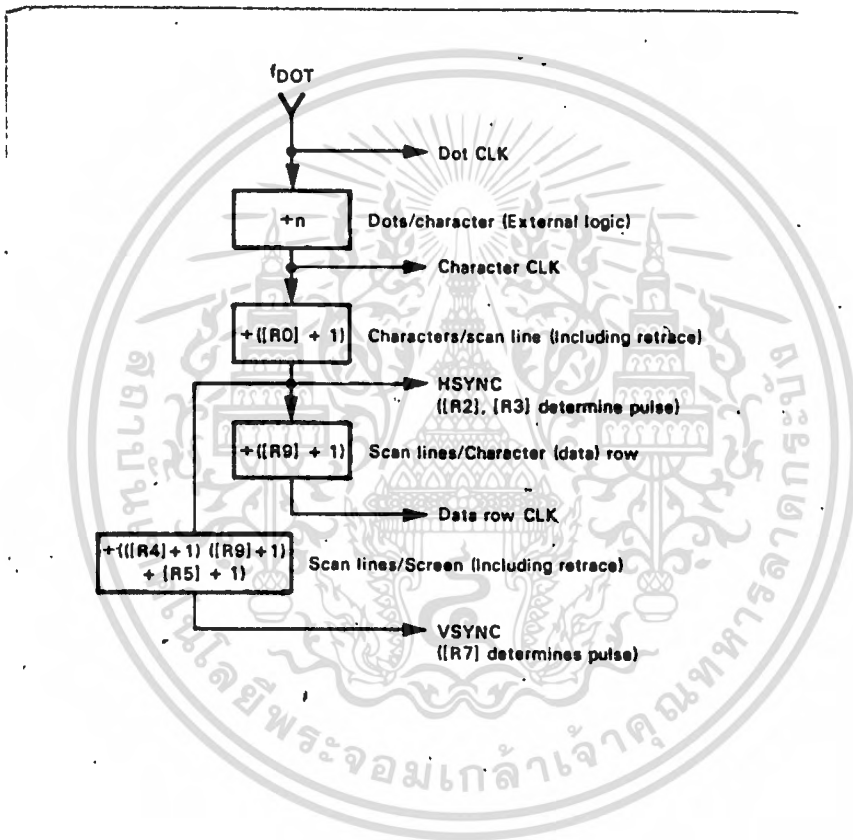
จากรูป 3.12

R9 เป็นจำนวน SCAN LINE PER CHARACTER ROW มี 5 BIT สามารถ SCAN ได้ 32 LINES MAXIMUM



รูป 3.13 6845 CRT CONTROLLER OPERATING REGISTERS (R12-R17)

จากรูป 3.13 R12-R17 เป็นการกำหนดให้ 6845 จะทำให้ทำงานตรงไหน โดย R12-R13 เป็น START ADDRESS สามารถจะ PROGRAM ได้ว่าเริ่มจาก ADDRESS เท่าไรของ SCREEN MEMORY R14-R15 เป็น CURSOR ADDRESS สามารถจะกำหนดตำแหน่งของ CURSOR ได้ R16-R17 เป็น 14 BIT LIGHT PEN REGISTER เป็นสัญญาณ INPUT ซึ่งจะถูกรับ LOAD SCREEN MEMORY ADDRESS เมื่อ LPSTB ถูก DETECTED และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.14

จากรูป 3.14 เป็นการคำนวณ TIMING ต่าง ๆ และ ความสัมพันธ์ของ
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยญาติเห็นาไปใช้ประโยชน์ในการค้า
 REGISTER ใน 6845
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การแสดงผลอักษร

ในการแสดงผลบนจอภาพนั้นสามารถที่จะทำได้ 2 แบบ คือ

1. CHARACTER MODE (TEXTMODE)

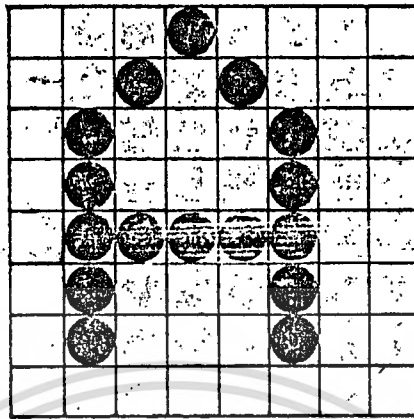
2. GRAPHICS MODE

โดยที่ในโครงสร้างนี้จะกล่าวถึงแบบ Character Mode เท่านั้น โดยจะออกแบบตัว CHARACTER นี้ให้เป็นตัวกราฟฟิก และจะนำแต่ละส่วนมากล่าวเสียก่อน

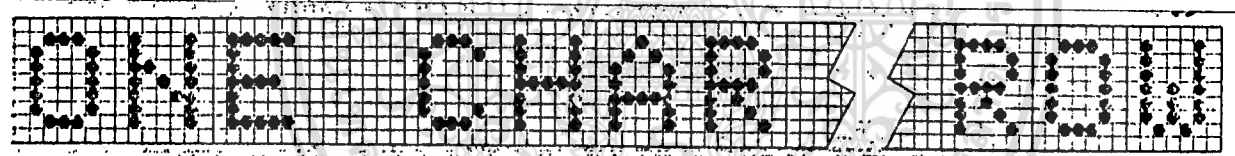
4.1 การแสดงผลทางดิจิตอล

การแสดงผลทางดิจิตอลมีแต่ขาวกับดำและเป็นจุด ๆ ไปไม่ต่อเนื่องเหมือนทางลิเนียร์ เช่นเครื่องรับทีวี โดยที่ในการแสดงผลในโครงงานนี้ใช้แบบเมตริกซ์ $8*8$ นั้นหมายถึงต้องใช้เส้นสแกนไลน์ 8 เส้นต่อการสแกน 1 บรรทัดอักษร และให้แสดงผลบรรทัดละ 80 ตัวอักษร ซึ่งมี 25 บรรทัดต่อ 1 จอภาพ นั่นคือต้องใช้สแกนไลน์ทั้งหมด $25*8=200$ เส้น โดยแต่ละเส้นสแกนไลน์จะประกอบด้วย $80*8=640$ จุดภาพ (คำที่กล่าวมาเป็นคำที่ต้องใช้ในการแสดงผลเท่านั้นยังไม่รวมส่วนที่เป็น Blank ซึ่งต้องเพิ่มขึ้นอีก)

ในเมตริกซ์ $8*8$ ที่กล่าวมาเรียกว่า 1 ช่องอักษร โดยที่คิดว่าบรรทัดหรือบรรทัดอักษรให้ถือว่าความหมายเหมือนกัน (Row และ Character row) ซึ่งเราจะใช้คำว่าโรว์คู่กับคำว่าคอลัมน์ เพื่อแสดงถึงตำแหน่งที่อักษรบนจอภาพ โดยโรว์จะมีค่า 0-24 และคอลัมน์มีค่า 0-79 (ในการแสดงผล 25 บรรทัด ๆ ละ 80 อักษร)



รูป 4.1 ภาพอักขระในแมทริกซ์ 8*8



รูป 4.2 ภาพอักขระที่ปรากฏบนจอภาพใน 1 ROW

4.2 รหัสอักขระ

รหัสอักขระที่ใช้บนเครื่องไมโครคอมพิวเตอร์ทั้งหมดจะเป็นรหัส ASCII โดยจะมีการกำหนดหมายเลขให้แก่อักขระต่าง ๆ ให้เป็นมาตรฐาน เช่นอักษร A มีค่า $\$41$, B มีค่า $\$42$ เป็นต้น

ในรหัส ASCII นี้จะประกอบด้วย อักษรภาพอังกฤษทั้งเล็กและใหญ่ เครื่องหมายต่าง ๆ รวมทั้งอักขระควบคุมอีก 32 ตัว รวมแล้วเท่ากับ 128 ตัว คือรหัส $0-\$7f$ นั่นเอง ส่วนรหัส $\$80-\ff นั้นไม่มีกำหนดเป็นมาตรฐานซึ่งขึ้นอยู่กับวัตถุประสงค์ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งของผู้ใช้ โดยในประเทศไทยจะนำมากำหนดเป็นรหัสภาษาไทยนั่นเอง แต่ของ IBM จะใช้เป็นแสดงเป็นภาพกราฟิก โดยที่ในโครงงานนี้จะใช้รหัส \$80-\$ff ที่ได้กำหนดหรือสร้างขึ้นมาใหม่ มาเป็นตัวกำหนดภาพกราฟิกบนจอภาพ

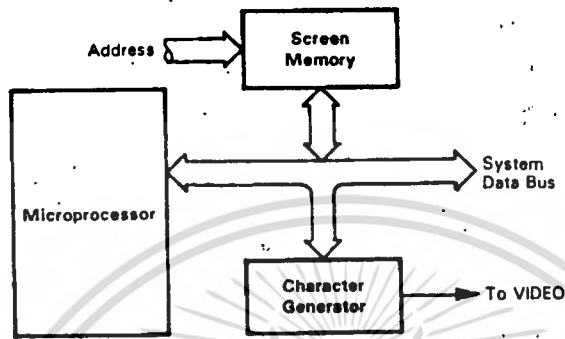
4.3 หน่วยความจำจอภาพ (SCREEN MEMORY)

ในการแสดงภาพอักขระทางจอภาพ คือการจัดหน่วยความจำหน่วยหนึ่งเป็นหน่วยความจำจอภาพ โดยที่ต้องจัดวงจรทางฮาร์ดแวร์ไว้ว่า ถ้าหากนำรหัสอักขระใดก็ตามมาไว้ที่ไบต์แรกของหน่วยความจำนี้ก็จะปรากฏอักขระนั้นขึ้นมาบนจอภาพตรงตำแหน่งซ้ายสุดหรือ Row = 0 , Col = 0 ถ้าไว้ที่ไบต์ต่อมาก็จะปรากฏเป็นตัวต่อมาที่บรรทัดเดียวกันเรียงกันไปจนครบ 80 อักขระ จากนั้นจะขึ้นบรรทัดใหม่เป็นเช่นนี้เรื่อยไปจนครบทั้งจอภาพ (ROW 24, COL =79)

ในการคำนวณขนาดของหน่วยความจำจอภาพนี้มีหลักการดังนี้

แสดงผล 80×25 นั่นคือต้องใช้หน่วยความจำ = 2,000 ไบต์ ซึ่ง Address นี้จะอยู่ที่ใดก็ได้แล้วแต่การออกแบบวงจร

มีเรื่องที่ต้องสังเกตด้วยคือรหัสอักขระ 1 รหัส คือ 1 ไบต์ ทำให้เกิดภาพอักขระ 1 ตัว ตามโครงงานใช้แบบเมตริกซ์ 8×8 จุดภาพ นั่นคือต้องใช้ 64จุดภาพ ต่อ 1 อักขระนั่นเอง และในการทำให้บังเกิดจุดภาพ 1 จุด จะต้องใช้หน่วยความจำ 1 บิต และส่วนมากจะให้เป็นจุดขาวเมื่อบิตนี้มีค่าเป็น 1 และเป็นจุดมืดเมื่อมีค่า 0 ดังนั้นในการแสดงภาพ 8×8 จุดภาพ จะต้องใช้หน่วยค่า 8 ไบต์ ฉะนั้นหน่วยความจำที่ต้องใช้ทั้งหมดคือ $80 \times 25 \times 8 = 16,000$ ไบต์ แต่ในตอนต้นกล่าวว่าใช้เพียง 2,000 ไบต์ นั้นเพราะใช้ แพทเทิร์นของภาพอักขระใน ROM เข้าช่วยนั่นเอง และถ้าไม่ใช้ แพทเทิร์นของภาพอักขระใน ROM แล้วจะต้องใช้หน่วยความจำถึง 16,000 ไบต์ซึ่งก็คือการแสดงภาพใน GRAPHICS MODE นั่นเอง



รูป 4.3 แสดงความสัมพันธ์ระหว่าง CHARACTER GENERATOR กับ SCREEN-MEMORY

4.4 แพทเทิร์นของภาพอักขระใน ROM

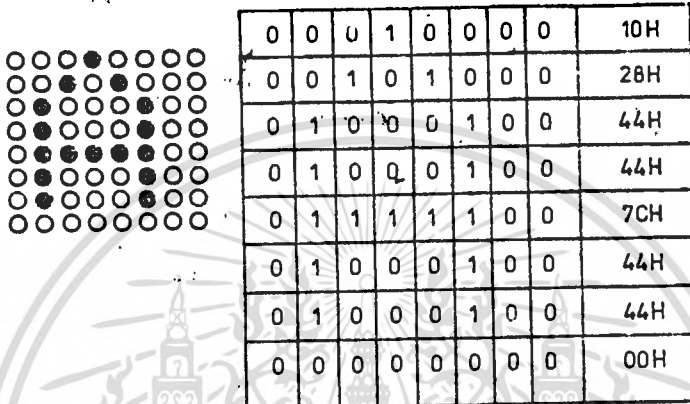
จากที่กล่าวมาแล้วว่าเราใช้แพทเทิร์นของภาพอักขระใน ROM เข้าช่วยในการแสดงผลในแบบ CHARACTER MODE แพทเทิร์นของภาพอักขระใน ROM ซึ่งปัจจุบันนิยมใช้แบบ EPROM ซึ่งถูกโปรแกรมแพทเทิร์นของภาพอักขระไว้ ดังรูป 4.4 ซึ่งเป็นการเตรียมข้อมูลเพื่อการโปรแกรม EPROM สำหรับการแสดงผลแบบเมตริกซ์ 8*8 โดยสมมติว่าได้ดำเนินการมาถึงตัวอักษร A

ในการแสดงผลแบบ 8*8 เราต้องใช้หน่วยความจำ 8 ไบต์ต่อแพทเทิร์นของภาพอักขระ 1 อักขระ ดังนั้นในการแสดงผลภาพอักขระทั้ง 256 อักขระต้องใช้หน่วยความจำ $256 * 8 = 2048$ ไบต์หรือ 2KBYTE ได้แก่ EPROM เบอร์ 2716 ในการโปรแกรมเราให้รหัส 001 อยู่ที่ไบต์ 8-15 และต่อ ๆ ไป ดังนั้นรหัส A คือ 041 จะอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าที่ไบต์ 520-527

ไมวากรณ์ใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่กล่าวมานี้จะเห็นว่าแพทเทิร์นของภาพอักขระเกินอยู่ในลักษณะเป็นไบต์
 คู่ 8 แต่การที่เราให้ค่าในแต่ละไบต์แก่จอภาพ เราจะต้องจับมันมาเข้าแถวเรียง
 เดียวกันออกไปซึ่งดำเนินการโดยใช้ SHIFT REGISTER ซึ่งจะกล่าวต่อไป



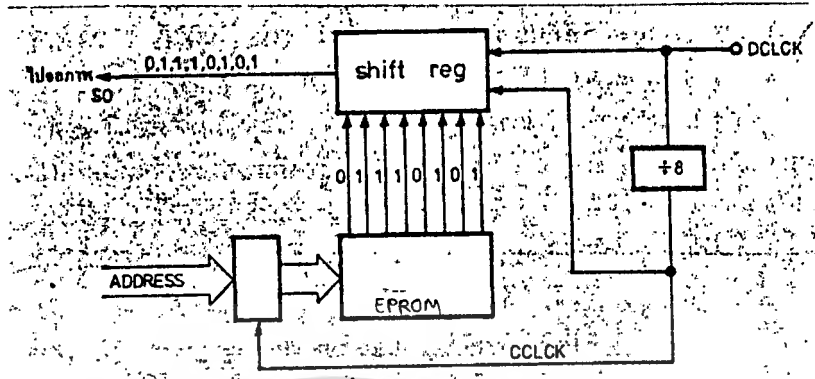
0	0	0	1	0	0	0	0	10H
0	0	1	0	1	0	0	0	28H
0	1	0	0	0	1	0	0	44H
0	1	0	0	0	1	0	0	44H
0	1	1	1	1	1	0	0	7CH
0	1	0	0	0	1	0	0	44H
0	1	0	0	0	1	0	0	44H
0	0	0	0	0	0	0	0	00H

รูป 4.4 แพทเทิร์นของอักขระ A

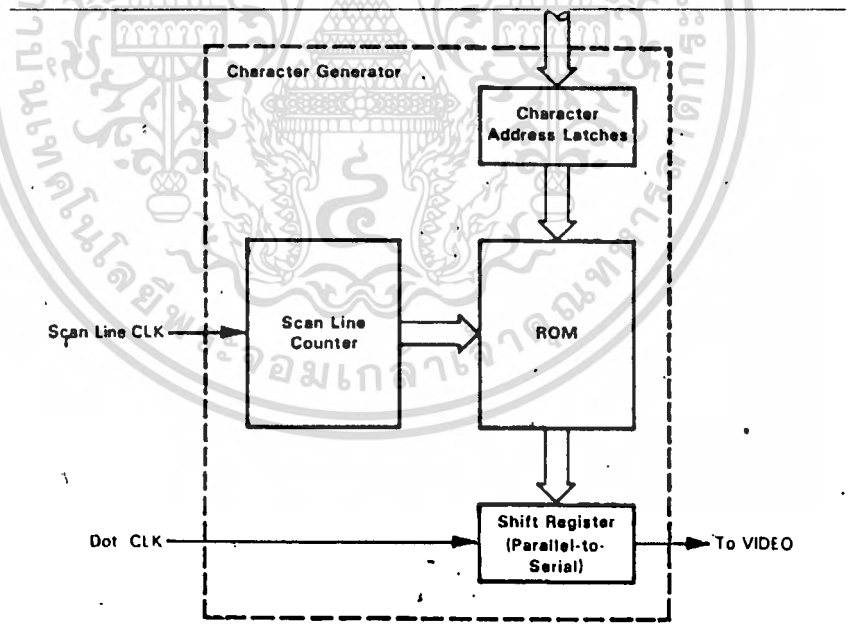
4.5 SHIFT REGISTER

จะทำให้หน้าที่เปลี่ยนข้อมูลจากขนานไปเป็นอนุกรม เพื่อจะนำข้อมูลจาก EPROM
 ไปออกจอภาพซึ่งจะดำเนินการครั้งละ 1Byte ได้ภาพบนจอภาพออกมา 8 จุดภาพ

ในรูปแสดงว่ามีอุปกรณ์ให้ค่าแอดเดรสแก่ EPROM ตามช่วงจังหวะของสัญญาณนาฬิกา
 CCLK (CHARACTER CLOCK) ซึ่งค่า Data ใน EPROM จะให้ค่าคาตาแก่ -
 SHIFT REGISTER นำไปชิฟท์ออกมาเป็นข้อมูลแบบอันดับตามช่วงจังหวะของสัญญาณ
 นาฬิกา DCLK ซึ่งจะเร็วกว่า CCLK 8 เท่า



รูป 4.5 แสดงการทำงานของ Shift Register กับ EPROM(Pattern)



รูป 4.6 แสดง CHARACTER GENERATOR

รูปที่ 4.6 จะเห็นว่า เป็นโครงสร้างการทำงานรวมของคาแรคเตอร์เจนเนอเรเตอร์
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 เรเตอร์ ซึ่งประกอบด้วย หน่วยความจำแบบ ROM สแกนไลน์เคาท์เตอร์(SCAN
 ไม่วากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหากมีการนำไปใช้

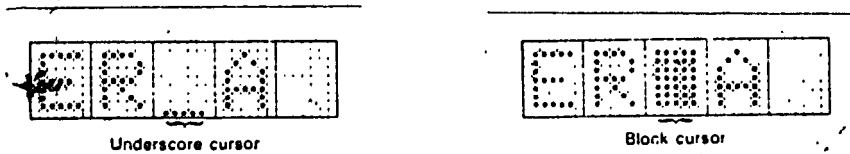
LINE COUNTER), คาแรคเตอร์แอดเดรสแลทช์ (CHARACTER ADDRESS LATCH) และชิฟต์รีจิสเตอร์ (SHIFT REGISTER) อุปกรณ์เหล่านี้เป็นส่วนมาตรฐานของ วงจรคาแรคเตอร์เอนเนอเรเตอร์ ดังนั้นจึงมีบริษัทผลิต IC มากมายที่ผลิต IC ที่มีคาแรคเตอร์เอนเนอเรเตอร์อยู่ในทั้งหมด โดยสามารถที่จะนำ IC ตัวนี้ไป ใช้กับรหัส ASCII ที่มีเมทริกซ์ 5*7 หรือ 8*8 ก็ได้ แต่บริษัทบางบริษัทมักจะทำ ลอจิก (LOGIC) ภายใน IC ที่แตกต่างกันออกไป เช่น IC บางตัวอาจรวมส่วน ของสแกนไลน์เคาท์เตอร์คาแรคเตอร์แอดเดรสแลช และชิฟต์รีจิสเตอร์ ในขณะที่ ส่วนของหน่วยความจำแบบ ROM จะแยกเป็น IC อีกตัวหนึ่ง เพื่อที่จะได้สามารถ เปลี่ยนแปลง รูปแบบของตัวอักษรได้โดยง่ายและการใช้งานที่กว้างขวางมากขึ้น แต่ ในโครงการนี้ ส่วนต่าง ๆ เหล่านี้จะสร้างขึ้นมาจาก IC ต่าง ๆ ที่จะ นำมาประกอบรวมกันเพื่อเป็นคาแรคเตอร์เอนเนอเรเตอร์

4.6 การจัดการเกี่ยวกับข้อมูลของ CRT สกรีน

(Manipulation Of CRT Screen Data)

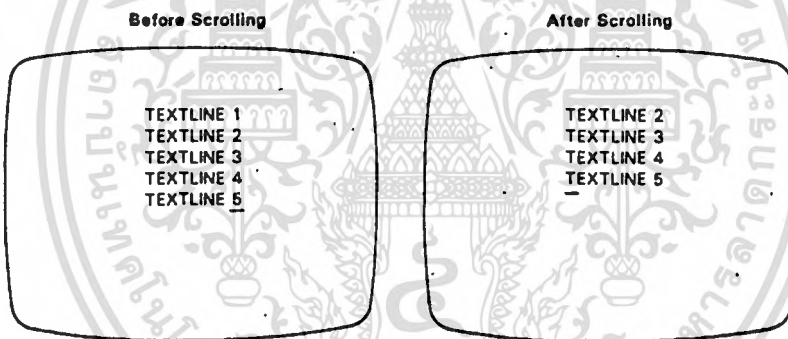
เคอร์เซอร์ (CURSOR) เมื่อมีข้อมูลขึ้นบนจอภาพแล้ว ถ้าต้องการเปลี่ยนข้อมูล ที่แสดงผลบนจอภาพ และจะเขียนข้อมูลลงในสกรีนเมมโมรี่ในตำแหน่งที่เหมาะสม โดยการใส่เคอร์เซอร์ ซึ่ง CRT คอนโทรลเลอร์ส่วนใหญ่ จะมีเคอร์เซอร์ให้ไว้ เคอร์เซอร์มักจะเห็นและเคลื่อนที่ได้ บางครั้งอาจจะ โปรแกรมให้ทำงานกระหนาบที่ แตกต่างกัน และ อาจจะมีลักษณะเป็นเส้นหรือบล็อก

ดังรูป 4.7



รูปที่ 4.7 แสดงลักษณะของเคอร์เซอร์ที่เป็นเส้นและบล็อก

การเลื่อนข้อมูลเป็นบรรทัดบนจอภาพ มันมีลักษณะคล้ายกับกระดาษที่นำมาเข้าเครื่องพิมพ์ดีด โดยเอกสารนี้จะอยู่ส่วนบนของจอภาพ และตามด้วยแถวต่อ ๆ ไป เมื่อถึงจุดต่ำสุดของจอภาพ เอกสารทั้งหมดถึงจะเลื่อนขึ้น วิธีการเลื่อนข้อมูลขึ้นมีอยู่หลายวิธี



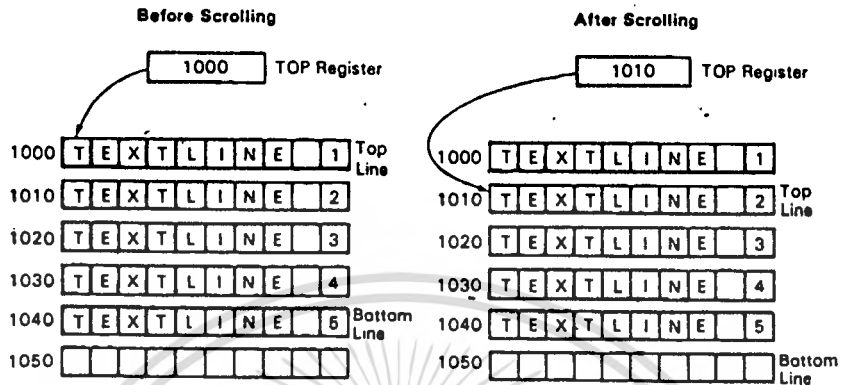
รูปที่ 4.8 แสดงการเลื่อนข้อมูลขึ้นแบบง่ายที่สุด

วิธีที่ง่ายที่สุด คือ การเลื่อนข้อมูลแถวบนสุดก่อนแล้วจึงเลื่อนข้อมูลของแถวถัดมาเรื่อย ๆ แต่วิธีนี้มีข้อเสียคือ ทำงานช้า ตามรูปที่ 4.8

แต่วิธีที่ใช้ใน CRT คอนโทรลเลอร์ ที่มีประสิทธิภาพ คือ การใช้ที่อปริจิสเตอร์ (Top Register) ซึ่งจะบรรจุแอดเดรสที่มีข้อมูลบนสุดของจอภาพแอดเดรสเหล่านี้ ถูกเปลี่ยนไป เมื่อมีการเลื่อนข้อมูลนี้ ตามรูปที่ 4.9

ก่อนการเลื่อนข้อมูลขึ้น

หลังการเลื่อนข้อมูลขึ้น



รูปที่ 4.9 แสดงการเลื่อนข้อมูลขึ้นโดยใช้ทอปรีจิสเตอร์

ปัญหาที่เห็นเด่นชัดในการเลื่อนข้อมูลขึ้นแต่ละครั้งที่มีการเลื่อนข้อมูล ก็จะมีผลเหมือนกับการเคลื่อนที่หน้าต่างซึ่งมีขนาดเท่ากับความจุของจอภาพ สมมติว่าถ้ามีจอภาพ 80 ตัวอักษร 25 แถว จะต้องใช้ 2000 ไบต์ (25×80) ในสกรีนเมมโมรี่แต่ละครั้งที่มีการเลื่อนข้อมูลขึ้น 80 ไบต์ ของหน่วยความจำจะถูกเพิ่มขึ้น หลังจากมีการเลื่อนจอภาพถึง 25 ครั้ง ทำให้ต้องใช้เมมโมรี่ถึง 4000 (2000×2) ไบต์ ดังนั้น การเลื่อนข้อมูลโดยใช้หน้าต่างนั้นจะไม่ค่อยนำมาใช้ในการประยุกต์ต่าง ๆ มากมายนัก

บทที่ 5

การออกแบบวงจร

วงจรในโครงงานนี้แบ่งเป็น 2 ส่วน คือ

1. ส่วน CPU
2. ส่วนแสดงผล

การทำงานของส่วน CPU

ในโครงงานนี้เป็นส่วนหนึ่งของโครงงาน

เป็นส่วนที่ทำหน้าที่ในการรับข้อมูลจากหน่วยประมวลผลมาแสดงบนจอภาพโดย CPU ในโครงงานนี้จะเป็นตัวอำนวยความสะดวกในการจัดการทั้งหมดบนจอภาพ เช่น ลบหน้าจอ เลือกรหัสหรือเพจที่จะแสดงผล เป็นต้น โดยรับข้อมูลเข้ามาทาง CONNECTOR ซึ่งสัญญาณต่าง ๆ เป็นมาตรฐานเดียวกับ PRINTER PORT CONNECTOR ที่ใช้จะมีลักษณะเดียวกับที่อยู่บน PRINTER สามารถนำไปต่อกับ PRINTER PORT ของ IBM เมื่อส่งข้อมูลออกมาโดยวิธีเดียวกันกับการส่งข้อมูลไปพิมพ์ ส่วน CPU ก็จะได้รับข้อมูลและนำไปพิมพ์บนจอภาพได้อย่างถูกต้อง ซึ่งในการออกแบบส่วนที่จะรับข้อมูลจาก PRINTER PORT โดยใช้ 8255 โดย PORT A โปรแกรมให้เป็นแบบ HANDSHAKE และใช้ขา PC6 เป็นสัญญาณ ACK ตัว CPU ได้ใช้เบอร์ Z80 ซึ่งรายละเอียดของ Z80 นี้จะไม่กล่าวถึงเพราะสามารถหาอ่านได้ทั่วไปและเป็น CPU ที่คุ้นเคยกันดีอยู่แล้วและใช้ EPROM เบอร์ 2764 และ RAM เบอร์ 6264 อย่างละตัวซึ่งเพียงพอที่นอกจากจะใช้เป็นตัวจัดการเกี่ยวกับการแสดงผลในโครงงานนี้แล้วยังสามารถนำไปใช้ในงานควบคุมขนาดเล็กอื่น ๆ ได้อีกด้วย

ในส่วน MEMORY ได้ออกแบบให้ EPROM อยู่ที่แอดเดรส 0000H-1FFFFH และ RAM อยู่ที่แอดเดรส 2000H-3FFFFH ส่วน PORT ที่ต่อกับ 8255 คือ 00H, 01H, 02H, 03H ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกหนึ่งที่มีให้คือแบบสื่อนี้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

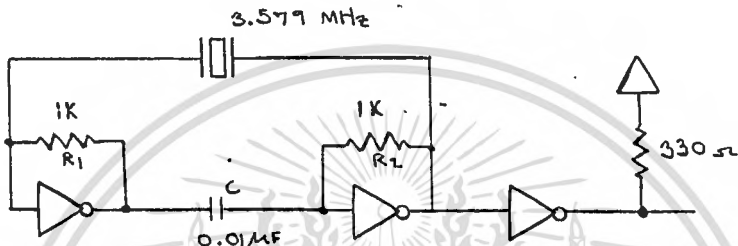
O1 = PORT B

O2 = PORT C

O3 = CONTROL

ขั้นตอนการออกแบบ

1. วงจร CLOCK



จาก

$$R1 = R2$$

$$R1 * C \geq 1 / F$$

$$T_{PHL} < 1 / F$$

เลือกใช้ความถี่ 3.57 MHz [X-TAL = 3.57 MHz]

INVERTER ที่เลือกใช้คือ 74LS04

เลือก R1=R2=1K

ดังนั้น C ≥ 1/[3.57MHz][1K]

C ≥ 0.0002 uF

เพื่อความสะดวกเลือกใช้ C=0.01 uF

จากDATA Z80 V_{ILC} = 0.45 V

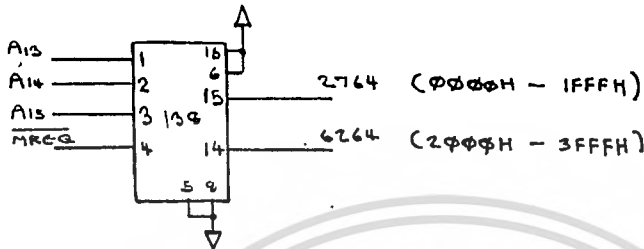
V_{IHC} = V - 0.6 V

ดังนั้นจึงต่อ R 330 Ω เป็น R pull up เพื่อดึงระดับ LOGIC 1 ให้

สูงเพียงพอสำหรับสำหรับ Z80

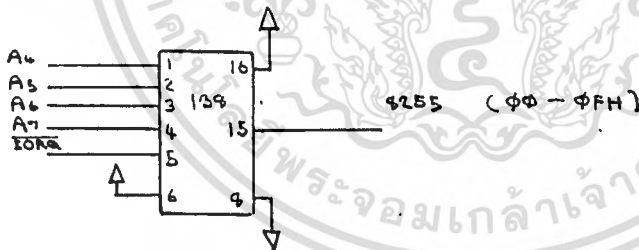
2. ส่วน DECODE MEMORY

จะเป็นดังรูป



3. ส่วน DECODE 8255

เพื่อป้องกันการอ้างถึง 8255 โดยไม่ต้องการจึงจัดให้ 74LS138 DECODE 8255 ACTIVE เฉพาะช่วง PORT ที่ต้องการเท่านั้นคือ PORT 0H-FH

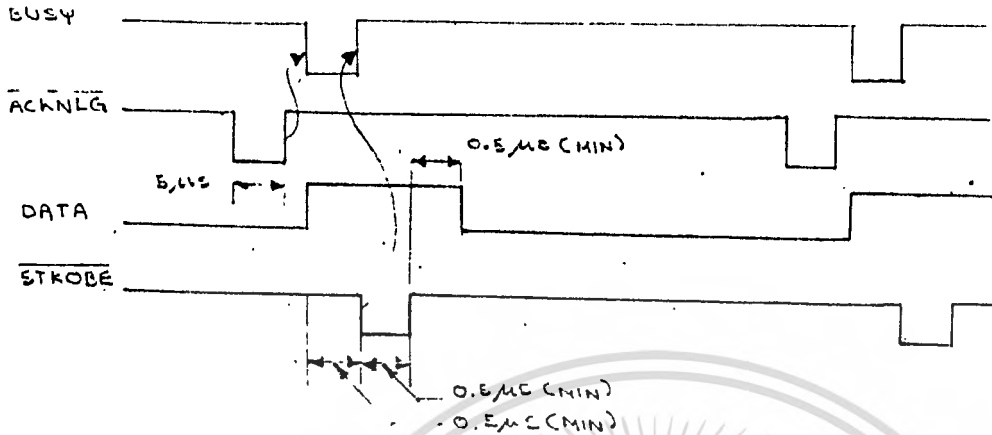


4. การจัดสัญญาณจาก 8255 เพื่อเชื่อมต่อกับ PRINTER ADAPTER

SPECIFICATION

- DATA TRANSFER RATE 1000 CPS (MAXIMUM)
- SYNCHRONIZATION : โดย STROBE PULSE
- HANDSHAKING ACKNLG OR BUSY SIGNAL

DATA TRANSFER SEQUENCE



เราจะใช้ PROT A เป็น INPUT PORT ในการรับ DATA โดยโปรแกรมให้เป็น MODE 1 (HANDSHAKING)

PC4	STB	==>	$\overline{\text{STROBG}}$
PC5	IBF	==>	BUSY
PC6	I/O	==>	$\overline{\text{ACK}}$
PC7	I/O		
PA0-PA7		==>	DATA

กำหนด PC6 เป็นสัญญาณ $\overline{\text{ACK}}$ โดยจัดการส่ง PULSE ออกไปโดย SOFTWARE ในการที่เราออกแบบให้ การรับข้อมูลแบบขนานสามารถส่ง DATA จาก PRINTER ADAPTER ได้ก็ต้องการให้เป็นมาตรฐาน คือสามารถจะรับ DATA จากรูปแบบของการส่งแบบ PRINTER ADAPTER ได้

วงจรส่วนแสดงผล

ในวงจรส่วนแสดงผลที่ได้ออกแบบนั้นต้องใช้ต่อกับจอแบบ COMPOSITE ที่ใช้กับ COMPUTER ทั่วไป ในการแสดงผลสามารถเลือกแสดงผลได้ทั้ง 80 COLUMN และ 40 COLUMN มีทั้งหมด 25 บรรทัด แบ่งเป็น 4 PAGE และสามารถเลือกโหมด หรือเลือก PAGE ได้ ในการแสดงผลจะใช้ SOFTWARE ทั้งหมดโดยส่ง DATA ไปที่ PORT OFDH ในส่วน CONNECTER ที่ต่อกับวงจรส่วนแสดงผลเป็นแบบ 40 ขา (แบบ ของ Z80J ซึ่งสามารถต่อกับ SINGLE BOARD ใด ๆ ที่ไม่ใช่ PORT OFFH, OFEH, OFDH และ MEMORY ส่วนที่ตำแหน่ง 8000H-9FFFH และมี Z80 เป็น CPU ในวงจรที่ได้ออกแบบใช้ CRTC เบอร์ 6845 ในการอ่านเขียน SCREEN MEMORY เราออกแบบให้อ่านเขียนได้เฉพาะช่วง RETRACE หรือ ในช่วงใด ๆ ที่อยู่นอกขอบเขตการแสดงผลบนจอภาพโดยเช็คได้จากสัญญาณ DISPEN ของ 6845 คือถ้า CPU ต้องการอ่านเขียน SCREEN MEMORY ในช่วงที่สัญญาณ DISPEN เป็น "1" คืออยู่ในขอบเขตของการแสดงผลบนจอภาพจะให้ CPU คอยก่อนจนกว่าสัญญาณ DISPEN เป็น "0" CPU จึงจะอ่านเขียนได้ด้วยวิธีการนี้จะไม่ทำให้เกิด SNOW บนจอภาพขณะส่งข้อมูลไปแสดงผล ในวงจรนี้ใช้ SCREEN MEMORY ขนาด 8K คือ เบอร์ 6264 แต่ในการแสดงผลครั้งหนึ่งจะใช้เพียง 2K จึงทำให้แบ่งเป็น 4 PAGE ได้คือ

PAGE 1 : 8000H-87FFH

PAGE 2 : 8800H-8FFFH

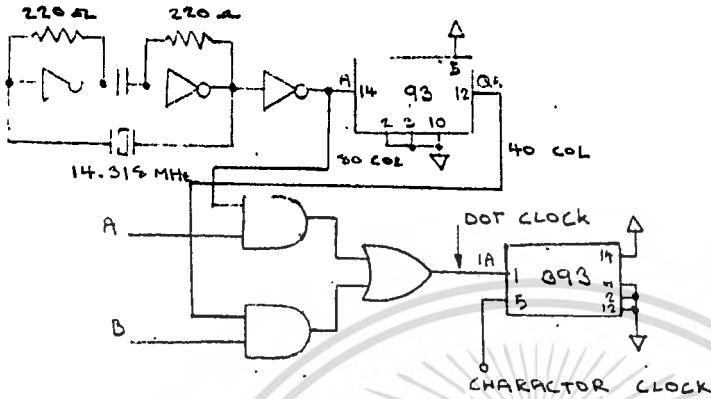
PAGE 3 : 9000H-97FFH

PAGE 4 : 9800H-9FFFH

CHARACTOR GENERATOR ใช้ขนาด 2K คือใช้ EPROM เบอร์ 2716

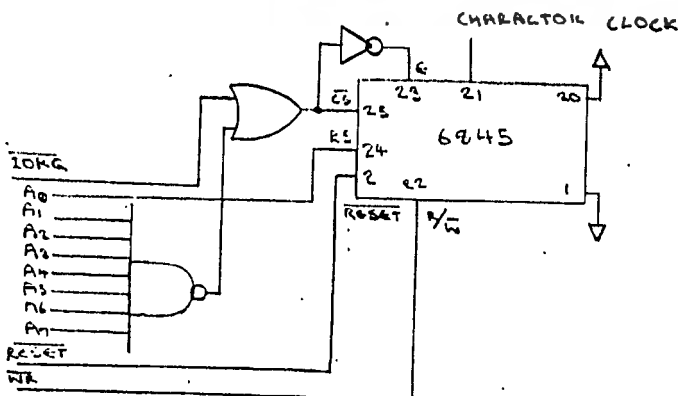
รายละเอียดของวงจร

1. วงจร CLOCK



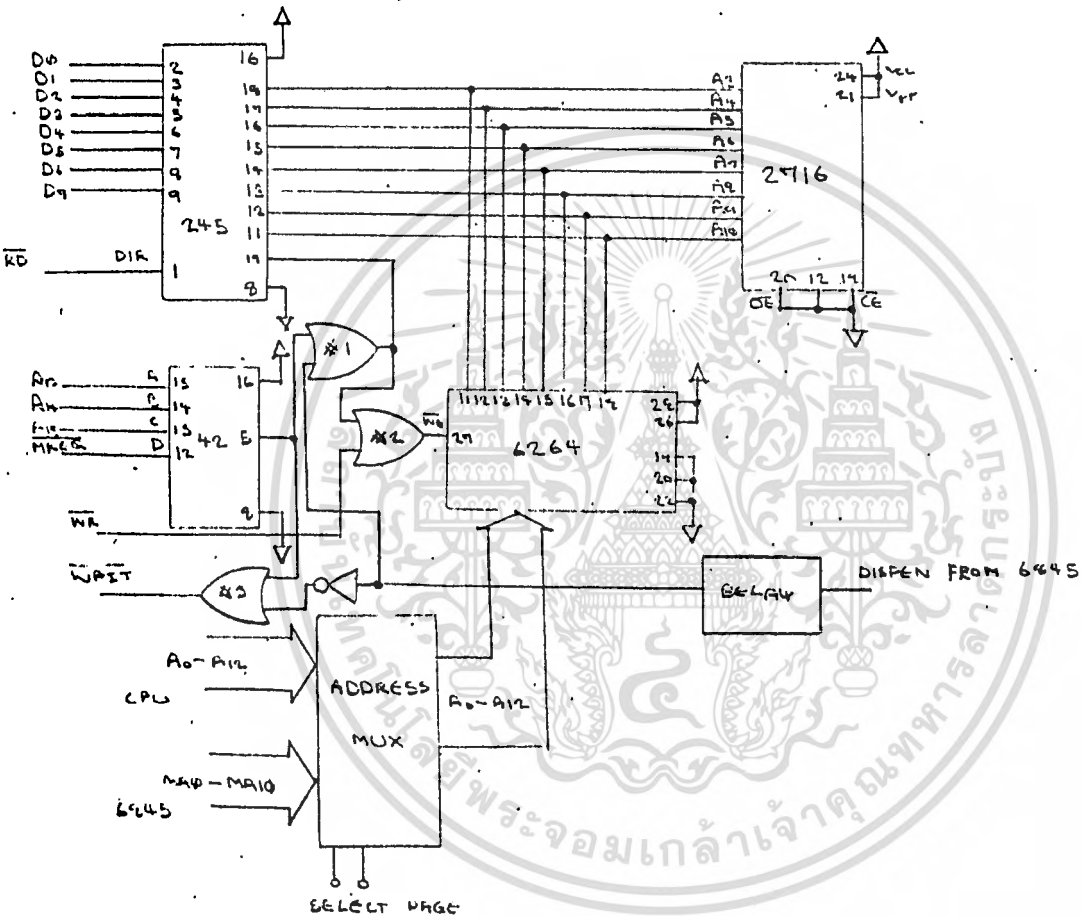
ในการที่เราจะเลือกโหมด 40 COL หรือ 80 COL จะใช้วิธีเปลี่ยนความถี่ของ DOT CLOCK คือในขณะ 80 COL จะใช้ 14.318 MHZ และในขณะ 40 COL จะใช้เพียงครึ่งหนึ่งโดยใช้ 74LS93 เป็นตัวหารและในการเลือกจะเลือกผ่าน PORT OFDH โดยในขณะขา A เป็น "1" ขา B เป็น "0" จะเป็นแบบ 80 COL ถ้า A เป็น "0" ขา B เป็น "1" จะเป็นแบบ 40 COL ซึ่งในการเปลี่ยนโหมด นอกจาก SET ตรงส่วนนี้แล้วจะต้องโปรแกรม CRTIC 6845 ใหม่ด้วย

2. ส่วนติดต่อและโปรแกรม CRTIC 6845



PORT ที่ใช้ติดต่อกับ CRTIC 6845 คือ PORT 0FFH และ PORT 0FEH โดย PORT 0FFH ใช้ในการเข้าถึง ADDRESS REGISTER ส่วน PORT 0FFH (A0=1) เป็นการเข้าถึง REGISTER โดยตรง.

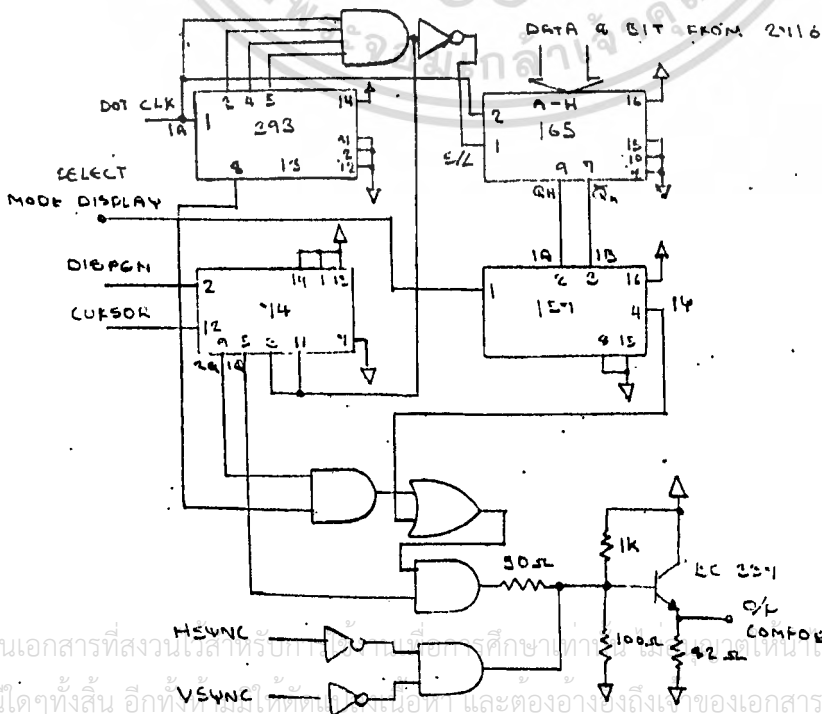
3. ส่วนควบคุมการเข้าถึง SCREEN MEMORY ของ CPU กับ CRTIC 6845



จากวงจรจะเห็นว่าในกรณีที่ CPU ไม่ต้องการอ่านเขียน SCREEN MEMORY 74LS245 และส่วนของ ADDRESS MUX ซึ่งก็คือ 74LS157 จำนวน 4 ตัว จะปิดกั้นขา DATA และ ADDRESS จาก CPU ฉะนั้น MA0-MA10 จะเข้าถึง SCREEN MEMORY ซึ่งก็คือ RAM เบอร์ 6264 ส่วน DATA จาก 6264 ก็จะถูกส่งให้ 2716 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาติให้นำไปใช้ประโยชน์ด้านการค้า ซึ่งก็คือ CHARACTER GENERATOR เพื่อดึงเอา รูปแบบของ CHARACTER ออกไป

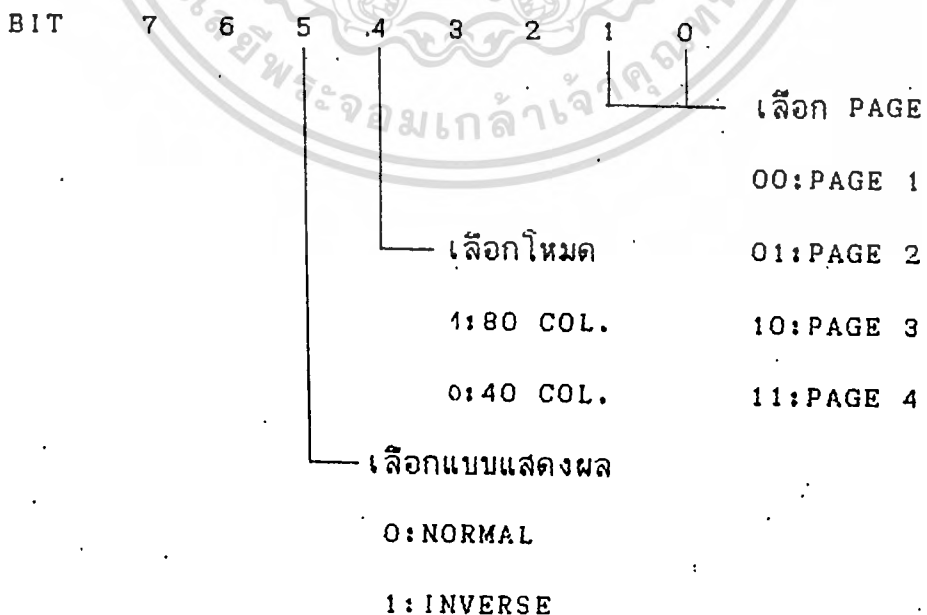
DISPLAY จะเห็นว่าเราใช้ MA0-MA10 เท่านั้น ในการเข้าถึง 6264 ซึ่งจะอ้างได้เพียง 2K ยังขาดสาย ADDRESS อีก 2 เส้น ซึ่งที่เหลืออีก 2 เส้นนี้เราใช้ในการเลือก PAGE ของ SCREEN MEMORY คือจะเป็นการเข้าถึง MEMORY ในช่วงของ ADDRESS ต่าง ๆ กัน ตามตำแหน่งของแต่ละ PAGE เมื่อ CPU ต้องการเข้าถึง RAM [SCREEN MEMORY] เมื่อ CPU อ้างถึง ADDRESS ในช่วง 8000H-9FFFH 74LS42 จะให้ขา 5 ACTIVE คือ เป็น '0' จากนั้นจะต้องดูที่ DISPEN ของ 6845 ด้วยว่าเป็น '1' หรือ '0' ถ้าเป็น '0' ก็จะแสดงว่าอยู่ในช่วง BLANK สามารถเข้าถึง SCREEN MEMORY ได้ OUTPUT ของ OR#1 จะเป็น '0' จะไปเปิด BUFFER 74LS245 พร้อมกันนั้นจะถูกส่งไปยังส่วนของวงจร ADDRESS MUX. เปิดให้ ADDRESS ของ CPU คือ A0-A12 ผ่านไปยัง 6264 ก็สามารถอ่านเขียนกับ SCREEN MEMORY ได้ แต่ถ้า DISPEN เป็น '1' ในช่วงนี้จะเห็นว่า OUTPUT ของ OR#1 จะเป็น '1' และ OUTPUT ของ OR#3 จะเป็น '0' ส่งสัญญาณ WAIT ไปให้ CPU คอยอยู่จนกว่า สัญญาณ DISPEN จะเป็น '0' ซึ่งในช่วงที่คอยนี้ 6845 ก็จะต้องคงให้ ADDRESS แก่ SCREEN MEMORY อยู่ จนกว่า DISPEN จะเป็น '0'

4. วงจร SHIFT DATA ออกไปแสดงผล



เมื่อ 6845 ให้ ADDRESS แก่ 6845 โดยจะนับไปเรื่อย ๆ จนครบจำนวนที่โปรแกรมได้ใน REGISTER, DATA จาก 6264 จะถูกส่งไปที่ 2716 ซึ่งเป็น CHARACTER GENERATOR นำไปประกอบกับ RAO-RA2 จาก 6845 ก็จะได้เป็นค่า ADDRESS ให้ 2716 เพื่อดึงเอา DATA ใน 2716 ซึ่งเป็นรูปแบบของตัวอักษรส่งไปที่ 74LS165 ซึ่งทำหน้าที่ เปลี่ยน DATA จากขนานไปเป็นอนุกรม OUTPUT ของ 74LS165 มี 2 ขา คือ Q_n และ \bar{Q}_n ซึ่งนำมาให้ 74LS157 และใช้ขา 1 ของ 74LS157 ในการเลือกว่าจะ DISPLAY แบบปกติหรือแบบอินเวอร์สจากนั้น OUTPUT ของ 74LS157 จะนำไป OR กับสัญญาณ CURSOR ที่ผ่านวงจร DELAY และควบคุมให้ กระพริบแล้วจากนั้นไป AND กับสัญญาณ DISPEN ที่ผ่านการ DELAY แล้วจากนั้นจะถูกส่งไปยังวงจรสร้างเป็นสัญญาณไปยังวงจรสร้างเป็นสัญญาณ COMPOSITE โดยใน ส่วนวงจรสร้างสัญญาณ COMPOSITE จะนำสัญญาณ VIDEO รวมกับสัญญาณ HSYNC และ VSYNC

การเลือกโหมดในการแสดงผลจะใช้วิธีผ่านค่าไปยัง PORT OFDH โดยแต่ละ BIT จะมีความหมายดังนี้



ค่าที่ใช้ในการโปรแกรม REGISTER ของ 6845

REGISTER	MODE 80 COL.	MODE 40 COL.
R0	78H	3EH
R1	50H	28H
R2	5AH	30H
R3	0EH	0DH
R4	26H	26H
R5	0H	0H
R6	19H	19H
R7	1FH	1FH
R8	0H	0H
R9	7H	7H
R10	0H	0H
R11	7H	7H
R12	0H	0H
R13	0H	0H
R12	0H	0H
R14	0H	0H
R15	0H	0H

การทำงานโดยรวมวงจรทั้งสองส่วน

เมื่อเริ่มเปิดเครื่องส่วน CPU จะทำการเริ่มต้น INITIALIZE ตัวเองเสร็จแล้วจะ
เริ่มทำงานในโปรแกรม MONITOR ที่ ADDRESS 0 ซึ่งจะทำการโปรแกรม CRTC
6845 ให้แสดงผลในโหมด 80 COL แบบ NORMAL และจะคอยรับ DATA จาก CONNE
CTOR ที่จะส่งมาโดยวิธี POLLING โดยจะมีค่าเป็น CONTROL CODE หรือ DATA
ถ้าเป็น DATA ก็จะส่งไปยัง SCREEN MEMORY ถ้าเป็น CONTROL CODE จะมีหน้า

ที่ต่าง ๆ มีดังนี้

00H	==>	CLEAR SCREEN
01H	==>	เลือก PAGE 1
02H	==>	เลือก PAGE 2
03H	==>	เลือก PAGE 3
04H	==>	เลือก PAGE 4
05H	==>	แสดงผลแบบ INVERSE
06H	==>	แสดงผลแบบ NORMAL
07H	==>	MODE 80 COLUMN
08H	==>	MODE 40 COLUMN
09H	==>	GOTOxy
0AH	==>	LINEFEED
0DH	==>	CARRIAGE RETURN
0EH	==>	CURSOR ON
0FH	==>	CURSOR OFF

สมมติว่า CPU รับ 41H เข้ามาจะเห็นว่าไม่ใช่พวก CONTROL CODE ฉะนั้น 41H จะถูกส่งไปยัง SCREEN MEMORY ในตำแหน่ง CURSOR ขณะนั้น และ CURSOR ก็ จะเลื่อนไปยังตำแหน่งถัดไป DATA ตัวที่ถูกส่งต่อมาก็จะถูกพิมพ์ต่อไป เราสามารถใช้ CODE 09H ซึ่งเป็นคำสั่ง GOTOxy คือกำหนดตำแหน่งที่ต้องการจะพิมพ์ตัวอักษรในตำแหน่งที่ต้องการบนจอภาพ ในการใช้คำสั่งนี้จะส่ง 09H และ ตามด้วย COLUMN และ ROW ส่วน CODE อื่นๆ จะเป็น CODE เต็มๆ ไม่ต้องตามด้วย DATA อะไรอีก

ถ้านำส่วนของ CPU ซึ่งจะมี CONNECTOR แบบเดียวกับเครื่อง PRINTER นำสาย PRINTER CABLE ไปต่อกับ PRINTER PORT ของ COMPUTER เมื่อเราส่งอะไรไปพิมพ์เหมือนกับส่งไปเครื่องพิมพ์ทั่วไป ส่วนของวงจรจะนำข้อมูลนั้นไปแสดงบนจอภาพทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดเห็นประโยชน์ในการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากผู้จัดทำเอกสารนี้ให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ช่วยในการพัฒนา SOFTWARE

อุปกรณ์ที่มีความสำคัญมากในการพัฒนา SOFTWARE คือ DOWNLOAD CARD และ CROSS ASSEMBLER ซึ่งในส่วนของ CPU ในโครงงานนี้ได้ใช้ Z80 ซึ่งพอติดบนเครื่อง IBM มี CROSS ASSEMBLER Z80 พอดี จึงได้ออกแบบ DOWNLOAD CARD บน IBM ในเครื่อง IBM มีหน่วยความจำว่างในช่วง A0000H-AFFFFH (ในกรณีมีหน่วยความจำ 640k) จึงสามารถใช้หน่วยความจำในช่วงนี้ได้ DOWNLOAD CARD ที่ได้ออกแบบขึ้นใช้หน่วยความจำ 8K ใช้ RAM เบอร์ 6264 เพื่อความสะดวกในการ DECODE แอดเดรส จะเลือกตำแหน่งที่ AC000H-ADFFFH โดยใช้ 74LS138 ในการดีโค๊ด ส่วน 74LS157 ใช้เป็นตัวเลือกว่าจะใช้แอดเดรส และสัญญาณ CONTROL อื่น ๆ จาก IBM หรือจาก DIP JUMPER เข้าถึง RAM 6264 โดยมีสัญญาณ DECODE จาก 74LS138 ไปเข้าที่ขา 1 ของ 74LS157 ทุกตัวในวงจรที่ออกแบบ ได้ออกแบบไว้ว่าในกรณีที่ IBM ไม่ได้อ่านเขียน RAM บน DOWNLOAD CARD 74LS157 จะเปิดให้สายแอดเดรสและสัญญาณ CONTROL จาก DIP JUMPER เข้าถึง RAM ส่วน 74LS245 เป็นตัวเปิด-ปิด DATA ในการอ่านเขียน RAM โดย 74LS245#1 จะถูกควบคุมจากสัญญาณ DECODE แอดเดรสจาก 74LS138 ส่วน 74LS245#2 จะใช้สัญญาณ CE จากภายนอกในการ อินาเบิล คือ ในกรณีที่วงจรที่เรา กำลังพัฒนา SOFTWARE ไม่ต้องการอ่าน DATA 74LS245#2 ก็จะทำให้สาย DATA ที่ต่อกับ DIP JUMPER เป็น HIGH IMPEDANT จะได้ไม่ไปรบกวนส่วนของวงจรส่วนอื่น ส่วน 74LS245#1 จะเปิดให้ DATA ผ่านก็ต่อเมื่อ IBM ต้องการอ่านเขียน RAM เท่านั้น ซึ่ง 74LS245 ทั้ง 2 ตัวจะเป็นอิสระต่อกัน

ส่วน DIP SWITCH มีไว้ในการเลือกเบอร์ของ EPROM ซึ่งจะใช้ในวงจรที่กำลังพัฒนาซึ่งทำให้ DOWNLOAD CARD มีขอบข่ายการใช้งานกว้างขึ้น ในการใช้งานนั้นเราจะใช้ DOWNLOAD CARD เสียบบน SLOT ใด SLOT หนึ่งบนเครื่อง

IBM ในการเขียนโปรแกรมเราจะใช้ SIDEKICK เขียนเสร็จแล้วจะ ASSEMB ด้วย CROSS ASSEMBLER ได้ MACHINE CODE จากนั้นจะใช้ DEBUG LOAD MACHINE

CODE นั้นแล้ว MOVE ไปไว้ที่ ADDRESS ACOO:0 ถึงตอนนี้เราก็จะใช้ DIP JUMPER ไปเขียนในตำแหน่ง EPROM ซึ่งอาจจะเป็นตัวทำหน้าที่เป็น MONITOR PROGRAM หรืออะไรก็ตามเมื่อวงจรที่กำลังพัฒนาอ่าน EPROM ตำแหน่งนี้ก็จะได้อัฒมจาก RAM บน DOWNLOAD CARD ไปซึ่งเราสามารถแก้ไขได้สะดวกเมื่อโปรแกรมที่เราพัฒนาสมบูรณ์จึงจะนำไปเก็บใน EPROM อีกทีหนึ่ง



บทที่ 6

สรุปผลและวิจารณ์

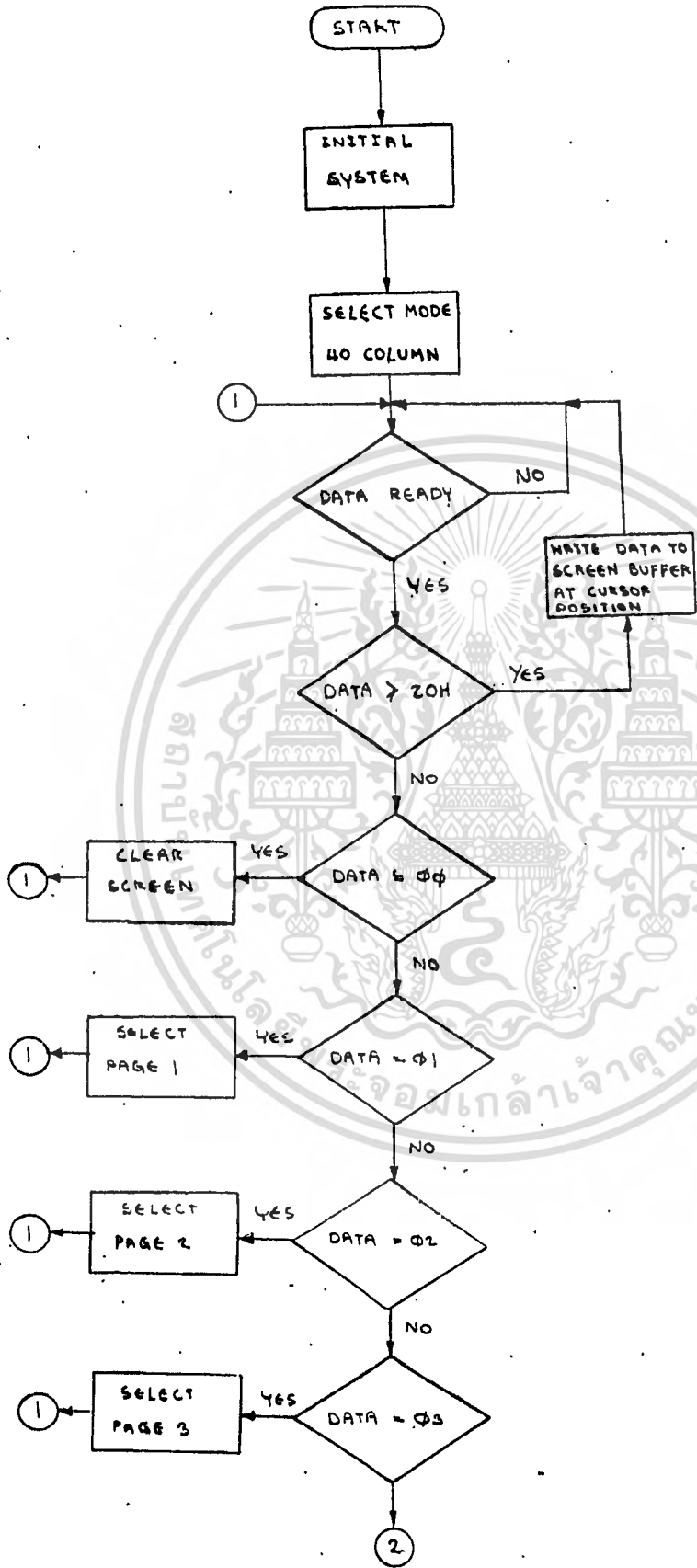
โครงการนี้ได้บรรลุผลตามเป้าหมายที่ตั้งไว้ทุกประการ คือจะทำการรับ DATA จาก PRINTER PORT เข้ามาแล้ว CPU Z80 จะทำการตรวจเช็คว่าเป็น CONTROL CODE ถ้าใช่ก็จะทำงานตาม CODE เหล่านั้น เช่น เปลี่ยน PAGE แสดงผลซึ่งมีอยู่ 4 PAGE, เปลี่ยนรูปแบบแสดงผล (INVERSE, NORMAL), ควบคุม CURSOR (ON, OFF, GOTOxy) เป็นต้น มิฉะนั้นแล้วจะนำเอา DATA ที่รับมานี้ไปปรากฏบนจอภาพที่ตำแหน่ง CURSOR ขณะนั้น จากนั้นจะคอยรับค่าเข้ามาใหม่เป็นเช่นนี้เรื่อยไป

ในตอนแรกการทำงานของวงจรมีปัญหาอยู่บ้างกล่าวคือมี NOISE หรือ SNOW มาปรากฏบนจอภาพด้วยและได้ทำการแก้ไขโดยการใส่ C (CAPACITOR) เข้าไปในบางจุดของวงจรก็ปรากฏว่า NOISE หรือ SNOW นี้ได้หายไปและทำงานได้ดี

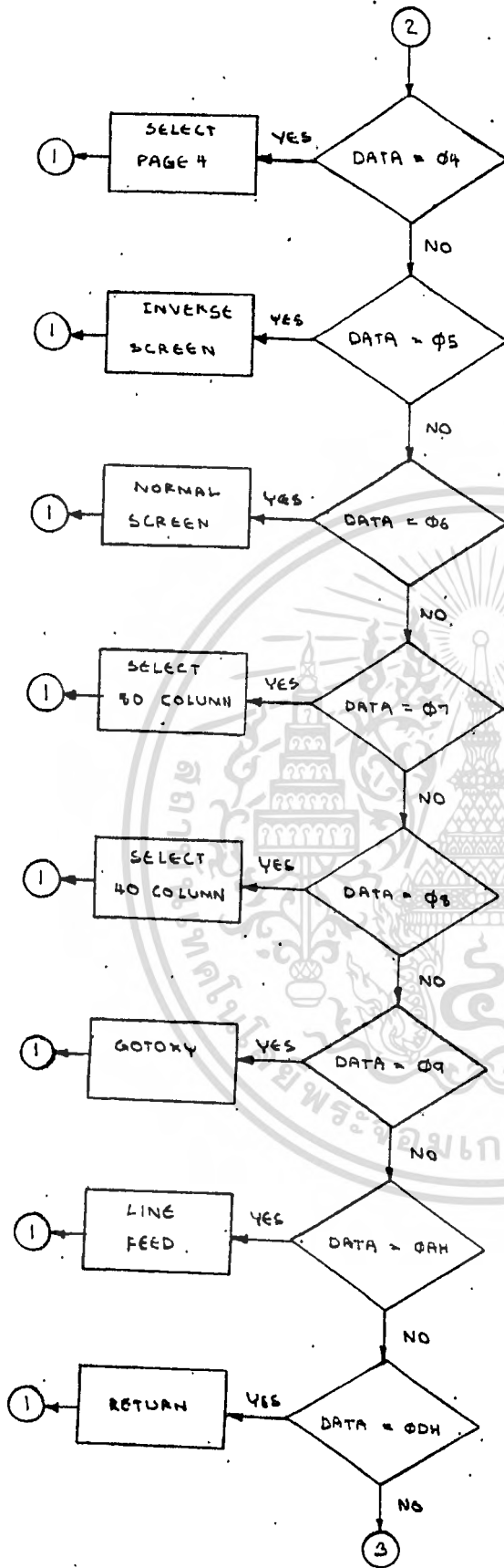
สำหรับโครงการนี้สามารถนำไปประยุกต์ใช้งานอื่น ๆ ได้อีกด้วยโดยไม่จำเป็นว่าจะต้องต่อร่วมกับอีก 3 โครงการที่กล่าวในตอนต้นดังนี้

1. ส่วน CPU สามารถนำไปต่อ I/O PORT ใช้งานได้ทันทีเพราะได้ออกแบบเพื่อไว้ให้แล้วเพียงแต่เปลี่ยน MONITOR PROGRAM ใน EPROM 2764 ใหม่เท่านั้น
2. ส่วนแสดงผล สามารถนำไปต่อกับ SINGLE BOARD ที่มี Z80 เป็น CPU และมี MEMORY ตำแหน่ง 8000H-9FFFH และ PORT FDH, FEH, FFH วางอยู่ก็สามารถใช้งานได้ทันที (MPF 1, ET BOARD)

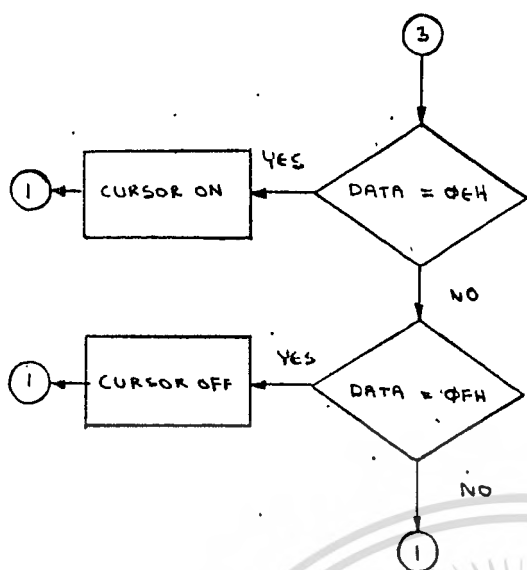
FLOW CHART. ของ MONITOR PROGRAM.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

..... MONITOR PROGRAM .....
PROJECT GRAPHIC DISPLAY SYSTEM
... DISPLAY PART...
SCREEN MEMORY 8000H-9FFFH (6264)
PORT 0FH = SET MODE
PORT FEH = ADDRESS REGISTER
PORT FFH = DATA REGISTER
FUNCTION
- 40 COLUME
- 80 COLUME
- 4 PAGE
- NOLMAL
- INVERSE
...CPU PART...
MEMORY
0000H-1FFFH => EPROM (2764)
2000H-3FFFH => RAM (6264)
ADDRESS PORT 8255 PIA
PORT 00 = PORT A
PORT 01 = PORT B
PORT 02 = PORT C
PORT 03 = CONTROL

```

```

PORT MODE EQU 0FDH
PORT_ADR EQU 0FEH
PORT_DATA EQU 0FFH
INIT_PAGE EQU 0
INIT_STATUS EQU 0
CONTROL EQU 0B0H
CLS EQU 00H
S_P1 EQU 01H
S_P2 EQU 02H
S_P3 EQU 03H
S_P4 EQU 04H
INV_SCREEN EQU 05H
NOR_SCREEN EQU 06H
COL_80 EQU 07H
COL_40 EQU 08H
B_YF EQU 09H
L_F EQU 0AH
R_T EQU 0BH
CUR_ON EQU 0EH
CUR_OFF EQU 0FH
; MAIN PROGRAM

```

```

.....
ORG 0000H
NOP
LD SP,3FFFH
LD A,CONTROL
; SET PORT A I/P (HANDSHAKING)
; PORT B O/P
; PC4 = -STB
; PC5 = +IBF
; PC6 = -ACK
OUT (03H),A
LD HL,DATA
LD (HL),INIT_PAGE
INC HL
LD (HL),INIT_STATUS
CALL MODE_80
; POLLING .....
POLL: CALL POLLING
CP 20H
JR NC,NON_CONTROL
CP CLS
JR NZ,NEW1
CALL CLSCREEN ; CLEAR SCREEN
JR POLL
NEW1: CP S_P1
JR NZ,NEW2

```



7179K312x 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL PAGE1      ; SELECT PAGE1
JR POLL
NEW2: CP S P2
JR NZ,NEW3
CALL PAGE2      ; SELECT PAGE2
JR POLL
NEW3: CP S P3
JR NZ,NEW4
CALL PAGE3      ; SELECT PAGE3
JR POLL
NEW4: CP S P4
JR NZ,NEW5
CALL PAGE4      ; SELECT PAGE4
JR POLL
NEW5: CP INV SCREEN
JR NZ,NEW6
CALL INVERSE    ; INVERSE SCREEN
JR POLL
NEW6: CP NOR SCREEN
JR NZ,NEW7
CALL NORMAL     ; NORMAL SCREEN
JR POLL
NEW7: CP COL 80
JR NZ,NEW8
CALL MODE_80    ; SELECT 80 COLUMN
JR POLL
NEW8: CP COL 40
JR NZ,NEW9
CALL MODE_40    ; SELECT 40 COLUMN
JR POLL
NEW9: CP G XY
JR NZ,NEW10
CALL GOTOXY     ; GOTOXY ... READ COL... ROW...
JR POLL
NEW10: CP L F
JR NZ,NEW11
CALL LINE_FEED  ; LINE FEED
JR POLL
NEW11: CP R T
JR NZ,NEW12
CALL RETURN     ; RETURN
JR POLL
NEW12: CP CUR ON
JR NZ,NEW13
CALL CURSOR_ON
JR POLL
NEW13: CP CUR OFF
JR NZ,POLL
CALL CURSOR_OFF
JP POLL
NON_CONTROL: CALL WRITE
JP POLL
HALT
; END MAIN PROGRAM....
; HIGH BYTE OF SCREEN BUFFER PAGE
DATA_P:  DB 80H
        DB 88H
        DB 90H
        DB 98H

; POLLING PORT A (RETURN VALUE IN A)
POLLING: IN A,(02H)
        BIT 5,A      ; CHECK IBF
        JR Z,POLLING ; IF NO DATA GOTO POLLING
        IN A,(00H)
        PUSH AF
        ; SENT -ACK SIGNAL
        -----
        IN A,(02H)
        RES 6,A

```

```
OUT (02H),A
SET 6,A
OUT (02H),A
```

```
-----
POP AF
RET
```

```
; SELECT MODE 40 COLUME
```

```
MODE_40: LD C,PORT DATA
LD HL,MODE40
XOR A
```

```
LOOP: OUT (PORT_ADR),A
OUTI
INC A
CP 10H
JR NZ,LOOP
LD HL,DATA+1
RES 4,(HL)
LD A,(HL)
OUT (PORT_MODE),A
CALL CLSCREEN
RET
```

```
MODE40: DB 3EH,28H,30H,0DH,26H,0H,19H,1FH,0H,7H,0H,7H,0H,0H,0H,0H
```

```
; SELECT MODE 80 COLUME
```

```
MODE_80: LD C,PORT DATA
LD HL,MODE80
XOR A
```

```
LOOP1: OUT (PORT_ADR),A
OUTI
INC A
CP 10H
JR NZ,LOOP1
LD HL,DATA+1
SET 4,(HL)
LD A,(HL)
OUT (PORT_MODE),A
CALL CLSCREEN
RET
```

```
MODE80: DB 78H,50H,5AH,0EH,26H,0H,19H,1FH,0H,7H,0H,7H,0H,0H,0H,0H
```

```
CURSOR_OFF: LD A,0AH
OUT (PORT_ADR),A
LD A,7
OUT (PORT_DATA),A
LD A,0BH
OUT (PORT_ADR),A
LD A,9
OUT (PORT_DATA),A
RET
```

```
CURSOR_ON: LD A,0AH
OUT (PORT_ADR),A
LD A,0
OUT (PORT_DATA),A
LD A,0BH
OUT (PORT_ADR),A
LD A,7
OUT (PORT_DATA),A
RET
```

```
; GOTOXY READ DATA 1. COLUME ,2.ROW FROM PORT A .
```

```
GOTOXY: CALL POLLING
LD C,A ; COLUME
CALL POLLING
LD B,A ; ROW
LD HL,DATA+1
BIT 4,(HL)
JR Z,MODE40
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD E,50H
JR G01
M040: LD E,20H
G01:  XOR A
      LD D,A
      LD HL,0H
G02:  CP B
      JR NC,G03
      ADD HL,DE
      DEC B
      JR G02
G03:  ADD HL,BC
      CALL S_CURSOR
      RET

```

; SEND CHARACTER IN A TO DISPLAY'

```

WRITE: EX AF,AF'
      CALL W_PAGE
      PUSH HC
      CALL W_CURSOR
      POP DE
      PUSH HL
      ADD HL,DE
      EX AF,AF'
      LD (HL),A
      POP HL
      INC HL
      PUSH HL
      LD HL,DATA+1
      BIT 4,(HL)
      JR Z,M5_40
      LD DE,07B0H
      LD BC,07B0H
      JR E1
M5_40: LD DE,03EBH
      LD BC,03C0H
E1:   AND A
      POP HL
      PUSH HL
      SBC HL,DE
      JR NC,E2
      POP HL
      CALL S_CURSOR
      RET
E2:   POP HL
      LD H,B
      LD L,C
      CALL S_CURSOR
      CALL SCROLL_UP
      RET

```

; SELECT PAGE TO DISPLAY

```

PAGE1: LD HL,DATA
      LD (HL),0
      INC L
      RES 0,(HL)
      RES 1,(HL)
      LD A,(HL)
      OUT (PORT_MODE),A
      RET

```

```

PAGE2: LD HL,DATA
      LD (HL),1
      INC L
      SET 0,(HL)
      RES 1,(HL)
      LD A,(HL)
      OUT (PORT_MODE),A
      RET

```

```

PAGE3: LD HL,DATA

```



```

LD (HL),2
INC L
RES 0,(HL)
SET 1,(HL)
LD A,(HL)
OUT (PORT_MODE),A
RET

```

```

PAGE4: LD HL,DATA
LD (HL),3
INC L
SET 0,(HL)
SET 1,(HL)
LD A,(HL)
OUT (PORT_MODE),A
RET

```

```

; INVERSE SCREEN
INVERSE: LD HL,DATA+1
SET 5,(HL)
LD A,(HL)
OUT (PORT_MODE),A
RET

```

```

; NORMAL DISPLAY
NORMAL: LD HL,DATA+1
RES 5,(HL)
LD A,(HL)
OUT (PORT_MODE),A
RET

```

```

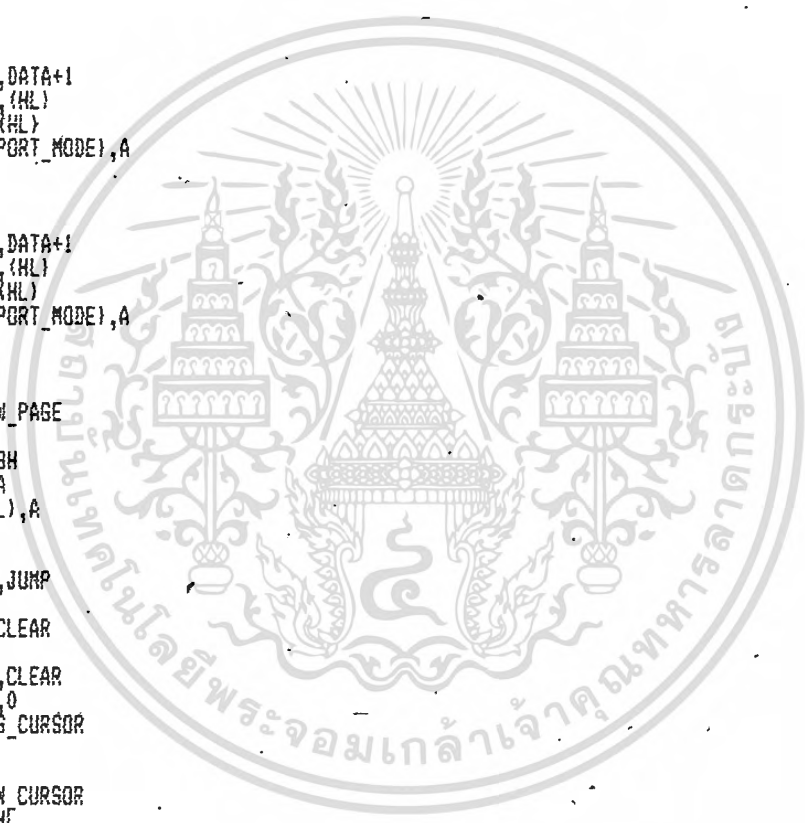
; CLEAR SCREEN
CLSSCREEN: CALL W_PAGE
XOR A
LD C,8H
LD B,A
LD (HL),A
INC L
CP L
JR NZ,JUMP
INC H
DJNZ CLEAR
DEC C
JR NZ,CLEAR
LD HL,0
CALL S_CURSOR
RET

```

```

RETURN: CALL W_CURSOR
PUSH HL
PUSH HL
LD HL,DATA+1
BIT 4,(HL)
JR Z,M40
LD C,50H
JR A1
M40: LD C,20H
A1: LD B,0H
AND A
POP HL
PUSH HL
SBC HL,BC
JR C,OK
POP DE
JR A2
OK: POP DE
POP HL
AND A

```



```
SBC HL,DE
CALL S_CURSOR
RET
```

```
LINE_FEED: CALL W_CURSOR
            PUSH HL
            LD HL,DATA+1
            BIT 4,(HL)
            JR Z,M 40
            LD C,50H
            LD DE,07D0H
            JR AS
M_40:      LD C,28H
            LD DE,03E8H
A3:        LD B,0
            POP HL
            ADD HL,BC
            PUSH HL
            AND A
            SBC HL,DE
            JR NC,OK1
            POP HL
            JR AS
OK1:       POP HL
            CALL SCROLL_UP
            JR EZ
AS:        CALL S_CURSOR
EZ:        RET
```

```
; FIND START ADDRESS (RETURN VALUE IN HL)
W_PAGE:    LD HL,DATA P
            LD A,(DATA)
            ADD A,L
            JR NC,NO_CARRY
            INC H
NO_CARRY:  LD L,A
            LD A,(HL)
            LD H,A
            XOR A
            LD L,A
            RET
```

```
; FIND CURSOR POSITION (RETURN VALUE IN HL)
W_CURSOR:  LD A,0EH
            OUT (PORT_ADR),A
            IN A,(PORT_DATA)
            LD H,A
            LD A,0FH
            OUT (PORT_ADR),A
            IN A,(PORT_DATA)
            LD L,A
            RET
```

```
; SET CURSOR POSITION (PASS VALUE IN HL)
S_CURSOR   LD A,0EH
            OUT (PORT_ADR),A
            LD A,H
            OUT (PORT_DATA),A
            LD A,0FH
            OUT (PORT_ADR),A
            LD A,L
            OUT (PORT_DATA),A
            RET
```

```
SCROLL_UP: CALL W_PAGE
            PUSH HL
            LD HL,DATA+1
            BIT 4,(HL)
            JR Z,M1 40
            LD DE,00
            LD BC,0780H
            JR A11
M1_40:     LD DE,40
```

M1_40: สารนี้เป็นลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

A11: LD BC,03C0H
      POP HL
      PUSH DE
      PUSH HL
      PUSH BC
      PUSH HL
      ADD HL,DE
      POP DE
      LDIR
      POP HL
      CALL S_CURSOR
      POP BC
      ADD HL,BC
      POP BC
      LD B,C
      XOR A
DEL_L: LD (HL),A
      INC L
      CP L
      JR NZ,J1
      INC H
J1:   DJNZ DEL_L
      RET

```

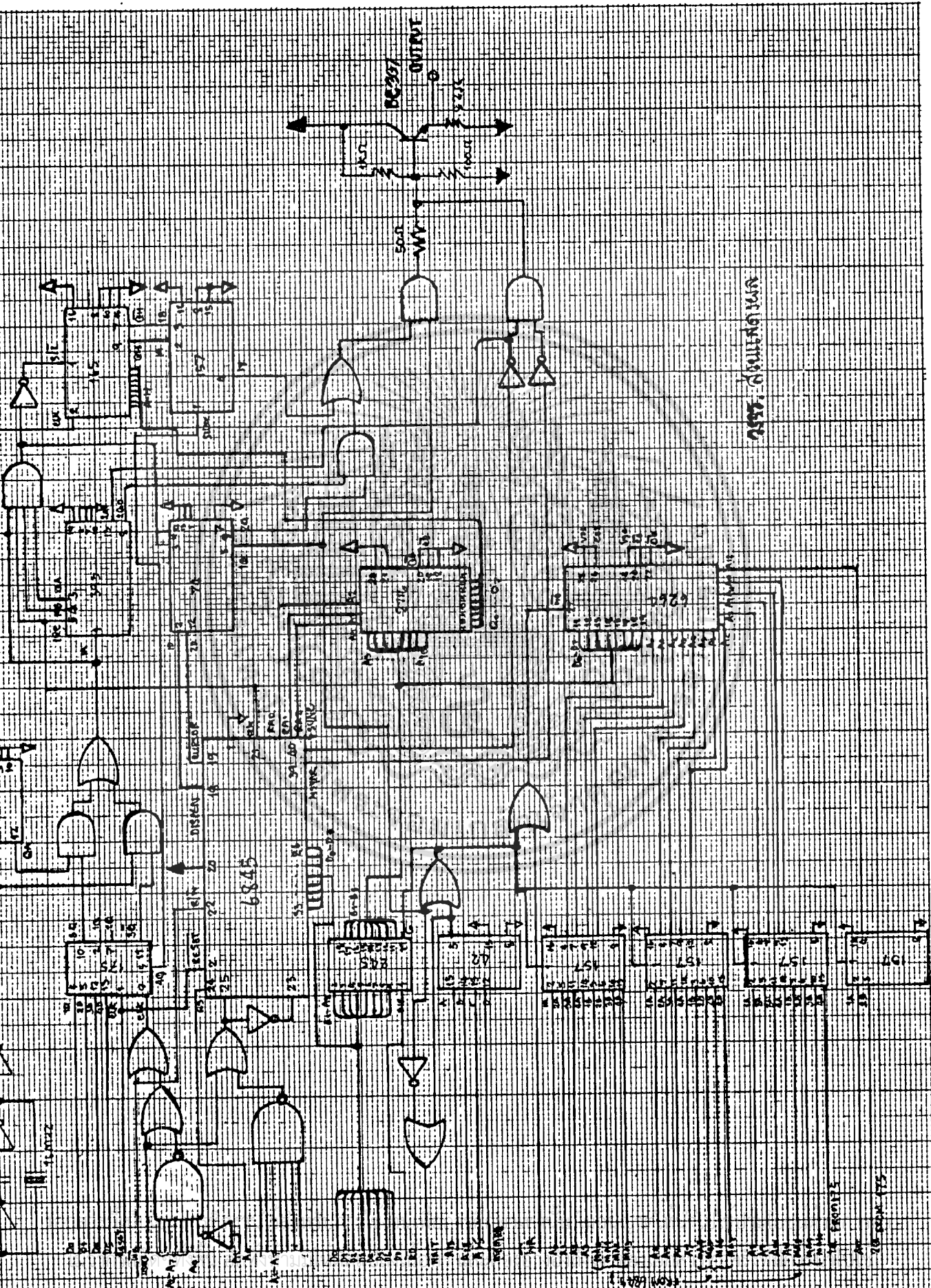
```

DATA: ORG 2000H
      END

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2585. 011110111011010

6845

FROM BOARD 1
 FROM BOARD 2
 FROM BOARD 3
 FROM BOARD 4
 FROM BOARD 5
 FROM BOARD 6
 FROM BOARD 7
 FROM BOARD 8
 FROM BOARD 9
 FROM BOARD 10
 FROM BOARD 11
 FROM BOARD 12
 FROM BOARD 13
 FROM BOARD 14
 FROM BOARD 15
 FROM BOARD 16
 FROM BOARD 17
 FROM BOARD 18
 FROM BOARD 19
 FROM BOARD 20
 FROM BOARD 21
 FROM BOARD 22
 FROM BOARD 23
 FROM BOARD 24
 FROM BOARD 25
 FROM BOARD 26
 FROM BOARD 27
 FROM BOARD 28
 FROM BOARD 29
 FROM BOARD 30
 FROM BOARD 31
 FROM BOARD 32
 FROM BOARD 33
 FROM BOARD 34
 FROM BOARD 35
 FROM BOARD 36
 FROM BOARD 37
 FROM BOARD 38
 FROM BOARD 39
 FROM BOARD 40
 FROM BOARD 41
 FROM BOARD 42
 FROM BOARD 43
 FROM BOARD 44
 FROM BOARD 45
 FROM BOARD 46
 FROM BOARD 47
 FROM BOARD 48
 FROM BOARD 49
 FROM BOARD 50
 FROM BOARD 51
 FROM BOARD 52
 FROM BOARD 53
 FROM BOARD 54
 FROM BOARD 55
 FROM BOARD 56
 FROM BOARD 57
 FROM BOARD 58
 FROM BOARD 59
 FROM BOARD 60
 FROM BOARD 61
 FROM BOARD 62
 FROM BOARD 63
 FROM BOARD 64
 FROM BOARD 65
 FROM BOARD 66
 FROM BOARD 67
 FROM BOARD 68
 FROM BOARD 69
 FROM BOARD 70
 FROM BOARD 71
 FROM BOARD 72
 FROM BOARD 73
 FROM BOARD 74
 FROM BOARD 75
 FROM BOARD 76
 FROM BOARD 77
 FROM BOARD 78
 FROM BOARD 79
 FROM BOARD 80
 FROM BOARD 81
 FROM BOARD 82
 FROM BOARD 83
 FROM BOARD 84
 FROM BOARD 85
 FROM BOARD 86
 FROM BOARD 87
 FROM BOARD 88
 FROM BOARD 89
 FROM BOARD 90
 FROM BOARD 91
 FROM BOARD 92
 FROM BOARD 93
 FROM BOARD 94
 FROM BOARD 95
 FROM BOARD 96
 FROM BOARD 97
 FROM BOARD 98
 FROM BOARD 99
 FROM BOARD 100

กิตติกรรมประกาศ

ในนามของผู้จัดทำปริญญาโทฉบับนี้ ขอขอบคุณอาจารย์สุทัศน์ันท์ ประมาธิกุล ที่ได้ให้คำแนะนำและคำปรึกษาทางด้านวิชาการเป็นอย่างมาก รวมทั้งบรรดาเพื่อน ๆ และท่านอาจารย์หลาย ๆ ท่านที่ให้คำปรึกษาและให้ความสะดวกในด้านเครื่องมือ วัสดุ อุปกรณ์ต่าง ๆ ในการทำโครงการนี้ จึงขอขอบคุณมา ณ. ที่นี้

ผู้จัดทำ



เอกสารอ้างอิง

1. บุญเลิศ เอี่ยมทัศนาศนา, "ออกจอกแก้ว ตอน 1", ไมโครคอมพิวเตอร์, ปีที่ 1, ฉบับที่ 2, หน้า 117-121, 2527.
2. บุญเลิศ เอี่ยมทัศนาศนา, "ออกจอกแก้ว ตอน 2", ไมโครคอมพิวเตอร์, ปีที่ 1, ฉบับที่ 3, หน้า 122-127, 2527.
3. บุญเลิศ เอี่ยมทัศนาศนา, "ออกจอกแก้ว ตอน 3", ไมโครคอมพิวเตอร์, ปีที่ 1, ฉบับที่ 4, หน้า 129-134, 2527.
4. บุญเลิศ เอี่ยมทัศนาศนา, "ออกจอกแก้ว ตอน 4", ไมโครคอมพิวเตอร์, ปีที่ 1, ฉบับที่ 5, หน้า 131-140, 2527.
5. ยืน ภู่วรรณ และ วัฒนา เชียงกุล, "ไมโครโปรเฟสเซอร์ไมโครคอมพิวเตอร์" เอช เอ็น การพิมพ์.
6. ชูชัย ฉนสารตั้งเจริญ, "การใช้งาน Z-80", นิลิกส์เซ็นเตอร์การพิมพ์, 2527.
7. Gerry Kene, "CRT CONTROLLER HANDBOOK", Osborn/McGrawHill, 1984.
8. Fodny Zaks, "Programing The Z80", Third Edition, Sybex INC, 1980.