การศึกษาตัวประมวลผลทางกราฟฟิก
GRAPHIC DISPLAY CONTROLLER

โดย

นาย ฉัตรชัย อัศวธีระธรรม

นาย นิวัฒน์ นิลสุวรรณ

นาย รอภาษ สมยาวงศาสุข

บริญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมอิเลคทรอนิคส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

ปริญญานิพนธ์สำหรับภาคการศึกษาที่ 2 ปีการศึกษา 2535

ภาควิชาวิศวกรรมไฟเลคทรอนิคส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง Graphic Display Controller.

(uPD 7220)

ผู้จัดทำ

นาย ฉัตรชัย อัศวธีระธรรม 32.1062

นาย นิวัฒน์ นิลสุวรรณ 32.1150

นาย รอภาษ สมยาวงศาสุข 32.1438

อาจารย์ที่ปรึกษา

.....

ผศ. พลผดุง ผดุงกุล

Graphic Display Controller(GDC)

นาย ฉัตรชัย อัศวธีระธรรม 32.1062

นาย นิวัฒน์ นิลสุวรรณ 32.1150

นาย รอภาษ สมยาวงศาสุข 32.1438

อาจารย์ที่ปรึกษา

ผศ. พลผดุง ผดุงกุล

บทคัดย่อ

โครงงานนี้เป็นการศึกษา uPD-7220 Graphic Display Controller(GDC)ที่มีความสามารถในการควบคุมการแสดงผลรูปแบบต่างๆ เช่น

- ทำงานได้เร็ว เนื่องจากมีตัวประมวลผลคำสั่ง (Command Processor) ในตัวเองจึงไม่ต้องอาศัยความสามารถของ CPU มากนัก
- สามารถย่อ-ขยายได้ถึง 16 ระดับ
- สามารถเลื่อนดูส่วนต่างๆของภาพได้สะดวก
- ทำงานกับจอแสดงผลได้หลายประเภท แล้วแต่การออกแบบ
- หน่วยความจำสำหรับการแสดงผล (Display Memory) ขยายได้สูงสุดถึง 256K 16bit word
- ทำงานในลักษณะ Multiple Controller หรือ Multi Plan ได้ถึง 4 Plan โดยได้ความละเอียดสูงถึง 1024 x 1024 จุด

ในปัจจุบันการใช้งาน uPD-7220 ยังไม่เป็นแพร่หลาย การนำไปใช้ส่วนใหญ่เป็นไปในลักษณะการควบคุมจอภาพแบบ Multi Plan เท่านั้น จึงทำการศึกษาคุณสมบัติต่างๆของ uPD-7220 ในส่วนอื่นๆเช่น ความสามารถด้านกราฟฟิก ความเร็วในการแสดงผล การย่อ-ขยายภาพ เป็นต้น โดยได้ทำเป็นการ์ดควบคุมจอสี ความละเอียด 640x350 จุด เพื่อใช้ในการศึกษาส่วน Hardware และ Software ที่เป็นชุดคำสั่งต่างๆ ที่ช่วยในการทำงานของ uPD-7220 นี้

Graphic Display Controller(GDC)

CHARTCHAI ASAVATEERATHAM 32.1062

NIWAT NINSUWAN 32.1150

OPAS SOMYAWONGSASUK 32.1438

ADVISOR

ASST.PROF POLPHADUNG PHADUNGKUL

ABSTRACT

The subject of this Project is to study uPD7220 (Graphic Display Controller Chip) about it's features such as

- high speed processing because of it has it's own processor (Command Processor).
- Zoom with 16 scales
- Can Pan to any part of picture
- Able to process in many display mode depends on Hardware which usere define
- Can proces Display memory up to 256K 16 Bit word
- Able to process in Multiple Controller mode up to 4 plans with the highest resvolution 1024 x 1024

But at present time,uPD 7220 is not widespread, it is only used in normal Control Display (Multi plan only). We would like to study it's other interesting capability such as Graphic Capability, Display Speed and Zoom - Contrast the picture.

So we design Color Graphic Display Card with 640x350 resvolution for study Hardware and Software of this Graphic Display Controllor Chip.

สารบัญ

เนื้อหา หน้า

บทที่ 1

บทนำ

ในปัจจุบันนี้ คอมพิวเตอร์ ได้เข้ามามีบทบาทในชีวิตประจำวันมาก ซึ่งนับวันก็จะมีการพัฒนาให้มีประสิทธิภาพสูงขึ้นเรื่อย ๆ เช่น ในด้านความเร็ว ความละเอียดของจอภาพ ฯลฯ และ ในการแสดงผลออกทางจอภาพนั้น จะต้องใช้ไอซีตัวหนึ่ง คือ ซีอาร์ที คอนโทรลเลอร์ (CRT Controller) เป็นตัว ควบคุมการแสดงผลออกทางจอถาพ แต่ซีอาร์ที คอนโทรลเลอร์นั้นยังมีข้อด้อยอยู่คือ ไม่มี ซีพียู ภายในตัว จึงทำให้ทำงานได้ช้าและยังต้องใช้ซอฟท์แวร์เป็นส่วนใหญ่ในการควบคุมเพื่อให้ระบบสามารถทำงานได้

วิทยาพนธ์ฉบับนี้ขอเสนอไอซีตัวใหม่ คือ Graphic Display Controller (GDC) เบอร์ uPD-7220 ซึ่งไอซีเบอร์นี้สามารถทำหน้าที่แทน ซีอาร์ที คอนโทรลเลอร์ ได้ แต่จะมีข้อดีกว่าคือ

1.เป็นไอซีที่มี ซีพียู ภายในตัว จึงสามารถประมวลผลเองได้ ทำให้ช่วยลดภาระ และเวลาในการทำงานของ ซีพียู หลัก เป็นผลให้ระบบทำงานได้เร็วขึ้น

2.สามารถแสดงผลความละเอียดสูงได้

3.การทำงานส่วนใหญ่เป็นหน้าที่ของ ฮาร์ดแวร์(HARDWARE) ทำให้ทำงานได้เร็วขึ้นกว่าระบบที่ใช้ ซอฟท์แวร์ (SOFTWARE) ในการควบคุมการทำงาน

4.ในการใช้งานร่วมกับไดนามิค แรม (Dynamic RAM) เราไม่ต้องสร้างสัญญาน RAS และ รีเฟรช(Refresh) เพราะ ไอซีเบอร์นี้ ได้สร้างสัญญาน RAS ให้แล้ว และ ยังทำการรีเฟรช ไดนามิคแรม ให้ด้วย

5.สามารถทำ มัลติแพลน (Muti-plan) ได้

จากข้อดีต่างๆ ที่กล่าวมานี้ ทำให้ผู้เสนอวิทยานิพนธ์สามารถนำข้อดีเหล่านี้มาใช้ในการศึกษา แนวทาง การนำ ไอซีตัวนี้มาสร้าง การ์ดจอเพื่อใช้กับจอ ชนิดต่าง ๆ ซึ่งทำให้สามารถแสดงผลได้ด้วยความเร็วและความละเอียดสูง

สำหรับในบทต่อไปจะเป็นการกล่าวถึงคุณสมบัติ และการทำงานของ uPD-7220

ส่วนในบทที่ 3 จะกล่าวถึง โครงสร้างโดยทั่วไปของวงจรโดยย่อ ซึ่งจะอธิบายถึงหน้าที่ของส่วนต่าง ๆ และวัตถุประสงค์ที่เลือกใช้หรือออกแบบอุปกรณ์ต่างๆ

บทที่ 4 จะอธิบายถึง หลักการในการออกแบบ และการทำงานของวงจรในส่วน GDC INTERFACE UNIT

บทที่ 5 จะกล่าวถึง การออกแบบทางซอฟท์แวร์และการเลือกค่าพารามิเตอร์ต่างๆ

บทที่ 6 เป็นการทดลองและผลที่ได้รวมทั้งแนวทางในการแก้ปัญหา

บทที่ 2

รายละเอียดของ uPD-7220

ไอซี uPD-7220(GDC)เป็นหน่วยประมวลผลเกี่ยวกับแสดงผลออกทางจอภาพทั้งหมด ทำหน้าที่สร้างสัญญาณควบคุมส่วนต่างๆ เช่นจอภาพ การอ่านและเขียนหน่วยความจำแสดงผล(VIDEO RAM) รวมทั้งควบคุมการรับส่งข้อมูลกับคอมพิวเตอร์ด้วย

ซึ่งรายละเอียดต่างๆของไอซี uPD-7220 จะมีดังนี้คือ

2.1 การพิจารณาระบบการทำงานของ uPD-7220

uPD-7220 นั้น ได้ถูกออกแบบมาให้ใช้งานร่วมกับ ไมโครโปรเซสเซอร์ (Micro-Processor) เพื่อนำมาใช้กับระบบคอมพิวเตอร์กราฟฟิก(Computer Graphic) ที่มีประสิทธิภาพสูง ๆ ซึ่งหน้าที่การทำงานของไอซีเบอร์นี้ แบ่งออกได้ 6 ระดับ ดังนี้คือ

2.1.1 สร้างสัญญาณ Basic Video Raster Timing , สัญญาณซิงค์ (SYNC) และ แบลงกิ้ง(Blanking) และรวมทั้งการแบ่งพื้นที่ของหน้าจอ(screen) และ การ zoom .

2.1.2 ในขั้นนี้ข้อมูลในส่วนหน่วยความแสดงผลจะถูกแก้ไข(Modify)ในระหว่างขบวนการวาดและการเคลื่อนย้ายข้อมูล

2.1.3 ในขั้นนี้จะใช้ในการคำนวนตำแหน่งแอดเดรสตำแหน่งต่อไปที่จะวาดในหน่วยความแสดงผล

2.1.4 ในขั้นนี้จะเป็นการคำนวนเบื้องต้นเพื่อเตรียมค่าพารามิเตอร์ต่าง ๆ ที่จะใช้ในการวาด

2.1.5 ในขั้นนี้ GDC จะทำการเตรียมข้อมูลที่จะวาด ให้อยู่ในรูปกราฟฟิค ซึ่งเป็นรูปแบบของข้อมูลที่จะสามารถวาดได้

2.1.6 ส่วนในขั้นสุดท้ายนี้จะเป็นการเก็บข้อมูลทั้งหมดที่เตรียมมา และส่งออกไปยังหน่วยความจำแสดงผล โดยใช้ 3 ขั้นตอนแรกที่กล่าวมา

2.2 รายละเอียดของขาต่างๆของ uPD-7220

uPD-7220 เป็นไอซีที่มี 40 ขา ซึ่งรายละเอียดของทั้ง 40 ขา จะมีดังนี้คือ

No.	Symbol	Direction	Function
1	2*Wclk	IN	clock input
2	-DBIN	OUT	display memory read input Flag
3	HSYNC	OUT	horizontal video sync output
4	V/EXT SYNC	IN/OUT	vertical video sync output or external VSYNC input
5	BLANK	OUT	CRT blanking output
6	ALE	OUT	address latch enable output
7	DRQ	OUT	DMA request output
8	-DACK	IN	DMA acknowledge input
9	-RD	IN	read strobe input for microprocessor interface
10	WR	IN	write strobe input for microprocessor interface
11	A0	IN	address select input for microprocessor interface

12-19	DB0-7	IN/OUT	bidirectional data bus to host microprocessor
20	GND	-	ground
21	LPEN	IN	light pen detect input
22-34	AD0-12	IN/OUT	address and data line to display memory
35-37	AD13-15	IN/OUT	utilization varies with mode of operation
38	A16	OUT	,,_____,,
39	A17	OUT	,,_____,,
40	V_{cc}	-	+5V

2.3 โครงสร้างภายในของ uPD-7220

โครงสร้างภายในแต่ละส่วนของ uPd-7220 จะมีหน้าที่ดังนี้ คือ

2.3.1 MICROPROCESSOR BUS INTERFACE

วงจรส่วนนี้จะทำหน้าที่ในการแลกเปลี่ยนข้อมูลที่มีขนาด 8 บิทระหว่าง ไมโครโปรเซสเซอร์กับ GDC ซึ่งค่าจากรีจิสเตอร์แสดงสถานะ(Status Register) จะสามารถอ่านได้ตลอดเวลา แต่สำหรับการติดต่อกับบัฟเฟอร์(BUFFER) ของ FIFO นั้น จะสามารถติดต่อได้เมื่อใดจะขึ้นอยู่กับค่าของแฟลก(FLAG) ในรีจิสเตอร์แสดงสถานะ และจะไม่ขึ้นกับการกระทำภายในของ GDC

2.3.2 COMMAND PROCESSOR

จะเป็นตัวแปลชุดข้อมูลที่บรรจุอยู่ใน FIFO

2.3.3 DMA CONTROLLER

เป็นวงจรที่ทำหน้าที่ร่วมกับตัวควบคุม DMA (DMA Controller) ภายนอกในการเคลื่อนย้ายข้อมูลระหว่าง หน่วยความจำของไมโครโปรเซสเซอร์ กับหน่วยความจำของส่วนแสดงผล

2.3.4 PARAMETER RAM

วงจรส่วนนี้ประกอบด้วย RAM ขนาด 16 บิท ซึ่งเอาไว้เก็บค่าพารามิเตอร์ (Parameter) ที่จะใช้ในการวาดและการแสดงผล

สำหรับการทำงานใน Character Mode RAM นี้จะถูกแบ่งออกเป็น 4 ส่วน เพื่อเอาไว้สำหรับเก็บค่าพารามิเตอร์ที่จะใช้ในการแสดงผลของแต่ละพื้นที่

สำหรับการทำงานใน Graphic Mode RAM จะถูกแบ่งเป็น 2 ส่วน เพื่อเอาไว้สำหรับเก็บรูปแบบของภาพที่จะวาด และข้อมูลของตัวอักษร

2.3.5 VIDEO SYNC GENERATOR

วงจรส่วนนี้จะทำหน้าที่สร้างสัญญาณ Raster Timing โดยข้อมูลของรูปแบบของสัญญาณที่จะสร้างนั้นจะถูกโปรแกรมต่อจากคำสั่งรีเซ็ท (RESET) ซึ่งขณะนั้น GDC จะอยู่ในสภาวะ idle

2.3.6 MEMORY TIMING GENERATOR

วงจรส่วนนี้จะทำหน้าที่สร้าง memory cycle โดยจะแบ่งเป็น 2 ชนิด คือ

- Two clock period ซึ่งจะใช้สำหรับการ Refresh Dynamic RAM

- Four clock period ซึ่งจะใช้สำหรับขบวนการ READ-MODIFY-WRITE CYCLE(RMW CYCLE)

โดยที่สัญญาณที่จะใช้ในการควบคุมการทำงานของหน่วยความจำนั้น จะสร้างได้จาก สัญญาณ ALE และ DBIN ของ GDC

2.3.7 ZOOM AND PAN CONTROLLER

Zoom & Pan Controller ใช้ค่าจาก Zoom Display factor และค่าที่กำหนดพื้นที่แสดงผล (Display area) ใน พารามิเตอร์แรม (Parameter RAM) ในการกำหนดว่า จะให้การรีเฟรชหน้าจอ ใช้ค่าในแอดเดรสถัดไปได้เมื่อไรและเมื่อไรจะไปยังพื้นที่แสดงผลถัดไป การขยายภาพ (Zoom) ทางแนวนอนเกิดจากการลดอัตราการรีเฟรชให้ช้าลง ส่วนการขยายภาพทางแนวตั้งเกิดจากการวาดเส้นเดิมซ้ำ เป็นจำนวนครั้งเท่ากับในทางแนวนอน ส่วนการ Pan ก็ทำโดยการแก้ไขแอดเดรสเริ่มต้นของพื้นที่แสดงผล ซึ่งสามารถทำได้โดยไม่จำกัดทิศทาง และเป็นอิสระจากพื้นที่แสดงผลอื่น ๆ

2.3.8 DRAWING CONTROLLER

หน่วยประมวลผลสำหรับการวาดจะทำหน้าที่เก็บข้อมูลที่จะใช้ในการคำนวณหาแอดเดรส และตำแหน่งของ จุด(PIXEL)ต่อไปที่จะวาด

โดยที่ผู้ใช้เพียงโปรแกรมตำแหน่งเริ่มต้นที่จะวาดและค่าพารามิเตอร์ต่างๆให้เหมาะสม วงจรส่วนควบคุมการวาด(DRAWING CONTROLLER) ก็จะสามารถทำการคำนวณต่อเพื่อให้สามารถวาดรูปตามต้องการได้

2.3.9 DISPLAY MEMORY CONTROLLER

จุดประสงค์หลักของส่วนนี้คือจะทำหน้าที่ในการแยก(Multiplex) แอดเดรสและข้อมูลที่จะส่งเข้าและออกจากหน่วยความจำแสดงผลออกจากกัน และยังทำหน้าที่ในการเก็บ 16-bit Logic Unit เพื่อใช้ในการแก้ไขข้อมูลในหน่วยความจำแสดงผล ทำหน้าที่เป็นตัวนับแถวในขณะวาดในโหมดคาแรคเตอร์(Character Mode) และทำการรีเฟรช

(Refresh) ตัวนับ(Counter)สำหรับไดนามิกแรม(Dynamic RAMS)

2.3.10 LIGHT PEN DELITCHER

ถ้า Light Pen จับสัญญาณขอบขาขึ้นติดต่อกันได้ 2 ตัวขึ้นไป ณ.จุดเดียวกัน บิทแสดงสถานะของ Light Pen จะแสดงให้รู้ว่า ขณะนี้ค่าที่เก็บอยู่ใน Light Pen Register คือค่าแอดเดรสที่ Light Pen อ่านขึ้นมาได้

2.4 คำแนะนำสำหรับการโปรแกรม GDC

GDC จะใช้แอดเดรส 2 ค่าในการติดต่อกับรีจิสเตอร์แสดงสถานะ และ FIFO โดยที่ถ้าบิท A_0 = 0 จะเป็นการอ่านค่าจากรีจิสเตอร์แสดงสถานะ หรือการส่งชุดพารามิเตอร์ ไปเก็บไว้ที่ FIFO ซึ่งจะสามารถสรุปได้ดังตารางที่ 2.1

A_0	READ	WRITE
0	Status Register	Parameter into FIFO
1	FIFO Read	Command into FIFO

ตารางที่ 2.1 GDC Microprocessor Bus Interface Registers

สำหรับคำสั่งต่างๆของ GDC จะประกอบด้วย คำสั่งที่มีขนาด 1 ไบท์ และตามด้วยชุดของพารามิเตอร์ ซึ่งชุดของพารามิเตอร์ นี้จะเป็นตัวกำหนดรายละเอียดของคำสั่งนั้นๆ

ในการทำงานตามคำสั่งต่างๆนั้น คำสั่งและชุดพารามิเตอร์ จะถูกดึงเข้าไปเก็บไว้ในรีจิสเตอร์(Register) ภายในของ GDC ก่อน แล้วจึงจะทำตามคำสั่งและชุดของพารามิเตอร์เหล่านั้น

คำสั่งต่างๆของ GDC จะแบ่งออกเป็น 5 ชนิดด้วยกันคือ

2.4.1 คำสั่งที่ใช้สำหรับควบคุมจอภาพ(Video Control Command)

ซึ่งจะประกอบด้วยคำสั่งต่างๆต่อไปนี้คือ

2.4.1.1 RESET

RESET	0	0	0	0	0	0	0	0

เป็นคำสั่งที่ใช้สำหรับปรับสภาพของ GDC ให้อยู่ในสภาวะ idle ซึ่งจะประกอบด้วยพารามิเตอร์ต่างๆต่อไปนี้คือ

P1	0	0	C	F	I	D	G	S

ซึ่งในการกำหนดค่าของบิทต่างๆจะเลือกได้จากตารางที่ 2.2-2.5

C	G	Display Mode
0	0	Mixed graphics and character Mode
0	1	Graphics Mode
1	0	Character Mode
1	1	Invalid

I	S	Video Framing
0	0	Noninterlaced
0	1	Invalid
1	0	INTERLACED Repeat Field for character display
1	1	Interlaced

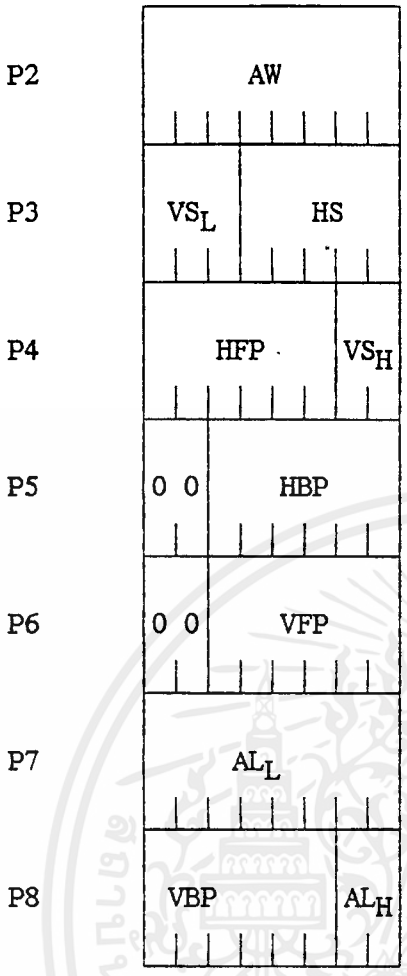
ตารางที่ 2.3 ตารางแสดงค่าต่างๆของ I และ S

D	Dynamic RAM Refresh Cycle Enable
0	No Refresh - STATIC RAM
1	Refresh - DYNAMIC RAM

ตารางที่ 2.4 ตารางแสดงค่าต่างๆของ D

F	Drawing Time Window
0	Drawing during active display time and replace blanking
1	Drawing only during retrace blanking

ตารางที่ 2.5 ตารางแสดงค่าต่างๆของ F



จาก P2 - P8 จะสามารถกำหนดค่าของบิทต่างๆได้ดังนี้คือ

AW = Active Word per Line - 2

เช่น ถ้าให้ความละเอียดของจอ = 640*350

Active Word per Line = 640/16 = 40

HBP (Horizontal Back Poarch)

ค่าของ HBP จะสามารถเลือกได้ 3 แบบ ซึ่งขึ้นอยู่กับโหมดที่เลือกใช้ ดังนี้

-สำหรับโหมดทั่งๆไป ให้ HBP >= 3 Display word cycles

-สำหรับ Image Mode ให้ HBP>= 5 Display word cycles

-สำหรับ Interlaced Mode ให้ HBP >= 5 Display word cycles

HFP (Horizontal front poarch)

ค่าของ HFP จะสามารถเลือกได้ 4 แบบ ซึ่งขึ้นอยู่กับรหมดที่เลือกใช้ดังนี้ คือ

-ถ้าในการ Zoom เราเลือกใช้การ Zoom มากกว่า 1 เท่า จะกำหนดให้ HFP >= 2 Display word cycles

-ถ้า GDC ถูกใช้เป็น Slave Mode จะกำหนดให้ HFP >= 4 Display word cycles

-ถ้ามีการใช้ Light Pen จะกำหนดให้ HFP >= 6 Display word cycles

-ถ้าใช้ Interlaced Mode จะกำหนดให้ HFP >= 3 Display word cycles

VS :คือค่า Verticle Sync Width

HS :คือค่า Horizontal Sync Width ซึ่งกำหนดไว้ว่า HS >= 5 Display Word Cycles

VFP :คือค่า Verticle Front Poarch Width

VBP :คือค่า Verticle Back Poarch Width

AL :คือค่า Active Display Lines per Video Field

2.4.1.2 SYNC

SYNC	0	0	0	0	1	1	1	DE

เป็นคำสั่งที่ใช้สำหรับกำหนดรูปแบบในการแสดงผลของจอภาพ

ซึ่งค่า DE จะกำหนดได้ดังนี้ คือ

ถ้า DE = 1 จะทำให้สามารถแสดงผลออกทางจอภาพได้

DE = 0 จะทำให้เกิดแบล็งค์(Blank) ทั้งจอภาพ

สำหรับค่าพารามิเตอร์ของคำสั่งนี้ จะเหมือนกับ พารามิเตอร์ของคำสั่ง RESET ทุกประการ

2.4.1.3 VSYNC

VSYNC

0	1	1	0	1	1	1	M

เป็นคำสั่งที่ใช้สำหรับให้ผู้ใช้เลือกว่า จะให้ GDC มีโหมดการทำงานเป็น Master Mode หรือ Slave Mode

ซึ่งค่าของ M จะเลือกได้ดังนี้ คือ

ถ้า M = 0 จะทำให้ GDC มีโหมดการทำงานเป็นแบบ Slave Mode ซึ่งจะทำให้ GDC รับสัญญาณ Verticle SYNC จากภายนอก

M = 1 จะทำให้ GDC มีโหมดการทำงานเป็นแบบ Master Mode ซึ่งจะทำให้ GDC ทำหน้าที่เป็นตัวสร้างสัญญาณ Verticle SYNC

ในกรณีที่ใช้ GDC มากกว่า 1 ตัวขึ้นไปในการสร้างภาพ 1 ภาพ จะต้องกำหนดให้ GDC ตัวหนึ่งมีโหมดการทำงานเป็นแบบ Master Mode ส่วนตัวอื่นๆที่เหลือจะต้องกำหนดให้มีโหมดการทำงานเป็นแบบ Slave Mode ให้หมด และขา VSYNC ของ GDC ทุกตัวจะต้องต่อเข้าด้วยกัน

2.4.1.4 CCHAR

CCHAR

0	1	0	0	1	0	1	1

เป็นคำสั่งที่ใช้ในการกำหนด Cursor และ ความสูงของตัวอักษร

ซึ่งจะประกอบด้วยพารามิเตอร์ต่างๆต่อไปนี้ คือ

P1

DC	0	0	LR

P2

BR_L	SC	CTOP

P3

CBOT	BR_H

ซึ่งค่าของบิทต่างๆจะกำหนดได้จาก ข้อกำหนดต่างๆต่อไปนี้ คือ

-DC : Display Cursor

บิทนี้จะถูกกำหนดให้เป็น 1 เมื่อต้องการให้แสดงเคอร์เซอร์ ขึ้นหน้าจอด้วย .

-LR :Lines Per Character Row

สำหรับค่านี้จะต้องกำหนดให้เป็น 0 ถ้าการทำงานเป็นแบบกราฟฟิคโหมด

-BR : Blink Rate

ค่า BR นี้จะต้องกำหนดให้เท่ากับ 3 ถ้าต้องการให้การแสดงผลเป็นแบบ Interlaced ในโหมดกราฟฟิค(Graphic Mode)

-SC :

ถ้า SC = 0 จะเป็นการกำหนดให้เคอร์เซอร์ กระพริบ

SC = 1 จะเป็นการกำหนดให้เคอร์เซอร์ ไม่กระพริบ

-CTOP : Cursor Top Line Number in The Row

เป็นการกำหนดตำแหน่งบนสุดของเคอร์เซอร์ ในแถว

-CBOT : Cursor Bottom Line Number in The Row

เป็นการกำหนดตำแหน่งล่างสุดของเคอร์เซอร์ ในแถว

2.4.2 คำสั่งสำหรับควบคุมการแสดงผล(DISPLAY CONTROL COMMAND)

จะประกอบด้วยคำสั่งต่างๆต่อไปนี้ คือ

2.4.2.1 START

START	0	1	1	0	1	0	1	1

เป็นการสั่งให้ GDC พ้นจากสภาวะ idle mode เริ่มเข้าสู่การทำงานตามคำสั่งต่างๆของ GDC

2.4.2.2 BCTRL

BCTRL	0	0	0	0	1	1	0	DE

เป็นคำสั่งที่ใช้ในการกำหนดว่าจะให้แสดงผลออกทางจอภาพหรือไม่ ซึ่งถ้าต้องการให้แสดงผลออกทางจอภาพ ให้กำหนดให้ DE = 1

2.4.2.3 ZOOM

ZOOM	0	1	0	0	0	1	1	0

เป็นคำสั่งที่ใช้สำหรับกำหนด Zoom Factor สำหรับการแสดงผลและการเขียนตัวอักษรแบบกราฟฟิค(Graphic Character)

ซึ่งจะประกอบพารามิเตอร์ต่าง ๆ ต่อไปนี้ คือ

P1	DISP				GCHR			

และในการกำหนดค่าของบิทต่างๆ จะกำหนดได้จากข้อกำหนดต่อไปนี้ คือ

-DISP :DISPLAY ZOOM FACTOR

GDC สามารถ Zoom ได้จาก 1-16 ซึ่งจะกำหนดว่าจะให้ Zoom ขนาดเท่าไรได้จากพารามิเตอร์ตัวนี้ ดังนี้คือ

ถ้าต้องการ Zoom 1 ให้กำหนดค่า DISP = 0

ต้องการ Zoom 2 ให้กำหนดค่า DISP =1

:

:

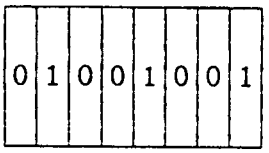
:

ต้องการ Zoom 16 ให้กำหนดค่า DISP = 15

-GCHR : ZOOM FACTOR for GRAPHICS CHARACTER WRITING AND AREA FILLING

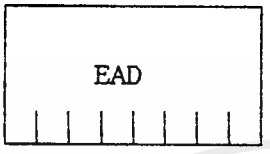
2.4.2.4 CURS

CURS

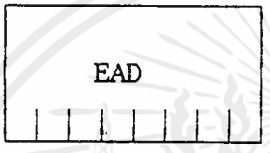


เป็นคำสั่งที่ใช้สำหรับกำหนดตำแหน่งของเคอร์เซอร์ในหน่วยความจำแสดงผล ซึ่งจะประกอบด้วยพารามิเตอร์ต่างๆต่อไปนี้คือ

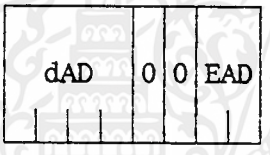
P1



P2



P3



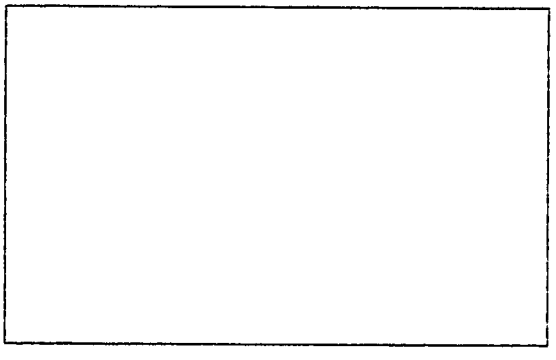
EAD : คือค่า EXECUTE WORD ADDRESS

dAD : คือค่า DOT ADDRESS within THE WORD

โดยมีวิธีคำนวณหาค่าต่างๆดังนี้ คือ

เนื่องจากการกำหนดตำแน่งการวาดนั้นกำหนดด้วยระบบแกน X และแกน Y ซึ่งเริ่มต้นที่ (0,0) และแกน X มีการเพิ่มค่าจากซ้ายไปขวา และแกน Y มีการเพิ่มค่าจากบนลงล่าง ดังในรูปที่ 2.1

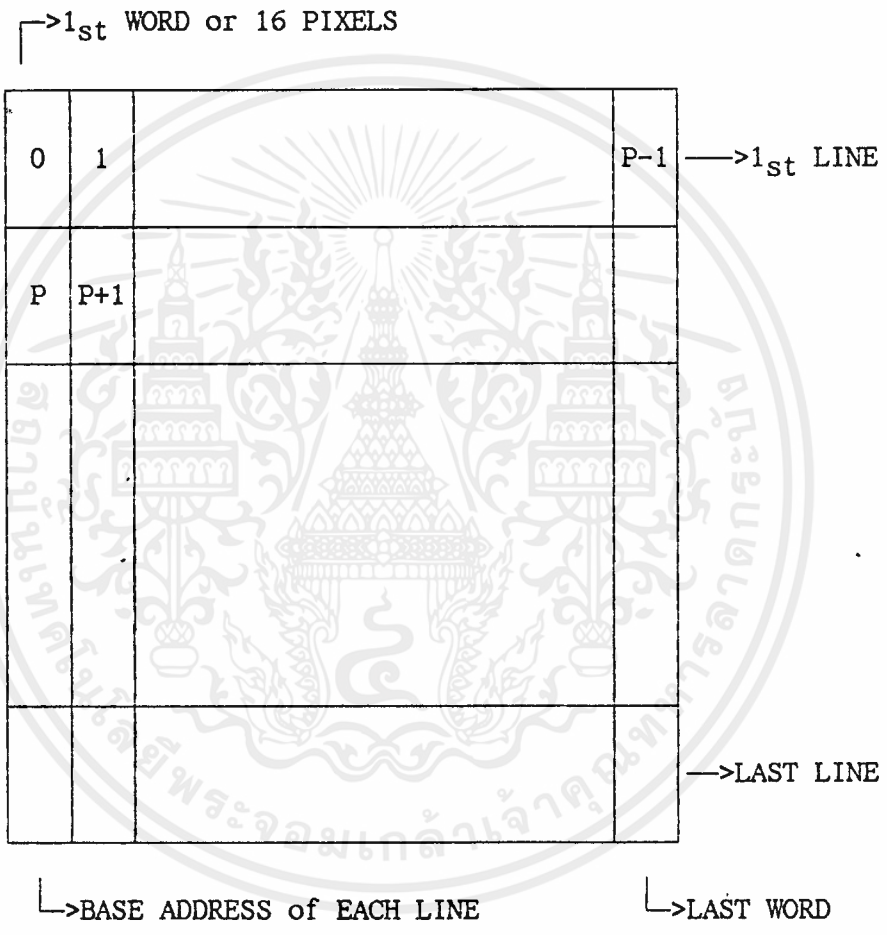
(0,0) (XMAX)



(0,YMAX) (XMAX,YMAX)

รูปที่ 2.1 แสดงการกำหนดตำแหน่ง X,Y ในการวาด

แต่การจัดแอดเดรสของหน่วยความจำแสดงผลนั้นจะแตกต่างกัน ดังนั้นจึงต้องมีการนำค่า X,Y ที่ต้องการมาทำการคำนวณเพื่อเปลี่ยนให้เป็นตำแหน่งแอดเดรสของหน่วยความจำแสดงผล เนื่องจาก GDC ต้องการ GDC ต้องการพารามิเตอร์ที่เป็นแอดเดรสของหน่วยความจำแสดงผลสำหรับการวาด การจัดตำแหน่งแอดเดรสของหน่วยความจำแสดงผลจะแสดงดังรูปที่ 2.2



รูปที่ 2.2 แสดงการจัดแอดเดรสของหน่วยความจำแสดงผล

จากรูปที่ 2.2 จะเห็นได้ว่าการจัดแอดเดรสของหน่วยความจำแสดงผลนั้นจะเรียงตามแกน X นั่นคือเพิ่มค่าจากซ้ายไปขวา และจากบนลงล่างเหมือนกับการจัดตำแหน่งในระบบ X-Y แต่หน่วยความจำแสดงผล 1 แอดเดรสนั้นประกอบด้วยจุด 16 จุด ดังนั้น

ถ้าให้ P = จำนวน 16 bIT WORDS ในหนึ่งเส้น

ดังนั้น $P = (X_{max} + 1)/16$

ถ้าให้ X_{max} มีค่าเท่ากับ 1023 ดังนั้น $P = 64_{10}$ หรือ 40_H

จะได้ Line Base Address (LBA) = P*Y

ดังนั้นแอดเดรสของหน่วยความจำแสดงผลหรือค่าของแต่ละบิทในพารามิเตอร์ต่างๆ คือ

$$EAD = LBA + TRUNC(X/16)$$

และตำแหน่งของบิทคือ $dAD = X \bmod 16$

2.4.2.5 PRAM

PRAM

0	1	1	1	SA

เป็นคำสั่งที่ใช้ในการกำหนดแอดเดรสเริ่มต้น ความยาวของพื้นที่ในการแสดงผล และรูปแบบของข้อมูลของตัวอักษรแบบกราฟฟิค

โดยที่ค่า SA จะเป็นค่าที่บอกว่าจะให้เริ่มอ่านจากพารามิเตอร์ตัวที่เท่าใด เช่น

ถ้า SA = 0 จะหมายถึงว่าให้เริ่มต้นอ่านจากพารามิเตอร์ตัวที่ 1(P1)

SA = 1 จะหมายถึงว่าให้เริ่มต้นอ่านจากพารามิเตอร์ตัวที่ 2(P2)

และจะเป็นเช่นนี้ไปเรื่อยๆ ซึ่งจะสามารถเลือกได้ทั้งหมด 16 ค่า

เมื่อ GDC เริ่มอ่านค่าจากพารามิเตอร์ ตัวที่ SA+1 แล้ว ก็จะอ่านค่าพารามิเตอร์ตัวถัดไป อ่านไปเรื่อยๆและจะหยุดอ่านเมื่อได้รับชุดคำสั่งต่อไป

สำหรับค่าของพารามิเตอร์ต่างๆ จะกล่าวถึงต่อไปในหัวข้อ PARAMETER RAM CONTENT

2.4.2.6 PITCH

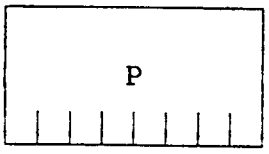
PITCH

0	1	0	0	0	1	1	1

เป็นคำสั่งที่ใช้สำหรับกำหนดความกว้างของ Display Memory ค่านี้จะถูกใช้ระหว่างขบวนการวาด โดยที่ Drawing Processor จะทำหน้าที่หาทิศทางของ word ต่อไปที่จะเขียน ว่าอยู่ทางทิศใดของ word ปัจจุบัน และในขณะที่แสดงผล ก็จะทำหน้าที่หาตำแหน่งเริ่มต้นของแถวต่อไป

สำหรับคำสั่งนี้จะมีพารามิเตอร์เพียงตัวเดียว

P1



P: Number of Word Addresses in Display Memory in The Horizontal Direction

2.4.3 คำสั่งที่ใช้ควบคุมการวาด (DRAWING CONTROL COMMANDS)

ซึ่งจะประกอบด้วยคำสั่งต่างๆต่อไปนี้ คือ

2.4.3.1 WDAT

WDAT

0	0	1	TYPE	0	MOD

เป็นคำสั่งที่ใช้ในการเขียนข้อมูลเป็น word หรือ byte ลงในหน่วยความจำแสดงผล(Display Memory)

ซึ่งจะกำหนดค่าของบิทต่างๆได้ดังนี้

TYPE : Data Transfer Type

00 : Word,Low then High byte

01 : Low byte of the Word

10 : High byte of the Word

11 : Invalid

MOD : RMW Memory Cycle,Logical Operation

00 :Replace with Pattern

01 :Complement

10 :Reset to Zero

11 :Set to 1

ส่วนค่าพารามิเตอร์ต่างๆก็คือ ข้อมูลที่จะส่ง โดยจะส่งทีละ 8 บิท

การทำงานของคำสั่งนี้จะเป็นดังนี้ คือ

-เมื่อ GDC ได้รับ 1 ชุด ของพารามิเตอร์ (2 ไบท์ ถ้ากำหนดให้เป็นการส่งเป็นเวิร์ด และ 1 ไบท์ ถ้ากำหนดให้ส่งเป็น ไบท์)แล้วหน่วยความจำของจอภาพ(Video Memory) ก็จะทำการ RMW ณ. ตำแหน่ง EAD ที่กำหนดไว้ในคำสั่ง CURS

-จากนั้นตัวชี้(Pointer) ของ EAD ก็จะชี้ไปยังเวิร์ดต่อไป ตามทิศทางที่ได้กำหนดเอาไว้

-จากนั้นก็จะสามารถรับพารามิเตอร์ชุดต่อไปได้

-สำหรับการเขียนเป็นไบท์นั้น ในขบวนการ RMW Cycle ไบท์ที่ไม่ได้ถูกกำหนดจะถูกแทนด้วยศูนย์ทั้งหมด

-สำหรับการเขียนแบบ graphic bit-map นั้น ค่า LSB ของพารามิเตอร์ต่างๆ เท่านั้นที่จะถูกใช้ในขบวนการ RMW Cycle ดังนั้นค่าพารามิเตอร์ต่างๆจะเป็นได้ 2 กรณี คือ เป็น 1 ทั้ง 8 บิท หรือเป็น 0 ทั้ง8 บิท

-สำหรับการเขียนในโหมดกราฟฟิค นั้น ทุกๆบิทของพารามิเตอร์จะถูกใช้หมด

-คำสั่งนี้จะต่างจากคำสั่งอื่นๆ คือ คำสั่งนี้จะต้องใช้พารามิเตอร์ในการกำหนด Pattern Register ในขณะที่คำสั่งอื่นๆจะใช้ค่าของพารามิเตอร์ที่เก็บอยู่ใน Parameter RAM

-ก่อนที่จะส่งคำสั่งนี้จะต้องส่งคำสั่ง FIGS และพารามิเตอร์ 3 ตัวแรกของคำสั่ง FIGS ก่อน เพื่อใช้ในการกำหนด รูปแบบของการวาด ทิศทางในการวาด และค่า DC

2.4.3.2 MASK

MASK

0	1	0	0	1	0	1	0

คำสั่งนี้ใช้สำหรับกำหนดค่าของ 16-bit Mask Register ซึ่ง Mask Register นี้จะใช้เป็นตัวกำหนดว่าจะให้ข้อมูลบิทใดบ้างในหน่วยความจำแสดงผลถูกเปลี่ยนแปลงแก้ไขในขบวนการ RMW Cycle

การโหลดค่าใน Mask Register จะทำได้ 2 วิธี คือ

1.ใช้คำสั่ง MASK คำสั่งนี้จะมีพารามิเตอร์ 2 ตัว ซึ่งจะใช้สำหรับกำหนดว่า จะให้บิทใดบ้างถูกเปลี่ยนแปลงแก้ไข โดยที่ทั้ง 16 บิทนี้จะถูกโหลดเข้าไปเก็บไว้ใน Mask Register

2.ใช้คำสั่ง CURS โดยกำหนดที่ dAD ของพารามิเตอร์ตัวที่ 3 ของคำสั่งนี้

ถ้าการวาดเป็นแบบ Graphic bit-map จะไม่ต้องใช้คำสั่ง MASK แต่จะกำหนดโดยใช้คำสั่ง CURS แทน โดยที่คำสั่ง CURS นี้จะเป็นตัวกำหนดตำแหน่งของจุดของแต่ละแอดเดรสที่เหมาะสม

2.4.3.3 FIGS

FIGS

0	1	0	0	1	1	0	0

เป็นคำสั่งที่ใช้ในการกำหนดพารามิเตอร์ที่จะใช้ในการควบคุมการวาด ซึ่งจะหาค่าของบิทต่างๆของพารามิเตอร์แต่ละตัวได้ดังนี้

P1

SL	R	A	GC	L	DIR

SL : Slanted Graphics Character

R : Rectangle

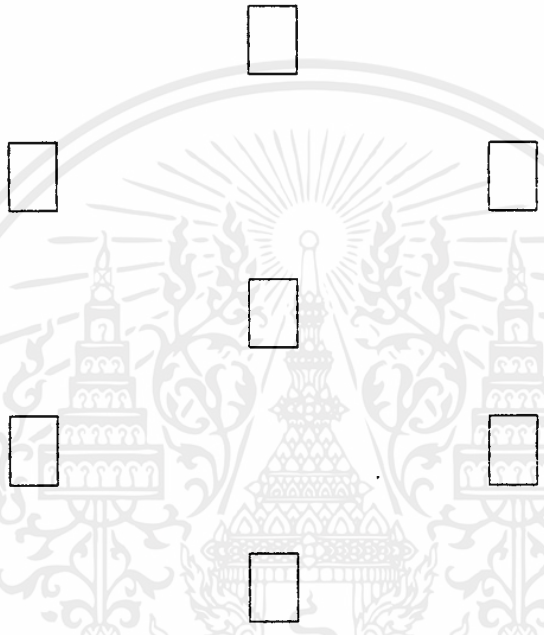
A : ARC/Circle

GC : Graphics Character

L : Line(Vector)

DIR: Drawing Direction Base

สำหรับไบท์แรกนี้จะใช้สำหรับกำหนดทิศทางในการวาด(DIR) จะเปรียบเทียบได้จากรูปที่ 2.3



รูปที่ 2.3 แสดงทิศทางการวาดของ GDC

หมายเลขแต่ละตัวจะแทนทิศทางการวาดแต่ละทิศ เช่น

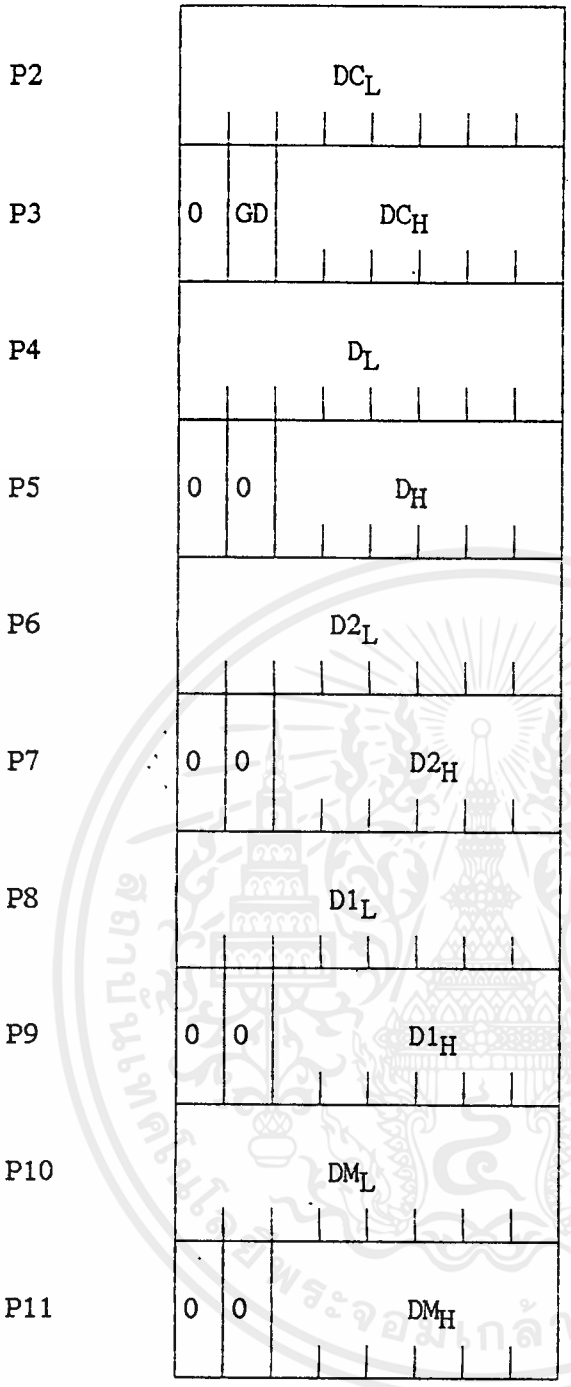
ถ้า ต้องการวาดในทิศทางลง จะกำหนดให้ DIR = 000

ต้องการวาดในทิศทางขึ้น จะกำหนดให้ DIR = 001 เป็นต้น

สำหรับ ค่า SL,R,A.GC,L จะใช้ในการกำหนดชนิดที่จะวาด ซึ่งสามารถเลือกค่าของบิทต่างๆได้ตามตารางที่ 2.6

SL	R	A	GC	L	Operation
0	0	0	0	0	Character Display Mode Drawing.Individual Dot Drawing.DMA,WDAT,and RDAT
0	0	0	0	1	Straight Line Drawing
0	0	0	1	0	Graphics Character Drawing and Area Filling with Graphics Character Pattern
0	0	1	0	0	Arc and Circle Drawing
0	1	0	0	0	Rectangle Drawing
1	0	0	1	0	Slanted Graphics Character Drawing and Slanted Area Filling

ตารางที่ 2.6 Valid Figure Type Select Drawing



สำหรับอีก 10 ไบท์ จะเป็นการกำหนดค่ารีจิสเตอร์ของ DIGITAL DIFFERENTIAL ANALYZER(DDA : เป็นฮาร์ดแวร์ที่อยู่ในตัว GDC ทำหน้าที่คำนวนทางคณิตศาสตร์ จะทำงานไปพร้อมกับการอ่านข้อมูลด้วยฮาร์ดแวร์ ซึ่งใน GDC เรียกว่า READ MODIFY WRITE (RMW))

ค่า GD จะกำหนดให้เท่ากับ 1 เมื่อ ต้องการวาดในโหมดกราฟฟิคและ จะกำหนดให้เท่ากับ 0 เมื่อต้องการวาดในโหมด Graphics and Mixed Mode สำหรับค่า DC,D,D2,D1 และ DM จะกำหนดตามชนิดของการวาดได้ตามตารางที่ 2.7

Drawing Type	DC	D	D2	D1	DM
Initial Value	0	0	0	-1	-1
Line	dI	2dD-dI	2(dD-dI)	2dD	–
ARC	rsinx	r-1	2(r-1)	-1	rsinz
Rectangle	3	A-1	B-1	-1	A-1
Area Fill	B-1	A	A	–	–
Graphic Character	B-1	A	A	–	–
Write Data	W-1	–	–	–	–
DMAW	D-1	C-1	–	–	–
DMAR	D-1	C-2	(C-2)2t	–	–
Read Data	W	–	–	–	–

ตารางที่ 2.7 ตารางแสดงค่า Drawing Parameter

- : No Parameter bytes sent to GDC for this Parameter

dI: The Larger at Dx or Dy

dD: The Smaller at Dx or Dy

r : Radius of Curvature in Pixels

x : Angle from Major Axis to end of the Arc , x <= 45

z : Angle from Major Axis to start of the Arc,z<= 45

A : Number of Pixels in The Initially Specified Direction

B : Number of Pixels in The Direction at Right Angles to The Initially Specified Direction

W : Number of Words to be Accessed

C : Number of Bytes to be Transferred in The Initially Specified Direction

D : Number of Word s to be Accessed in The Direction at Right Angles to The Initially Specified Direction

DC: Drawing Count Parameter which is One Less Than The Number of RMW Cycles to be Executed

DM: Dots Masked from Drawing During Arc Drawing

t : Needs only for Word Resds

สำหรับค่า DC,D,D2,D1 จะเป็นจุดเริ่มต้นและจุดสิ้นสุดของเวคเตอร์ และในการวาดแบบ LINE จะต้องมีการคำนวนหาค่าต่างๆเหล่านี้ด้วย ซึ่งสามารถทำได้โดย

ขั้นที่ที่ 1 คำนวนหาระยะห่างเป็นจำนวนจุดทางด้านแกน X และแกน Y ด้วยสมการ

$$dX := X_2 - X_1$$

$$dY := Y_2 - Y_1$$

เมื่อ X_2 คือจุดสุดท้ายของเวคเตอร์ทางด้านแกน X

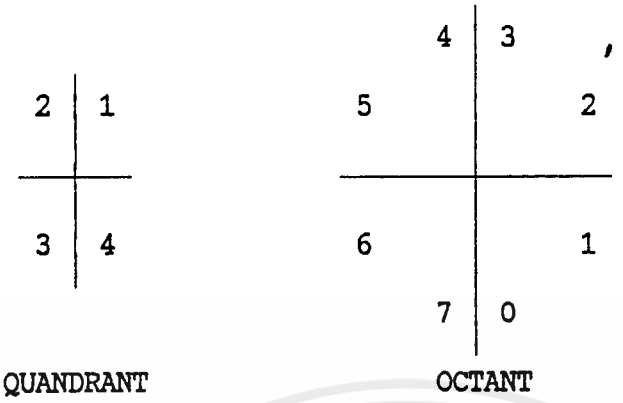
X_1 คือจุดเริ่มต้นของเวคเตอร์ทางด้สนแกน X

Y_2 คือจุดสุดท้ายของเวคเตอร์ทางด้านแกน Y

Y_1 คือจุดเริ่มต้นของเวคเตอร์ทางด้านแกน Y

ค่าของ dX และ dY สามารถเป็นได้ทั้งบวกและลบ ค่าทั้งสองคือระยะห่างเป็นจำนวนจุดของทั้งสองแกน แต่ไม่ใช่จำนวนจุดที่จะวาด ตัว GDC จะทำการวาดจุดแรกลงไปในหน่วยความจำในตำแหน่งของเคอร์เซอร์ ขณะที่ DDA ทำการคำนวนแอดเดรสของจุดที่ 2 เมื่อ GDC ทำการวาดจุดที่ 2 DDA ก็จะทำการคำนวนจุดต่อไป เป็นเช่นนี้ไปเรื่อยๆ จนกระทั่งครบทุกจุดในเวคเตอร์หรือในเส้นนั้น โดยจำนวนจุดที่จะวาดทั้งหมดจะอยู่ในรีจิสเตอร์ DC และเมื่อ GDC ทำการวาดถึงจุดสุดท้าย ค่าในรีจิสเตอร์ DC จะมีค่าเป็นศูนย์ และเคอร์เซอร์(CURSOR) จะเคลื่อนที่ไปอยู่ที่จุดสุดท้าย

ขั้นที่ 2 เป็นการหาค่า QUANDRANT และ OCTANT จากค่า dX และ dY ซึ่งการกำหนด QUANDRANT และ OCTANT ของ GDC มีการกำหนดดังในรูปที่ 2.4



รูปที่ 2.4 แสดงการกำหนด QUADRANT และ OCTANT

การหาค่า QUADRANT จะได้จากการเปรียบเทียบเครื่องหมายของ dX และ dY ตามตารางที่ 2.8

เครื่องหมายของ dX	เครื่องหมายของ dY	QUADRANT
+	−	1
−	−	2
−	+	3
+	+	4

ตารางที่ 2.8 ตารางแสดงการกำหนด QUANDRANT

ส่วนการหาค่า OCTANT จะนำค่า QUADRANT ที่หาได้มาทำการประมวลผลร่วมกับการเปรียบเทียบค่าสัมบูรณ์ (ABSOLUTE) ของ dX และ dY ($|dX|, |dY|$) ตามตารางที่ 2.9

$|dX| > |dY|$

QUADRANT	OCTANT
1	2
2	5
3	6
4	1

$|dX| < |dY|$

QUDRANT	OCTANT
1	3
2	4
3	7
4	0

ตารางที่ 2.9 ตารางแสดงการกำหนด OCTANT

ถ้า $|dX| = |dY|$ แสดงว่าเวคเตอร์หรือเส้นที่จะวาดอยู่ในแนวเส้นทะแยงมุมพอดี ดังนั้นค่าที่ได้จะคาบอยู่ระหว่าง 2 OCTANT ให้ใช้ค่าเลขคู่เป็นค่า OCTANT

ขั้นที่ 3 เป็นการกำหนดแกนการวาดอิสระ (INDEPENDENT DRAWING AXIS) แกนที่เป็นแกนวาดอิสระจะเป็นแกนที่มีความยาวมากกว่าอีกแกนหนึ่ง เช่น แกน X จะเป็นแกนวาดอิสระเมื่อ $|dX| > |dY|$ จะเป็นแกน Y เมื่อ $|dX| <= |dY|$ ดังตารางที่ 2.10

OCTANT	INDEPEDENT AXIS	DEPEDENT AXIS
0	Y	X
1	X	Y
2	X	Y
3	Y	X
4	Y	X
5	X	Y
6	X	Y
7	Y	X

ตารางที่ 2.10 ตารางกำหนดแกนการวาดอิสระ

ขั้นที่ 4 การคำนวนค่าพารามิเตอร์ต่างๆในการวาด ค่าพารามิเตอร์ที่ใช้ในการวาดประกอบด้วยค่า DC,D,D2,D1 ซึ่งจะสามารถคำนวนได้จากค่าของ $|dX|$ และ $|dY|$ ตามสมการต่อไปนี้ คือ

$$DC = |dI|$$

$$D = (2*|dD|) - |dI|$$

$$D2 = 2*(|dD| - |dI|)$$

$$D1 = 2*|dD|$$

เมื่อ dI เป็นระยะห่างของแกนการวาดอิสระ

dD เป็นระยะห่างของแกนการวาดไม่อิสระ

2.4.3.4 FIGD

FIGD	0	1	1	0	1	1	0	0

เป็นคำสั่งให้เริ่มวาดตามรูปแบบที่กำหนดมาจากคำสั่งต่างๆเหนือคำสั่ง FIGD เมื่อสั่งคำสั่งนี้ GDC จะทำการโหลดค่าพารามิเตอร์จาก PARAMETER RAM เข้าไปเก็บไว้ใน DRAWING PROCESSOR และเริ่มขบวนการวาด ณ.ตำแหน่งที่กำหนดไว้ด้วยค่า EAD และ dAD ในคำสั่ง CURS

2.4.3.5 GCHRD

GCHRD	0	1	1	0	1	0	0	0

เป็นคำสั่งที่ใช้สำหรับการวาดข้อมูลแบบกราฟฟิคเข้าไปเก็บไว้ในหน่วยความจำแสดงผล

2.4.4 คำสั่งที่ใช้ในการอ่านข้อมูล (DATA READ COMMAND)

ประกอบด้วยคำสั่งต่างๆต่อไปนี้

2.4.4.1 RDAT

RDAT	1	0	1	TYPE	0	MOD	

เป็นคำสั่งที่ใช้ในการอ่านข้อมูลเป็น word หรือ byte จากหน่วยความจำแสดงผล โดยที่ TYPE และ MOD จะมีวิธีกำหนดเหมือนกับคำสั่ง WDAT

2.4.4.2 CURD

CURD	1	1	1	0	0	0	0	0

เป็นคำสั่งที่ใช้ในการอ่านตำแหน่งของ CURSOR โดยที่เมื่อ GDC ทำตามคำสั่งนี้แล้ว ค่าพารามิเตอร์ต่างๆต่อไปนี้จะถูกส่งกลับคืนมาที่ FIFO

P1	A_7			EAD_L				A_0
P2	A_{15}			EAD_H				
P3	0	0	0	0	0	0	EAD_H	
P4	dAD_L							
P5	dAD_H							

2.4.4.3 LPRD

LPRD

1	1	0	0	0	0	0	0

เป็นคำสั่งที่ใช้สำหรับอ่านค่าตำแหน่งแอดเดรสของ Light Pen โดยที่เมื่อ GDC ทำตามคำสั่งนี้ ค่าพารามิเตอร์ต่างๆต่อไปนี้จะถูกส่งออกมาที่ FIFO

P1	A_7			EAD_L				A_0
P2	A_{15}			EAD_H				
P3	0	0	0	0	0	0	EAD_H	
P4				dAD_L				
P5				dAD_H				

2.4.5 คำสั่งที่ใช้ควบคุม DMA (DMA CONTROL COMMAND)

จะประกอบด้วยคำสั่งต่างๆต่อไปนี้

2.4.5.1 DMAR

DMAR

1	0	1	TYPE	1	MOD

เป็นคำสั่งที่ใช้สำหรับบอก GDC ว่าต้องการอ่านข้อมูลโดยใช้บวนการ DMA โดยที่การกำหนด TYPE และ MOD จะมีวิธีการกำหนดเหมือนคำสั่ง WDAT

2.4.5.2 DMAW

DMAW

0	0	1	TYPE	1	MOD

เป็นคำสั่งที่ใช้สำหรับบอก GDC ว่าต้องการเขียนข้อมูลโดยใช้ขบวนการ DMA โดยที่การกำหนด TYPE และ MOD จะมีวิธีการกำหนดเหมือนคำสั่ง WDAT

2.5 STATUS REGISTER FLAG

ค่าแฟลกใน STATUS REGISTER เป็นค่าที่ใช้บอกสภาวะการทำงานของ GDC ซึ่งมีขนาด 8 บิท และแต่ละบิทจะมีความหมายดังนี้ คือ

STATUS REGISTER FLAG

7	6	5	4	3	2	1	0

2.5.1 SR-7 :LIGHT PEN DETECT

ถ้าบิทนี้ถูกเซ็ทเป็น 1 หมายถึงว่าขณะนั้นสามารถอ่านค่าตำแหน่งแอดเดรสของ LIGHT PEN ได้ และเมื่อ LAD (Light Pen Address) ถูกย้ายเข้าไปใน FIFO แล้ว บิทนี้จะถูกรีเซ็ทเป็น 0

2.5.2 SR-6 :HORIZONTAL BLANKING ACTIVE

ขณะที่เกิดสัญญาณฮอริซอนทอล รีเทรส แบลงกิ้ง(Horizontal Retrace Blanking) บิทนี้จะถูกเซ็ทเป็น 1

2.5.3 SR-5 :VERTICAL SYNC

ขณะที่เกิดสัญญาณเวอร์ทิคอล รีเทรสซิงค์(Vertical Retrace Sync) บิทนี้จะถูกเซ็ทเป็น 1

2.5.4 SR-4 :DMA EXECUTE

บิทนี้จะเป็น "1" เมื่อมีการเคลื่อนย้ายข้อมูลโดยใช้ขบวนการ DMA

2.5.5 SR-3 :DRAWING in PROGRESS

บิทนี้จะเป็น "1" ในขณะที่ GDC ทำการวาดภาพแบบกราฟฟิค

2.5.6 SR-2 :FIFO EMPTY

เมื่อบิทนี้เป็น "1" แสดงว่า ชุดคำสั่งและพารามิเตอร์ที่ถูกส่งเข้าไปใน GDC ทั้งหมดได้ถูกตีความและพร้อมที่จะทำตามคำสั่งและพารามิเตอร์เหล่านั้นเรียบร้อยแล้ว

2.5.7 SR-1 :FIFO FULL

ถ้าบิทนี้เป็น "1" แสดงว่า ขณะนั้น FIFO เต็ม ไม่สามารถรับชุดคำสั่งและพารามิเตอร์ได้

ถ้าบิทนี้เป็น "0" แสดงว่า ขณะนั้น FIFO สามารถรับชุดคำสั่งและพารามิเตอร์ได้อย่างน้อย 1 ไบท์ และบิทนี้จะต้องถูกเช็คทุกครั้งก่อนที่จะทำการส่งค่าใดๆเข้า GDC

2.5.8 SR-0 :DATA READ

ถ้าบิทนี้เป็น "1" แสดงว่า ขณะนั้นมีข้อมูลพร้อมที่จะส่งไปให้ไมโครโปรเซสเซอร์ และบิทนี้จะเป็น "0" ขณะที่ข้อมูลถูกย้ายจาก FIFO ไปยัง Microprocessor Interface Data Register และจะต้องทำการเช็คบิทนี้ทุกครั้งก่อนที่จะทำการอ่านแต่ละครั้ง

2.6 ข้อกำหนด ในการใช้คำสั่ง และการทำงาน ของ FIFO

FIFO (First In , First Out) ทำหน้าที่ เป็นตัวกลางในการโต้ตอบคำสั่งกับระบบไมโครโปรเซสเซอร์ โดยมีการทำงานในลักษณะ Half Duplex คือ สามารถส่งผ่านข้อมูลได้ทั้ง 2 ทิศทาง แต่ครั้งละทิศทางเดียวเท่านั้น ทิศทางการทำงานของ FIFO ถูกควบคุมโดยระบบไมโครโปรเซสเซอร์ ผ่านทางชุดคำสั่งของไมโครโปรเซสเซอร์หลัก จะจับคู่การส่งผ่านเหล่านี้ โดยการตรวจสถานะที่เหมาะสมของ บิทของรีจิสเตอร์แสดงสถานะหรือ รีจิสเตอร์บิทที่แสดงสถานะต่าง ๆ

ชุดคำสั่งในการสั่งงาน GDC ต้องมี ไบท์แรกที่แตกต่างกันออกไป ซึ่งไบท์แรกนี้ก็คือ คำสั่ง(Operation Code) ส่วนไบท์ต่อๆไป ก็คือ พารามิเตอร์ของคำสั่งนั้นที่ตามมา

การสั่งเขียนลงไปบน GDC จะทำให้ FIFO เก็บค่าแฟลก ไปคู่กับค่าของข้อมูล เป็นตัวบอกว่า ข้อมูลที่เขียนลงไปนั้นเป็น คำสั่ง หรือ พารามิเตอร์ ซึ่งตัวประมวลผลคำสั่ง (Command Processor) ใน GDC จะตรวจสอบบิทนี้ ขณะที่แปลค่าที่อยู่ใน FIFO

เมื่อตัวประมวลผลคำสั่ง ได้รับคำสั่งใหม่เข้ามา จะถือเป็นการสิ้นสุดคำสั่งก่อนหน้านี้ และถือเป็นการสิ้นสุดการรับค่าพารามิเตอร์ของคำสั่งนั้นๆด้วย

FIFO จะเปลี่ยนทิศทางการทำงานโดยการควบคุมของระบบไมโครโปรเซสเซอร์ การส่งคำสั่งเข้าไปยัง GDC จะเป็นการสั่งให้ FIFO ทำงานใน Mode Write เสมอ ซึ่งถ้า FIFO ยังอยู่ใน Mode Read ข้อมูลที่ค้างอยู่จะหายหมด เมื่อสั่งคำสั่งใหม่เข้าไป คำสั่งที่ต้องการการตอบสนอง เช่น RDAT, CURD, LPRD จะทำให้ FIFO กลายเป็น Read Mode

2.7 Read-Modified-Write Cycle

การส่งข้อมูลระหว่าง GDC และ หน่วยความจำแสดงผล (Display Memory) ต้องใช้ Read-Modified-Write memory cycle (RMW) ซึ่งเป็นสัญญาณที่ประกอบด้วย 4 ช่วงการทำงาน คือ

1. ส่งค่าแอดเดรส ไปยังหน่วยความจำแสดงผล
2. อ่านค่าข้อมูลจากหน่วยความจำ
3. ปรับปรุง-แก้ไข-เปลี่ยนแปลงข้อมูล
4. เขียนข้อมูลกลับลงไปที่ตำแหน่งที่เลือกไว้ในตอนแรก

การทำงานของ RMW ขณะที่ทำการแก้ไขข้อมูล นั้น ต้องประกอบด้วยส่วนพื้นฐานหลัก ๆ 3 ส่วนด้วยกันคือ

- Pattern Register เก็บรูปแบบของข้อมูลที่จะย้ายไปยังหน่วยความจำ โดยพารามิเตอร์ ของคำสั่ง WDAT หรือในการวาดจาก PRAM
- MASK Register กำหนดว่าบิทใด ในข้อมูลที่อ่านเข้ามาจะถูกแก้ไข
- 16 Bit Logic Unit เลือกการทำงานว่าจะ Replace , Complement , Set หรือ Clear Data ที่อ่านมาจากหน่วยความจำแสดงผล

1. Pattern Register

- ข้อมูลข้างใน จะถูก AND กับ ข้อมูลใน Mask Register ซึ่งกำหนดมาว่าบิทใดสามารถแก้ไขได้

- ในการวาดภาพกราฟฟิค ข้อมูลใน Pattern Register จะถูกคิดกับ Mask Register ครั้งละบิท เท่านั้น

- เมื่อมีการ AND กับบิทที่เป็น 1 ใน Mask Register บิทนั้นจะถูก แก้ไข โดย Logic Unit สำหรับจุดถัดไปในภาพ บิทถัดไปใน Pattern Register จะถูกเลือกมาทำงานต่อไป

- Execute Word Address Pointer Register (EAD) จะทำหน้าที่กำหนดแอดเดรสของเวิร์ด ที่บรรจุจุดนั้น

- ใน Character Mode ทุกบิทใน Pattern Register จะถูกใช้แบบขนานในการกำหนดลักษณะการแก้ไขข้อมูลทั้งเวิร์ด ซึ่งต่างกับ ลักษณะการทำงานครั้งละบิทต่อบิท ในการวาดภาพ ตรงที่ จะทำพร้อมกันทุก ๆ บิท ในหน่วยความจำ word Mask Register ต้องกำหนดเป็น1 ในตำแหน่งที่จะทำการแก้ไข

2. Make Register

สามารถถูก Load ได้ 2 ทาง คือ

- ใน Graphics Mode คำสั่ง CURS จะกำหนดค่า หรือ ตำแหน่ง Dot Address(dAD) ลักษณะ 4 บิท ซึ่งตัวประมวลผลคำสั่ง จะทำการแปลงให้เป็นแบบ 1 ใน 16 แบบ ทันที คือ แปลงเป็นขนาด 16 บิท และ จะมีเพียงบิทเดียว ที่จะแอคทีฟ (active) ใน 16 บิทนั้น

- ใน Character Mode คำสั่ง Mask จะสามารถ Load MaskRegister เข้าไปได้ ในลักษณะ Full 16 Bits

3. Logic Unit

จะใช้ข้อมูลจาก - ข้อมูลที่อ่านขึ้นมาจากหน่วยความจำแสดงผล

- Pattern Register

- Mask Register

นำมาประมวลผลร่วมกับ Logic Unit ให้ได้ผลลัพธ์เป็นข้อมูลที่จะทำการเขียนกลับลงหน่วยความจำแสดงผล โดยมีการทำงาน 4 ลักษณะ คือ Replace, Complement, Clear, Set ทั้ง 4 แบบนี้ ถ้า Mask bit เป็น 0 ข้อมูลที่อ่านขึ้นมาจะไม่ถูกแก้ไข แต่ถ้า Mask bit เป็น 1 และ Logic Unit เป็น Replace แล้ว ข้อมูลที่อ่านขึ้นมาได้จะถูกแทนด้วย Pattern Register ส่วนแบบอื่นอีก 3แบบนั้น ค่า 0 ในข้อมูลแสดงว่าให้ เขียนค่านั้นกลับลงไป เหมือนเดิม ส่วนค่า 1 ในข้อมูลจะถูกทำงานร่วมกับLogic Unit และ Mask bit

2.8 Figure Drawing

GDC จะวาดภาพแบบกราฟฟิค ในอัตรา 1 จุดต่อหนึ่ง Read-Modified-Write (RMW) Cycle Display Memory Cycle ซึ่งประกอบด้วย 4 คาบเวลา ดังนั้นที่สัญญาณนาฬิกา 5 MHz RMW Cycle จะใช้ เวลา 800 nS

ในระหว่างการวาด GDC จะหาจุดถัดไปที่จะวาด เป็นจุดที่อยู่ติดกัน ซึ่งสามารถกำหนดได้ถึง 8 จุด หรือ 8 ทิศทาง รอบ ๆ จุดเดิม โดย GDC จะกำหนดทิศทาง เหล่านี้เป็นหมายเลข 0-7 เริ่มจากด้านล่าง แล้ววนทวนเข็มนาฬิกา ดังรูป

O_5 O_4 O_3

O_6 X O_2

O_8 O_0 O_1

X : จุดปัจจุบัน

O : จุดต่อไป

ในการวาดภาพ สิ่งที่ต้องการคือแอดเดรส , ตำแหน่งของจุดในแอดเดรสนั้น และทิศทางในการวาด ในการย้ายในทิศขึ้นหรือลง จะใช้การลบหรือบวกด้วยจำนวนword ต่อเส้น ในหน่วยความจำแสดงผล ซึ่งจะมีพารามิเตอร์ตัวหนึ่ง ชื่อ Pitch ที่จะเก็บค่าจำนวนเวิร์ดต่อเส้นนี้ไว้ ในการย้ายไปเวิร์ดทางซ้ายหรือขวา Execute Word Address (EAD) จะถูกเพิ่ม หรือ ลดลง ในขณะที่ Dot Address จะย้ายไปอยู่ที่ LSB หรือ MSB ของ Mask Register ในการย้ายไปทาง ขวา หรือ ซ้ายของ word เดียวกัน ใช้วิธีหมุน Dot Address Pointer Register ไปทางขวาหรือ ซ้าย

ตารางแสดงการทำงาน ในการวาดภาพ ในแต่ละทิศทาง

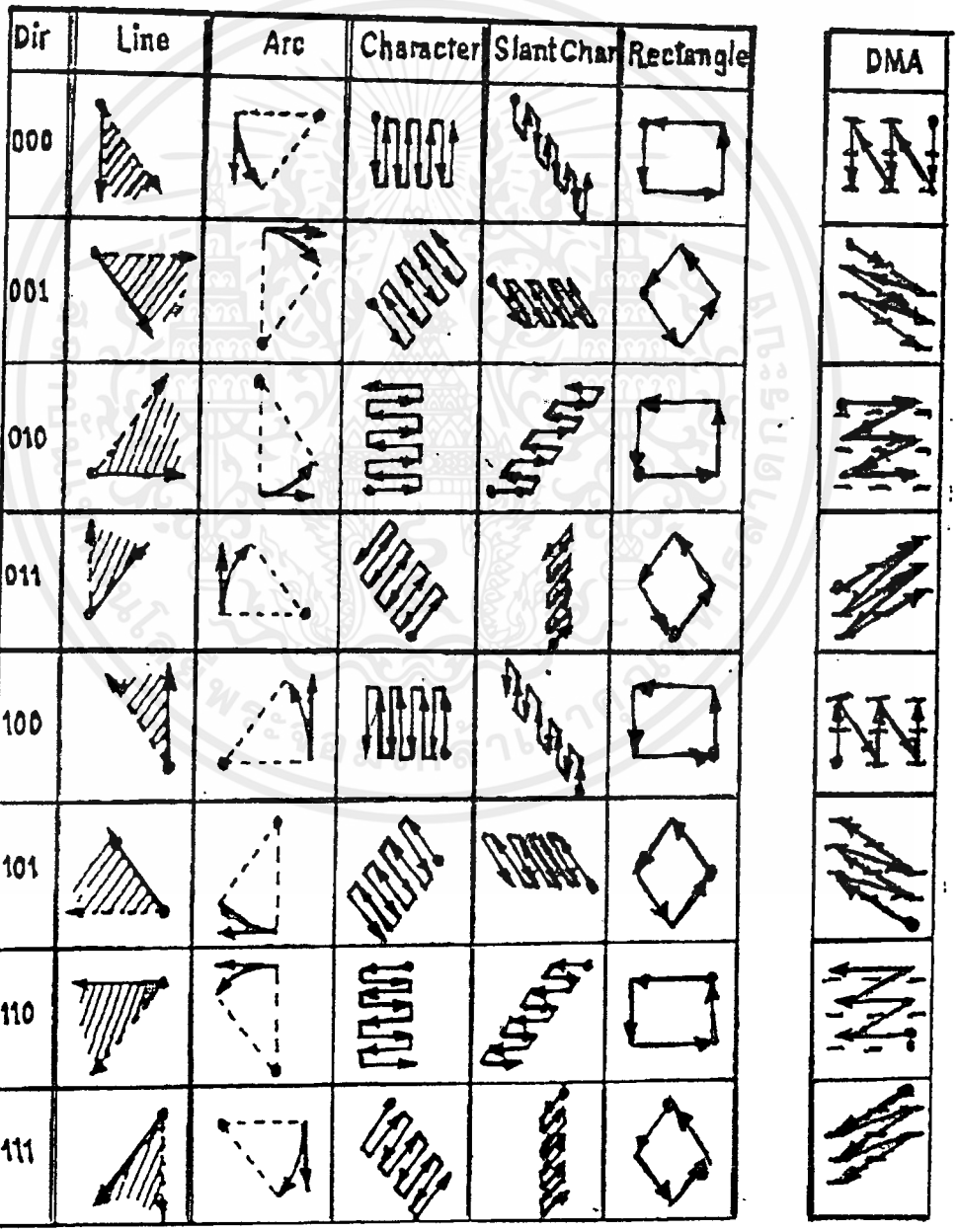
DIR	OPERATION TO ADDRESS THE NEXT PIXEL	
000	EAD + P -> EAD	
001	EAD + P -> EAD	
	dAD(MSB) = 1:EAD + 1 -> EAD	dAD -> LR
010	dAD(MSB) = 1:EAD + 1 -> EAD	dAD -> LR
011	EAD - P -> EAD	
	dAD(MSB) = 1:EAD + 1 -> EAD	dAD -> LR
100	EAD - P -> EAD	
101	EAD - P -> EAD	
	dAD(LSB) = 1:EAD - 1 -> EAD	dAD -> RR
110	dAD(LSB) = 1:EAD - 1 -> EAD	dAD -> RR
111	EAD + P -> EAD	
	dAD(LSB) = 1:EAD - 1 -> EAD	dAD -> RR

Where P = Pitch LR = Left Rotate RR = Right Rotate

EAD = Execute Word Address

dAD = Dot Address stored in the Mask Register

การกำหนดทิศทาง มีผลต่อการวาดภาพแบบต่าง ๆ ดังภาพ



รูปที่ 2.5 ภาพแสดงผลของการกำหนดทิศทางที่มีต่อการวาดภาพ แบบต่าง ๆ

จะเห็นว่าในการลากเส้นตรงนั้น มุมของเส้นจะเป็นไปได้ภายในช่วง 45 องศา ในทิศทางทวนเข็มนาฬิกา วัดจากทิศทางของค่า DIR

ในการลากส่วนโค้ง จะเริ่มวาดในทิศทาง ของค่า DIR และ โค้งเป็นรูปเส้นโค้งซึ่งส่วนโค้งหนึ่ง ๆ สามารถกำหนดให้มีความยาวได้ถึง 45 องศา

2.9 พารามิเตอร์ที่ใช้ในการกำหนดการวาดภาพ

ในการเตรียมการวาดภาพ GDC ต้องการ ชนิดของภาพ, ทิศทาง, พารามิเตอร์ในการวาด ,จุดเริ่มต้น และ รูปแบบ(Pattern) เมื่อค่าเหล่านี้ถูกกำหนดเรียบร้อยแล้ว ก็ใช้คำสั่ง FIGD ในการสั่งให้เริ่มวาด

ลำดับขั้นตอน(Algorithm) ที่ใช้ในการวาดภาพของ GDC ถูกพัฒนามาเพื่อเพิ่มความเร็วในการวาด โดยการใช้การกำหนดรายละเอียดแบบพิเศษ ซึ่งทำให้ CPU ลดงานในการคำนวณหาจุดในการวาด และ โอนงานนี้ไปให้ GDC ทำแทน ซึ่งทำให้ การวาดภาพแบบ จุดต่อจุด (Pixel-by-Pexel) มีความเร็วสูงขึ้น ซึ่งพารามิเตอร์ ต่างๆ มีดังตารางนี้

Drawing Type	DC	D	D2	D1	DW
Initial Value*	0	0	0	-1	-1
Line	$\lvert dL \rvert$	$2\lvert dD \rvert - \lvert dL \rvert$	$2(\lvert dD \rvert - \lvert dL \rvert)$	2dD	-
Arc**	r sin a	r-1	2(r-1)	-1	r sin b
Rectangle	3	A-1	B-1	-1	A-1
Area Fill	B-1	A	A	-	-
Graphic Character***	B-1	A	A	-	-
Write Data	W-1	-	-	-	-
DMAW	D-1	C-1	-	-	-
DMAR	D-1	C-2	(C-2)2^	-	-
Read Data	W	-	-	-	-

* Initial values for the various parameters remain aseach drawing process ends

** Circles are drawn with 8 arcs,each of which span 45°,so that sin a =1\2 and sin b =0

***Graphic characters are as special case of bit-map area filling in which B and A < 8 ,if A=<8 there is no need to load D and D2

Where

1= all ONES value

All numbers are shown in vase 10 for convenience. The GDC accepts base 2 numbers(2s complement notation where appropriate)

-= No parameter bytes sent to GDC for thes parameter

dL= The larger at dx or dy

dD= The smaller at dx or dy

r= Radius of curvature, in pixels

a= Angle from major axis to end of the arc a =< 45°

b= Angle from major axis to start of the arc b =< 45°

^= Round up to the next higher integer

A= Number of pixels in the initially specified direction

B= Number of pixels in the direction at right angles to the initially specified direction

W= Number of words to be accessed

C= Number of words to be transferred in the initially specified direction (Two bytes per word if word transfer mode is selected)

D= Number of words to be accessed in the direction at right angles to the initially specified direction

DC= Drawing count parameter which is one less than the number of RMW cycles to be executed

DM= Dots masked from drawing during arc drawing

2.10 การวาดตัวอักขระกราฟฟิค

ในการวาดอักษรกราฟฟิก GDC จะวาดแบบ pixel-by-pixel โดยการ Load ตัวอักษรที่จะวาดนี้จะถูก Load เข้าไปเก็บไว้ใน parameter RAM ซึ่งเก็บได้ใหญ่สุดถึงขนาด 8x8 และสามารถกำหนดการวาดอักขระได้ไม่จำกัดรูปแบบ โดยการเปลี่ยนแปลง Parameter ในการวาด, ทิศทางในการวาด และโดยการใช้คุณสมบัติในการ Fill area ก็จะสามารถเขียนอักขระลงซ้ำ ๆ กันไม่จำกัดจำนวนครั้ง โดยไม่ต้อง load parameter RAM ใหม่

เมื่อข้อมูลของอักขระขนาด 8 Byte ถูก Load เข้าสู่ Parameter RAM เรียบร้อยแล้ว ก็สามารถใช้คำสั่ง GCHRD เพื่อสั่งให้ GDC เริ่มทำการเขียนตัวหนังสือลงหน่วยความจำแสดงผลได้ นอกจากนี้ยังมีคำสั่ง Zoom ซึ่งมีพารามิเตอร์ที่สามารถกำหนดขนาดของอักขระที่จะแสดงผลได้ โดยสามารถขยายอักขระได้ถึง 1-16 เท่า โดยค่าแต่ละ bit ที่ถูกเก็บไว้จะถูกทำซ้ำ ในแนวตั้ง และ แนวนอน ตามจำนวนครั้งที่กำหนดในคำสั่ง Zoom การย้าย Pram ไปยังหน่วยความจำแสดงผล ถูกกำหนดด้วยคำสั่ง FIGS

ในการเขียนอักขระขนาด 8x8 pixel นั้น pixel แรกจะเริ่มเขียนจาก LSB ของ RA-15 ไปเรื่อย ๆ จนถึง MSB ของ RA-15 จากนั้นก็จะกระโดดข้ามไป RA-14 ที่บิทที่ตรงกัน (ในขณะนี้คือ MSB) แล้วขยับไปเรื่อย ๆ จนถึง LSB ของ RA-14 และทำแบบสลับฟันปลาเช่นนี้เรื่อย ๆ อีก 6 PRAM Byte

ถ้าพื้นที่ที่จะวาด แคบกว่า 8 pixel ขั้นตอนของการสลับฟันปลานี้ ก็จะกระโดดข้ามไปไบท์ ถัดไป ก่อนถึง MSB

2.11 การกำหนดค่าใน Parameter RAM ตำแหน่งที่ 0 ถึง 15 (RA 0 to 15)

ค่าพารามิเตอร์ ที่ถูกเก็บไว้ใน PRAM นั้น GDC สามารถนำมาใช้ซ้ำ ได้เรื่อย ๆ และสามารถแก้ไขค่าภายใน เฉพาะบาง Byte ได้ โดยไม่มีผลกระทบต่อ Byte อื่น ๆ

PRAM เก็บรายละเอียดได้ 2 ชนิด คือ รายละเอียดของการทำงาน และรูปแบบของภาพที่จะวาด

รายละเอียดของการทำงาน จะแบ่งเป็น 2 บล็อก บล็อกละ 4 Byte โดยจะเก็บ

1. แอดเดรสเริ่มต้นในหน่วยความจำแสดงผล ของแต่ละพื้นที่การแสดงผล (Display Area)

2. ความยาวของ พื้นที่การแสดงผล

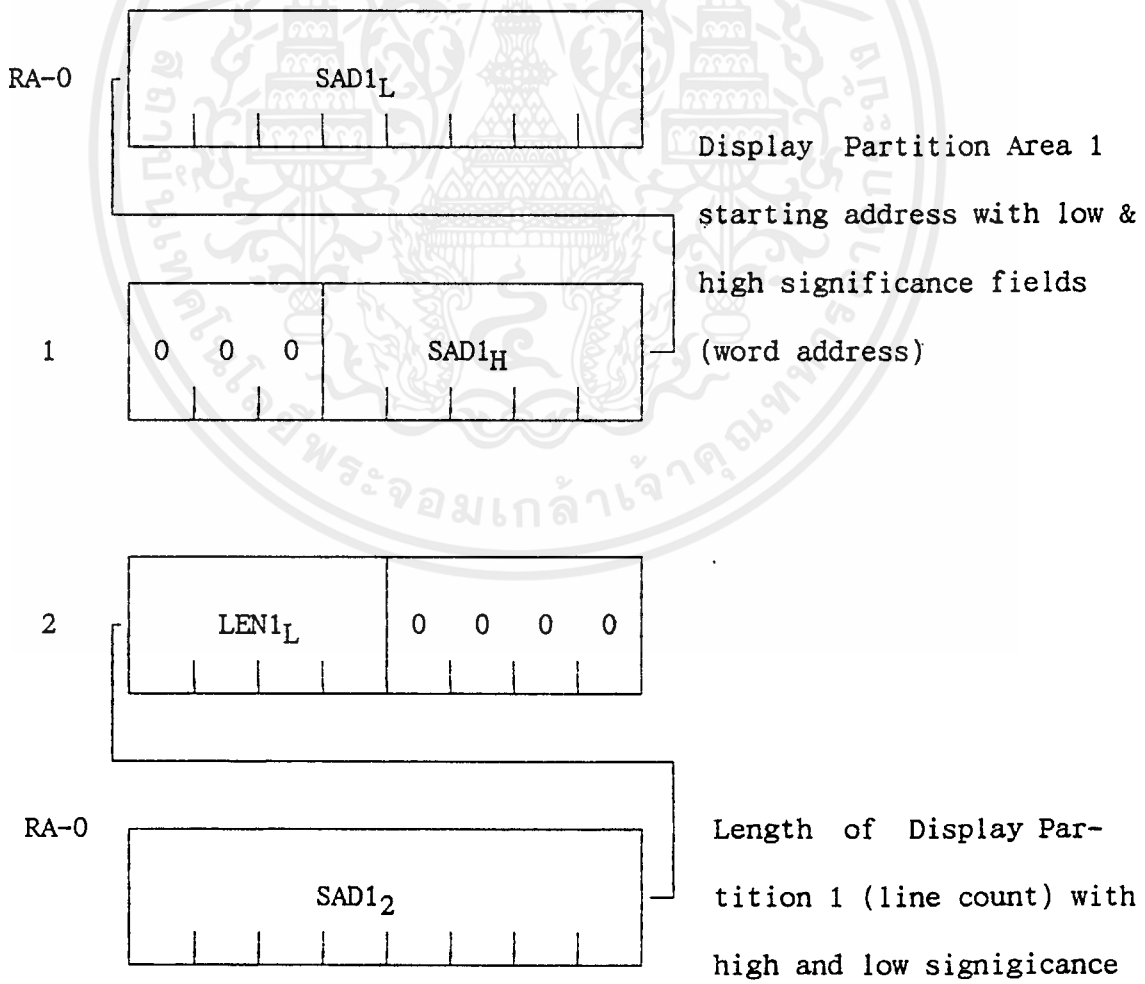
นอกจากนี้ ในแต่ละพื้นที่ ยังสามารถแบ่งเป็น 2 แบบ คือ แบบพื้นที่สำหรับ

BitMap Graphics และพื้นที่สำหรับ Coded Character และยังแบ่งออกได้เป็นแบบ 16 bits หรือ 32 bits wide display cycle

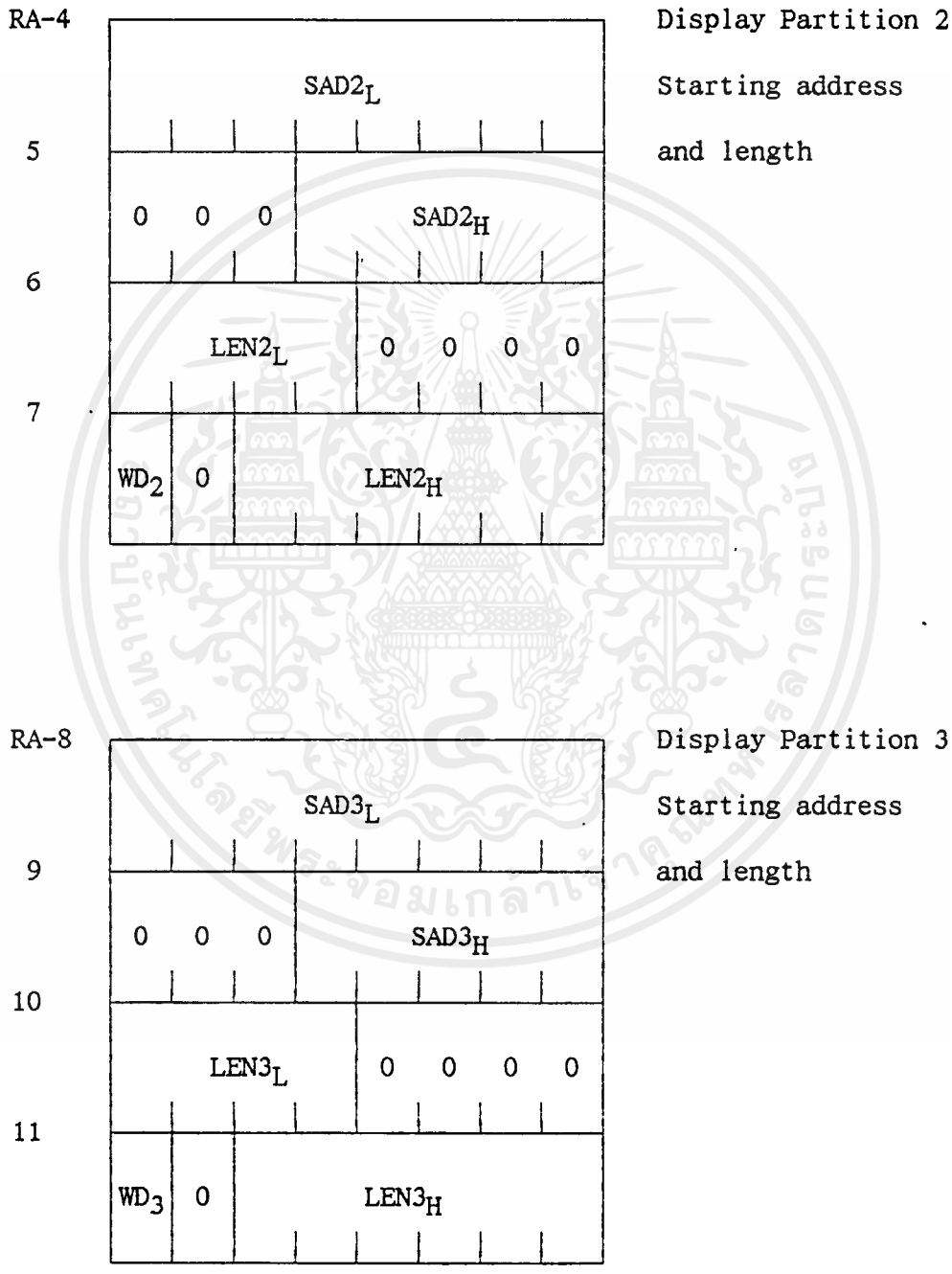
การใช้งาน PRAM นอกจากนี้ ก็เช่นการกำหนดรูปแบบ สำหรับภาพที่จะวาด ใน Graphics Mode ซึ่ง PRAM Byte ที่ 8-15 จะเก็บรูปแบบนี้ไว้ ซึ่งถ้าเป็นการวาดภาพลักษณะเชิงเส้น เช่น เส้นตรง สี่เหลี่ยม เส้นโค้ง แล้ว ก็จะใช้เฉพาะไบท์ 8,9 ในการกำหนดว่าจะวาดเป็นเส้นทึบ หรือเส้นปรับ หรือจุด แต่ถ้าเป็นการ Fill Area และการเขียนอักขระกราฟฟิกแล้ว จะใช้ตั้งแต่ไบท์ 8 ถึง ไบท์ 15 ในการกำหนดภาพที่จะวาด

รายละเอียดของการกำหนดค่าต่าง ๆ ในแต่ละโหมด

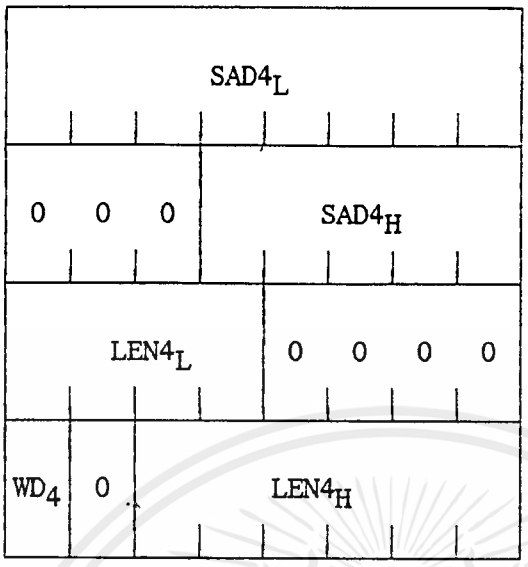
CHARACTER MODE



A Wide Display cycle width of two words per memory cycle is selected for this display area if this bit is set to a 1. The display address counter is then incremented by 2 for each display scan cycle. Other memory cycle types are not influenced.



RA-12



Display Partition 4

Starting address

and length

บทที่ 3

HARDWARE

ในการที่จะออกแบบส่วนวงจรของเครื่อง Multi Screen เราจะต้องคำนึงถึงความสามารถของเครื่อง Multi Screen จุดมุ่งหมายของ โครงงานนี้ จะเป็นการออกแบบเครื่อง Multi Screen เพื่อใช้ในการศึกษา ค่าพารามิเตอร์ต่าง ๆ และส่วนการติดต่อกับเครื่องคอมพิวเตอร์ ดังนั้น คุณสมบัติของเครื่องสำคัญคือ มีความสามารถแก้ไขภาพที่แสดง และแสดงข้อมูลออกได้ที่จอ Monochrome หรือ EGA สามารถย่อขยายภาพได้ ซึ่งการควบคุมสามารถกระทำได้โดยผ่านเครื่องคอมพิวเตอร์

จากคุณสมบัติคร่าว ๆ ดังนี้ พร้อมทั้งจากการศึกษาคุณสมบัติทาง Hardware ของ GDC จะได้ว่า ส่วนวงจรทั้งหมด สามารถแบ่งออกได้เป็น 5 ส่วน

1. ส่วนติดต่อเครื่องคอมพิวเตอร์
2. ส่วนสัญญาณนาฬิกา
3. ส่วนหน่วยความจำ
4. ส่วนสัญญาณควบคุม
5. ส่วนนำข้อมูลแสดงจอภาพ
6. ส่วนประมวลผล

ซึ่งรายละเอียดจะกล่าวเป็นส่วน ๆ ดังต่อไปนี้

ส่วนติดต่อเครื่องคอมพิวเตอร์

ในส่วนของการเชื่อมต่อระหว่างไมโครคอมพิวเตอร์กับการ์ดคอนโทรเลอร์ (GDC 7220)นี้ แบ่งเป็น 2 ส่วนคือ

1. การส่ง/รับข้อมูลให้กับ IC GDC มี IC เบอร์ 74LS245 เป็นบัฟเฟอร์ให้กับสัญญาณข้อมูล(D0-D7)ในที่นี้ได้กำหนดให้เป็นพอร์ตแอดแดรส 308H และ 309H

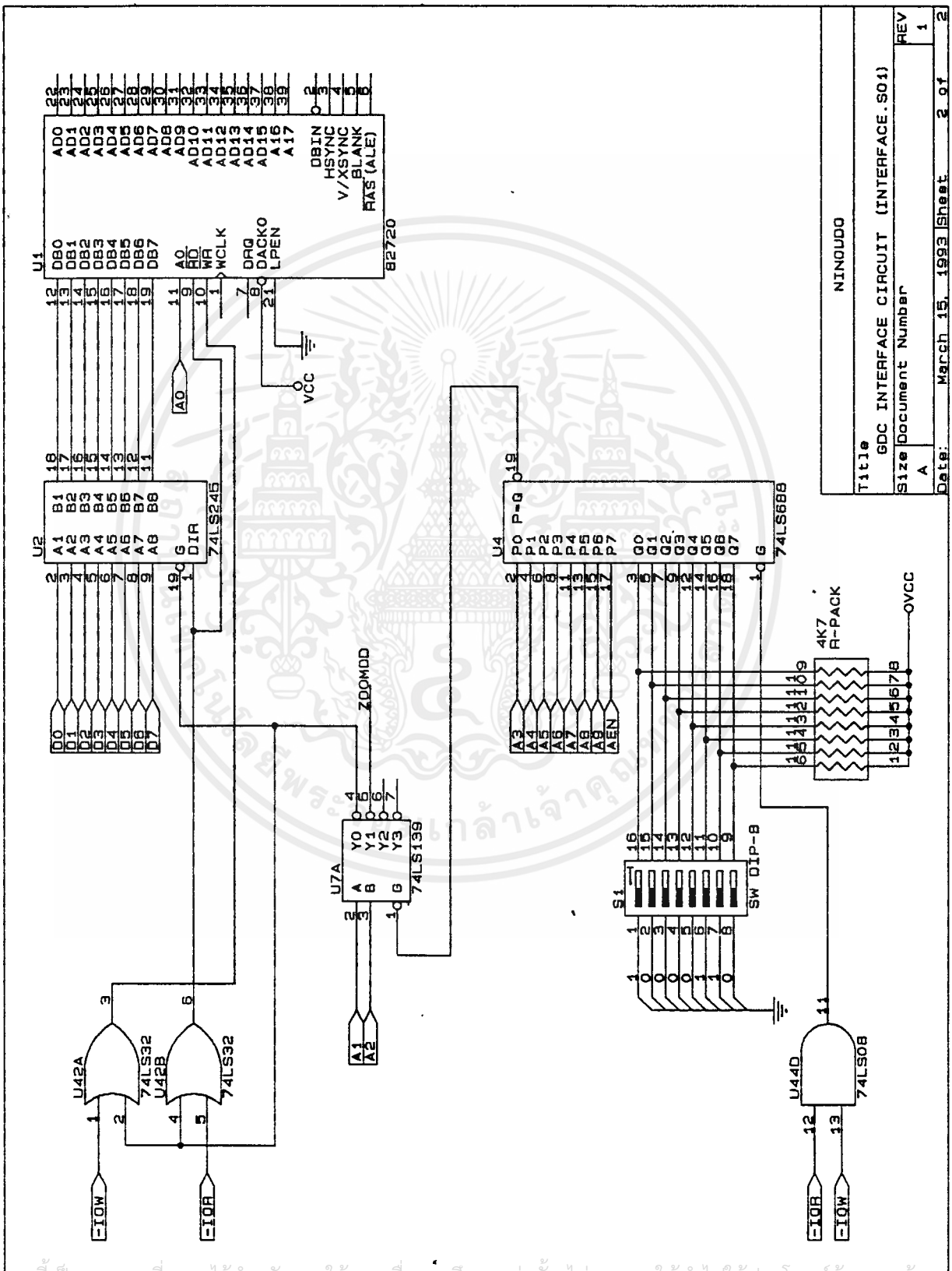
2. การส่งสัญญาณให้ D FLIP-FLOP เพื่อเป็นการกำหนดความถี่สัญญาณ DOTCLK ในวงจรแสดงผลสู่จอภาพ (ซึ่งรายละเอียดอยู่ในหัวข้อที่ 2 ส่วนสัญญาณนาฬิกาและวงจรย่อขยาย) ได้กำหนดให้เป็นพอร์ตเบอร์ 30aH

สัญญาณแอดแดรสจะถูกถอดรหัสด้วยส่วนเปรียบเทียบไอซีคอมพาราเตอร์ 74LS688 และ 74LS139

หลักการทำงาน

จากรูปที่ เมื่อจะทำการส่ง/รับข้อมูล สัญญาณ-IOW หรือสัญญาณ-IOR จะแอคทีฟเป็น LOW ทำให้ U44D มีเอาร์พุต LOW ไปทริกขา GATE ที่ตัวคอมพาราเตอร์ U4 U4 จะทำการเปรียบเทียบสัญญาณแอดแดรสที่ส่งมา กับสัญญาณจากดิพสวิทช์ที่ตั้งไว้ (ขาดิพสวิทช์ที่ต่อขา18(Q7) จะต้องเป็น LOW เพราะในการ INTERFACE จะ ACTIVE เมื่อ AEN เป็น LOW) ถ้าแอดแดรสที่ส่งมาถูกต้อง U4 จะส่งสัญญาณ (ขา 19) ไปทริกตัวถอดรหัส U7A U7A ก็จะส่งสัญญาณ Y0 หรือ Y1 (ขึ้นกับค่า A1,A2 ที่ส่งมา)ไปทริกขา GATE ของไอซีเบอร์ 74LS245 หรือ สัญญาณ ZOOMDD ออกไปตามลำดับ

วงจรส่วนติดต่อ คอมพิวเตอร์ (308H 309H 30AH)

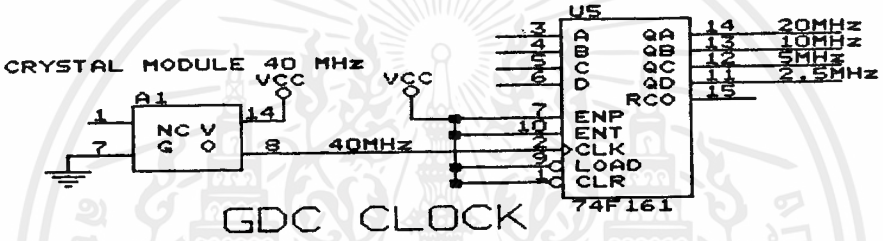


ส่วนสัญญาณนาฬิกา

ในส่วนนี้ สามารถแบ่งได้เป็น วงจรนาฬิกาสำหรับไอซี GDC(uPD 7220) และวงจรนาฬิกาสำหรับการแสดงผลสู่จอภาพ

วงจรนาฬิกาสำหรับ GDC uPD 7220

จากการศึกษาคุณสมบัติ GDC เลือกใช้วงจรนาฬิกาขนาด 2.5 MHz โดยสร้างจากวงจรความถี่ คริสตัล โมดูล 40 MHz มาหาร 8 ด้วยไอซี 74F161 ดังรูป



GDC CLOCK

ในเวลาการทำงานปกติ การนำภาพออกสู่หน้าจอจะใช้ความถี่20 MHz คำนวณมาจาก Timing ในการอ่านข้อมูล 16 บิท จากแรมไปสู่จอภาพจะใช้เวลา 2 clock(state s1,s2)

เวลา 2 state = เวลานำข้อมูลสู่จอภาพ 16 bit

(1/2.5 MHz) x 2 = 16 x (1/Dotclk)

Dotclk = (2.5 MHz/2) x 16

= 20 MHz

เมื่อจะต้องการขยายภาพ จาก 1 ไปเป็น 2,3,...,16 เท่านั้น

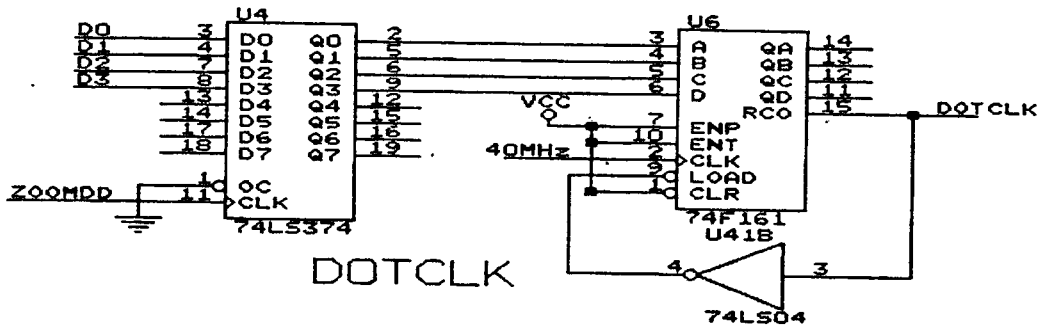
Timing ในการอ่านข้อมูล 16 bit จาก RAM สู่จอภาพจะใช้เวลา 4 clock (state S1-S4)

โดยการคำนวณวิธีเดิมจะได้ Dotclk = 10 MHz

สรุปเป็นตาราง Dotclk ได้ดังนี้

Zoom	Dotclk
1	20MHz
2	10MHz
3	6.67MHz
4	5.00MHz
5	4.00MHz
6	3.34MHz
7	2.86MHz
8	2.50MHz
9	2.22MHz
10	2.00MHz
11	1.82MHz
12	1.67MHz
13	1.54MHz
14	1.43MHz
15	1.33MHz

การออกแบบวงจรได้ออกแบบให้ใช้ IC Counter 74F161 เป็นตัวนับ clk 40 MHz แล้วทำการเซ็ตค่าเริ่มต้นด้วย 74LS374 (ผ่านทางพอร์ท 30AH PC) เมื่อ Counter นับมาถึง 0000 ค่า RCO จะเป็น 0 ส่งเป็นสัญญาณ LOAD ให้ 74F161 ใหม่ดังภาพ



สรุปตาราง

ค่าเริ่มต้น	ค่าความถี่	Zoom
0EH	20MHz	x1
0DH	10MHz	x2
0CH	6.67MHz	x3
0BH	5.00MHz	x4
0AH	4.00MHz	x5
09H	3.33MHz	x6
08H	2.86MHz	x7
07H	2.50MHz	x8
06H	2.22MHz	x9
05H	2.00MHz	x10
04H	1.82MHz	x11
03H	1.67MHz	x12
02H	1.54MHz	x13
01H	1.43MHz	x14
00H	1.33MHz	x15

ส่วนหน่วยความจำ

เป็นการออกแบบหน่วยความจำให้มีขนาดเพียงพอต่อการแสดงผลในโครงงานนี้มีจุดมุ่งหมายเพื่อแสดงจอภาพได้ 2 แบบ คือ แบบโมโน(MONO)และแบบอีจีเอ(EGA)

จากคุณสมบัติของระบบจอ Monochrome ซึ่งมีวิธีการแสดงผล 1 จุดต่อ 1 บิทข้อมูล เพราะฉะนั้น

ขนาดจอมีความละเอียด 720 x 348 จุด ต้องการหน่วยความจำ = 250,560 บิท

เราจะต้องใช้หน่วยความจำขนาด 250,560 / 16 บิท ซึ่งประมาณเท่ากับ 15 K word (เนื่องจากการติดต่อกับ GDC uPD7200 จะใช้ขนาดข้อมูล Word = 16 บิท)

จากคุณสมบัติของจอ EGA 16 สี ความละเอียด 640 x 350 จุด

วิธีการแสดงผล 1 จุดต่อ 4 บิท ข้อมูล(บิทสีแดง,บิทสีเขียว,บิทสีน้ำเงิน,บิทความเข้มสี) ได้แยกวงจรส่วน RAM เป็น 3 Page สี(บิทความเข้มต่อ High ไว้ทำให้ผสมสีได้ 8 สีโดยแต่ละ Page สี(แดง,เขียว,น้ำเงิน) มีความละเอียด 640 x 350 จุด

ขนาดจอความละเอียด 640 x 350 จุด ใช้หน่วยความจำ = 224,000 บิท ต่อ สี

เราจะใช้หน่วยความจำ 224,000 / 16 ประมาณ 14 K word

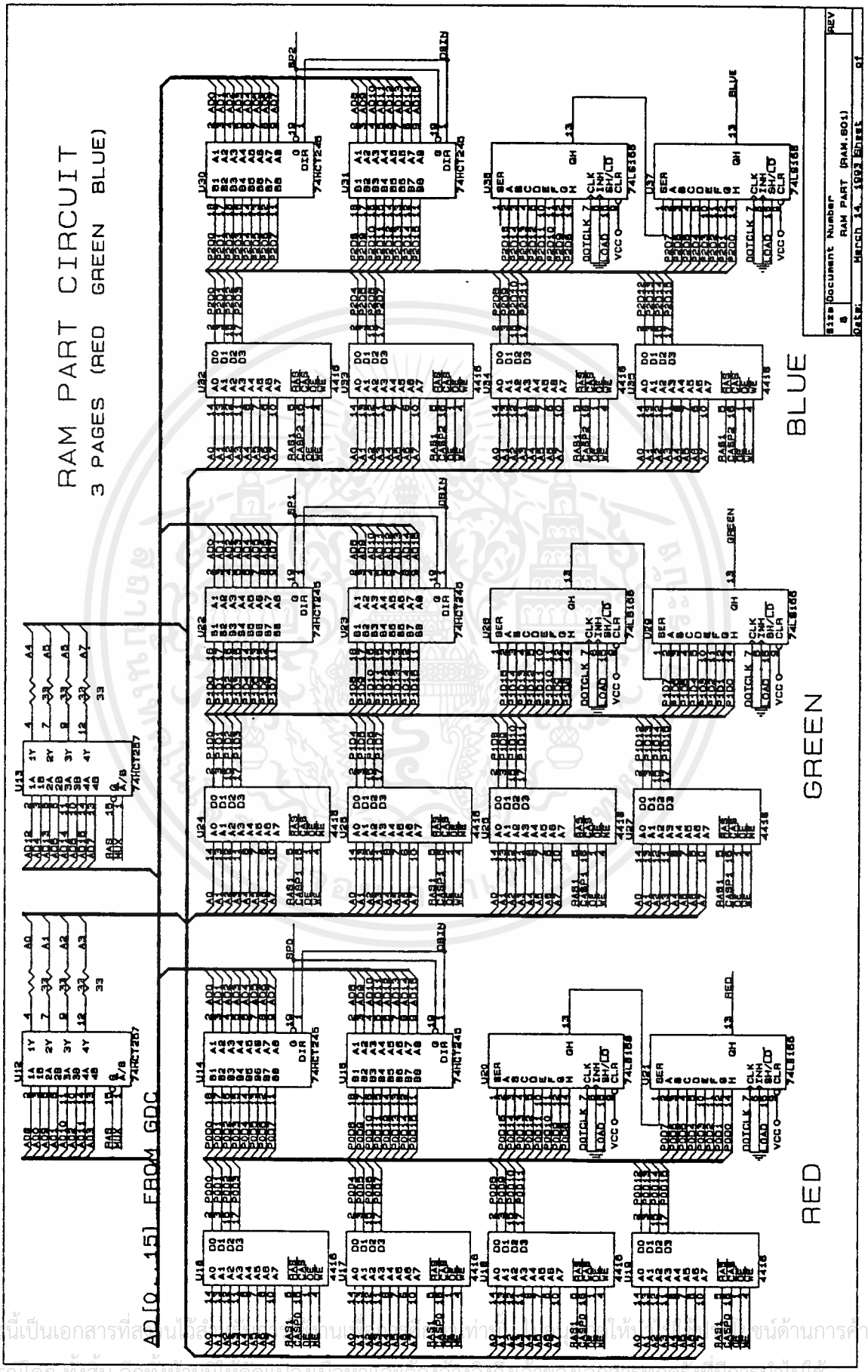
หมายเหตุ

เนื่องจากเราต้องการที่จะออกแบบระบบไว้เพื่อความสามารถในด้านรายละเอียดเพิ่มเติมในอนาคตได้จึงได้ออกแบบหน่วยความจำไว้ถึง 64 K word 16 บิท (คุณสมบัติของหน่วยความจำที่เหมาะสม)

ในการออกแบบจึงใช้ IC Memory ชนิด Dynamic บิทขนาด 64 กิโลไบท์ 4 ตัว มาต่อให้เป็นหน่วยความจำขนาด 64 K word word ละ 16 บิท (SAMSUNG KM41464A-10)

วงจรแสดง หน่วยความจำ ขนาด 64 Kword (Word ละ 16 bit)

3 Page สี (สีแดง สีเขียว และ สีน้ำเงิน)



ส่วนสัญญาณควบคุม

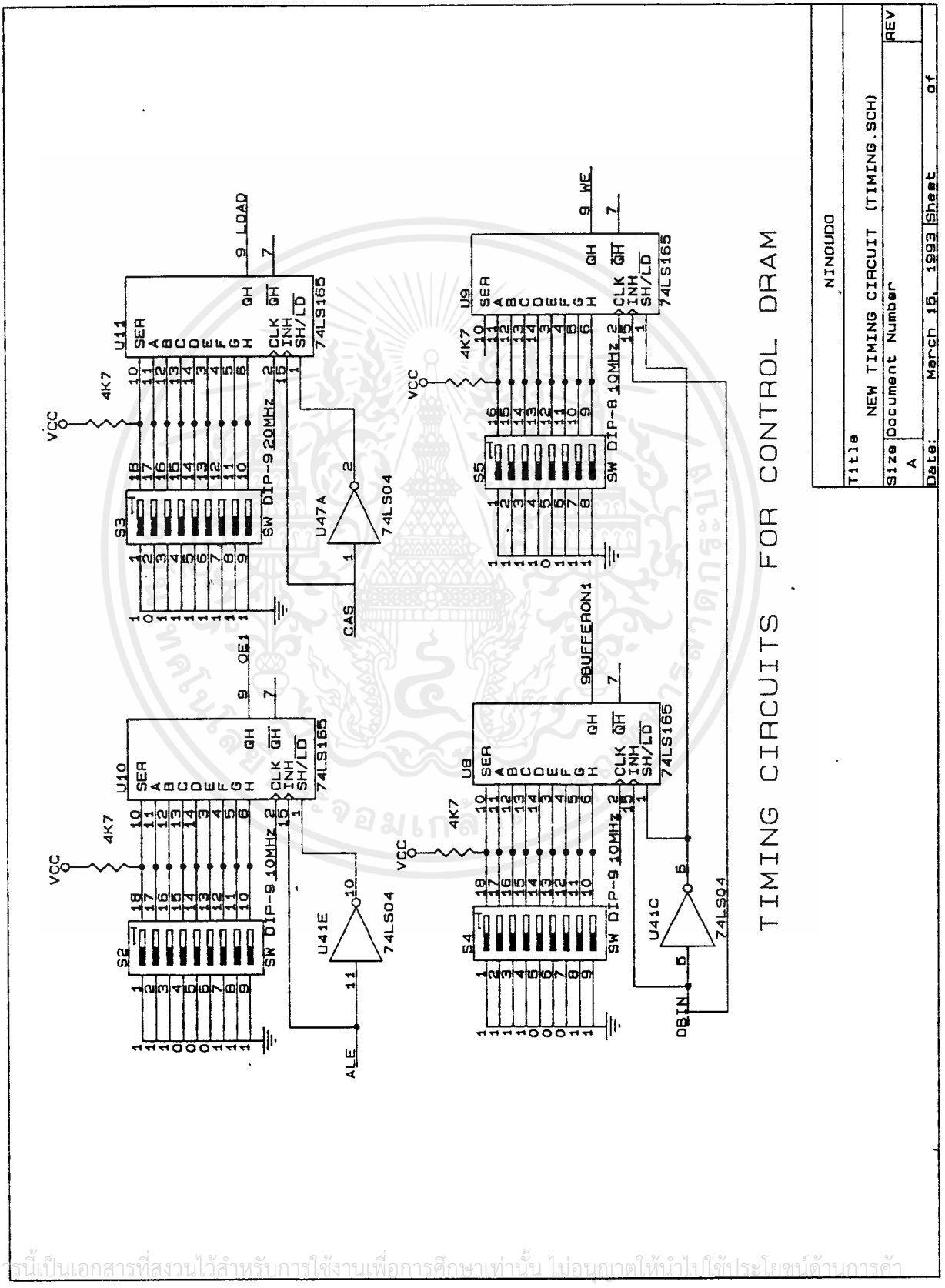
ในการติดต่อ RAM นั้น จะมีอยู่สองช่วงเวลา คือ

- READ CYCLE
- READ MODIFY WRITE CYCLE

จะต้องมีสัญญาณ ควบคุมทิศการไหลของข้อมูล สัญญาณ READ - WRITE และสัญญาณ LOAD สู่จอภาพ

ดังนั้น เราจะต้องสร้างสัญญาณ เหล่านั้น โดยใช้ ไอซี Pararell to Serial เบอร์ 741s165 ซึ่งจากคุณสมบัติของ ไอซีตัวนี้ เราจะสามารถควบคุมระยะเวลาหน่วงความยาวของสัญญาณ ได้

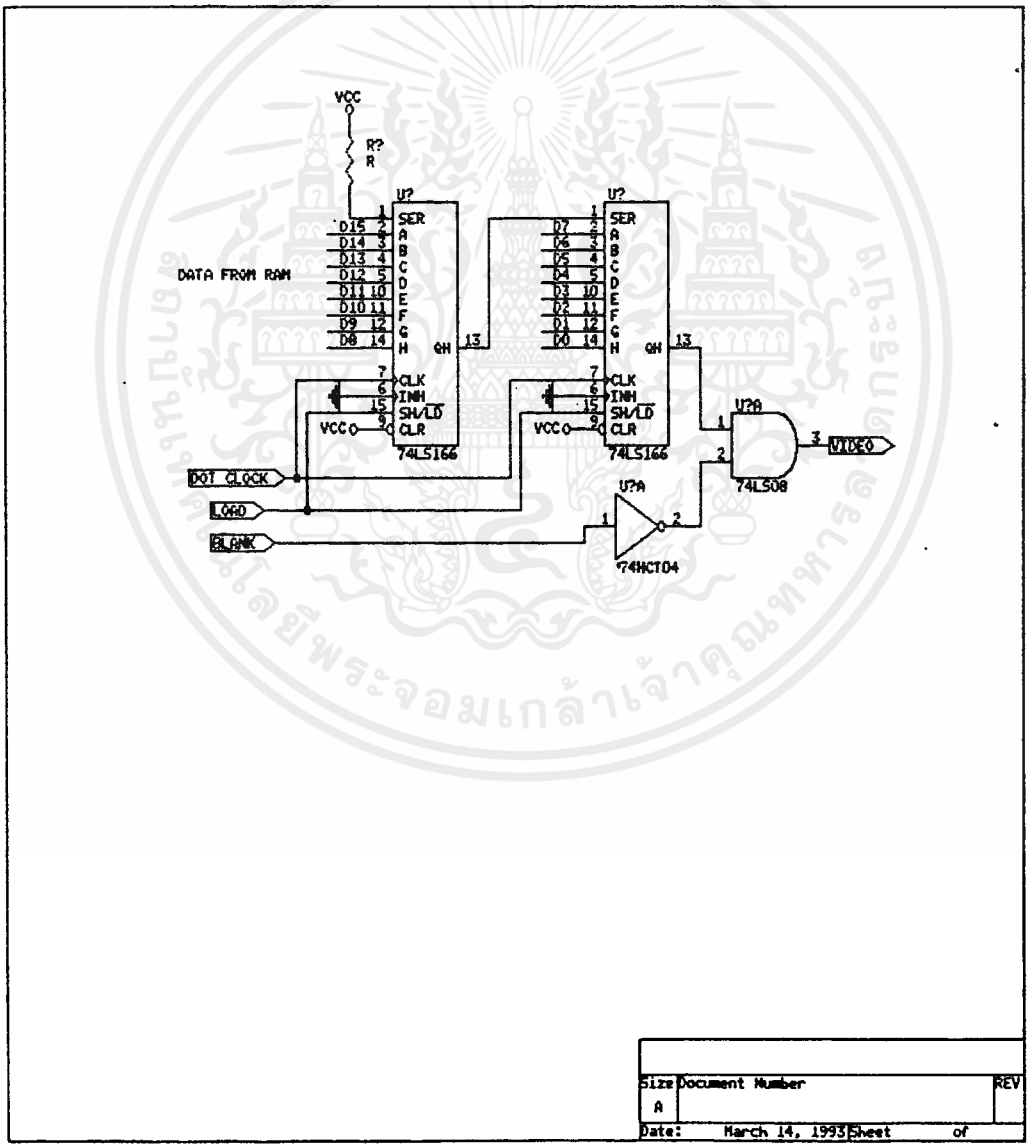
วงจรส่วนสร้างสัญญาณ ควบคุม จาก ไอซี 74ls165



ส่วนนำข้อมูลสู่จอภาพ

ข้อมูลที่จะนำสู่จอภาพ เป็น สัญญาณ แบบอนุกรม โดยที่ข้อมูลนั้น จะมาจาก Dynamic Ram 16 bit และมีความถี่ในการแสดงผล (Dotclk) เป็นค่าต่าง ๆ ตามอัตราการขยายจอภาพ

ในการออกแบบ ได้ใช้ ไอซี Pararell to Serial เบอร์ 741s165 โดยมีสัญญาณในการนำข้อมูลจาก Dynamic Ram (สัญญาณ Load) เป็นสัญญาณ ควบคุม ดังรูป



ส่วนประมวลผล

ในการออกแบบระบบทั้งหมดนั้น จะต้องศึกษาสัญญาณที่ตัว GDC ส่งออกมาเพื่อควบคุม และสัญญาณที่จะใช้ควบคุมการแสดงผลที่จอ Monochrome และ จอภาพ EGA

สัญญาณควบคุมที่ GDC สร้างขึ้นมา (โดยสรุปที่จะใช้ในโครงงานนี้เท่านั้น)

1) AD0 - AD15 เป็นสัญญาณบัสแอดเดรส และบัสข้อมูลในการติดต่อกับส่วนหน่วยความจำ

2) A16 - A17 เป็นสัญญาณควบคุมเลือกเพส ของส่วนหน่วยความจำ ซึ่งสามารถใช้ในการแสดงข้อมูลสีได้

3) DBIN (Display bus Input) เป็นสัญญาณสเตรบอ่านข้อมูล จากส่วนหน่วยความจำเข้าสู่ GDC (จะเกิดในช่วงที่มีการเปลี่ยนแปลงข้อมูล ในหน่วยความจำ Read Modify Write)

4) HSYNC (Horizontal Sync) เป็นสัญญาณสบัดกลับของเส้นที่กวาดบนจอภาพในแนวตั้ง

5) VSYNC (Vertical Sync) เป็นสัญญาณสบัดกลับของเส้นที่กวาดบนจอภาพในแนวนอน

6) BLANK เป็นสัญญาณเอ้าท์พุท เพื่อป้องกันไม่ให้เกิดสัญญาณ Video ปรากฏที่จอภาพ ในตอนที่มีสัญญาณสบัดกลับ

7) ~RAS (ALE) (Row Address Strobe หรือ Address Latch Enable) เป็นสัญญาณในการควบคุม ส่วนหน่วยความจำชนิด Dynamic และใช้ในการ Multiplex สัญญาณบัสแอดเดรส / บัสข้อมูล

สัญญาณที่ใช้ควบคุมจอภาพ Monochrome

1) VSYNC เป็นสัญญาณควบคุมการสบัดกลับในแนวนอน

2) HSYNC เป็นสัญญาณควบคุมการสบัดกลับในแนวตั้ง

3) VIDEO เป็นสัญญาณข้อมูลที่จะแสดงออกจอภาพ ซึ่งจะมาจากส่วนหน่วยความจำทีละ 16 บิท จะต้องมาผ่านวงจรอนุกรม ออกเป็นทีละ 1 บิท

สัญญาณที่ใช้ควบคุมจอภาพ EGA

1) VSYNC เป็นสัญญาณควบคุมการสบัดกลับในแนวนอน

2) HSYNC เป็นสัญญาณควบคุมการสบัดกลับในแนวตั้ง

3) RED VIDEO เป็นสัญญาณข้อมูลที่จะแสดงออกจอภาพ สีแดง

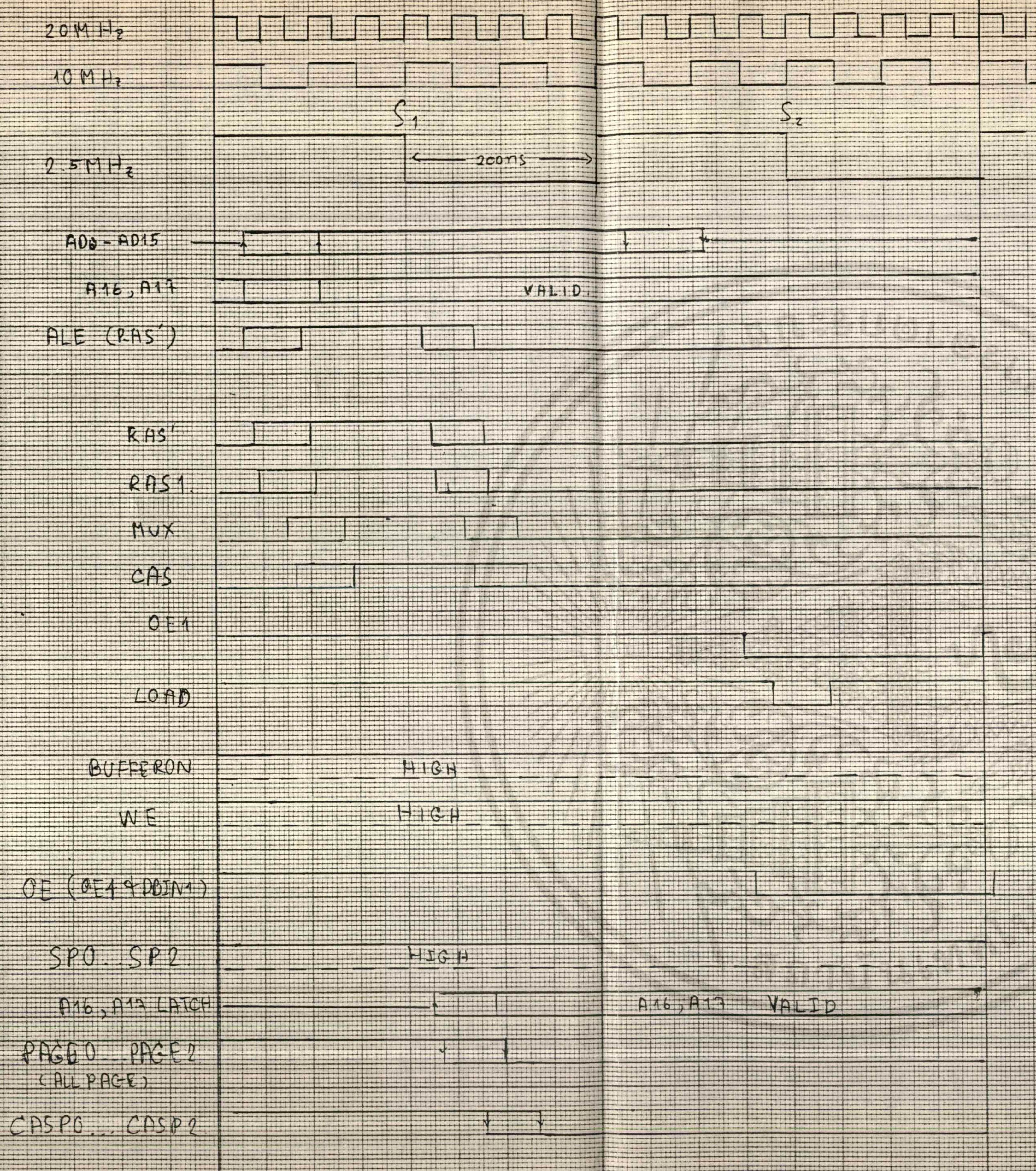
4) GREEN VIDEO เป็นสัญญาณข้อมูลที่จะแสดงออกจอภาพ สีเขียว

5) BLUE VIDEO เป็นสัญญาณข้อมูลที่จะแสดงออกจอภาพ สีน้ำเงิน

6) INTENSITY เป็นสัญญาณแสดงความเข้มของสี

จะพบว่า GDC นั้นจะให้สัญญาณ VSYNC และ HSYNC เท่านั้นที่สามารถควบคุมจอ Monochrome ได้โดยตรง ส่วนสัญญาณ Video นั้น เราจะต้องสร้างขึ้นมา โดยนำสัญญาณต่าง ๆ ที่ออกมาจาก GDC มาผ่านวงจร Hardware ที่เหมาะสม เพื่อที่จะให้ได้สัญญาณ Video ที่มาจากส่วนหน่วยความจำ ที่เป็นสัญญาณอนุกรม ที่มีความถี่เหมาะสม ดังที่แสดงไว้ในส่วนวงจรนาฬิกา

ได้แสดงเวลาสัญญาณต่าง ๆ ของวงจร ซึ่งรวมทั้ง GDC สร้าง และสัญญาณที่ได้ออกแบบ



READ CYCLE

สัญญาณ LOAD DATA จาก RAM สู่วงจรควบคุมข้อมูลจอภาพ.

สัญญาณเปิด Buffer ระหว่าง GDC และ RAM

20MHz

10MHz

2.5MHz

AD0-AD15

DBIN

A16, A17

ALE (RAS')

RAS

RAS1

MUX

CAS

OE1

LOAD

BUFFER ON

WE

OE (OE1 & DBIN1)

A16, A17 LATCH

PAGE 0 ... PAGE 2
ALLPAGE

CAS P0 ... CAS P2

SP0 ... SP2

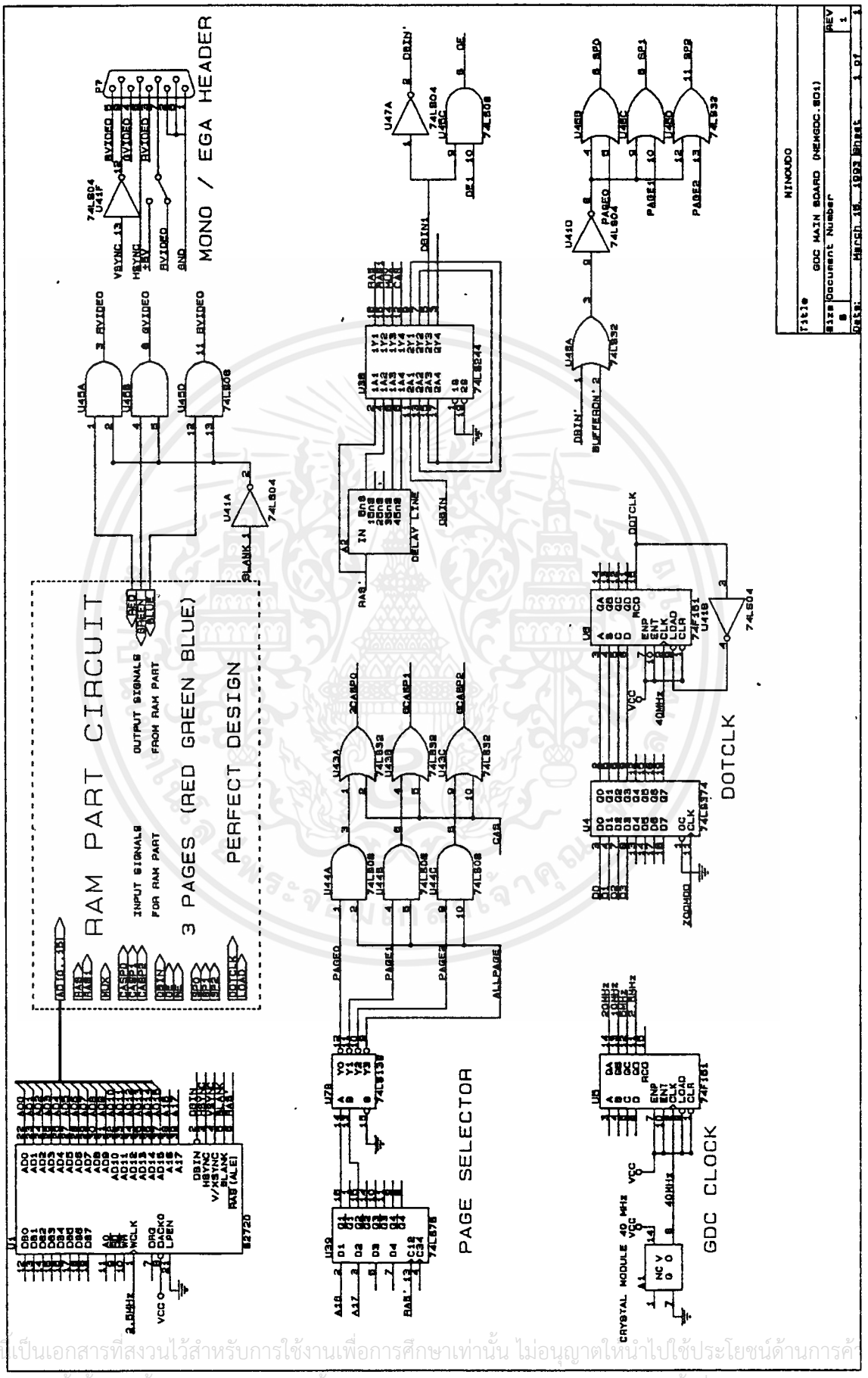
Valid output Address

VALID

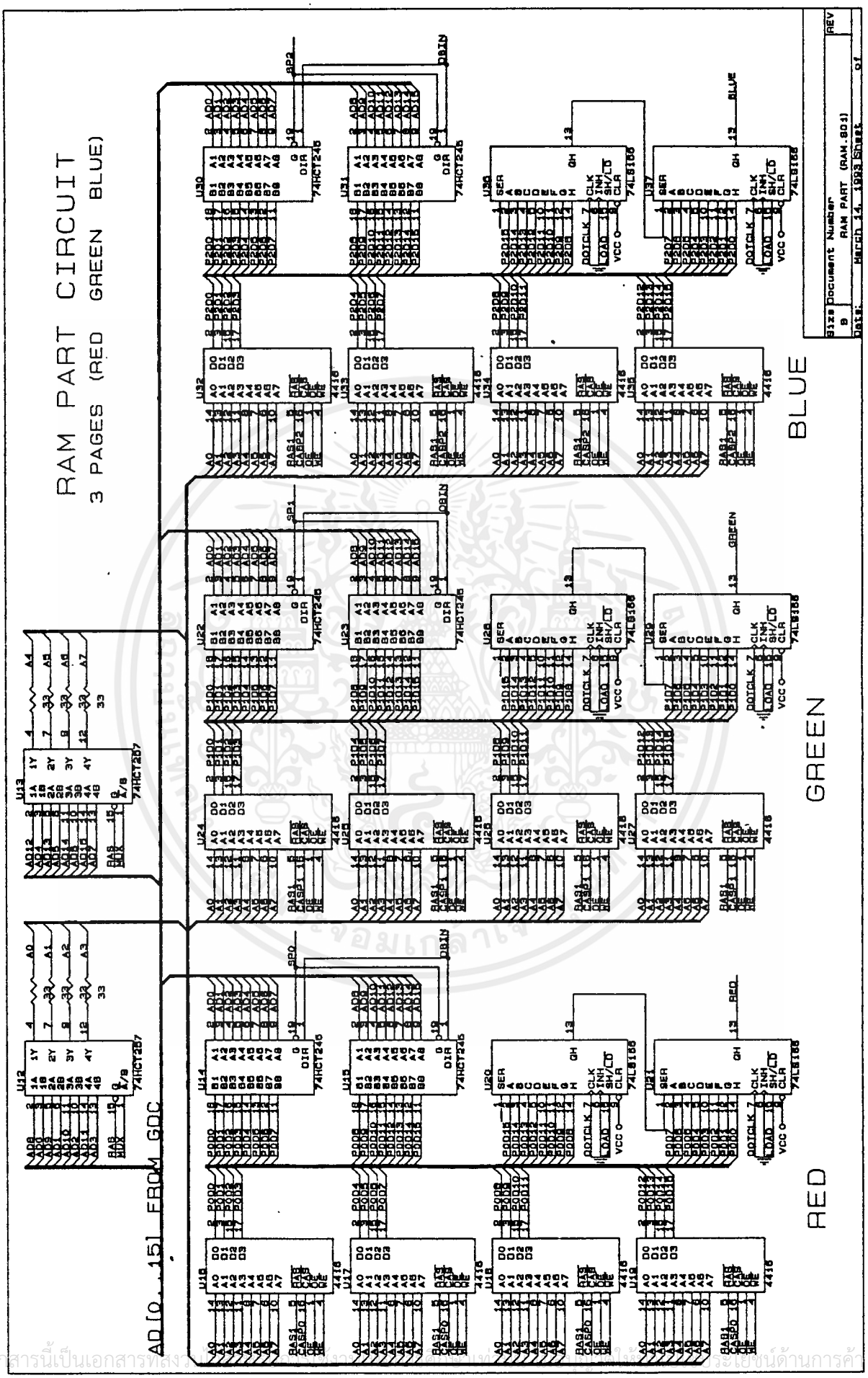
VALID OUTPUT DATA

READ MODIFY WRITE CYCLE

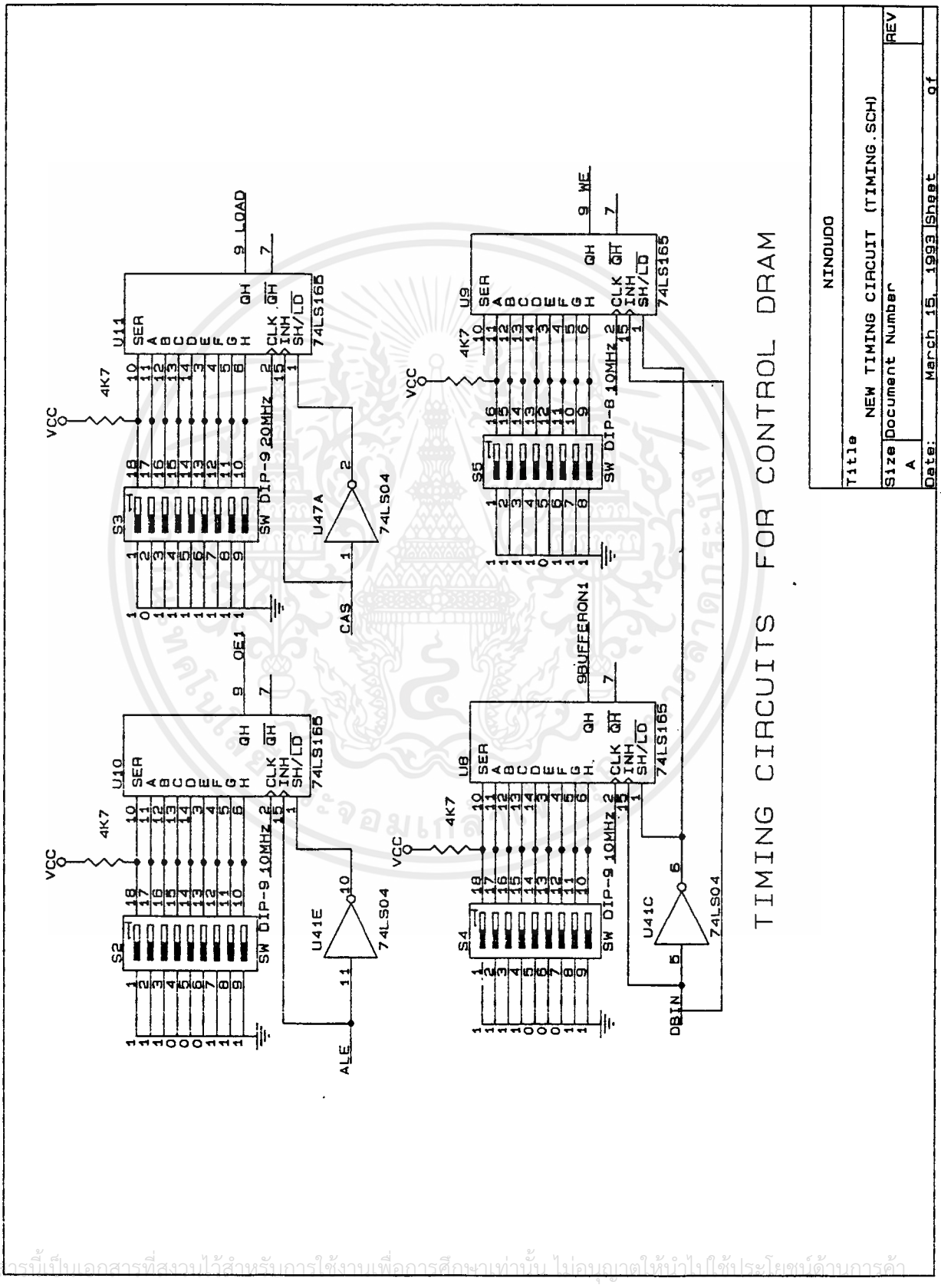
สรุป วงจรหลักของ Graphic Display Controller Board



วงจรส่วนหน่วยความจำ 3 สี (สีแดง สีเขียว สีน้ำเงิน)



วงจรส่วนสร้างสัญญาณควบคุม Dynamic Ram



บทที่ 4

การออกแบบโปรแกรมสำหรับ GDC

ในการทำงาน GDC ต้องการชุดคำสั่งที่เรียงกันเป็นลำดับดังนี้

1 ชุดคำสั่งสำหรับกำหนดสภาวะเริ่มต้นของ GDC ในการควบคุมจอภาพเรียงตามลำดับดังนี้

reset	สำหรับกำหนดขนาดของ Vsync Hsync Blanking และ
sync	โหมดการทำงาน
vsync	
pitch	กำหนดความกว้างของ display memory
pram	กำหนดพื้นที่แสดงผล
cchar	กำหนดรูปแบบของ cursor
zoom	กำหนดขนาดของภาพ

2 คำสั่งสำหรับเริ่มการแสดงผล

start	เริ่มการแสดงผล
bctrl	ควบคุม blanking

3 คำสั่งสำหรับการวาดภาพ

curs	กำหนดตำแหน่งที่จะวาด
pram	รูปแบบของภาพที่จะวาด
figs	รูปแบบของภาพที่จะวาด
figd	วาดภาพที่กำหนดโดย figs
gchrd	วาดข้อมูลกราฟฟิคใน pram
wdat	วาดข้อมูลเป็น ไบท์ หรือ เวอร์ด ลงในหน่วยความจำ

การเลือกค่าพารามิเตอร์ของคำสั่งต่างๆสำหรับกำหนดสภาวะเริ่มต้นของ GDC

RESET : มีการกำหนดพารามิเตอร์ต่างๆดังนี้

P1 ประกอบด้วยแฟลก 0 0 C F I D G S เลือกค่าเป็น 00010110 (16H)

CG เป็น 01 = แสดงผลแบบกราฟฟิคโหมด

IS เป็น 00 = noninterlaced

D เป็น 1 = มีการรีเฟรชไดนามิคส์แรม

F เป็น 1 = GDC ทำการวาดเฉพาะในช่วง retrace blanking

P2 เป็นการกำหนดค่า active word per line สำหรับจอโมโนโครมให้แสดงผลแถวละ 720 จุด ดังนั้น ค่าพารามิเตอร์ = 720/16 = 45 = 2DH จากข้อมูลของ GDC ได้ค่า AW = 2DH - 2 = 2BH

สำหรับจอสีระดับ EGA ให้แสดงผลแถวละ 640 จุด ดังนั้นค่าพารามิเตอร์จึงกำหนดเป็น 640/16 = 40 = 28H ได้ค่า AW = 26H

P3-P8 สามารถคำนวนค่าต่างๆได้ดังนี้

- HFP,HS,HBP ในการทดลองนี้ใช้สัญญาณนาฬิกาควบคุมการส่งสัญญาณภาพ (dot clock) ขนาด 20 MHz จึงมีความสามารถในการแสดงผล 20,000,000 จุดต่อวินาที และมีความต้องการความถี่ในการสแกนในแนวนอน (Hsync) มีค่าเท่ากับ 18.432 KHz หรือ 18432 แถวต่อวินาที ดังนั้นในแต่ละแถวจะมีจำนวนจุด = 20 MHz/18.432 KHz = 1085 จุดต่อแถว ถ้าเป็นจอโมโนโครมแสดงผลเพียง 720จุด จึงมีจุดเหลือที่ไม่ได้แสดงผลในแต่ละแถวเป็น 1085 - 720 = 365 จุด ซึ่งถ้าเลือกค่า HFP,HBP,HS เป็น 7,7,8 word ก็จะคิดเป็นจำนวนจุด (7 x 16) + (7 x 16) + (8 x 16) =352 ซึ่งใกล้เคียงกับค่าที่ต้องการ ถ้าเป็นจอสี EGA จะมีจุดเหลือ 1085 - 640 = 445 จุด จึงเลือกค่า HFP HBP HS เป็น 9 9 9 word จะได้จำนวนจุดเป็น 432 จุด

- VFP,VS,VBP การสแกนในแนวตั้ง (Vsync) มีความถี่ 50 Hz หรือ 50 เฟรมต่อวินาที ดังนั้นในแต่ละเฟรมสามารถแสดงผลได้ 50Hz/18432Hz = 369 แถว แต่ต้องการแค่ 350 แถว เหลือ 19 แถว นำไปกำหนดเป็นค่า VFP,VBP,VS = 6,6,6

SYNC : มีการกำหนดค่าพารามิเตอร์เหมือนคำสั่ง RESET ทุกอย่าง

VSYNC: เลือกค่า M เป็น 1 เพื่อให้มีการทำงานแบบ Master mode

PITCH: จำนวนเวิร์ดต่อแถว สำหรับจอโมโนโครม 720/16 = 45 = 2DH สำหรับจอสี 640/16 = 40 = 28H

PRAM : กำหนดพื้นที่แสดงผลเป็น 300H

CCHAR: ในการทดลองนี้ไม่ต้องการทำงานเกี่ยวกับการแสดงเคอร์เซอร์ จึงกำหนดพารามิเตอร์ต่างๆเป็น 0 ทั้งหมด

ZOOM: กำหนด เป็น 1 เพราะเป็นสถานะเริ่มต้นยังไม่ต้องการขยายภาพ

สามารถเขียน algorithm ได้ดังนี้

```
PROCEDURE reset;
     begin
          sendcommand(00H);
          sendparameter(16H);
          sendparameter((dotperline div 16)-2);
          sendparameter(((Vsync SHL 5) AND e0H) OR
                         ((Hsync-1) AND 1fH));
          sendparameter(((HFP-1) SHL 2) AND fcH) OR
                         (( Vsync SHR 3) AND 03H));
          sendparameter((HBP -1)AND 3fH);
          sendparameter(VFP AND 3fH);
          sendparameter(LO(lineperfield));
          sendparameter(((VBP SHL 2) AND fcH) OR
                         (Hi(lineperfield) AND 03H);
     end;

PROCEDURE sync;
     begin
          sendcommand(0fH);
          sendparameter(16H);
          sendparameter((dotperline div 16)-2);
          sendparameter(((Vsync SHL 5) AND e0H) OR
                         ((Hsync-1) AND 1fH));
          sendparameter(((HFP-1) SHL 2) AND fcH) OR
                         (( Vsync SHR 3) AND 03H));
          sendparameter((HBP -1)AND 3fH);
          sendparameter(VFP AND 3fH);
          sendparameter(LO(lineperfield));
```

```
            sendparameter(((VBP SHL 2) AND fcH) OR
                          (Hi(lineperfield) AND 03H);
      end;

PROCEDURE Vsync;
      begin
            sendcommand(6fH);
      end;

PROCEDURE CCHAR;
      begin
            sendcommand(4bH);
            sendparameter(0);
            sendparameter(0);
            sendparameter(0);
      end;

PROCEDURE ZOOM;
      begin
            sendcommand(46H);
            sendparameter((((Disp-1) SHL 4) AND f0H) OR
                          ((Gchr-1)AND 0fH));
      end;

PROCEDURE PITCH;
      begin
            sendcommand(47h);
            sendparameter((dotperline div 16));
      end;
```

```
PROCEDURE PRAM;
     begin
          sendcommand(70H + (startaddress AND 0fH);
          for i=0 to NumberofParameter
               sendparameter(data[i]);
     end;
```

คำสั่งสำหรับเริ่มการแสดงผลมีสองคำสั่งสามารถเขียนเป็น algorithm ได้ดังนี้

```
PROCEDURE START;
begin
     SendCommand($6B);
end;

PROCEDURE BCTRL;
begin
        SendCommand($0C+DE)
end;
```

การเลือกค่าพารามิเตอร์ต่างๆสำหรับคำสั่งในการวาด

CURS : สามารถคำนวนค่า EAD และ dAD ได้ดังนี้คือ

ให้ P เป็นจำนวนเวิร์ดในหนึ่งแถว ดังนั้น $P = (X_{max} + 1)/16 = 45 = 2DH$

จะได้ค่า line base address (LBA) = P x Y

เมื่อ X คือ ตำแหน่งของแถวที่ต้องการวาด และ Y คือตำแหน่งที่ต้องการวาดในแนวตั้ง เราสามารถหาตำแหน่งของหน่วยความจำได้ดังนี้

LBA = Y x 2DH

EAD = LBA + trunc(X/16)

dAD = X mod 16

สำหรับรายละเอียดของคำสั่งอื่นๆสามารถอ่านได้ในบทที่ 2

สามารถเขียนเป็น algorithm ได้ดังนี้

```
PROCEDURE CURS;
begin
     SendCommand($49);
     SendParameter(lo(EAD));
     SendParameter(hi(EAD));
     SendParameter ((lo(dAD) shl 4) or  (hi(EAD shr 8)));
end;



PROCEDURE FIGS;
begin
SendCommand($4C);
if FigType = 'WDAT' then SendParameter($00  or direction) else
if FigType = 'RDAT' then SendParameter($00  or direction) else
if FigType = 'CHAR' then SendParameter($00  or direction) else
if FigType = 'LINE' then SendParameter($08  or direction) else
if FigType = 'GCHR' then SendParameter($10  or direction) else
if FigType = 'AREA' then SendParameter($10  or direction) else
if FigType = 'ARC'  then SendParameter($20  or direction) else
if FigType = 'CIR'  then SendParameter($20  or direction) else
if FigType = 'RECT' then SendParameter($40  or direction) else
if FigType = 'SLAN' then SendParameter($90  or direction)
```



```
     SendParameter(lo(DC));
```

```
SendParameter(hi(DC) AND $3F) ;
SendParameter(lo(D));
SendParameter(hi(D) AND $3F);
SendParameter(lo(D2));
SendParameter(hi(D2) AND $3F);
SendParameter(lo(D1));
SendParameter(hi(D1) AND $3F);
SendParameter(lo(DM));
SendParameter(hi(DM) AND $3F);
end;



PROCEDURE    WDAT;
begin
     Case WrType of
          0 : SendCommand($20 + WrMode);
          2 : SendCommand($30 + WrMode);
          3 : SendCommand($38 + WrMode);
     for i := 0 to NumOfByte-1 do
     begin
          sendparameter(data[i]);
     end;
end;

PROCEDURE FIGD;    {figure draw start }
begin
     sendcommand($6C);
end;
```

```
PROCEDURE GCHRD;
begin
     sendcommand($68);
end;
```

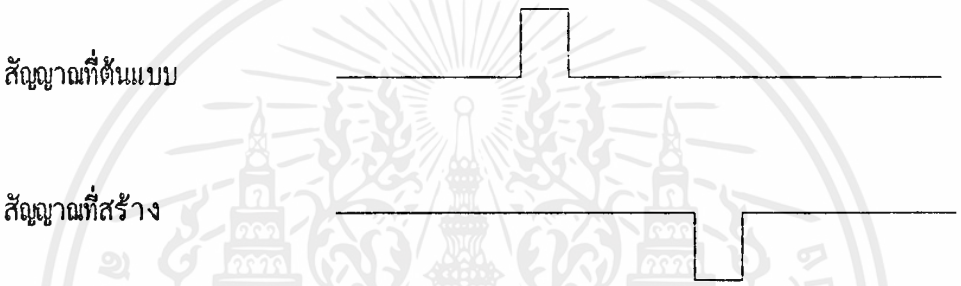
บทที่ 5

การทดลอง และการแก้ไขปัญหา

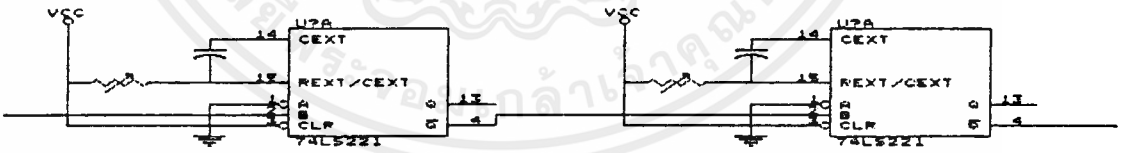
ในการออกแบบวงจร ได้ประสพปัญหามากมาย ดังนี้

1 ปัญหาในการสร้างสัญญาณควบคุม

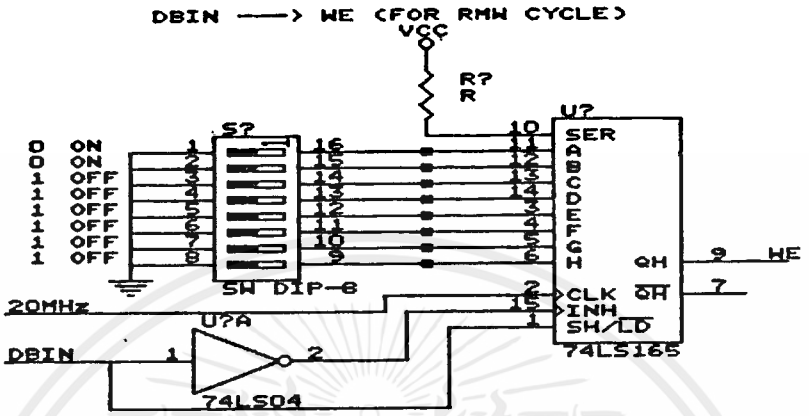
การสร้างสัญญาณควบคุมให้กับไดนามิกแรม มักจะเป็นสัญญาณที่ถูกหน่วงเวลาออกไปช่วงหนึ่งดังรูป



ในการออกแบบวงจรในเทอมแรกได้ใช้ไอซี รีมรันสเตเบิล 74LS221 มาต่อเป็นวงจรดังรูป



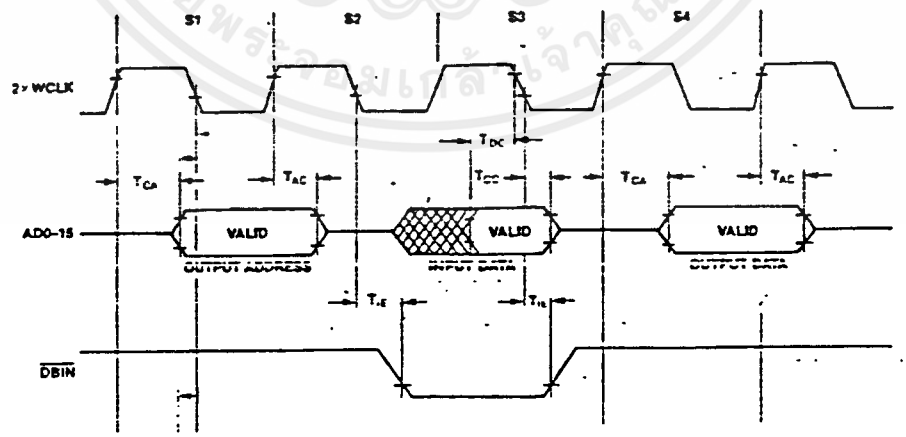
ซึ่งมีปัญหาในการหาตัวต้านทานและตัวเก็บประจุที่มีค่าตามต้องการ ซึ่งจะทำให้ได้คาบสัญญาณที่ไม่แน่นอนหรือไม่ตรงตามต้องการ ในเทอมสองจึงได้ออกแบบให้ใช้ไอซีนานเป็นอนุกรม 74LS165 มาสร้างเป็นวงจรดังรูป



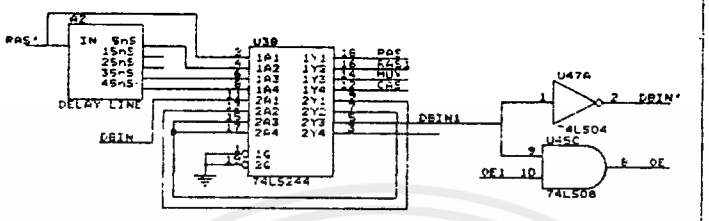
วงจรดังกล่าวสามารถจะควบคุมระยะเวลาในการควบคุมได้ง่ายและค่อนข้างแน่นอนเพราะความถี่ที่ใช้ควบคุมเป็นฐานเวลาเดียวกับสัญญาณนาฬิกา

2 ปัญหาระยะเวลาของข้อมูลที่ GDC ต้องการ

ปัญหาที่พบคือการไหลของข้อมูล ในช่วงเวลา READ MODIFY WRITE มีสัญญาณออกจาก GDC ดังรูป

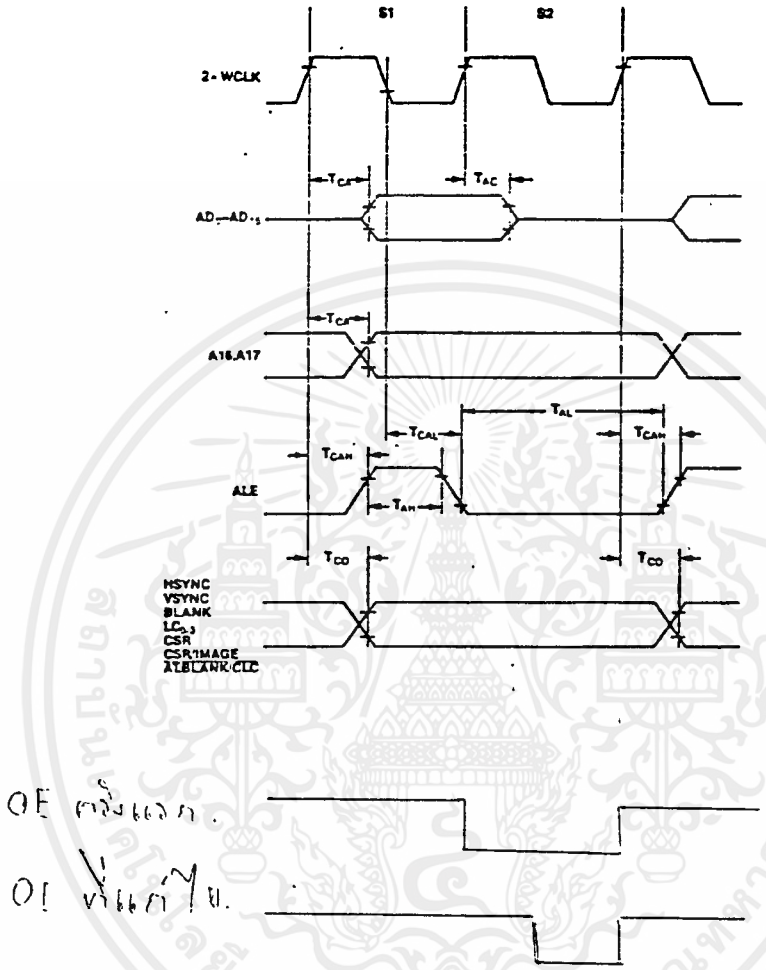


การออกแบบได้นำ DBIN มาเป็นสัญญาณอ่านข้อมูลจากแรม แต่ GDC ต้องการเวลาอีกประมาณ 30 nS หลังจาก DBIN หมดไป การออกแบบในขั้นแรกจึงประสบปัญหาว่า GDC ไม่สามารถอ่านข้อมูลได้ทัน จึงแก้ปัญหาโดยการหน่วงข้อมูลออกไปอีก โดยการใช้ 74LS244 ที่เหลือดังรูป



และยังผ่านเกทตัวอื่นอีกทำให้สามารถหน่วงเวลาออกไป 50 nS

3 ในช่วงเวลาการอ่านข้อมูลออกสู่จอภาพได้ออกแบบ timing ดังรูป พบว่าข้อมูลและแอดเดรสชนกันบนบัสทำให้ ไอซี 74LS245ร้อน จึงได้แก้โดยให้สัญญาณ OE เล็กลงและห่างออกไปจากสัญญาณเอดเดรสดังรูปและยังได้สร้างวงจรควบคุมไม่ให้ 74LS245 ทำงานในช่วงเวลาการนำข้อมูลออกสู่จอภาพ

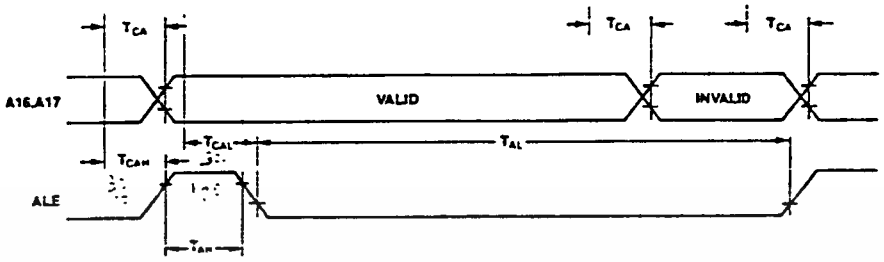


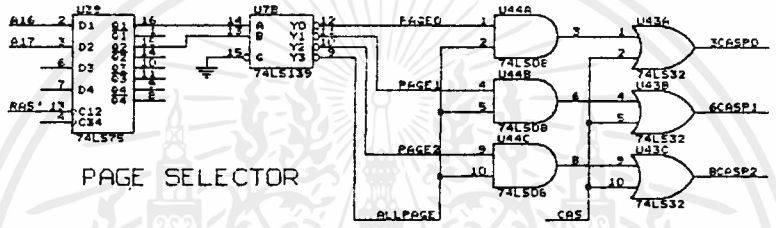
OE ก่อนแก้.

OE หลังแก้ไข.

4 ปัญหาแอดเดรสในการเลือก page ของแรม

ในการออกแบบวงจรได้ใช้ A16 A17 ในการเลือก page ของแรม แต่ภายหลังพบว่าสัญญาณ A16 A17 ไม่แน่นอนในบางช่วงเวลาดังรูปทำให้เกิดปัญหาในการเขียนข้อมูลลงหน่วยความจำ จึงทำการแก้ไขโดยการแลทค่า A16 A17 เอาไว้ด้วย D flip-flop ดังรูป





5 ปัญหาอื่นๆที่พบ

ที่พบมากก็คือปัญหาในการ หน่วงเวลาของสัญญาณเนื่องจากต้องผ่านเกทหลายตัว จึงต้องทำการแก้ไขโดยการใช้ ไอซีที่มีความเร็วสูง

บทที่ 6

สรุป และ วิจารณ์

จากการศึกษา Graphic Display Controller (GDC) uPD 7220 และได้พัฒนาการ์ด ต้นแบบขึ้นมา ทำให้สามารถใช้ในการศึกษา ถึงการออกแบบทั้งด้าน Hardware และ Software สำหรับควบคุมและ ใช้งาน GDC ตัวนี้ได้เป็นอย่างดี สำหรับการ์ดต้นแบบที่ทำการพัฒนาขึ้น มีความสามารถดังนี้

1. สามารถอ้างหน่วยความจำได้ถึง 192 Kword word ละ 16 บิท โดยแบ่งออกเป็น 3 Pageสี Page ละ 64 Kword word ละ 16 บิท
2. ใช้ได้กับจอ Monochrome และจอสี ระดับ EGA 8 สี
3. สามารถใช้ทดสอบคำสั่ง และพารามิเตอร์ต่าง ๆ ของ GDC ได้
4. สามารถ ขยาย และ เลื่อนตำแหน่งจอภาพ (PAN) ดูส่วนต่าง ๆ ของภาพได้ด้วยความเร็วสูง

สำหรับการ์ดที่ได้พัฒนาขึ้น การแสดงผลในระดับจอสี ยังมีปัญหาเล็กน้อย เนื่องจากภาพที่ได้ จะไม่ชัดเจน เนื่องจากการ์ดที่ทำใช้การ ไวร์แรป จึงมีปัญหาเกี่ยวกับการดึงกระแสสัญญาณ และ สัญญาณรบกวน

โดยสรุปแล้ว Graphic Display Controller ตัวนี้ เป็น อุปกรณ์ที่น่าสนใจมากตัวหนึ่ง เนื่องมาจาก การทำงานที่รวดเร็วมาก เมื่อเทียบกับการประมวลผลโดยวิธีอื่น ๆ

กิตติกรรมประกาศ

คณะผู้จัดทำขอขอบพระคุณ อาจารย์ พลผดุง ผดุงกุล อาจารย์ที่ปรึกษา ซึ่งเป็นผู้ริเริ่มโครงงานนี้ โดยท่านได้ทุ่มเทสติปัญญาความคิดในการวิจัย ค้นคว้าให้คำปรึกษาในการแก้ไขปัญหาทั้งทางด้านทฤษฎี และปฏิบัติอย่างเต็มกำลังความสามารถ จนทำให้ปริญญานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี รวมทั้งอาจารย์ วิชชา สำนักวิจัยคอมพิวเตอร์ ที่ได้ให้ความช่วยเหลือ และช่วยไขปัญหาต่าง ๆ

คณะผู้จัดทำ

82720
GRAPHICS DISPLAY CONTROLLER

- Displays Low-to-High Resolution Images
- Draws Characters, Points, Lines, Arcs, and Rectangles
- Supports Monochrome, Gray Scale, or Color Displays
- Zooms, Pans and Windows Through a 4 Mpixel Display Memory
- Extremely Flexible Programmable Screen Display, Blanking, and Sync Formats
- Compatible with Intel's Microprocessor Families
- High-Level Commands Off Load Host Processor from Bit Map Loading and Screen Refresh Tasks
- Supports Graphics, Character, and Mixed Display Modes

FUNCTIONAL DESCRIPTION

Introduction

The 82720 Graphics Display Controller (GDC) is an intelligent microprocessor peripheral designed to drive high-performance raster-scan computer graphics and character CRT displays. Positioned between the video display memory and Intel microprocessor bus, the GDC performs the tasks needed to generate the raster display and manage the display memory. Processor software overhead is minimized by the GDC's sophisticated instruction set, graphics figure drawing, and DMA transfer capabilities. The display memory directly supported by the GDC can be configured in any number of formats and sizes up to 256K 16-bit words. The display can be zoomed and partitioned screen areas can be independently scrolled and panned. With its light pen input and multiple controller capability, the GDC is ideal for most computer graphics applications. Systems implemented with the GDC can be designed to be compatible with standards such as VDI, NAPLPS, GKS, Core, or custom implementations.

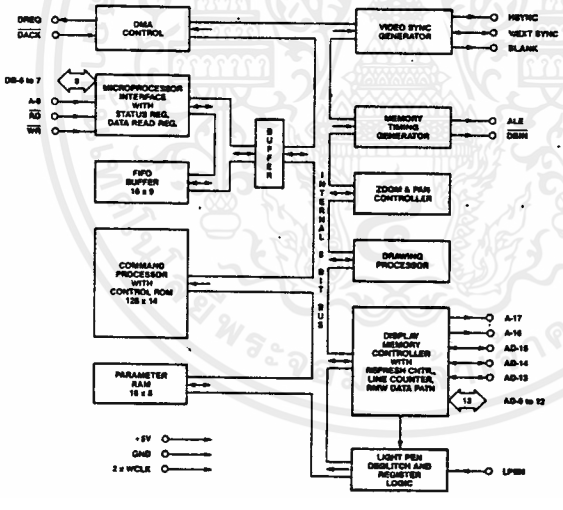


Figure 1. Block Diagram

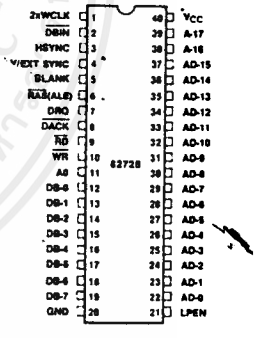


Figure 2. Pin Configuration

JUNE 1983
ORDER NUMBER: 210655-002

Table 1. Pin Description

Symbol	Pin No.	Type	Name and Description
2XWCLK	1	I	**Clock Input**
$\overline{DBIN}$	2	O	**Display Bus Input:** Read strobe output used to read display memory data into the GDC.
HSYNC	3	O	**Horizontal Sync:** Output used to initiate the horizontal retrace of the CRT display.
V/EXT SYNC	4	I/O	**Vertical Sync:** Output used to initiate the vertical retrace of the CRT display. In slave mode, this pin is an input used to synchronize the GDC with the master raster timing device.
BLANK	5	O	**Blank:** Output used to suppress the video signal.
$\overline{RAS}$ (ALE)	6	O	**Row Address Strobe (Address Latch Enable):** Output used to start the control timing chain when used with dynamic RAMs. When used with static RAMs, this signal is used to demultiplex the display address/data bus.
DRQ	7	O	**DMA Request:** Output used to request a DMA transfer from a DMA controller (8237) or I/O processor (8089).
$\overline{DACK}$	8	I	**DMA Acknowledge:** Input used to acknowledge a DMA transfer from a DMA controller or I/O processor.
$\overline{RD}$	9	I	**Read:** Input used to strobe GDC Data into the microprocessor.
$\overline{WR}$	10	I	**Write:** Input used to strobe microprocessor data into the GDC.
A0	11	I	**Register Address:** Input used to select between commands and data read or written.
DB0	12	I/O	**Bidirectional Microprocessor Data Bus Line:** Input enabled by $\overline{WR}$. Output enabled by $\overline{RD}$.
DB1	13		
DB2	14		
DB3	15		
DB4	16		
DB5	17		
DB6	18		
DB7	19		
GND	20		**Ground.**
V_{CC}	40		**+5V Power Supply**
A_{17}	39	O	**Graphics Mode:** Display Address Bit 17 Output **Character Mode:** Cursor and Line Counter Bit 4 Output **Mixed Mode:** Cursor and Image Mode Flag
A_{16}	38	O	**Graphics Mode:** Display Address Bit 16 Output **Character Mode:** Line Counter Bit 3 Output **Mixed Mode:** Attribute Blink and Line Counter Reset
AD_{15}	37	I/O	**Graphics Mode:** Display Address/Data Bits 13–15 **Character Mode:** Line Counter Bits 0–2 Output **Mixed Mode:** Display Address/Data Bits 13–15
AD_{14}	36		
AD_{13}	35		
AD_{12}	34	I/O	**Display Address/Data Bits 0–12**
AD_{11}	33		
AD_{10}	32		
AD_9	31		
AD_8	30		
AD_7	29		
AD_6	28		
AD_5	27		
AD_4	26		
AD_3	25		
AD_2	24		
AD_1	23		
AD_0	22		
LPEN	21	I	**Light Pen Detect Input**

FUNCTIONAL DESCRIPTION (Continued)

Microprocessor Bus Interface

Control of the GDC by the system microprocessor is achieved through an 8-bit bidirectional interface. The status register is readable at any time. Access to the FIFO buffer is coordinated through flags in the status register.

Command Processor

The contents of the FIFO are interpreted by the command processor. The command bytes are decoded, and the succeeding parameters are distributed to their proper destinations within the GDC. The bus interface has priority over the command processor when both access the FIFO simultaneously.

DMA Control

The DMA Control circuitry in the GDC coordinates data transfers when using an external DMA controller. The DMA Request and Acknowledge handshake lines interface with an 8257 or 8237 DMA controller or 8089 I/O processor, so that display data can be moved between the microprocessor memory and the display memory.

Parameter RAM

The 16-byte RAM stores parameters that are used repetitively during the display and drawing processes. In character mode, the RAM holds the partitioned display area parameters. In graphics mode, the RAM also holds the drawing pattern and graphics character.

Video Sync Generator

Based on the clock input, the sync logic generates the raster timing signals for almost any interlaced, non-interlaced, or "repeat field" interlaced video format. The generator is programmed during the idle period following a reset. In video sync slave mode, it coordinates timing between the GDC and another video source.

Memory Timing Generator

The memory timing circuitry provides two memory cycle types: a two-clock period refresh cycle and the read-modify-write (RMW) cycle which takes four clock periods. The memory control signals needed to drive the display memory devices are easily generated from the GDC's $\overline{RAS}$(ALE) and $\overline{DBIN}$ outputs.

Zoom and Pan Controller

Based on the programmable zoom display factor and the display area parameters in the parameter RAM, the zoom and pan controller determines when to advance to the next memory address for display refresh and when to go on to the next display area. A horizontal zoom is produced by slowing down the display refresh rate while maintaining the video sync rates. Vertical zoom is accomplished by repeatedly accessing each line a number of times equal to the horizontal repeat. Once the line count for a display area is exhausted, the controller accesses the starting address and line count of the next display area from the parameter RAM. The system microprocessor, by modifying a display area starting address, allows panning in any direction, independent of the other display areas.

Drawing Processor

The drawing processor contains the logic necessary to calculate the addresses and positions of the pixels of the various graphics figures. Given a starting point and the appropriate drawing parameters, the drawing-processor needs no further assistance to complete the figure drawing.

Display Memory Controller

The display memory controller's tasks are numerous. Its primary purpose is to multiplex the address and data information in and out of the display memory. It also contains the 16-bit logic units used to modify the display memory contents during RMW cycles, the character mode line counter, and the refresh counter for dynamic RAMs. The memory controller apportions the video field time between the various types of cycles.

Light Pen Debouncer

Only if two rising edges on the light pen input occur at the same point during successive video fields are the pulses accepted as a valid light pen detection. A status bit indicates to the system microprocessor that the light pen register contains a valid address.

System Operation

The GDC is designed to work with Intel microprocessors to implement high-performance computer graphics systems. System efficiency is maximized through partitioning and a pipelined architecture. At the lowest level, the GDC generates the basic video

raster timing, including sync and blanking signals. Partitioned areas on the screen and zooming are also accomplished at this level. At the next level, video display memory is modified during the figure drawing operations and data moves. Third, display memory address are calculated pixel by pixel as drawing progresses. Outside the GDC at the next level, preliminary calculations are done to prepare drawing parameters. At the fifth level, the picture must be represented as a list of graphics figures drawable by the GDC. Finally, this representation must be manipulated, stored and communicated. The GDC takes care of the high-speed and repetitive tasks required to implement graphics systems.

GENERAL OVERVIEW

In order to minimize system bus loading, the 82720 uses a private video memory for storage of the video image. Up to 512K bytes of video memory can be directly supported. For example, this is sufficient capacity to store a 2048 × 2048 pixel × 1 bit image. Images can be generated on the screen by:

—Drawing Commands
—Program-Controlled Transfers
—DMA Transfers from System Memory

The 82720 can be configured to support a wide variety of graphics applications. It can support:

—High Dot Rates
—Color Planes
—Horizontal Split Screen
—Character-oriented Displays
—Multiplexed Graphic and Character Display

GRAPHIC DISPLAY CONFIGURATIONS

The 82720 provides the flexibility to handle a wide variety of graphic applications. This flexibility results from having its own private video memory for storage of the graphics image. The organization of this memory determines the performance, the number of bits/pixel and the size of the display. Several different video memory organizations are examined in the following paragraphs.

In the simplest 82720 system, the memory can store up to a 2048 × 2048 × 1 bit image. It can display a 1024 × 1024 × 1 bit section of the image at a maximum dot rate of 44 MHz, or 88 MHz in wide mode. In this configuration, only 1 bit/pixel is used.

By partitioning the memory into multiple banks, color, gray scale and higher bandwidth displays can be supported. By adding various amounts of external logic, many cost/performance tradeoffs for both display and drawing are realizable.

The video memory can be partitioned into 4 banks, each 1024 x 1024 bits. By selecting all 4 memory banks during display, 4 bits/pixel can be provided by a single 82720. Each bank of video memory contributes 1 bit to each pixel. This configuration can support color monitors, again with a maximum dot shift rate of 44 or 88 MHz.

Higher performance may be achieved by using multiple 82720s. Multiple 82720s can be used to support mutliple display windows, increased drawing speed, or increased bits per pixel. For display windows, each 82720 controls one window of the display. For increased drawing speed, multiple 82720s are operated in parallel. For increased bits/pixel, each 82720 contributes a portion of the number of bits necessary for a pixel.

CHARACTER DISPLAY CONFIGURATION

Although the 82720 is intended primarily for raster-scan graphics, it can be used as a character display controller. The 82720 can support up to 8K by 13 bits of private video memory in this configuration (1 character = 13 bits). This is sufficient memory to store 4 screens of data containing 25 rows by 80 characters. The 82720 can display up to 256 characters per row. Smooth vertical scrolling of each of 4 independent display partitions is also supported.

MIXED DISPLAY CONFIGURATION

The GDC can support a mixed display system for both graphic and character information. This capability allows the display screen to be partitioned between graphic and character data. It is possible to switch between one graphic display window and one character display window with raster line resolution. A maximum of 256K bytes of video memory is supported in this mode: half is for graphic data, half is for character data. In graphic mode, a one megapixel image can be stored and displayed. In character mode, 64K, 16-bit characters can be stored.

DETAILED OPERATIONAL DESCRIPTION

The GDC can be used in one of three basic modes —Graphics Mode, Character Mode and Mixed Mode. This section of the data sheet describes the following for each mode:

1. Memory organization
2. Display timing
3. Special Display functions
4. Drawing and writing

Graphics Mode Memory Organization

The Display Memory is organized into 16-bit words (32-bit words in wide mode). Since the display memory can be larger than the CRT display itself, two width parameters must be specified: display memory width and display width. The Display width (in words) is selected by a parameter of the Reset command. The Display memory width (in words) is selected by a parameter of the Pitch command. The height of the Display memory can be larger than the display itself. The height of the Display is selected by a parameter of the Reset command. The GDC can directly address up to 4Mbits (0.5Mbytes) of display RAM in graphics mode.

Graphics Mode Display Timing

All raster blanking and display timings of the GDC are a function of the input clock frequency. Sixteen or 32 bits of data are read from the RAM and loaded into a shift register in each two clock period display cycle. The Address and Data busses of the GDC are multiplexed. In the first part of the cycle, the address of the word to be read is latched into an external demultiplexer. In the second part of the cycle the data is read from the RAM and loaded into the shift register. Since all 16 (32) bits of data are to be displayed, the dot clock is 8× (16×) the GDC clock or 16× (32×) the Read cycle rate.

Parameters of the Reset or Sync command determine the horizontal and vertical front porch, sync pulse, and back porch timings. Horizontal parameters are specified as multiples of the display cycle time, and vertical parameters as a multiple of the line time.

Another Reset command parameter selects interlaced or non-interlaced mode. A bit in the parameter RAM can define Wide Display Mode. In this mode, while data is being sent to the screen, the display address counter is incremented by two rather than one. This allows the display memory to be configured to deliver 32 bits from each display read cycle.

The V Sync command specifies whether the V Sync Pin is an input or an output. If the V Sync Pin is an output, the GDC generates the raster timing for the display and other CRT controllers can be synchronized to it. If the V Sync pin is an input, the GDC can be synchronized to any external vertical Sync signal.

Graphics Mode Special Display Functions:

WINDOWING

The GDC's Graphics Mode Display can be divided into two windows on the screen, upper and lower. The windows are defined by parameters written into the GDC's parameter RAM. Each window is specified by a starting address and a window length in lines. If the second window is not used, the first window parameters should be specified to be the same as the active display length.

ZOOMING

A parameter of the GDC's zoom command allows zooming by effectively increasing the size of the dots on the screen. This is accomplished vertically by repeating the same display line. The number of times it is repeated is determined by the display zoom factor parameter. Horizontally, zoom is accomplished by extending each display word cycle and displaying fewer words per line, according to the zoom factor. It is the responsibility of the microprocessor controlling the GDC to provide the shift register clock circuitry with the zoom factor required to slow down the shift registers to the appropriate speed. The frequency of the 2XWCLK should not be changed. The zoom factor must be set to a known state upon initialization.

PANNING

Panning is accomplished by changing the starting address of the display window. In this way, panning is possible in any direction, vertically on a line by line basis and horizontally on a word by word basis.

Graphics Mode Drawing and Writing

The GDC can draw solid or patterned lines, arcs, circles, rectangles, slanted rectangles, characters, slanted characters, filled rectangles. Direct access to the bit map is also provided via the DMA Commands and the Read or Write data commands.

MEMORY MODIFICATION

All drawing and writing functions take place at the location in the display RAM specified by the cursor. The cursor is not displayed in Graphics Mode. The cursor location is modified by the execution of drawing, reading or writing commands. The cursor will move to the bit following the last bit accessed.

Each bit is drawn by executing a Read-Modify-Write cycle on the display RAM. These R/M/W cycles normally require four 2XWCLK cycles to execute. If the display zoom factor is greater than two, each R/M/W cycle will be extended to the width of a display cycle. Write Data (WDAT), Read Data (RDAT), DMA write (DMAW) and DMA read (DMAR) commands can be used to examine or modify one to 16 bits in each word during each R/M/W cycle. All other graphics drawing commands modify one bit per R/M/W cycle.

An internal 16-bit Mask register determines which bit(s) in the accessed word are to be modified. A one in the Mask register allows the corresponding bit in the display RAM to be modified by the R/M/W cycle. A zero in the Mask register prevents the GDC from modifying the corresponding bit in the display RAM.

The mask must be set by the Mask Command prior to issuing the WDAT or DMAW command. The Mask register is automatically set by the CURS command and manipulated by the graphics commands.

The display RAM bits can be modified in one of four ways. They can be set to 1, reset to 0, complemented or replaced by a pattern.

When replace by a pattern mode is selected, lines, arcs and rectangles will be drawn using the 16-bit pattern in parameter RAM bytes 8 and 9.

In set, reset, or complement mode, parameter RAM bytes 8 and 9 act as another level of masking for line arc and rectangle drawing. As each 16-bit segment of the line or arc is drawn, it is checked against the pattern in the parameter RAM. If the pattern RAM bit is a one, the display RAM bit will be set, reset, or complemented per the proper modes. If the pattern RAM bit is a zero, the display RAM bit won't be modified.

When replace by pattern mode is selected, the graphics character and fill commands will cause the 8 x 8 pattern in parameter RAM bytes 8 to 15 to be written directly into the display RAM in the appropriate locations.

In set, reset, or complement mode, the 8 x 8 pattern in parameter RAM bytes 8 to 15 act as a mask pattern for graphics character or fill commands. If the appropriate parameter RAM bit is set, the display RAM bit will be modified. If the parameter RAM bit is zero, the display RAM bit will not be modified. These modes are selected by issuing a WDAT command without parameters before issuing graphics commands. The pattern in the parameter RAM has no effect on WDAT, RDAT, DMAW, or DMAR operations.

READING AND DRAWING COMMANDS

After the modification mode has been set and the parameter RAM has been loaded, the final drawing parameters are loaded via the figure specify (FIGS) command. The first parameter specifies the direction in which drawing will occur and the figure type to be drawn. This parameter is followed by one to five more parameters depending on the type of character to be drawn.

The direction parameter specifies one of eight octants in which the drawing or reading will occur. The effect of drawing direction on the various figure types is shown in Figure 9.

RDAT, WDAT, DMAR, and DMAW Operations move through the Display memory as shown in the "DMA" Column.

The other parameters required to set up figure reading or drawing are shown in Figure 3.

DRAWING TYPE	DC	D	D2	D1	DM
INITIAL VALUE*	0	8	8	−1	−1
LINE	|Δl|	2|ΔD| − |Δl|	2(|ΔD| − |Δl|)	2|ΔD|	−
ARC**	rsin θ!	r−1	2(r−1)	−1	rsin θ!
RECTANGLE	3	A−1	B−1	−1	A−1
AREA FILL	B−1	A	A	−	−
GRAPHIC CHARACTER***	B−1	A	A	−	−
WRITE DATA	W−1	−	−	−	−
DMAW	D−1	C−1	−	−	−
DMAR	D−1	C−2	(C−2)/2†	−	−
READ DATA	W	−	−	−	−

*INITIAL VALUES FOR THE VARIOUS PARAMETERS ARE LOADED WHEN THE FIGS COMMAND BYTE IS PROCESSED.
**CIRCLES ARE DRAWN WITH 8 ARCS, EACH OF WHICH SPAN 45°, SO THAT SIN o = 1/√2 AND SIN θ = 0.
***GRAPHIC CHARACTERS ARE A SPECIAL CASE OF BIT-MAP AREA FILLING IN WHICH B AND A ≤8. IF A = 8 THERE IS NO NEED TO LOAD D AND D2.

WHERE:
−1 = ALL ONES VALUE.
ALL NUMBERS ARE SHOWN IN BASE 10 FOR CONVENIENCE. THE GDC ACCEPTS BASE 2 NUMBERS (2s COMPLEMENT NOTATION WHERE APPROPRIATE).
− = NO PARAMETER BYTES SENT TO GDC FOR THIS PARAMETER.
Δl = THE LARGER OF Δx OR Δy.
ΔD = THE SMALLER OF Δx OR Δy.
r = RADIUS OF CURVATURE, IN PIXELS.
o = ANGLE FROM MAJOR AXIS TO END OF THE ARC. o ≤45°.
θ = ANGLE FROM MAJOR AXIS TO START OF THE ARC. θ ≤45°.
! = ROUND UP TO THE NEXT HIGHER INTEGER.
! = ROUND DOWN TO THE NEXT LOWER INTEGER.
A = NUMBER OF PIXELS IN THE INITIALLY SPECIFIED DIRECTION.
B = NUMBER OF PIXELS IN THE DIRECTION AT RIGHT ANGLES TO THE INITIALLY SPECIFIED DIRECTION.
W = NUMBER OF WORDS TO BE ACCESSED.
C = NUMBER OF BYTES TO BE TRANSFERRED IN THE INITIALLY SPECIFIED DIRECTION. (TWO BYTES PER WORD IF WORD TRANSFER MODE IS SELECTED.)
D = NUMBER OF WORDS TO BE ACCESSED IN THE DIRECTION AT RIGHT ANGLES TO THE INITIALLY SPECIFIED DIRECTION.
DC = DRAWING COUNT PARAMETER WHICH IS ONE LESS THAN THE NUMBER OF RMW CYCLES TO BE EXECUTED.
DM = DOTS MASKED FROM DRAWING DURING ARC DRAWING.
† = NEEDED ONLY FOR WORD READS.

Figure 3. Drawing Parameter Details

After the parameters have been set, line, arc, circle, rectangle or slanted rectangle drawing operations are initiated by the Figure Draw (FIGD) command. Character, slanted character, area fill and slanted area fill drawing operations are initiated by the Graphics Character Draw (GCHRD) command. DMA transfers are initiated by the DMA Read or Write (DMAR or DMAW) commands. Data Read Operations are initiated by the Read Data (RDAT) Command. Data Write Operations are initiated by writing a parameter after the WDAT command.

The area fill operation steps and repeats the 8 × 8 graphics character pattern draw operation to fill a rectangular area. If the size of the rectangle is not an integral number of 8 x 8 pixels, the GDC will automatically truncate the pattern at the edges furthest from the starting point.

The Graphics Character Drawing capability can be modified by the Graphics Character Write Zoom Factor (GCHR) parameter of the zoom command. The zoom write factor may be set from 1 to 16 (by using from 0 to 15 in the parameter). Each dot will be repeated in memory horizontally and vertically (adjusted for drawing direction) the number of times specified by the zoom factor.

The WDAT command can be used to rapidly fill large areas in memory with the same value. The mask is set to all 1's, and the least significant bit of the WDAT parameter replaces all bits of each word written.

Character Mode Memory Organization

In character mode, the Display memory is organized into up to 8K characters of up to 13 bits each. Wide mode is also available for characters of up to 26 bits.

The display memory can be larger than the display itself. The display width (in characters) is a parameter of the reset command. The display memory width (in characters) is a parameter of the Pitch Command. The height of the display (in lines) is a parameter of the Reset Command. The display memory height is determined by dividing the number of display memory words by the pitch.

In character mode, the display works almost exactly as it does in graphics mode. The differences lie in the fact that data read from the display RAM is used to drive a character generator as well as attribute logic if desired. In Character mode, address bits 13–16 become line counter outputs used to select the proper line of the character generator, and the address 17 output becomes the cursor and line counter MSB output.

Character Mode Display Timing

In character mode, the display timing works as it does in graphics mode. In addition, the Address 17 output becomes cursor output. The characteristics of the cursor are defined by parameters of the cursor and Character Characteristics (CCHAR) command. One bit allows the cursor output to be enabled or disabled. The height of the cursor is programmable by selecting the top and bottom line between which the cursor will appear. The blink rate is also programmable. The parameter selects the number of frame times that the cursor will be inactive and active, resulting in a 50% duty cycle cursor blinking at 2× the period specified by the parameter.

The cursor output pin also provides the line counter bit 4 signal, which is valid 10 clocks after the trailing edge of HSYNC.

Character Mode Special Display Functions

WINDOWING

The GDC's Character Mode display can be partitioned into one to four windows on the screen. The windows are defined by parameters written into the GDC's Parameter RAM. Each window is specified by a starting address and a window length in lines.

If windowing is not required, the first window length should be specified to be the same as the active display length.

ZOOMING AND PANNING

In character mode, zooming and pan handling commands function the same way as in Graphics Mode.

Character Mode Drawing and Writing

The GDC can read or write characters of up to 13 bits into or out of the Display RAM.

All reading and writing functions take place at the display RAM location specified by the cursor. The cursor location can be read by issuing the CURD command. The cursor can be moved anywhere within the display memory by the CURS command. The cursor location is also modified by the execution of character read or write commands.

Each character is written or read via a Read/Modify/Write cycle. The mask register contents determine which bit(s) in the character are modified. The mask register can be used to change character codes without modifying attribute bits or vice-versa. The Replace with pattern, Set, Reset and Complement

modes work exactly as they do in graphics mode, with the exception that the parameter RAM Pattern is not used. The pattern used is a parameter of the WDAT command.

The Figure Specify (FIGS) command must be set to Character Display mode, as well as specify the direction the cursor will be moved by read or write data commands.

In character mode, the FIGD and GCHRD commands are not used.

Mixed Mode Memory Organization

In mixed mode, the display memory is organized into two banks of up to 64K words of 16 bits each (32 bits in wide mode).

The display height and width are programmable by the same Reset or Sync command parameters as in the graphics and character modes. The display memory width (in words) is a parameter of the Pitch Command and the height of the display memory is determined by dividing the number of display memory words by the pitch.

An image mode signal is used to switch the external circuitry between graphics and character modes in two display windows.

In a graphics window, the GDC works as it does in pure graphics mode, but on a smaller total memory space (64K words vs 512K words).

In a character window, the GDC works as it does in pure character mode, but the line counter must be implemented externally. The counter is clocked by the horizontal sync pulse and reset by a signal supplied by the GDC.

In mixed mode, the GDC provides both a cursor and an attribute blink timing signal.

Mixed Mode Display Timing

In mixed mode, each word in a graphic area is accessed twice in succession. The AW parameter of the Reset or Sync command should be set to twice its normal value, and the video shift register load signal must be suppressed during the extra access cycle.

In addition, A16 becomes a Multiplexed Attribute and Clear Line Counter signal and A17 becomes a multiplexed cursor and image mode signal. A16 provides an active high line counter reset signal which is valid 10 clocks after the trailing edge of HSYNC. During the active display line time, A16 provides blink timing for external attribute circuitry. This signal blinks at 1/2 the blink rate of the cursor with a 75% on, 25% off duty cycle. A17 provides a signal which selects between graphics or character display, which is also valid 10 clocks after the trailing edge of HSYNC. During the active display time, A17 provides the cursor signal. The cursor timing and characteristics are defined in exactly the same way as in pure character mode.

Mixed Mode Special Display Functions

WINDOWING

The GDC supports two display windows in mixed mode. They can independently be programmed into either graphics or character mode determined by the state of two bits in the parameter RAM. The window location in display memory and size are also determined by parameters in the parameter RAM.

ZOOMING AND PANNING

In mixed mode, zooming and panning commands function the same as in graphics and character mode.

Mixed Mode Drawing and Writing

In mixed mode, the GDC can write or draw in exactly the same ways as in both graphics and character modes. In addition, the FIGS command has a parameter GD (Graphics Drawing Flag) which sets the image mode signal to select the proper RAM bank.

DEVICE PROGRAMMING

The GDC occupies two addresses on the system microprocessor bus through which the GDC's status register and FIFO are accessed. Commands and parameters are written into the GDC FIFO and are differentiated by address bit A0. The status register or the FIFO can be read as selected by the address line.

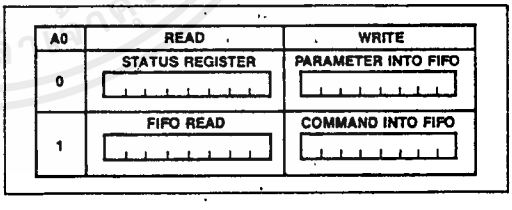


Figure 4. GDC Microprocessor Bus Interface Registers

Commands to the GDC take the form of a command byte followed by a series of parameter bytes as needed for specifying the details of the command. The command processor decodes the commands, unpacks the parameters, loads them into the appropriate registers within the GDC and initiates the required operations.

The commands available in the GDC can be organized into five categories as described in figure 5.

```
VIDEO CONTROL COMMANDS
   1. RESET:   RESETS THE GDC TO ITS IDLE STATE.
   2. SYNC:    SPECIFIES THE VIDEO DISPLAY FORMAT.
   3. VSYNC:   SELECTS MASTER OR SLAVE VIDEO
               SYNCHRONIZATION MODE
   4. CCHAR:   SPECIFIES THE CURSOR AND CHARACTER ROW
               HEIGHTS.
DISPLAY CONTROL COMMANDS
   1. START:   ENDS IDLE MODE AND UNBLANKS THE DISPLAY.
   2. BCTRL:   CONTROLS THE BLANKING AND UNBLANKING OF
               THE DISPLAY.
   3. ZOOM:    SPECIFIES ZOOM FACTORS FOR THE DISPLAY AND
               GRAPHICS CHARACTERS WRITING.
   4. CURS:    SETS THE POSITION OF THE CURSOR IN DISPLAY
               MEMORY.
   5. PRAM:    DEFINES STARTING ADDRESSES AND LENGTHS OF
               THE DISPLAY AREAS AND SPECIFIES THE EIGHT
               BYTES FOR THE GRAPHICS CHARACTER.
   6. PITCH:   SPECIFIES THE WIDTH OF THE X DIMENSION OF
               DISPLAY MEMORY.
DRAWING CONTROL COMMANDS
   1. WDAT:    WRITES DATA WORDS OR BYTES INTO DISPLAY
               MEMORY.
   2. MASK:    SETS THE MASK REGISTER CONTENTS.
   3. FIGS:    SPECIFIES THE PARAMETERS FOR THE DRAWING
               PROCESSOR.
   4. FIGD:    DRAWS THE FIGURE AS SPECIFIED ABOVE.
   5. GCHRD:   DRAWS THE GRAPHICS CHARACTER INTO DISPLAY

MEMORY DATA READ COMMANDS
   1. RDAT:    READS DATA WORDS OR BYTES FROM DISPLAY
               MEMORY.
   2. CURD:    READS THE CURSOR POSITION.
   3. LPRD:    READS THE LIGHT PEN ADDRESS.
DMA CONTROL COMMANDS
   1. DMAR:    REQUESTS A DMA READ TRANSFER.
   2. DMAW:    REQUESTS A DMA WRITE TRANSFER.
```

Figure 5. GDC Command Summary

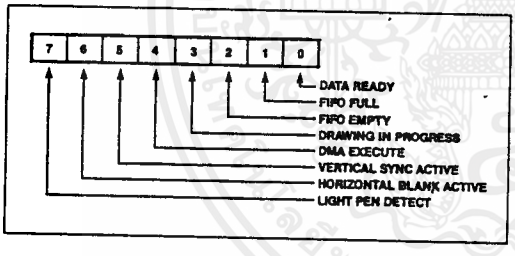


Figure 6. Status Register (SR)

Status Register Flags

SR-7: Light Pen Detect: When this bit is set to 1, the light pen address (LAD) register contains a deglitched value that the system microprocessor may read. This flag is reset after the 3-byte LAD is moved into the FIFO in response to the light pen read command.

SR-6: Horizontal Blanking Active: A 1 value for this flag signifies that horizontal retrace blanking is currently underway.

SR-5: Vertical Sync: Vertical retrace sync occurs while this flag is a 1. The vertical sync flag coordinates display format modifying commands to the blanked interval surrounding vertical sync. This eliminates display disturbances.

SR-4: DMA Execute: This bit is a 1 during DMA data transfers.

SR-3: Drawing In Progress: While the GDC is drawing a graphics figure, this status bit is a 1.

SR-2: FIFO Empty: This bit and the FIFO Full flag coordinate system microprocessor accesses with the GDC FIFO. When it is 1, the Empty flag ensures that all the commands and parameters previously sent to the GDC have been processed.

SR-1: FIFO Full: A 1 at this flag indicates a full FIFO in the GDC. A 0 ensures that there is room for at least one byte. This flag needs to be checked before each write into the GDC.

SR-0: Data Ready: When this flag is a 1, it indicates that a byte is available to be read by the system microprocessor. This bit must be tested before each read operation. It drops to a 0 while the data is transferred from the FIFO into the microprocessor interface data register.

FIFO Operation & Command Protocol

The first-in, first-out buffer (FIFO) in the GDC handles the command dialogue with the system microprocessor. This flow of information uses a half-duplex technique, in which the single 16-location FIFO is used for both directions of data movement, one direction at a time. The FIFO's direction is controlled by the system microprocessor through the GDC's command set. The microprocessor coordinates these transfers by checking the appropriate status register bits.

The command protocol used by the GDC requires the differentiation of the first byte of a command sequence from the succeeding bytes. This first byte contains the operation code and the remaining bytes carry parameters. Writing into the GDC causes the FIFO to store a flag value alongside the data byte to signify whether the byte was written into the command or the parameter address. The command processor in the GDC tests this bit as it interprets the entries in the FIFO.

The receipt of a command byte by the command processor marks the end of any previous operation. The number of parameter bytes supplied with a command is cut short by the receipt of the next command byte. A read operation from the GDC to the microprocessor can be terminated at any time by the next command.

The FIFO changes direction under the control of the system microprocessor. Commands written into the GDC always put the FIFO into write mode if it wasn't in it already. If it was in read mode, any read data in the FIFO at the time of the turnaround is lost. Commands which require a GDC response, such as RDAT, CURD and LPRD, put the FIFO into read mode after the command is interpreted by the GDC's command processor. Any commands and parameters behind the read-evoking command are discarded when the FIFO direction is reversed.

Read-Modify-Write Cycle

Data transfers between the GDC and the display memory are accomplished using a read-modify-write (RMW) memory cycle. The four clock period timing of the RMW cycle is used to: 1) output the address, 2) read data from the memory, 3) modify the data, and 4) write the modified data back into the initially selected memory address. This type of memory cycle is used for all interactions with display memory including DMA transfers, except for the two clock period display and RAM refresh cycles.

The operations performed during the modify portion of the RMW cycle merit additional explanation. The circuitry in the GDC uses three main elements: the Pattern register, the Mask register, and the 16-bit Logic unit. The Pattern register holds the data pattern to be moved into memory. It is loaded by the WDAT command or, during drawing, from the parameter RAM. The Mask register contents determine which bits of the read data will be modified. Based on the contents of these registers, the Logic unit performs the selected operations of REPLACE, COMPLEMENT, SET, or CLEAR on the data read from display memory.

The Pattern register contents are ANDed with the Mask register contents to enable the actual modification of the memory read data, on a bit-by-bit basis. For graphics drawing, one bit at a time from the Pattern register is combined with the Mask. When ANDed with the bit set to a 1 in the Mask register, the proper single pixel is modified by the Logic Unit. For the next pixel in the figure, the next bit in the Pattern register is selected and the Mask register bit is moved to identify the pixel's location within the word. The Execution word address pointer register, EAD, is also adjusted as required to address the word containing the next pixel.

In character mode, all of the bits in the Pattern register are used in parallel to form the respective bits of the modify data word. Since the bits of the character code word are used in parallel, unlike the one-bit-at-a-time graphics drawing process, this facility allows any or all of the bits in a memory word to be modified in one RMW memory cycle. The Mask register must be loaded with 1s in the positions where modification is to be permitted.

The Mask register can be loaded in either of two ways. In graphics mode, the CURS command contains a four-bit dAD field to specify the dot address. The command processor converts this parameter into the one-of-16 format used in the Mask register for figure drawing. A full 16 bits can be loaded into the Mask register using the MASK command. In addition to the character mode use mentioned above, the 16-bit MASK load is convenient in graphics mode when all of the pixels of a word are to be set to the same value.

The Logic unit combines the data read from display memory, the Pattern register, and the Mask register to generate the data to be written back into display memory. Any one of four operations can be selected: REPLACE, COMPLEMENT, CLEAR or SET. In each case, if the respective Mask bit is 0, that particular bit of the read data is returned to memory unmodified. If the Mask bit is 1, the modification is enabled. With the REPLACE operation, the modify data simply takes the place of the read data for modification enabled bits. For the other three operations, a 0 in the modify data allows the read data bit to be returned to memory. A 1 value causes the specified operation to be performed in the bit positions with set Mask bits.

Figure Drawing

The GDC draws graphics figures at the rate of one pixel per read-modify-write (RMW) display memory cycle. These cycles take four clock periods to complete. At a clock frequency of 5 MHz, this is equal to 800 ns. During the RMW cycle the GDC simultaneously calculates the address and position of the next pixel to be drawn.

The graphics figure drawing process depends on the display memory addressing structure. Groups of 16 horizontally adjacent pixels form the 16-bit words

which are handled by the GDC. Display memory is organized as a linearly addressed space of these words. Addressing of individual pixels is handled by the GDC's internal RMW logic.

During the drawing process, the GDC finds the next pixel of the figure which is one of the eight nearest neighbors of the last pixel drawn. The GDC assigns each of these eight directions a number from 0 to 7, starting with straight down and proceeding counterclockwise.

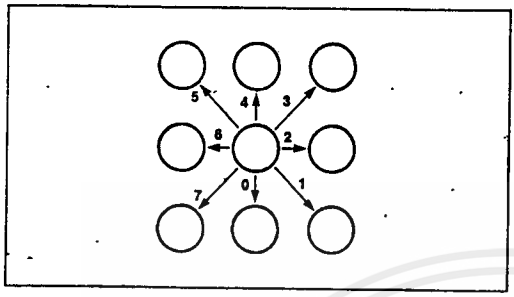


Figure 7. Drawing Directions

Figure drawing requires the proper manipulation of the address and the pixel bit position according to the drawing direction to determine the next pixel of the figure. To move to the word above or below the current one, it is necessary to subtract or add the number of words per line in display memory. This parameter is called the pitch. To move to the word to either side, the Execute word address cursor, EAD, must be incremented or decremented as the dot address pointer bit reaches the LSB or the MSB of the Mask register. To move to a pixel within the same word, it is necessary to rotate the dot address pointer register to the right or left.

Figure 8 summarizes these operations for each direction.

Whole word drawing is useful for filling areas in memory with a single value. By setting the Mask register to all 1s with the MASK command, both the LSB and MSB of the dAD will always be 1, so that the EAD value will be incremented or decremented for each cycle regardless of direction. One RMW cycle will be able to affect all 16 bits of the word for any drawing type. One bit in the Pattern register is used per RMW cycle to write all the bits of the word to the same value. The next Pattern bit is used for the word, etc.

DIR	ADDRESS OPERATION(S)
0	EAD = EAD + P
1	EAD = EAD + P If dAD.MSB = 1 then EAD = EAD + 1 dAD = LR(dAD)
2	If dAD.MSB = 1 then EAD = EAD + 1 dAD = LR(dAD)
3	EAD = EAD −P If dAD.MSB = 1 then EAD = EAD + 1 dAD =LR(dAD)
4	EAD = EAD − P
5	EAD = EAD − P If dAD.LSB = 1 then EAD = EAD − 1 dAD =RR(dAD)
6	If dAD.LSB = 1 then EAD = EAD − 1 dAD = RR(dAD)
7	EAD = EAD + P If dAD.LSB = 1 then EAD = EAD − 1 dAD = RR(dAD)

WHERE
P = PITCH, LR = LEFT ROTATE, RR = RIGHT ROTATE
CAD = CURSOR ADDRESS
dAD = DOT ADDRESS
LSB = LEAST SIGNIFICANT BIT
MSB = MOST SIGNIFICANT BIT

Figure 8. Address Calculation Details

For the various figures, the effect of the initial direction upon the resulting drawing is shown in figure 9.

Note that during line drawing, the angle of the line may be anywhere within the shaded octant defined by the DIR value. Arc drawing starts in the direction initially specified by the DIR value and veers into an arc as drawing proceeds. An arc may be up to 45 degrees in length. DMA transfers are done on word boundaries only, and follow the arrows indicated in the table to find successive word addresses. The slanted paths for DMA transfers indicate the GDC changing both the X and Y components of the word address when moving to the next word. It does not follow a 45 degree diagonal path by pixels.

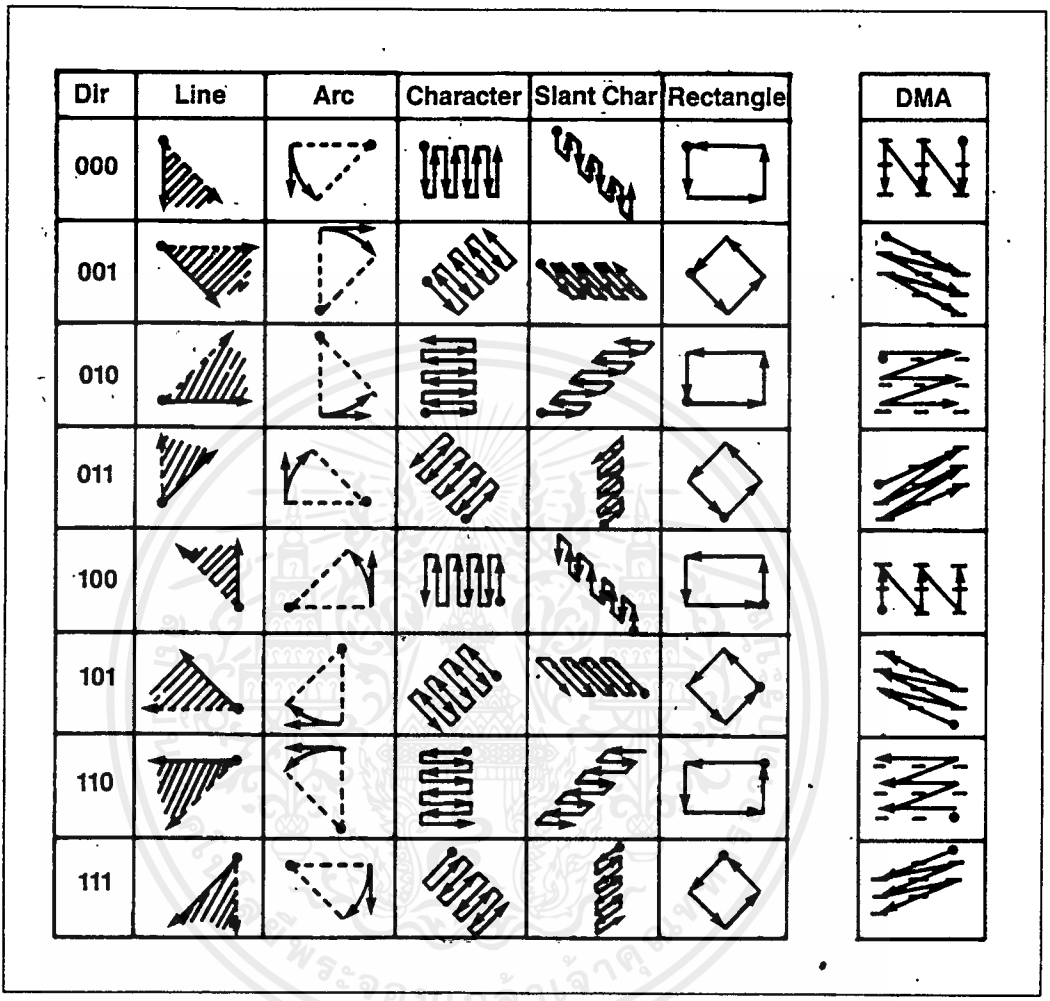


Figure 9. Effect of the Direction Parameter

Drawing Parameters

In preparation for graphics figure drawing, the GDC's Drawing Processor needs the figure type, direction and drawing parameters, the starting pixel address, and the pattern from the microprocessor. Once these are in place within the GDC, the Figure Draw command, FIGD, initiates the drawing operation. From that point on, the system microprocessor is not involved in the drawing process. The GDC Drawing Processor coordinates the RMW circuitry and address registers to draw the specified figure pixel by pixel.

The algorithms used by the processor for figure drawing are designed to optimize its drawing speed. To this end, the specific details about the figure to be drawn are reduced by the microprocessor to a form conducive to high-speed address calculations within the GDC. In this way the repetitive, pixel-by-pixel calculations can be done quickly, thereby minimizing the overall figure drawing time. Figure 3 summarizes the parameters.

Graphics Character Drawing

Graphics characters can be drawn into display memory pixel-by-pixel. The up to 8-by-8 character is loaded into the GDC's parameter RAM by the system microprocessor. Consequently, there are no limitations on the character set used. By varying the drawing parameters and drawing direction, numerous drawing options are available. In area fill applications, a character can be written into display memory as many times as desired without reloading the parameter RAM.

Once the parameter RAM has been loaded with up to eight graphics character bytes by the appropriate PRAM command, the GCHRD command can be used to draw the bytes into display memory starting at the cursor. The zoom magnification factor for writing, set by the zoom command, controls the size of the character written into the display memory in integer multiples of 1 through 16. The bit values in the PRAM are repeated horizontally and vertically the number of times specified by the zoom factor.

The movement of these PRAM bytes to the display memory is controlled by the parameters of the FIGS command. Based on the specified height and width of the area to be drawn, the parameter RAM is scanned to fill the required area.

For an 8-by-8 graphics character, the first pixel drawn uses the LSB of RA-15, the second pixel uses bit 1 of RA-15, and so on, until the MSB of RA-15 is reached. The GDC jumps to the corresponding bit in RA-14 to continue the drawing. The progression then advances toward the LSB of RA-14. This snaking sequence is continued for the other 6 PRAM bytes. This progression matches the sequence of display memory addresses calculated by the drawing processor as shown in figure 9. If the area is narrower than 8 pixels wide, the snaking will advance to the next PRAM byte before the MSB is reached. If the area is less than 8 lines high, fewer bytes in the parameter RAM will be scanned. If the area is larger than 8 by 8, the GDC will repeat the contents of the parameter RAM in two dimensions.

Parameter RAM Contents

The parameters stored in the parameter RAM, PRAM, are available for the GDC to refer to repeatedly during figure drawing and raster-scanning. In each mode of operation the values in the PRAM are interpreted by the GDC in a predetermined fashion. The host microprocessor must load the appropriate parameters into the proper PRAM locations. PRAM loading command allows the host to write into any location of the PRAM and transfer as many bytes as desired. In this way any stored parameter byte or bytes may be changed without influencing the other bytes.

The PRAM stores two types of information. For specifying the details of the display area partitions, blocks of four bytes are used. The four parameters stored in each block include the starting address in display memory of each display area, and its length. In addition, there are two mode bits for each area which specify whether the area is a bit-mapped graphics area or a coded character area, and whether a normal or wide display cycle is to be used for that area.

The other use for the PRAM contents is to supply the pattern for figure drawing when in a bit-mapped graphics area or mode. In these situations, PRAM bytes 8 through 16 are reserved for this patterning information. For line, arc, and rectangle drawing (linear figures) locations 8 and 9 are loaded into the Pattern register to allow the GDC to draw dotted, dashed, etc. lines. For area filling and graphics bit-mapped character drawing locations 8 through 15 are referenced for the pattern or character to be drawn.

Details of the bit assignments are shown on the following pages for the various modes of operation.

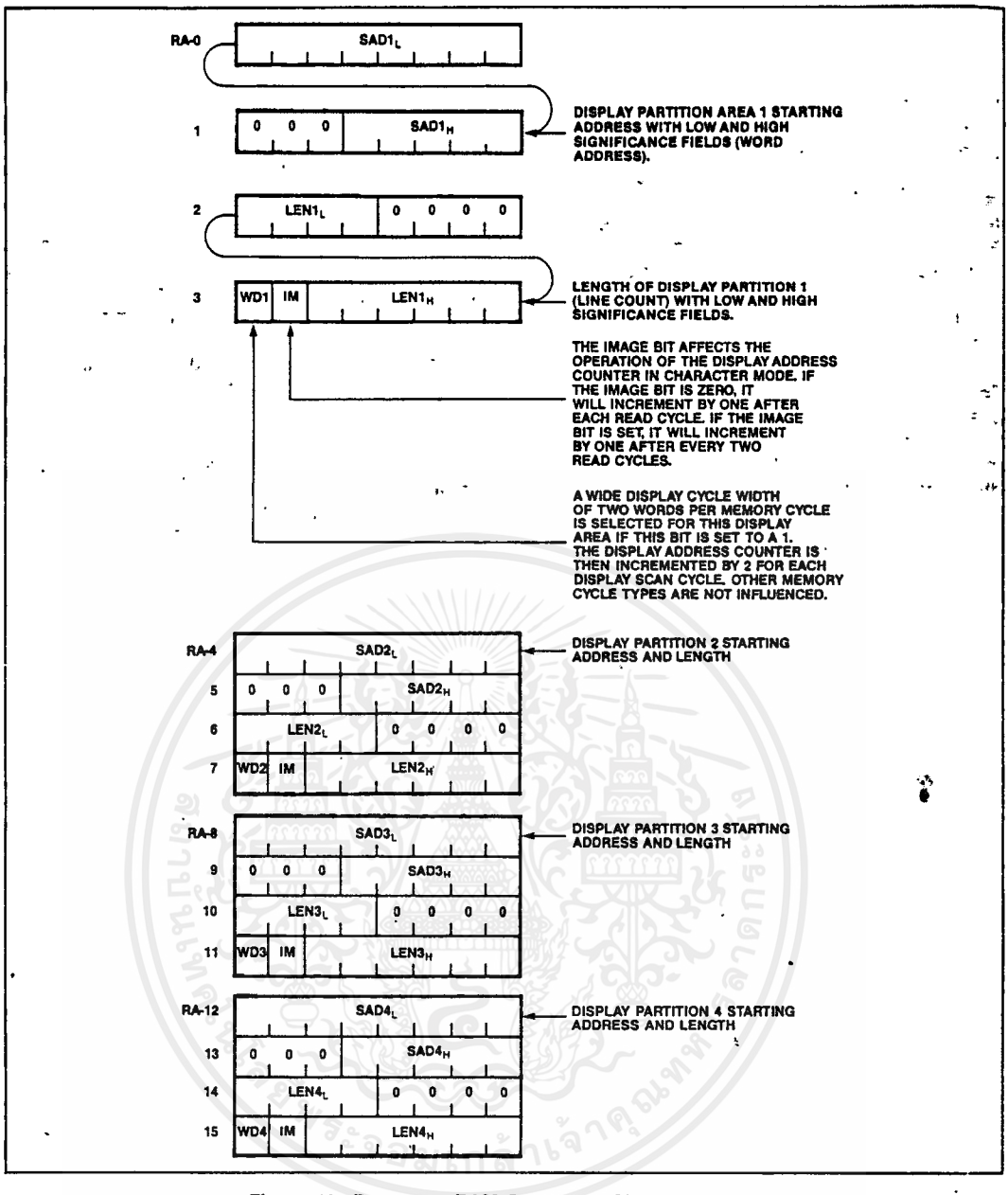


Figure 10. Parameter RAM Contents—Character Mode

210655-002

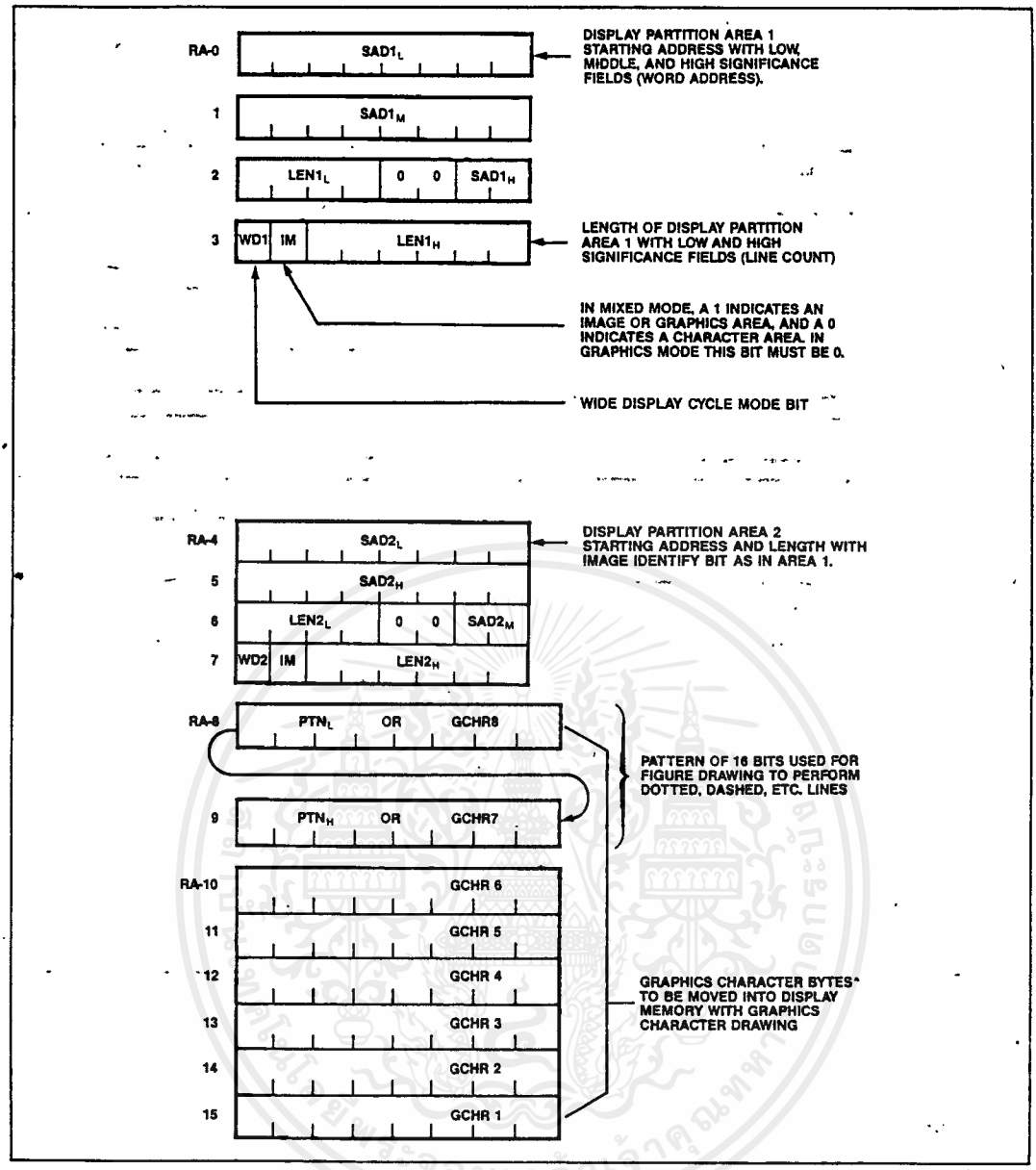


Figure 11. Parameter RAM Contents—Graphics and Mixed Graphics and Character Modes

210655-002

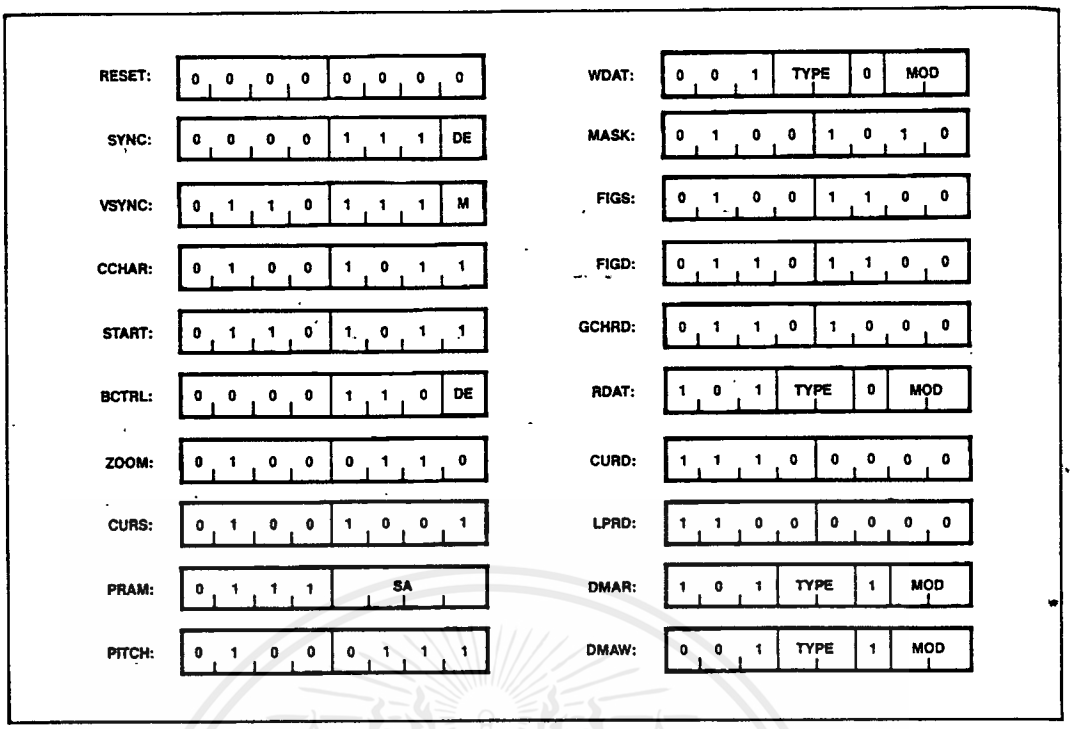


Figure 12. Command Bytes Summary

VIDEO CONTROL COMMANDS

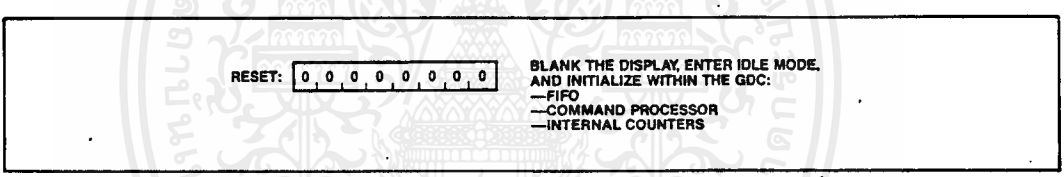


Figure 13. Reset Command

RESET COMMAND

This command can be executed at any time and does not modify any of the parameters already loaded into the GDC.

If followed by parameter bytes, this command also sets the sync generator parameters as described below. Idle mode is exited with the START command.

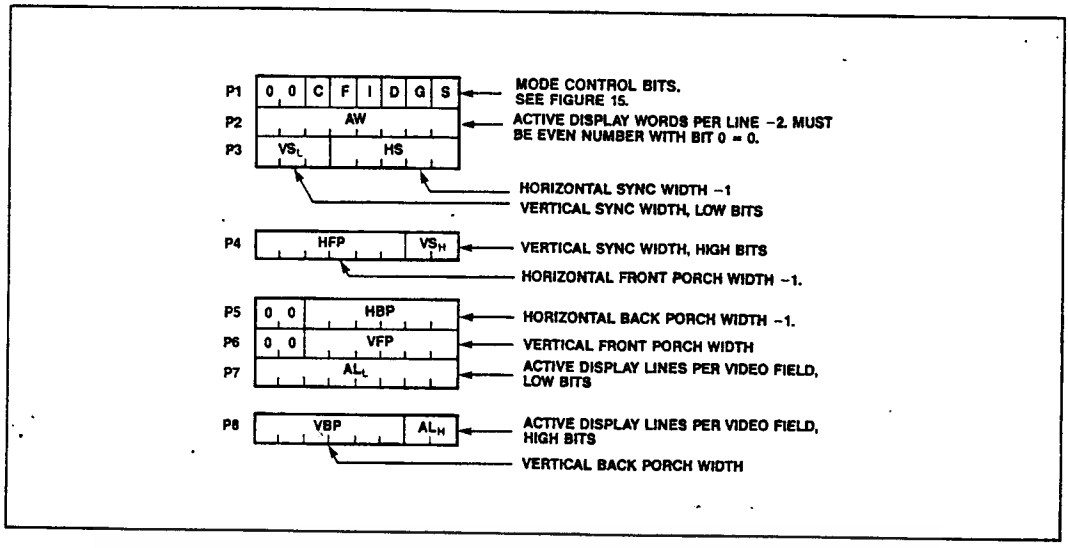


Figure 14. Optional Reset Parameters

In graphics mode, a word is a group of 16 pixels. In character mode, a word is one character code and its attributes, if any.

The number of active words per line must be an even number from 2 to 256.

An all-zero parameter value selects a count equal to 2^n where n = number of bits in the parameter field for vertical parameters.

All horizontal widths are counted in display words. All vertical intervals are counted in lines.

Sync Parameter Constraints

HORIZONTAL FRONT PORCH CONSTRAINTS

1. In general:
   HFP ≥2 words
2. If DMA is used, or the display zoom factor is greater than one in interlaced display mode:
   HFP ≥3 words
3. If the GDC is used in slave mode:
   HFP ≥4 words
4. If the light pen input is used:
   HFP ≥6 words

HORIZONTAL Sync CONSTRAINTS

1. If dynamic RAM refresh is used:
   HS ≥2 words
2. If interlaced display mode is used:
   HS ≥5 words

HORIZONTAL BACK PORCH CONSTRAINTS

1. In general:
   HBP ≥3 words
2. If interlaced display mode is used, or the IMAGE or WIDE mode bits change within one video field:
   HBP ≥ 5 words

MODE CONTROL BITS (FIGURE 15)

Repeat Field Framing:	2 Field Sequence with ½ line offset between otherwise identical fields.
Interlaced Framing:	2 Field Sequence with ½ line offset. Each field displays alternate lines.
Noninterlaced Framing:	1 field brings all of the information to the screen.

Total scanned lines in interlace mode is odd. The sum of VFP + VS + VBP + AL should equal one less than the desired odd number of lines.

Dynamic RAM refresh is important when high display zoom factors or DMA are used in such a way that not all of the rows in the RAMs are regularly accessed during display raster generation and for otherwise inactive display memory.

Access to display memory can be limited to retrace blanking intervals only, so that no disruptions of the image are seen on the screen.

C G	DISPLAY MODE
0 0	MIXED GRAPHICS & CHARACTER
0 1	GRAPHICS MODE
1 0	CHARACTER MODE
1 1	INVALID

I S	VIDEO FRAMING
0 0	NONINTERLACED
0 1	INVALID
1 0	INTERLACED REPEAT FIELD FOR CHARACTER DISPLAYS
1 1	INTERLACED

D	DYNAMIC RAM REFRESH CYCLES ENABLE
0	NO REFRESH—STATIC RAM
1	REFRESH—DYNAMIC RAM

F	DRAWING TIME WINDOW
0	DRAWING DURING ACTIVE DISPLAY TIME AND RETRACE BLANKING
1	DRAWING ONLY DURING RETRACE BLANKING

Figure 15. Mode Control Bits

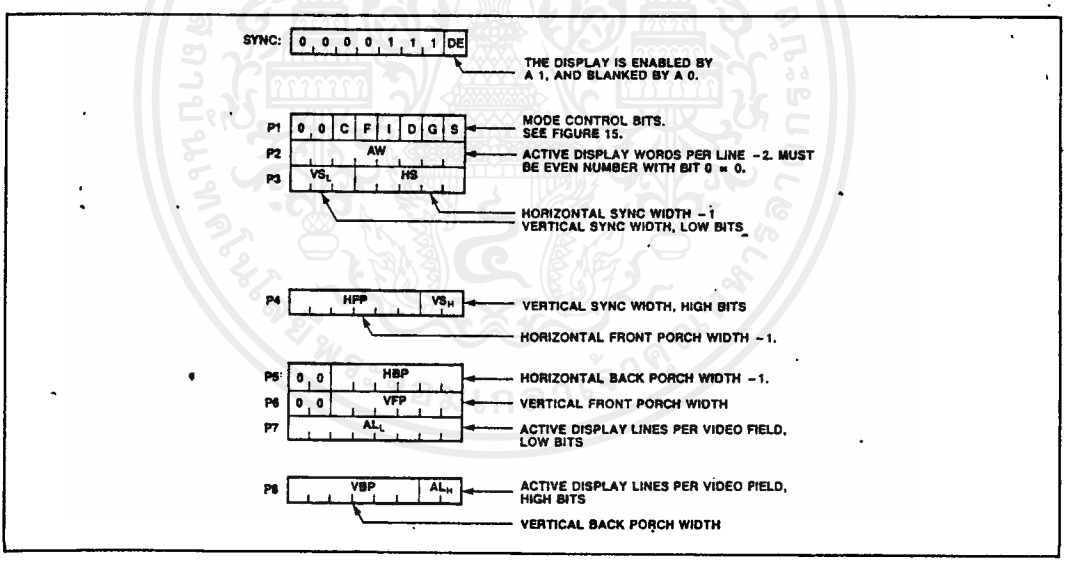


Figure 16. Sync Command

SYNC Format Specify Command

This command loads parameters into the sync generator. The various parameter fields and bits are identical to those at the RESET command. The GDC is not reset nor does it enter idle mode.

Vertical Sync Mode Command

When using two or more GDCs to contribute to one image, one GDC is defined as the master sync generator, and the others operate as its slaves. The VSYNC pins of all GDCs are connected together.

Slave Mode Operation

A few considerations should be observed when synchronizing two or more GDCs to generate overlayed video via the VSYNC INPUT/OUTPUT pin. As mentioned above, the Horizontal Front Porch (HFP) must be 4 or more display cycles wide. This is equivalent to eight or more clock cycles. This gives the slave GDCs time to initialize their internal video sync generators to the proper point in the video field to match the incoming vertical sync pulse (VSYNC). This resetting of the generator occurs just after the end of the incoming VSYNC pulse, during the HFP interval. Enough time during HFP is required to allow the slave GDC to complete the operation before the start of the HSYNC interval.

Once the GDCs are initialized and set up as Master and Slaves, they must be given time to synchronize. It is a good idea to watch the VSYNC status bit of the Master GDC and wait until after one or more VSYNC pulses have been generated before the display process is started. The START command will begin the active display of data and will end the video synchronization process, so be sure there has been at least one VSYNC pulse generated for the Slaves to synchronize to.

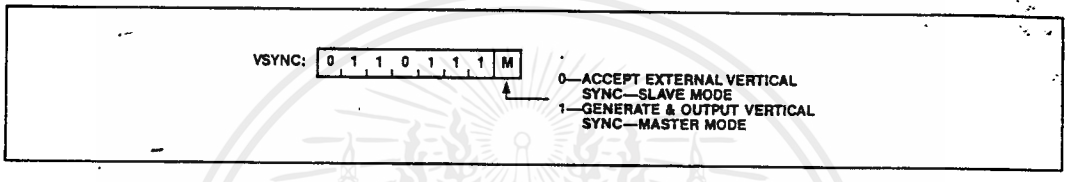


Figure 17. Vertical Sync Mode Command

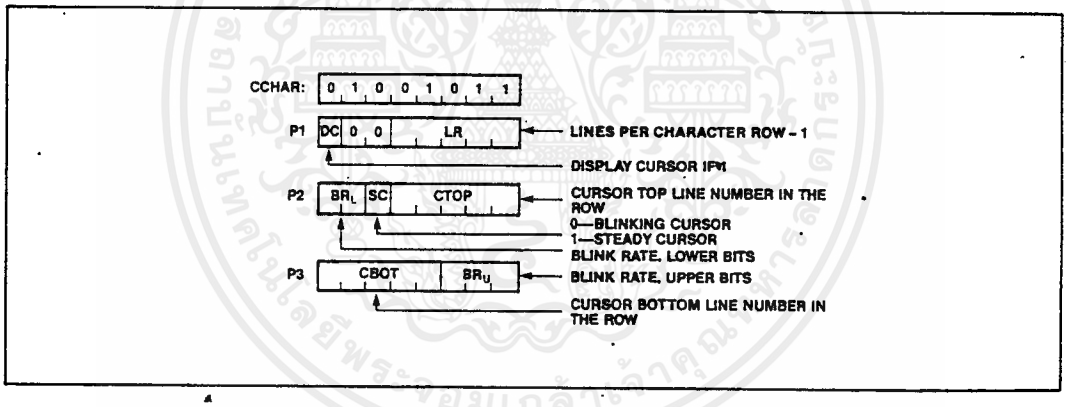


Figure 18. Cursor & Character Characteristics Command

Cursor and Character Characteristics Command

In graphics mode, LR should be set to 0. For interlaced displays in graphics mode, BR should be set to 3. The blink rate parameter controls both the cursor and attribute blink rates. The cursor blink-on-time = blink-off-time = 2 × BR (video frames). The attribute blink rate is always ½ the cursor rate but with a ¾ on-¼ off duty cycle.

DISPLAY CONTROL COMMANDS

Zoom Factors Specify Command

Zoom magnification factors of 1 through 16 are available using codes 0 through 15, respectively.

Cursor Position Specify Command

In character mode, the third parameter byte is not needed. The cursor is displayed for the word time in which the display scan address (DAD) equals the cursor address. In graphics mode, the cursor word address specifies the word containing the starting pixel of the drawing; the dot address value specifies the pixel within that word.

Parameter RAM Load Command

From the starting address, SA, any number of bytes may be loaded into the parameter RAM at incrementing addresses, up to location 15. The sequence of parameter bytes is terminated by the next command byte entered into the FIFO. The parameter RAM stores 16 bytes of information in predefined locations which differ for graphics and character modes. See the parameter RAM discussion for bit assignments.

Pitch Specification Command

This value is used during drawing by the drawing processor to find the word directly above or below the current word, and during display to find the start of the next line.

The Pitch parameter (width of display memory) is set by two different commands. In addition to the PITCH command, the RESET (or SYNC) command also sets the pitch value. The "active words per line" parameter, which specifies the width of the raster-scan display, also sets the Pitch of the display memory. In situations in which these two values are equal there is no need to execute a PITCH command.

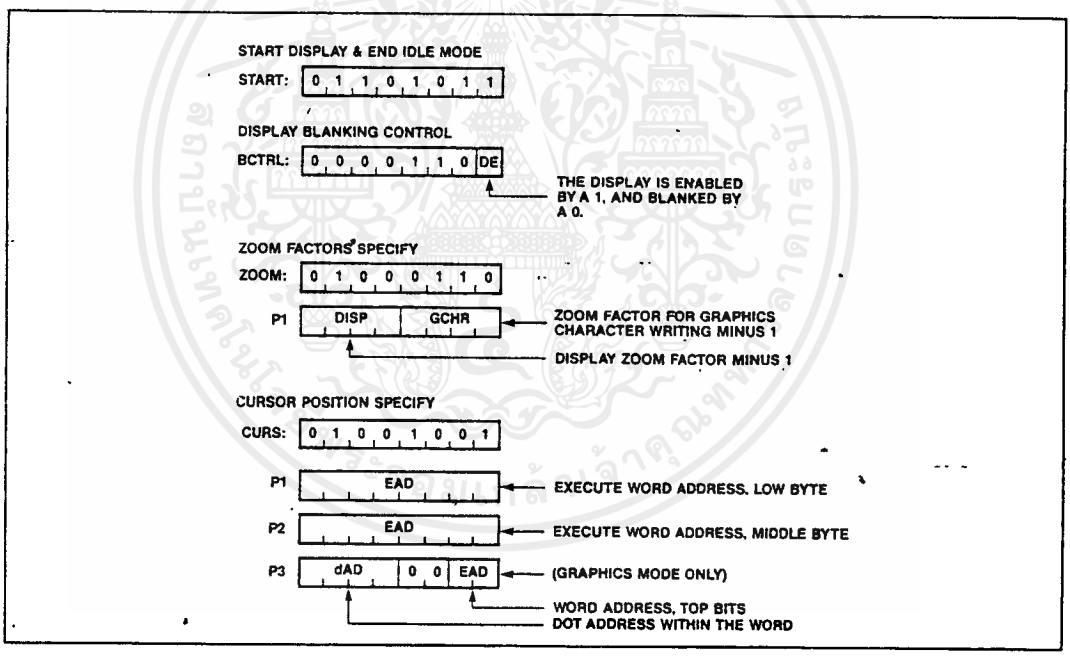


Figure 19. Display Control Commands

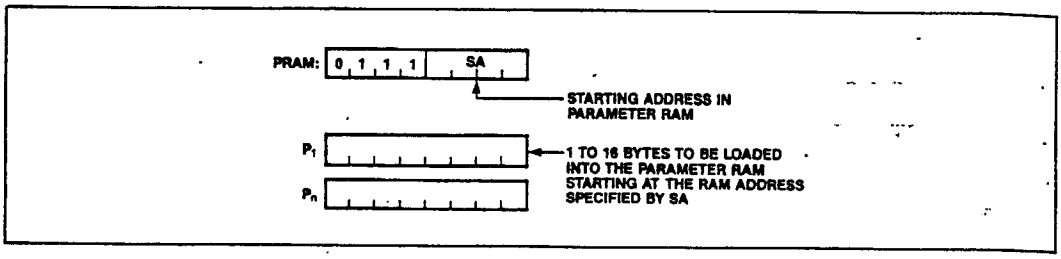


Figure 20. Parameter RAM Load Command

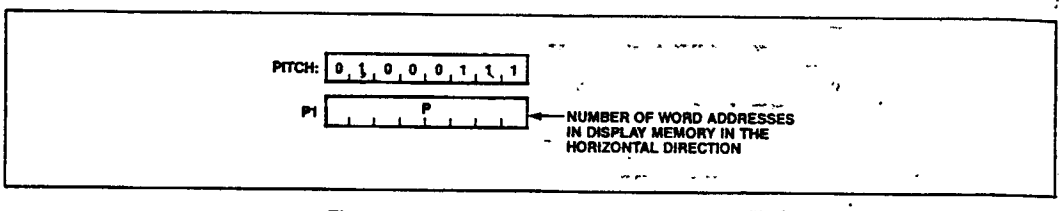


Figure 21. Pitch Specification Command

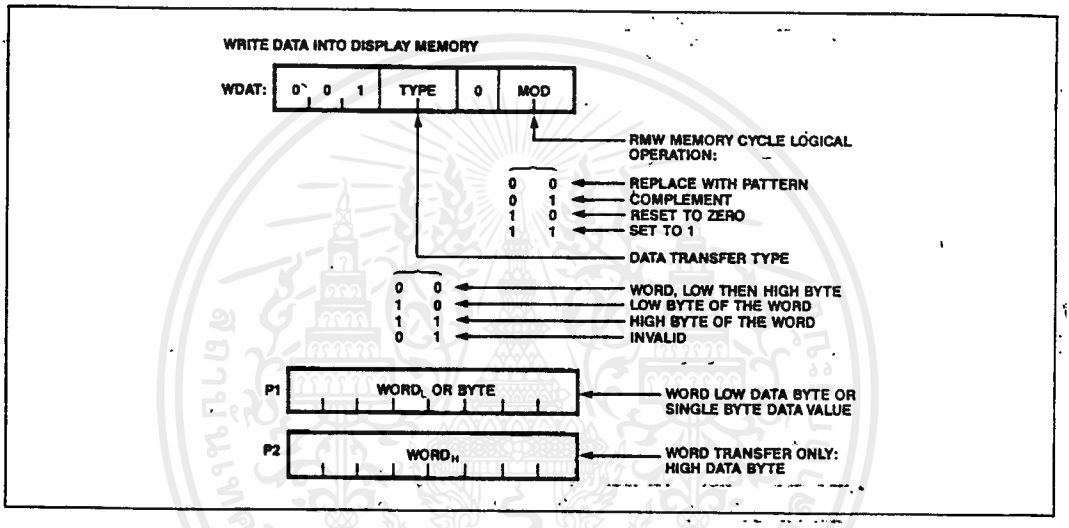


Figure 22. Write Data Command

DRAWING CONTROL COMMANDS

Write Data Command

Upon receiving a set of parameters (two bytes for a word transfer, one for a byte transfer), one RMW cycle into Video Memory is done at the address pointed to by the cursor EAD. The EAD pointer is advanced to the next word, according to the previously specified direction. More parameters can then be accepted.

For byte writes, the unspecified byte is treated as all zeros during the RMW memory cycle.

In graphics bit-map situations, only the LSB of the WDAT parameter bytes is used as the pattern in the RMW operations. Therefore it is possible to have only an all ones or all zeros pattern. In coded character applications all the bits of the WDAT parameters are used to establish the drawing pattern.

The WDAT command operates differently from the other commands which initiate RMW cycle activity. It requires parameters to set up the Pattern register while the other commands use the stored values in the parameter RAM. Like all of these commands, the

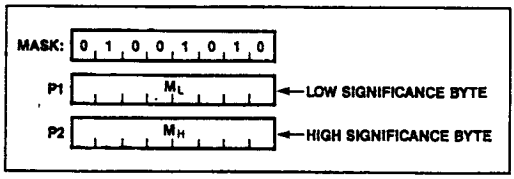
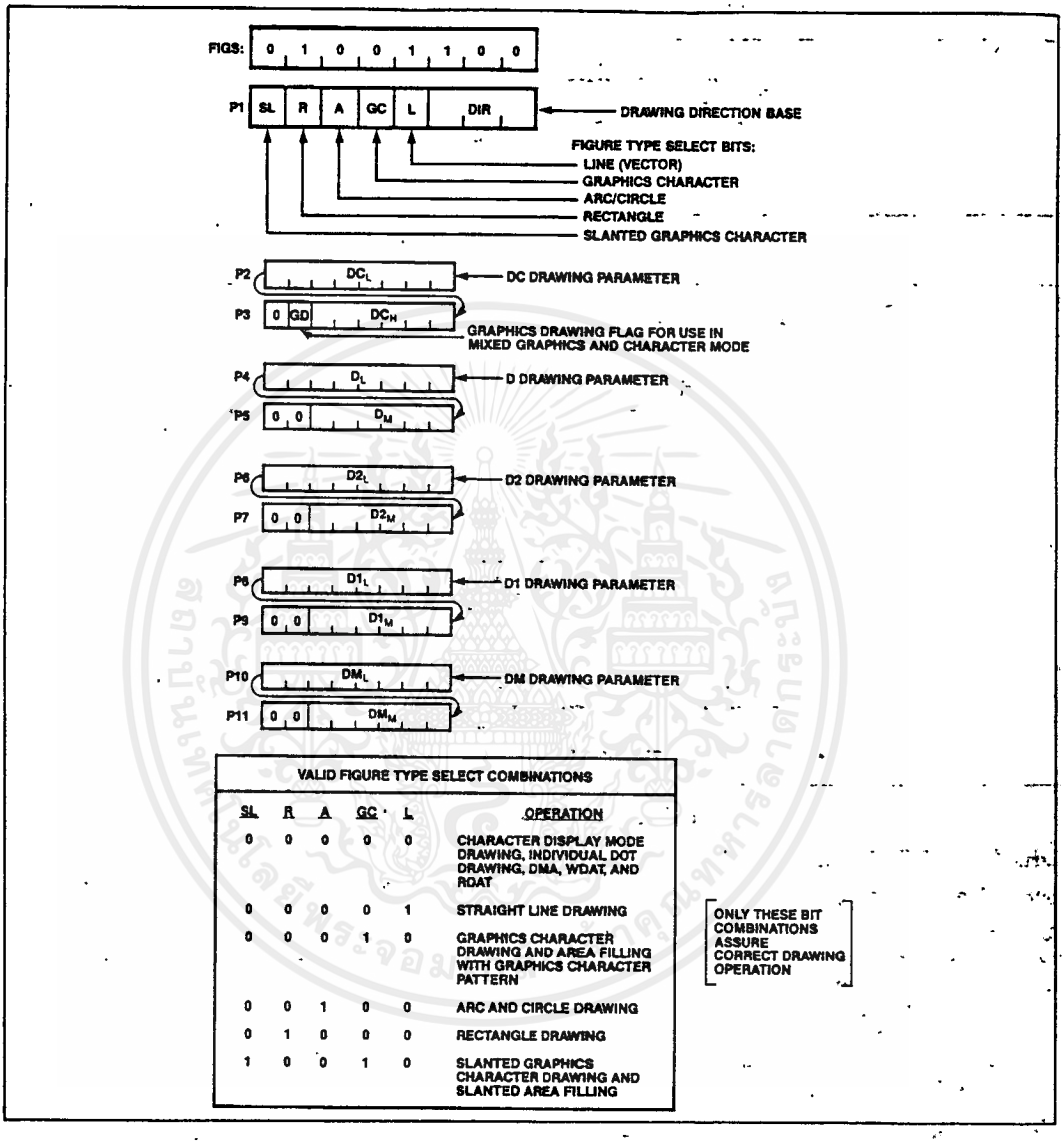


Figure 23. Mask Register Load Command

WDAT command must be preceded by a FIGS command and its parameters. Only the first three parameters need be given following the FIGS opcode, to set up the type of drawing, the DIR direction, and the DC value. The DC parameter +1 will be the number of RMW cycles done by the GDC with the first set of WDAT parameters. Additional sets of WDAT parameters will see a DC value of 0 which will cause only one RMW cycle to be executed.



VALID FIGURE TYPE SELECT COMBINATIONS

SL	R	A	GC	L	OPERATION
0	0	0	0	0	CHARACTER DISPLAY MODE DRAWING, INDIVIDUAL DOT DRAWING, DMA, WDAT, AND RDAT
0	0	0	0	1	STRAIGHT LINE DRAWING
0	0	0	1	0	GRAPHICS CHARACTER DRAWING AND AREA FILLING WITH GRAPHICS CHARACTER PATTERN
0	0	1	0	0	ARC AND CIRCLE DRAWING
0	1	0	0	0	RECTANGLE DRAWING
1	0	0	1	0	SLANTED GRAPHICS CHARACTER DRAWING AND SLANTED AREA FILLING

ONLY THESE BIT COMBINATIONS ASSURE CORRECT DRAWING OPERATION

Figure 24. Figure Drawing Parameters Specify Command

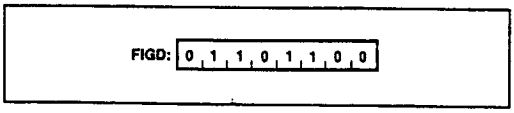


Figure 25. Figure Draw Start Command

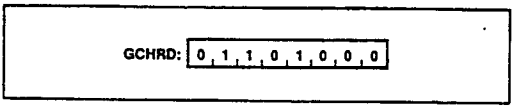


Figure 26. Graphics Character Draw and Area Filling Start Command

Mask Register Load Command

This command sets the value of the 16-bit Mask register of the figure drawing processor. The Mask register controls which bits can be modified in the display memory during a read-modify-write cycle.

The Mask register is loaded both by the MASK command and the third parameter byte of the CURS command. The MASK command accepts two parameter bytes to load a 16-bit value into the MASK register. All 16 bits can be individually one or zero, under program control. The CURS command on the other hand, puts a "1 of 16" pattern into the Mask register based on the value of the Dot Address value, dAD. If normal single-pixel-at-a-time graphics figure drawing is desired, there is no need to do a MASK command at all since the CURS command will set up the proper pattern to address the proper pixels as drawing progresses. For coded character DMA, and screen setting and clearing operations using the WDAT command, the MASK command should be used after the CURS command if its third parameter byte has been output. The Mask register should be set to all ones for any "word-at-a-time" operation.

Figure Draw Start Command

On execution of this instruction, the GDC loads the parameters from the parameter RAM into the drawing processor and starts the drawing process at the pixel pointed to by the cursor, EAD, and the dot address, dAD.

Graphics Char. Draw and Area Fill Start Command

Based on parameters loaded with the FIGS command, this command initiates the drawing of the graphics character or area filling pattern stored in Parameter RAM. Drawing begins at the address in display memory pointed to by the EAD and dAD values.

DATA READ COMMANDS

Read Data Command

Using the DIR and DC parameters of the FIGS command to establish direction and transfer count, multiple RMW cycles can be executed without specification of the cursor address after the initial load (DC = number of words or bytes).

As this instruction begins to execute, the FIFO buffer direction is reversed so that the data read from display memory can pass to the microprocessor. Any commands or parameters in the FIFO at this time will be lost. A command byte sent to the GDC will immediately reverse the buffer direction back to write mode, and all RDAT information not yet read from the FIFO will be lost. MOD should be set to 00.

Cursor Address Read Command

The Execute Address, EAD, points to the display memory word containing the pixel to be addressed.

The Dot Address, dAD, within the word is represented as a 1-of-16 code.

Light Pen Address Read Command

The light pen address, LAD, corresponds to the display word address, DAD, at which the light pen input signal is detected and deglitched.

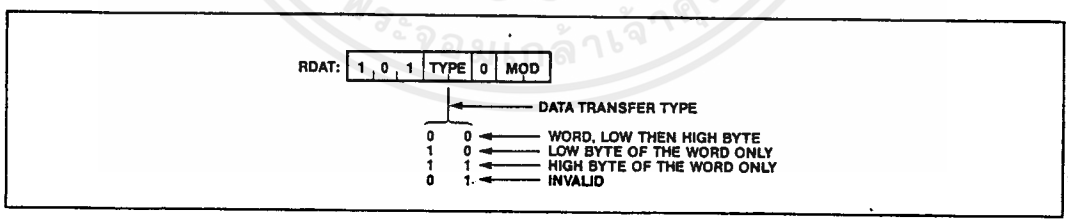


Figure 27. Read Data from Display Memory Command

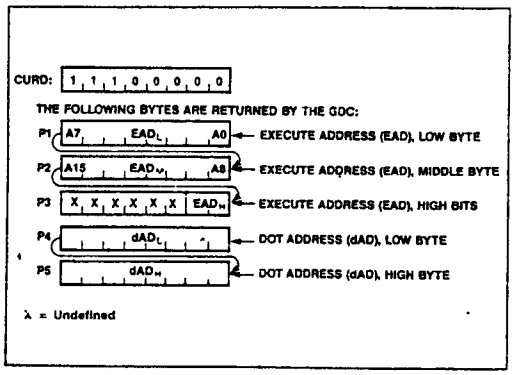


Figure 28. Cursor Address Read Command

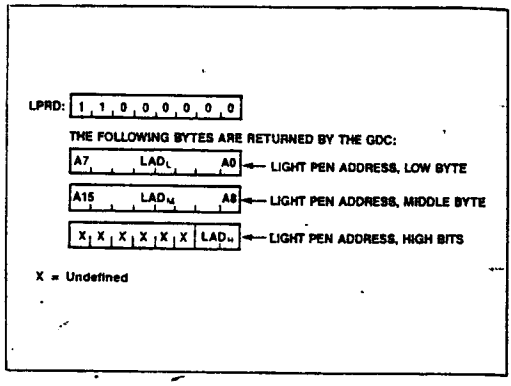


Figure 29. Light Pen Address Read Command

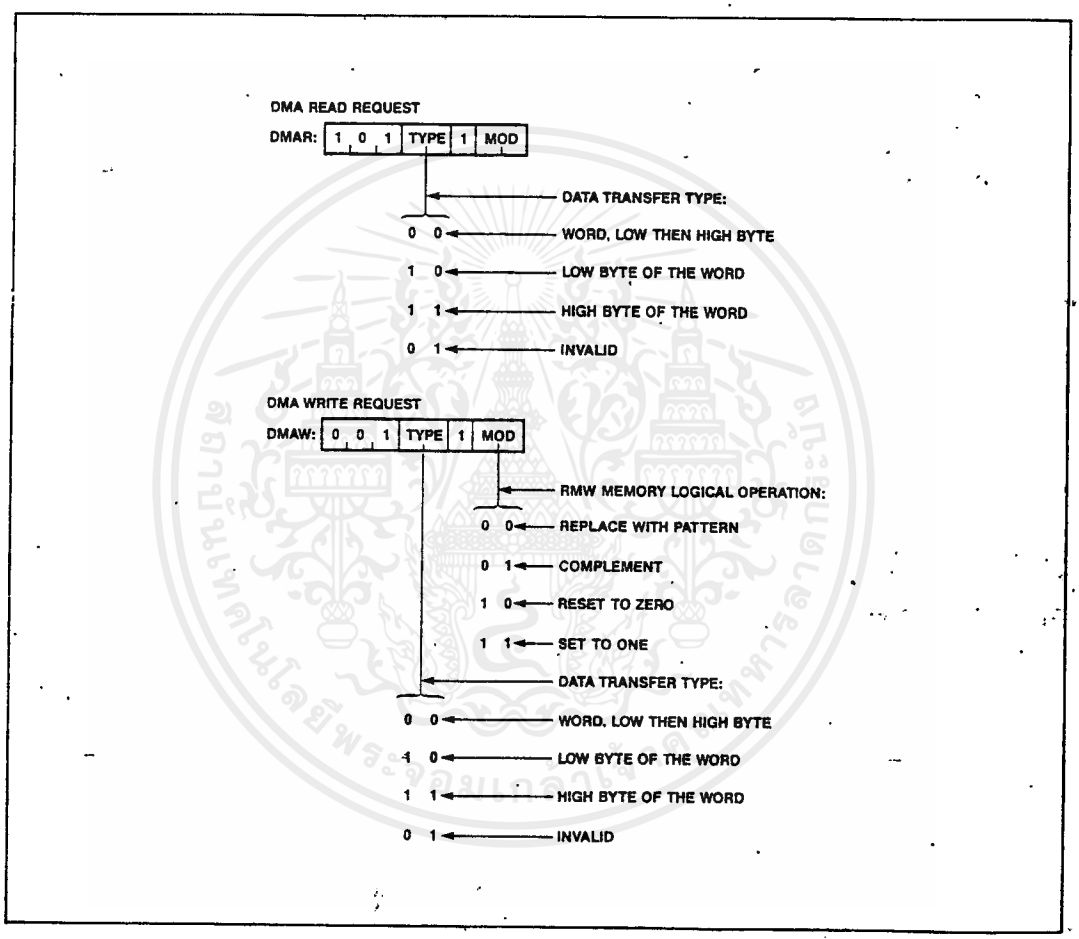


Figure 30. DMA Control Commands

210655-002

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ..... 0°C to 70°C
Storage Temperature ..... −65°C to 150°C
Voltage on any Pin with Respect to Ground ..... −0.5V to +7V
Power Dissipation ..... 1.5 Watt

COMMENT: Exposing the device to stresses above those listed in Absolute Maximum Ratings could cause permanent damage. The device is not meant to be operated under conditions outside the limits described in the operational sections of this specification. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC CHARACTERISTICS

T_A = 0°C to 70° C; V_{CC} = 5V ± 10%; GND = 0V

Symbol	Parameter	Limits		Unit	Conditions
		Min.	Max.		
V_{IL}	Input Low Voltage	−0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC} + 0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = −400 μA
I_{OZ}	Output Leakage Current		±10	μA	$V_{SS}+0.45 \leq V_I \leq V_{CC}$
I_{IL}	Input Leakage Current		±10	μA	$V_{SS} \leq V_I \leq V_{CC}$
V_{CL}	Clock Input Low Voltage	−0.5	0.6	V	
V_{CH}	Clock Input High Voltage	3.5	V_{CC} + 0.5	V	
I_{CC}	V_{CC} Supply Current		270	mA	Typical = 150 mA

CAPACITANCE

T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Limits		Unit	Conditions
		Min.	Max.		
C_{IN}	Input Capacitance		10	pF	fc = 1 MHz V = 0
C_{IO}	I/O Capacitance		20	pF	
C_{OUT}	Output Capacitance		20	pF	
C_O	Clock Input Capacitance		20	pF	

A.C. CHARACTERISTICS (T_A = 0°C to +70°C, V_{SS} = 0V, V_{CC} = +5V ±10%)

DATA BUS READ CYCLE

Symbol	Parameter	82720		82720-1		82720-2		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
T_{AR}	A_0 setup to $\overline{RD}$ ↓	0		0		0		ns	
T_{RA}	A_0 hold after $\overline{RD}$ ↑	0		0		0		ns	
T_{RR}	$\overline{RD}$ Pulse Width	$T_{RD}+20$		$T_{RD}+20$		$T_{RD}+20$		ns	
T_{RD}	$\overline{RD}$ ↓ to Data Out Delay		120		80		70	ns	CL = 50pF
T_{DF}	$\overline{RD}$ ↑ to Data Float Delay	0	120	0	100	0	90	ns	
T_{RV}	$\overline{RD}$ Recovery Time	4 T_{CY}		4 T_{CY}		4 T_{CY}		ns	

DATA BUS WRITE CYCLE

Symbol	Parameter	82720		82720-1		82720-2		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
T_{AW}	A_0 Setup to $\overline{WR}$ ↓	0		0		0		ns	
T_{WA}	A_0 Hold after $\overline{WR}$ ↑	0		0		10		ns	
T_{WW}	$\overline{WR}$ Pulse Width	120		100		90		ns	
T_{DW}	Data Setup to $\overline{WR}$ ↑	100		80		70		ns	
T_{WD}	Data Hold after $\overline{WR}$ ↑	0		0		10		ns	
T_{RV}	$\overline{WR}$ Recovery Time	4 T_{CY}		4 T_{CY}		4 T_{CY}		ns	

DISPLAY MEMORY TIMING

Symbol	Parameter	82720		82720-1		82720-2		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
T_{CA}	Address/Data Delay from 2XWCLK ↑	30	160	30	130	30	110	ns	CL = 50pF
T_{AC}	Address/Data Hold Time	30	160	30	130	30	110	ns	CL = 50pF
T_{DC}	Data Setup to 2XWCLK ↓	0		0		0		ns	
T_{CD}	Data Hold Time	$T_{IE}+20$		$T_{IE}+20$		$T_{IE}+20$		ns	
T_{IE}	2XWCLK ↑ to $\overline{DBIN}$	30	120	30	90	30	80	ns	CL = 50pF
T_{CAH}	2XWCLK ↑ to ALE ↑	30	125	30	100	30	90	ns	CL = 50pF
T_{CAL}	2XWCLK ↑ to ALE ↓	30	100	30	80	30	70	ns	CL = 50pF
T_{AL}	ALE Low Time	$T_{CY}+30$		$T_{CY}+30$		$T_{CY}+30$		ns	
T_{AH}	ALE High Time	$T_{CH}-20$		$T_{CH}-20$		$T_{CH}-20$		ns	
T_{CO}	Video Signal Delay from 2XWCLK ↑		150		120		100	ns	

A.C. CHARACTERISTICS (Continued)

OTHER TIMING

Symbol	Parameter	82720		82720-1		82720-2		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
T_{PC}	LPEN or VSYNC Input Setup to 2XWCLK↑	30		20		15		ns	
T_{PP}	LPEN or VSYNC Input Pulse Width	T_{CY}		T_{CY}		T_{CY}		ns	

CLOCK TIMING

Symbol	Parameter	82720		82720-1		82720-2		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
T_{CY}	Clock Period	250	2000	200	2000	180	2000	ns	
T_{CH}	Clock High Time	105		80		70		ns	
T_{CL}	Clock Low Time	105		80		70		ns	
T_R	Rise Time		20		20		20	ns	
T_F	Fall Time		20		20		20	ns	

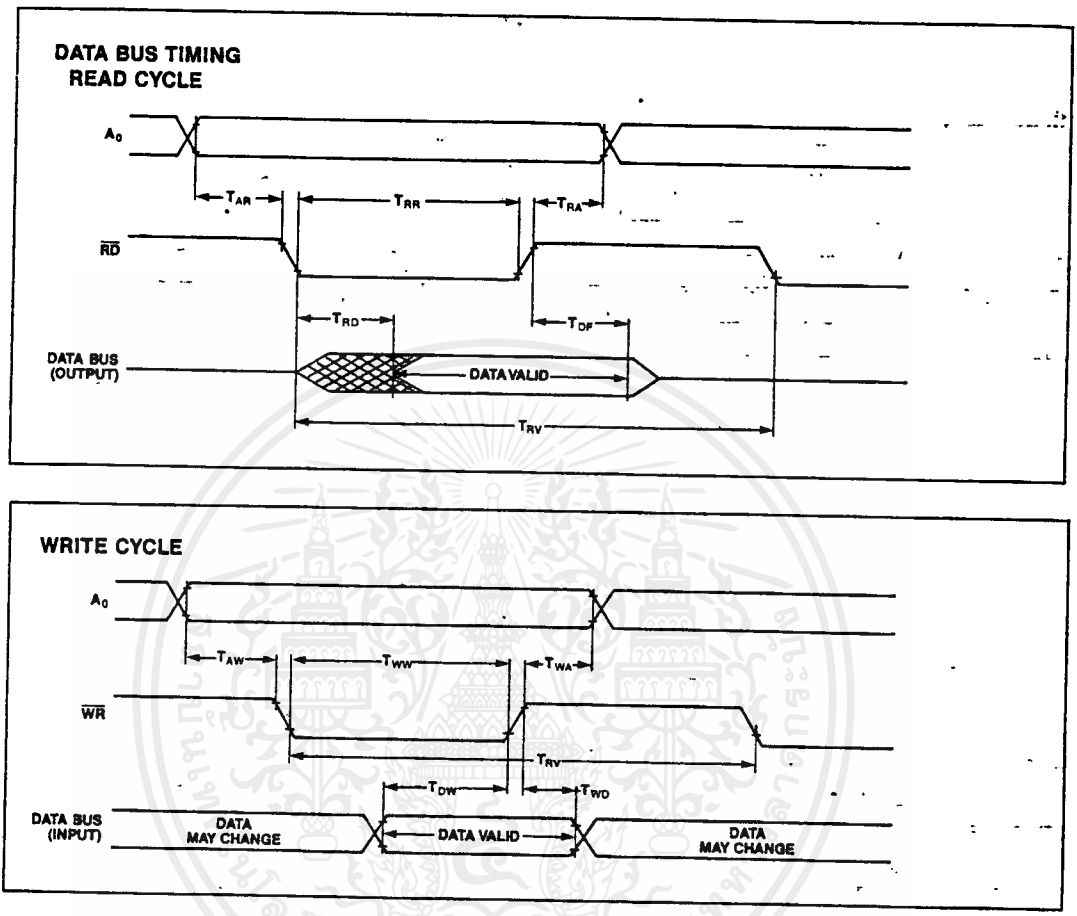
DMA TIMING

Symbol	Parameter	82720		82720-1		82720-2		Units	Test Conditions
		Min.	Max.	Min.	Max.	Min.	Max.		
T_{ACC}	$\overline{DACK}$ Setup to $\overline{RD}$↓ or $\overline{WR}$↓	0		0		0		ns	
T_{CAC}	$\overline{DACK}$ Hold from $\overline{RD}$↑ or $\overline{WR}$↑	0		0		0		ns	
T_{RR1}	$\overline{RD}$ Pulse Width	$T_{RD1}+20$		$T_{RD1}+20$		$T_{RD1}+20$		ns	
T_{RD1}	$\overline{RD}$↓ to Data Out Delay		1.5 T_{CY} + 120		1.5 T_{CY} + 80		1.5 T_{CY} + 70	ns	CL = 50pF
T_{KQ}	2XWCLK↑ to DREQ Delay		150		120		100	ns	CL = 50pF
T_{RQAK}	DREQ Setup to DACK↓	0		0		0		ns	
T_{AKRQ}	$\overline{DACK}$↓ to DREQ↓ Delay		$T_{CY}+150$		$T_{CY}+120$		$T_{CY}+100$	ns	CL = 50pF
T_{AKH}	$\overline{DACK}$ High Time	T_{CY}		T_{CY}		T_{CY}		ns	
T_{AK1}	$\overline{DACK}$ Cycle Time, Word Mode	4 T_{CY}		4 T_{CY}		4 T_{CY}		ns	
T_{AK2}	$\overline{DACK}$ Cycle Time, Byte Mode	5 T_{CY}		5 T_{CY}		5 T_{CY}		ns	

A.C. TEST CONDITIONS

Input Pulse Levels (except 2XWCLK) ..... 0.45V to 2.4V
Input Pulse Levels (2XWCLK) ..... 0.3V to 3.9V
Timing Measurement Reference Levels (except 2XWCLK) ..... 0.8V to 2.0V
Timing Measurement Reference Levels (2XWCLK) ..... 0.6V to 3.5V

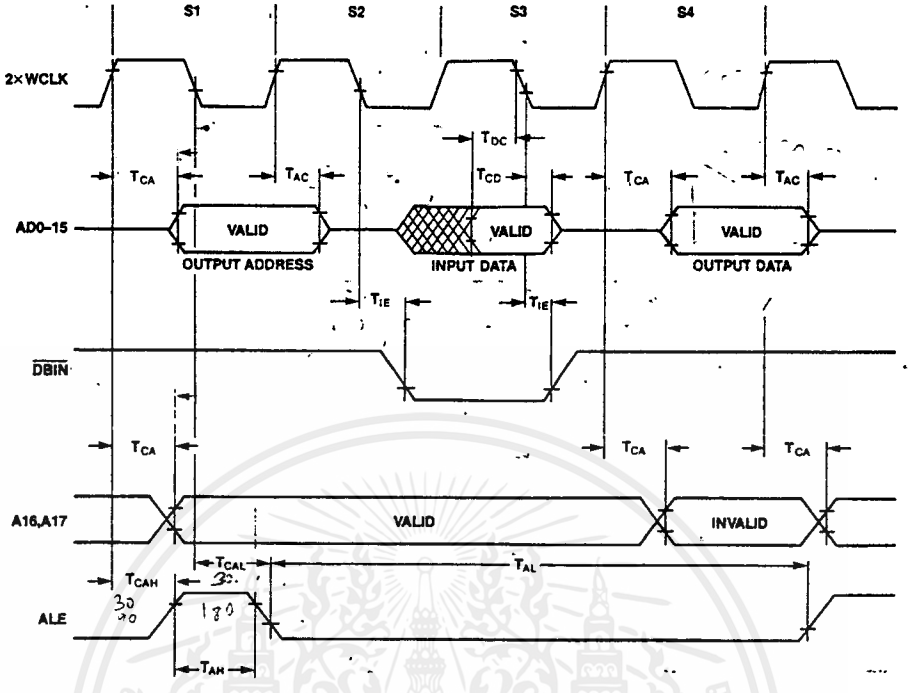
WAVEFORMS



210655-002

WAVEFORMS (Continued)

DISPLAY MEMORY TIMING
READ/MODIFY/WRITE CYCLE

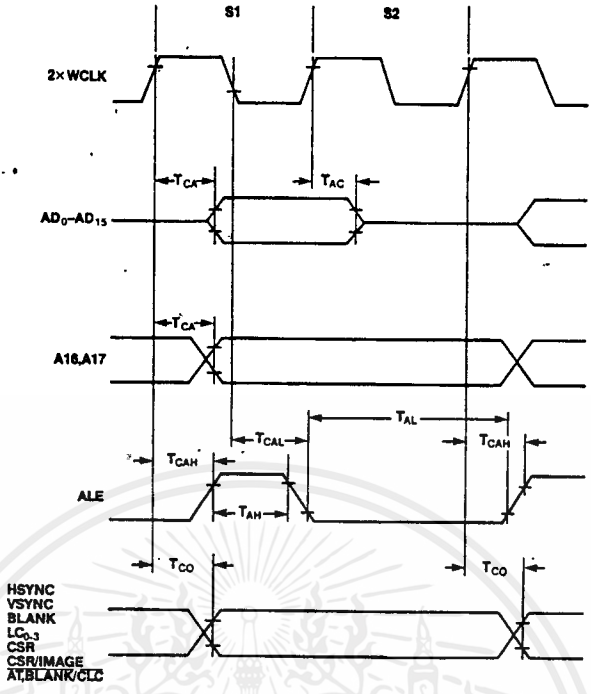


210655-002

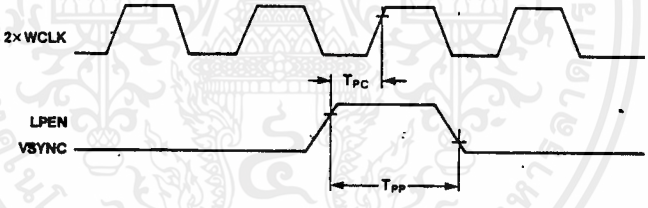
WAVEFORMS (Continued)

DISPLAY MEMORY TIMING (Continued)

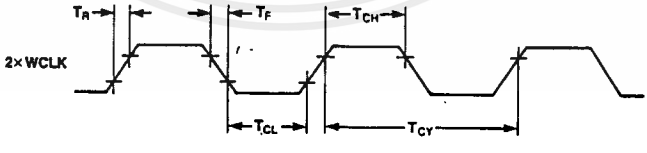
READ CYCLE



OTHER TIMING

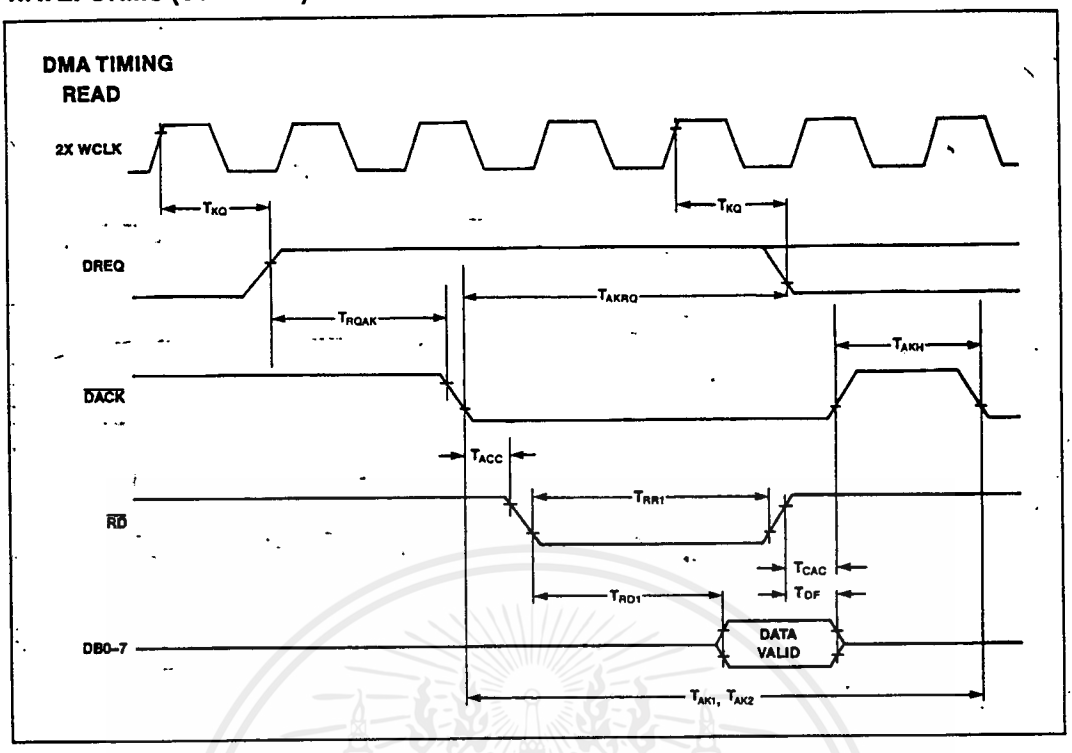


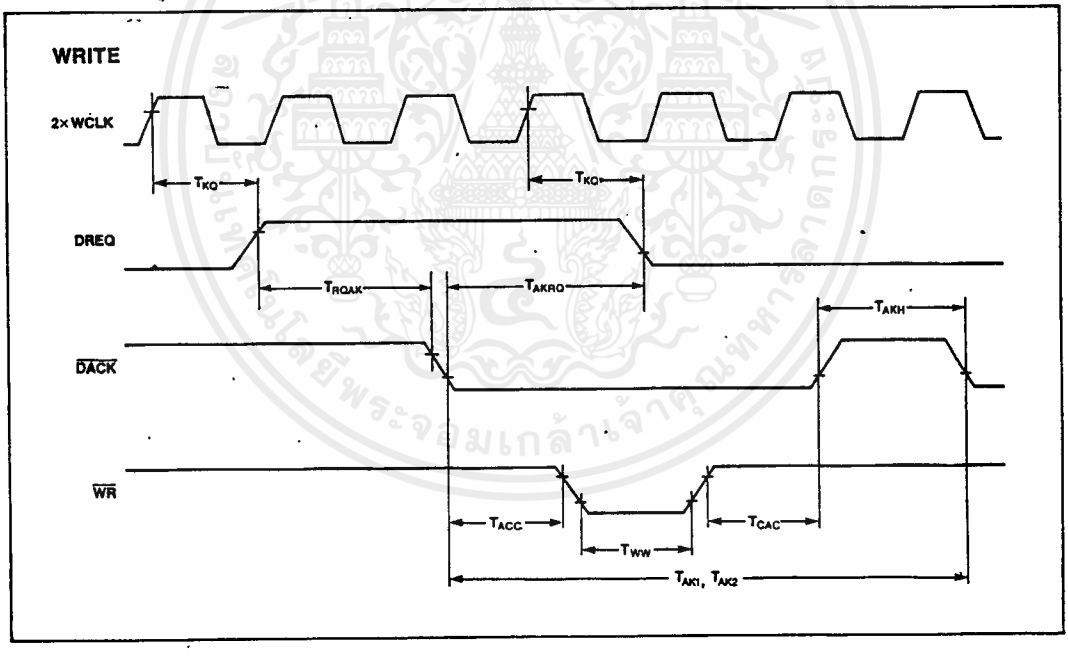
CLOCK TIMING



210655-002

WAVEFORMS (Continued)

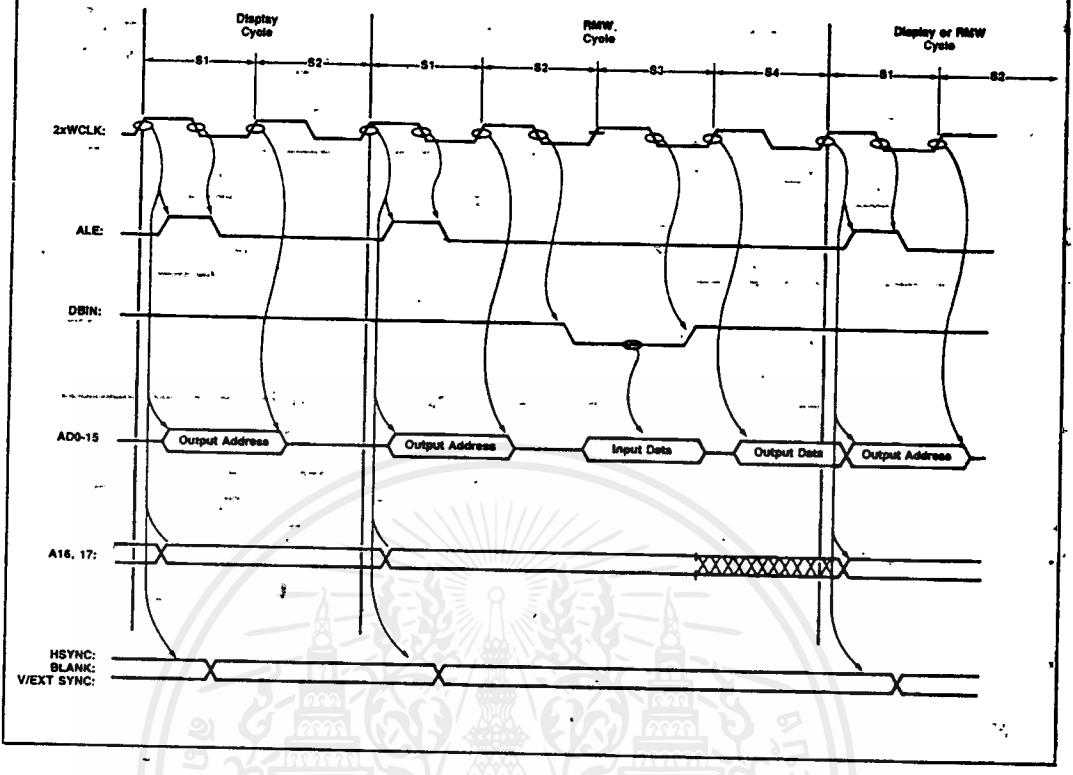




210655-002

WAVEFORMS (Continued)

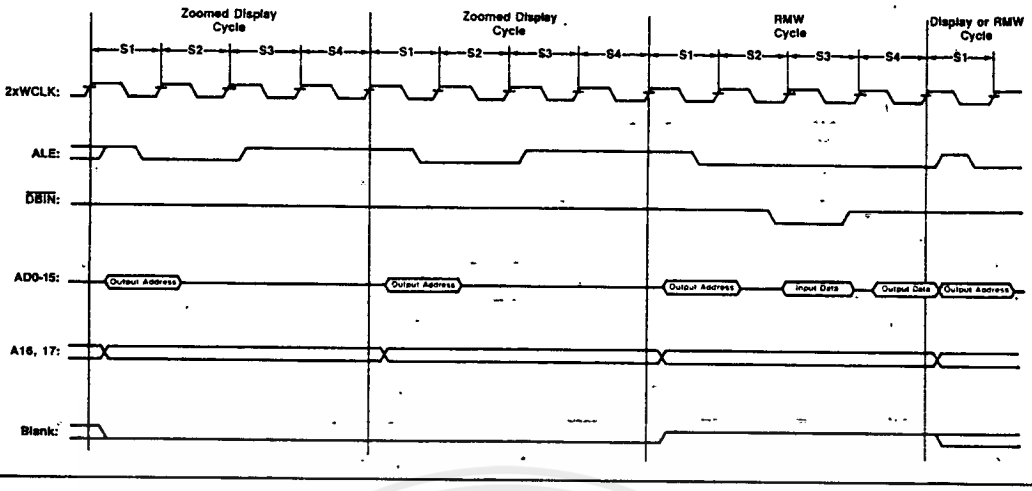
DISPLAY AND RMW CYCLES (1X ZOOM)



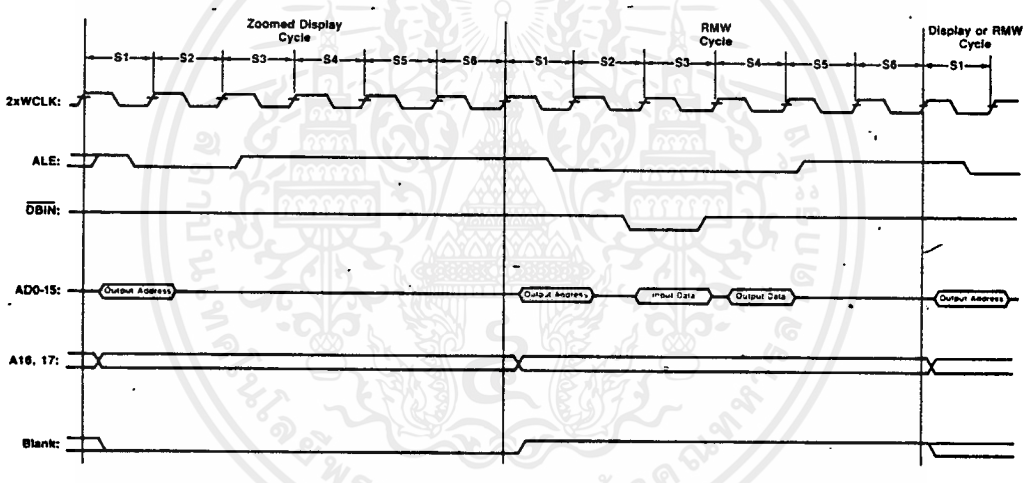
210655-002

WAVEFORMS (Continued)

DISPLAY AND RMW CYCLES (2X ZOOM)



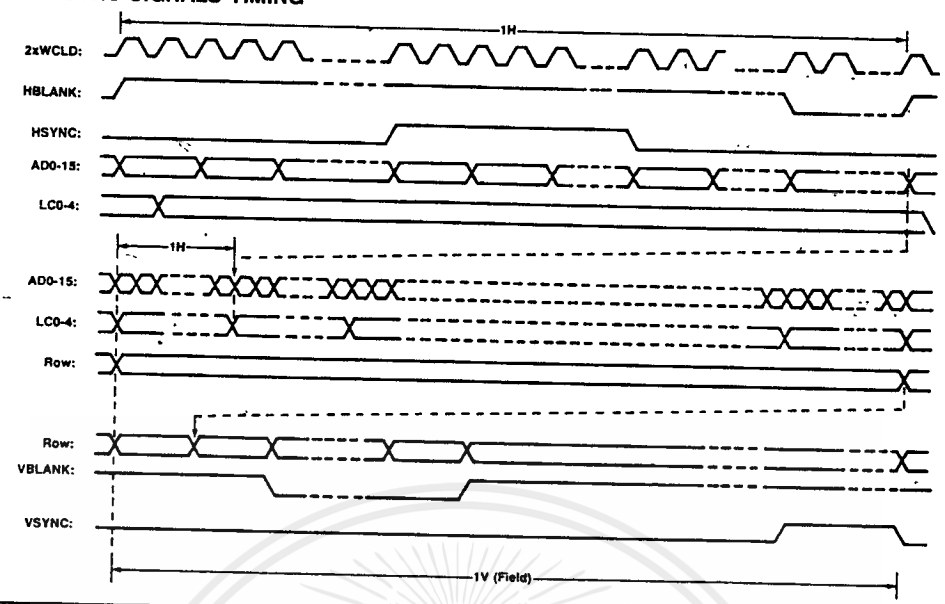
ZOOMED DISPLAY OPERATION WITH RMW CYCLE (3X ZOOM)



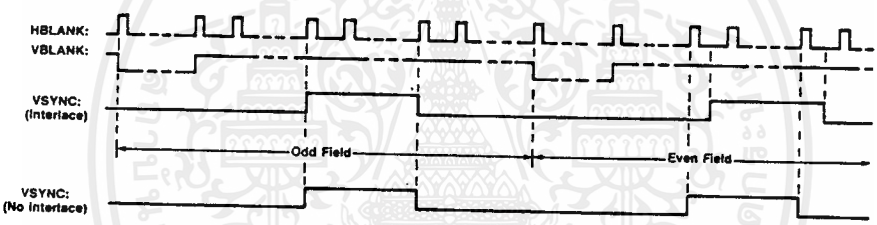
210655-002

WAVEFORMS (Continued)

VIDEO SYNC SIGNALS TIMING



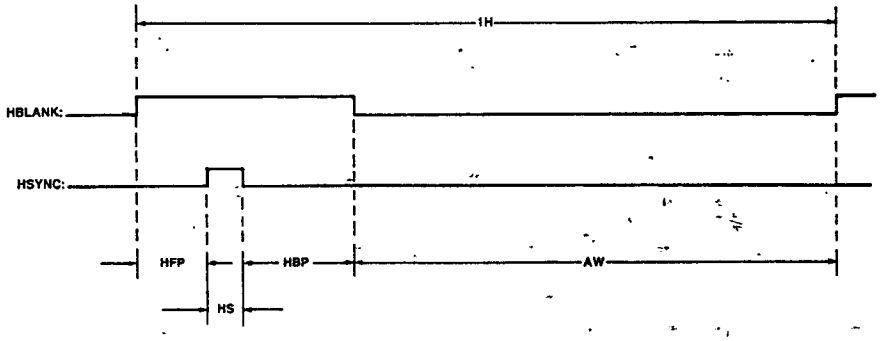
INTERLACED VIDEO TIMING



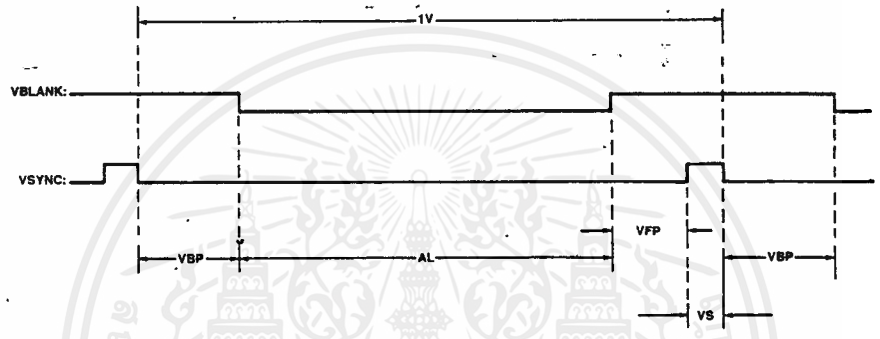
210655-002

WAVEFORMS (Continued)

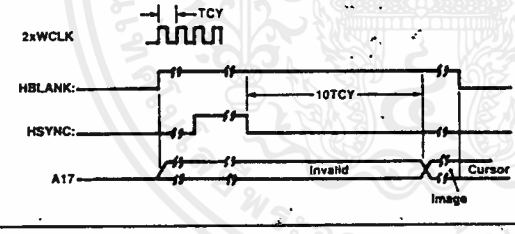
VIDEO HORIZONTAL SYNC GENERATOR PARAMETERS



VIDEO VERTICAL SYNC GENERATOR PARAMETERS

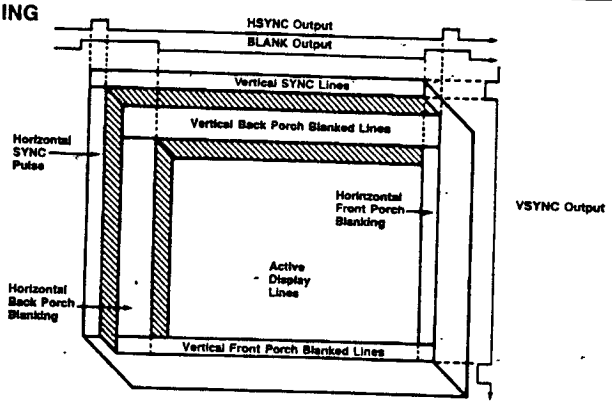


CURSOR—IMAGE BIT FLAG

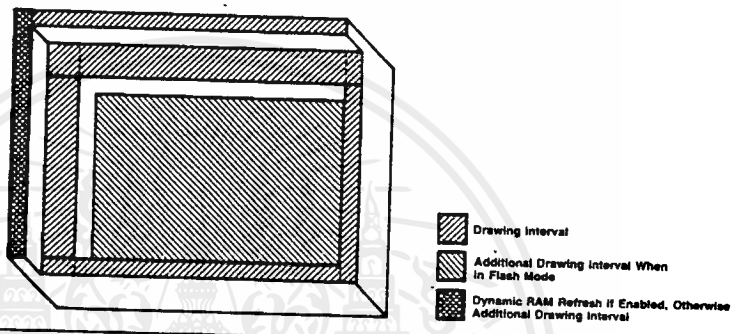


210655-002

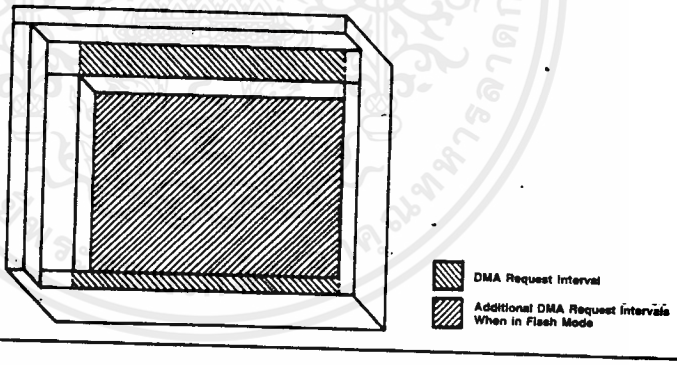
VIDEO FIELD TIMING



DRAWING INTERVALS



DMA REQUEST INTERVALS



210655-002

64K×4 Bit Dynamic RAM with Page Mode

FEATURES

- **Performance range**

	t_{RAC}	t_{CAC}	t_{RC}
KM41464A-10	100ns	50ns	190ns
KM41464A-12	120ns	60ns	220ns
KM41464A-15	150ns	75ns	260ns

- **Page Mode capability**
- **$\overline{CAS}$-before-$\overline{RAS}$ Refresh capability**
- **$\overline{RAS}$-only and Hidden Refresh capability**
- **TTL compatible inputs and outputs**
- **Early Write or Output Enable Controlled Write**
- **Single +5V±10% power supply**
- **256 cycle/4ms refresh**
- **JEDEC standard pinout in 18-pin DIP, 18-lead PLCC and 20-pin ZIP.**

GENERAL DESCRIPTION

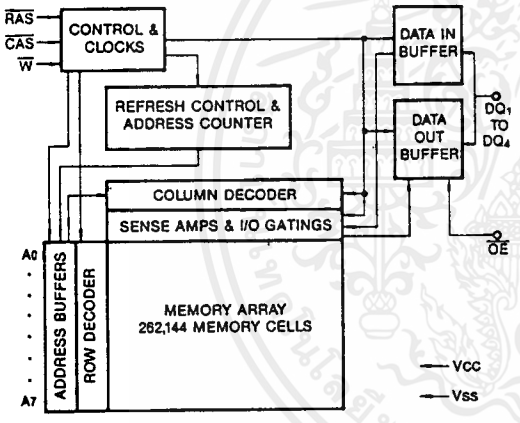
The KM41464A is a fully decoded 65,536×4 NMOS Dynamic Random Access Memory. The design is optimized for high speed, high performance applications such as computer memory, buffer memory, peripheral storage and environments where low power dissipation and compact layout are required.

The KM41464A features page mode which allows high speed random access of memory cells within the same row. $\overline{CAS}$-before-$\overline{RAS}$ refresh capability provides on-chip auto refresh as an alternative to $\overline{RAS}$-only refresh. Multiplexed row and column address inputs permit the KM41464A to be housed in a standard 18-pin DIP.

The KM41464A is fabricated using Samsung's advanced silicon gate NMOS process. This process, coupled with single transistor memory storage cells, permits maximum circuit density and minimal chip size.

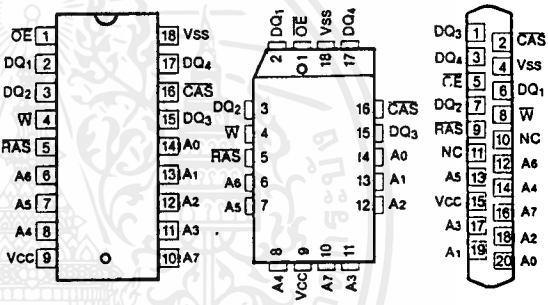
Clock timing requirements are noncritical, and power supply tolerance is very wide. All inputs and outputs are TTL complible.

FUNCTIONAL BLOCK DIAGRAM



PIN CONFIGURATION

- **KM41464AP**
- **KM41464AJ**
- **KM41464AZ**



Pin Name	Pin Function
A_0-A_7	Address Inputs
$\overline{RAS}$	Row Address Strobe
$\overline{CAS}$	Column Address Strobe
$\overline{W}$	Read/Write Input
$\overline{OE}$	Output Enable
DQ_1-DQ_4	Data In/Out
V_{CC}	Power (+5V)
V_{SS}	Ground

ABSOLUTE MAXIMUM RATINGS*

Parameter	Symbol	Rating	Units
Voltage on any pin relative to V_{SS}	V_{IN}, V_{OUT}	−1 to +7.0	V
Voltage on V_{CC} supply relative to V_{SS}	V_{CC}	−1 to +7.0	V
Storage Temperature	T_{stg}	−55 to +150	°C
Power Dissipation	P_D	1.0	W
Short Circuit Output Current	I_{OS}	50	mA

*Note: Permanent device damage may occur of ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to the conditions as detailed in the operational sections of this data sheet. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED OPERATING CONDITIONS (Voltages referenced to V_{SS}, T_A = 0 to 70°C)

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage	V_{CC}	4.5	5.0	5.5	V
Ground	V_{SS}	0	0	0	V
Input High Voltage	V_{IH}	2.4	—	V_{CC}+1	V
Input Low Voltage	V_{IL}	−1.0	—	0.8	V

DC AND OPERATING CHARACTERISTICS

(Recommended operating conditions unless otherwise noted.)

Parameter		Symbol	Min	Max	Units
OPERATING CURRENT* ($\overline{RAS}$ and $\overline{CAS}$ cycling; @t_{RC} = min.)	KM41464A-12 KM41464A-15	I_{CC1}	— —	75 65	mA mA
STANDBY CURRENT ($\overline{RAS}$ = $\overline{CAS}$ = V_{IH} after 8 $\overline{RAS}$ cycles min.)		I_{CC2}	—	4.5	mA
$\overline{RAS}$-ONLY REFRESH CURRENT* ($\overline{CAS}$ = V_{IH}, $\overline{RAS}$ cycling; @t_{RC} = min.)	KM41464A-12 KM41464A-15	I_{CC3}	— —	65 60	mA mA
PAGE MODE CURRENT* ($\overline{RAS}$ = V_{IL}, $\overline{CAS}$ cycling; @t_{PC} = min.)	KM41464A-12 KM41464A-15	I_{CC4}	— —	55 45	mA mA
$\overline{CAS}$-BEFORE-$\overline{RAS}$ REFRESH CURRENT ($\overline{RAS}$ cycling; @t_{RC} = min.)	KM41464A-12 KM41464A-15	I_{CC5}	— —	65 60	mA mA
INPUT LEAKAGE CURRENT (Any input 0≤V_{IN}≤5.5V, V_{CC} = 5.5V, V_{SS} = 0V, all other pins not under test = 0 volts.)		I_{IL}	−10	10	μA
OUTPUT LEAKAGE CURRENT (Data out is disabled, 0V≤V_{OUT}≤5.5V		I_{DOL}	−10	10	μA
OUTPUT HIGH VOLTAGE LEVEL (I_{OH} = −5mA)		V_{OH}	2.4	—	V
OUTPUT LOW VOLTAGE LEVEL (I_{OL} = 4.2mA)		V_{OL}	—	0.4	V

*Note: I_{CC} is dependent on output loading and cycle rates. Specified values are obtained with the output open. I_{CC} is specified as an average current.

CAPACITANCE ($T_A = 25°C$)

Parameter	Symbol	Min	Max	Unit
Input Capacitance (A_0-A_7)	C_{IN1}	—	7	pF
Input Capacitance ($\overline{RAS}$, $\overline{CAS}$, $\overline{W}$, $\overline{OE}$)	C_{IN2}	—	10	pF
Output Capacitance (DQ_1-DQ_4)	C_{DQ}	—	7	pF

AC CHARACTERISTICS ($0°C \leq T_A \leq 70°C$, $V_{CC} = 5.0V \pm 10\%$. See notes 1,2)

KM41464A STANDARD OPERATION

Parameter	Symbol	KM41464A-12		KM41464A-15		Unit	Notes
		Min	Max	Min	Max		
Random read or write cycle time	t_{RC}	220		260		ns	
Read-modify-write cycle time	t_{RWC}	305		355		ns	
Access time from $\overline{RAS}$	t_{RAC}		120		150	ns	3, 4
Access time from $\overline{CAS}$	t_{CAC}		60		75	ns	3, 5
Output buffer turn-off delay time	t_{OFF}	0	30	0	40	ns	6
Transition time (rise and fall)	t_T	3	50	3	50	ns	
$\overline{RAS}$ precharge time	t_{RP}	90		100		ns	
$\overline{RAS}$ pulse width	t_{RAS}	120	10,000	150	10,000	ns	
$\overline{RAS}$ hold time	t_{RSH}	60		65		ns	
$\overline{CAS}$ precharge time (all cycles except page mode)	t_{CPN}	30		35		ns	
$\overline{CAS}$ pulse width	t_{CAS}	60	10,000	75	10,000	ns	
$\overline{CAS}$ hold time	t_{CSH}	120		150		ns	
$\overline{RAS}$ to $\overline{CAS}$ delay time	t_{RCD}	25	60	25	75	ns	4
$\overline{CAS}$ to $\overline{RAS}$ precharge time	t_{CRP}	10		10		ns	
Row address set-up time	t_{ASR}	0		0		ns	
Row address hold time	t_{RAH}	15		15		ns	
Column address set-up time	t_{ASC}	0		0		ns	
Column address hold time	t_{CAH}	20		25		ns	
Column address hold time referenced to $\overline{RAS}$	t_{AR}	80		100		ns	
Read command set-up time	t_{RCS}	0		0		ns	
Read command hold time referenced to $\overline{CAS}$	t_{RCH}	0		0		ns	
Read command hold time referenced to $\overline{RAS}$	t_{RRH}	20		20		ns	
Write command set-up time	t_{WCS}	0		0		ns	7
Write command hold time	t_{WCH}	40		45		ns	
Write command pulse width	t_{WP}	40		45		ns	
Write command to $\overline{RAS}$ lead time	t_{RWL}	40		45		ns	
Write command to $\overline{CAS}$ lead time	t_{CWL}	40		45		ns	

KM41464A STANDARD OPERATION (Continued)

Parameter	Symbol	KM41464A-12		KM41464A-15		Units	Notes
		Min	Max	Min	Max		
Data-in set-up time	t_{DS}	0		0		ns	
Data-in hold time	t_{DH}	40		45		ns	
$\overline{CAS}$ to write enable delay time	t_{CWD}	100		120		ns	7
$\overline{RAS}$ to write enable delay time	t_{RWD}	160		195		ns	7
Write command hold time referenced to $\overline{RAS}$	t_{WCR}	100		120		ns	
Data-in hold time referenced to $\overline{RAS}$	t_{DHR}	100		120		ns	
Access time from $\overline{OE}$	t_{OEA}		30		40	ns	
$\overline{OE}$ to Data in delay time	t_{OED}	30		40		ns	
Output Buffer turn off delay from $\overline{OE}$	t_{OEZ}	0	30	0	40	ns	
$\overline{OE}$ hold time referenced to $\overline{W}$	t_{OEH}	25		25		ns	
$\overline{OE}$ to $\overline{RAS}$ inactive setup time	t_{OES}	0		0		ns	
Din to $\overline{CAS}$ delay time	t_{DZC}	0		0		ns	8
Din to $\overline{OE}$ delay time	t_{DZO}	0		0		ns	8
Refresh period (256 cycles)	t_{REF}		4		4	ms	

KM41464A $\overline{CAS}$-BEFORE-$\overline{RAS}$ REFRESH

$\overline{CAS}$ setup time ($\overline{CAS}$-before-$\overline{RAS}$ Refresh)	t_{CSR}	25		30		ns	
$\overline{CAS}$ hold time ($\overline{CAS}$-before-$\overline{RAS}$ Refresh)	t_{CHR}	55		60		ns	
$\overline{RAS}$ precharge to $\overline{CAS}$ hold time	t_{PRC}	20		20		ns	

KM41464A PAGE MODE

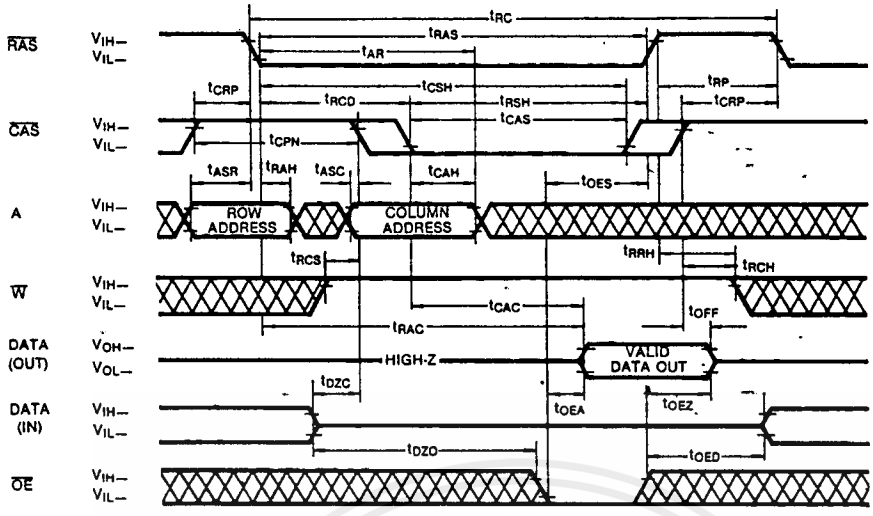
Page mode cycle time	t_{PC}	120		145		ns	
$\overline{CAS}$ precharge time (page mode only)	t_{CP}	50		60		ns	

NOTES

1. An initial pause of 100μs is required after power-up followed by any 8 $\overline{RAS}$ cycles before proper device operation is achieved.
2. V_{IH}(min) and V_{IL}(max) are reference levels for measuring timing of input signals. Transition times are measured between V_{IH}(min) and V_{IL}(max) and are assumed to be 5ns for all inputs.
3. Measured with a load equivalent to 2 TTL loads and 100pF.
4. Operation within the t_{RCD}(max) limit insures that t_{RAC}(max) can be met. t_{RCD}(max) is specified as a reference point only. If t_{RCD} is greater than the specified t_{RCD}(max) limit, then access time is controlled exclusively by t_{CAC}.
5. Assumes that $t_{RCD} \geq t_{RCD}$(max).
6. This parameter defines the time at which the output achieves the open circuit condition and is not referenced to V_{OH} or V_{OL}.
7. t_{CWD} and t_{RWD} are restrictive operating parameters for the read-modify-write cycle only. If $t_{WCS} \geq t_{WCS}$(min), the cycle is an early write cycle and the data output will remain open circuit throughout the entire cycle. If $t_{CWD} \geq t_{CWD}$(min) and $t_{RWD} > t_{RWD}$(min), the cycle is a late write cycle and the data output will contain data read from the selected cell. If neither of the above conditions are met, the condition of the data out (at access time until $\overline{CAS}$ goes back to V_{IH}) is indeterminate.
8. Either t_{OZC} or t_{OZO} must be satisfied for all cycles.

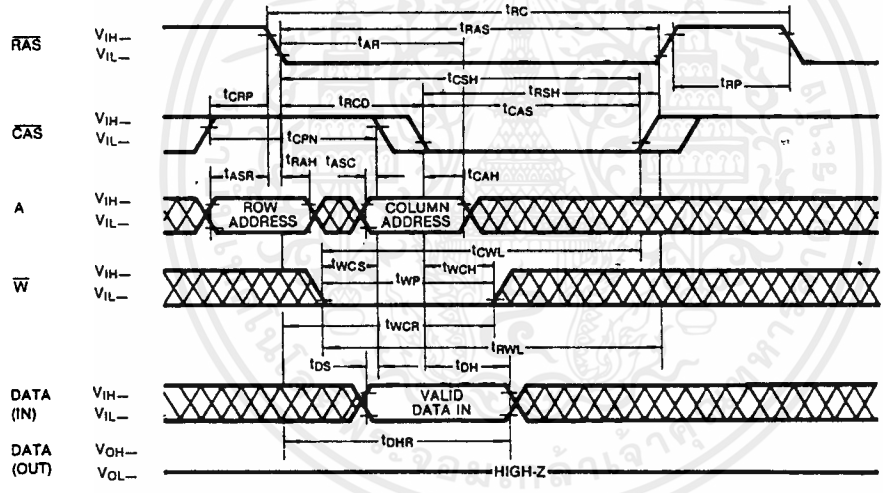
TIMING DIAGRAMS

READ CYCLE



WRITE CYCLE (EARLY WRITE)

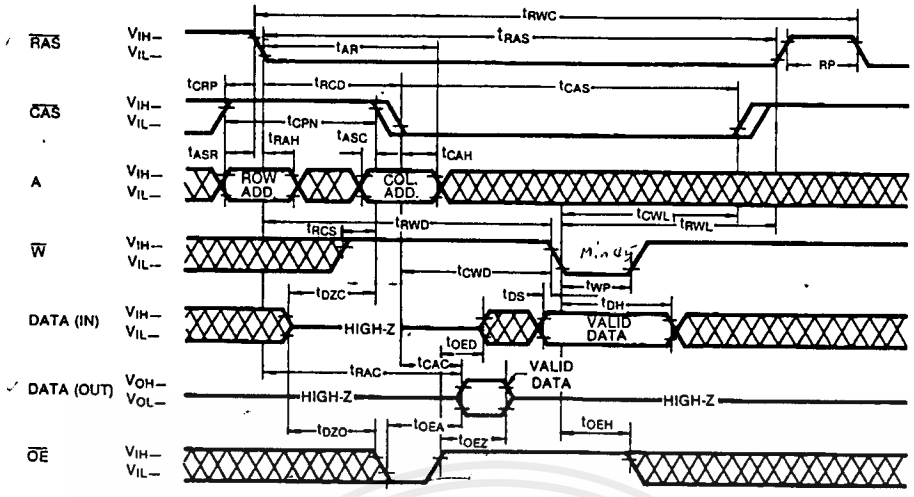
$\overline{OE}$ = Don't Care



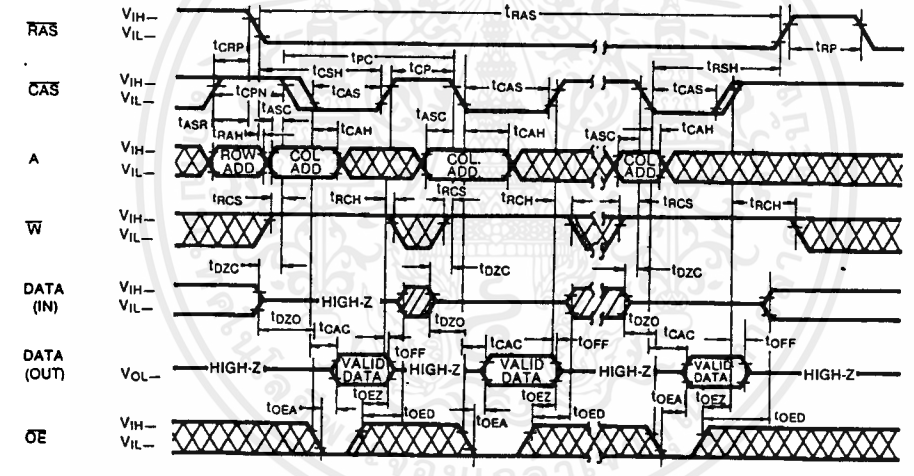
DON'T CARE

TIMING DIAGRAMS (Continued)

READ-WRITE/READ-MODIFY-WRITE CYCLE



PAGE MODE READ CYCLE

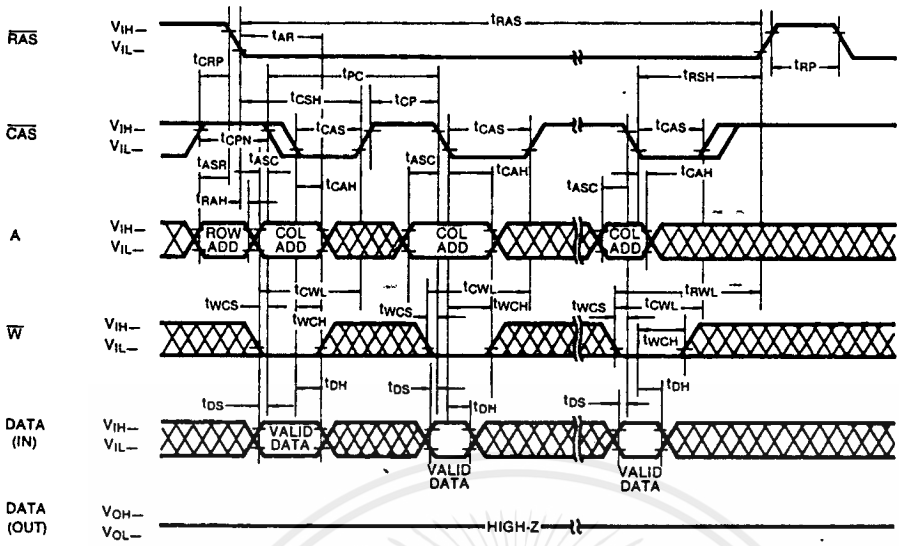


DON'T CARE

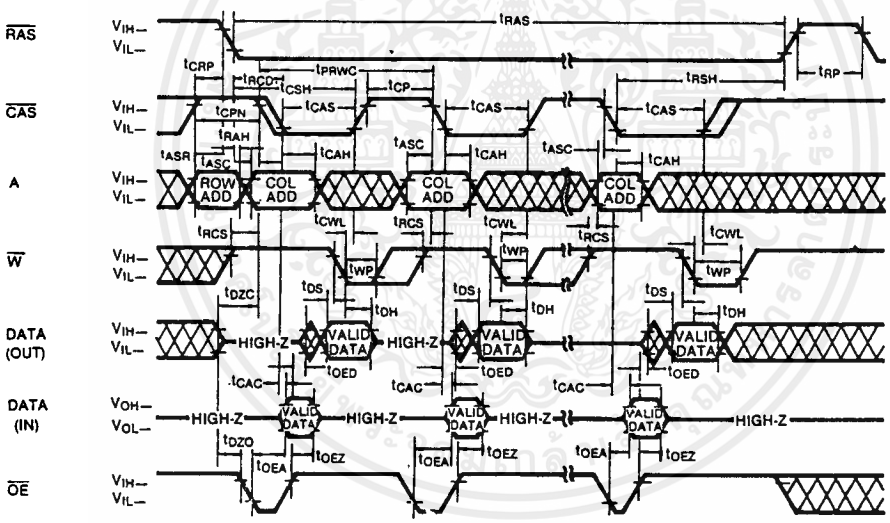
TIMING DIAGRAMS (Continued)

PAGE MODE WRITE CYCLE

$\overline{OE}$ = Don't Care



PAGE MODE READ-MODIFY-WRITE CYCLE

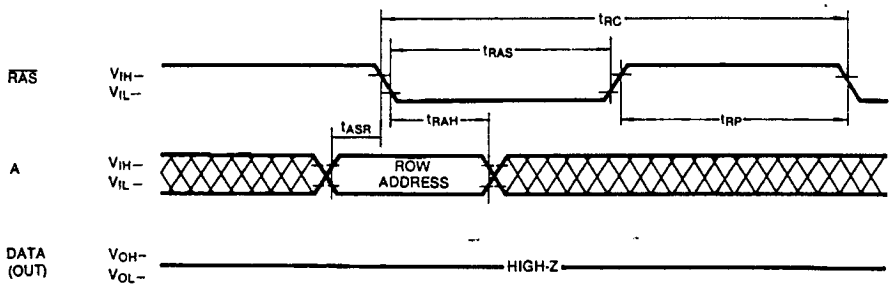


DON'T CARE

TIMING DIAGRAMS (Continued)

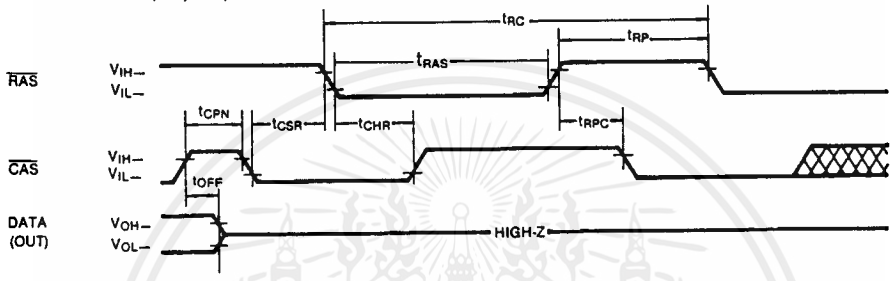
$\overline{RAS}$-ONLY REFRESH CYCLE

NOTE: $\overline{CAS} = V_{IH}$; $\overline{W}$, $\overline{OE}$, D = Don't Care

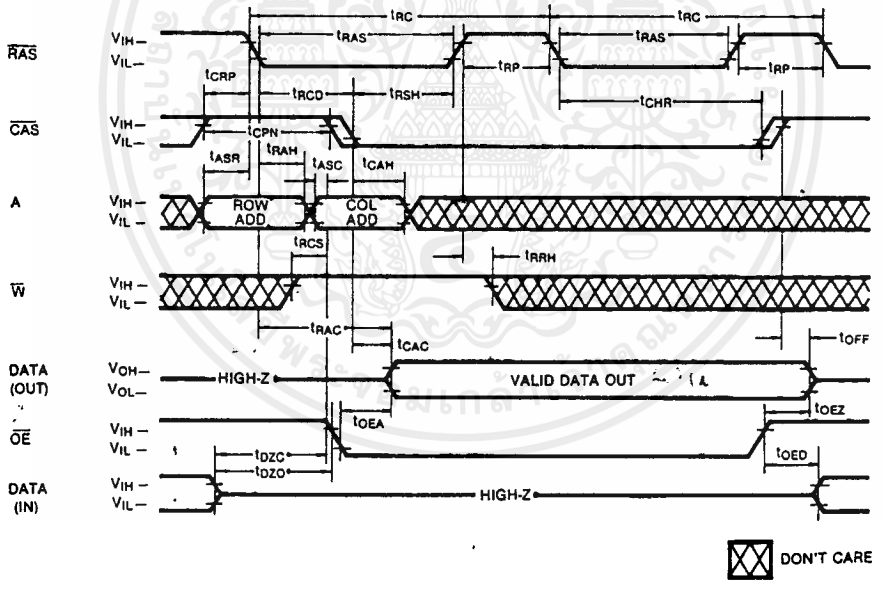


$\overline{CAS}$-BEFORE-$\overline{RAS}$ REFRESH CYCLE

NOTE: Address, $\overline{W}$, $\overline{OE}$, D = Don't Care



HIDDEN REFRESH CYCLE



KM41464A OPERATION

Device Operation

The KM41464A contains 262,144 memory locations organized as 65,536 × 4-bit words. Sixteen address bits are required to address a particular 4-bit word in the memory array. Since the KM41464A has only 8 address input pins, time multiplexed addressing is used to input 8 row and 8 column addresses. The multiplexing is controlled by the timing relationship between the row address strobe ($\overline{RAS}$), the column address strobe ($\overline{CAS}$) and the valid address inputs.

Operation of the KM41464A begins by strobing in a valid row address with $\overline{RAS}$ while $\overline{CAS}$ remains high. Then the address on the 8 address input pins is changed from a row address to a column address and is strobed in by $\overline{CAS}$. This is the beginning of any KM41464A cycle in which a memory location is accessed. The specific type of cycle is determined by the state of the write enable pin and various timing relationships. The cycle is terminated when both $\overline{RAS}$ and $\overline{CAS}$ have returned to the high state. Another cycle can be initiated after $\overline{RAS}$ remains high long enough to satisfy the $\overline{RAS}$ precharge time (t_{RP}) requirement.

$\overline{RAS}$ and $\overline{CAS}$ Timing

The minimum $\overline{RAS}$ and $\overline{CAS}$ pulse width are specified by t_{RAS}(min) and t_{CAS}(min) respectively. These minimum pulse widths must be satisfied for proper device operation and data integrity. Once a cycle is initiated by bringing $\overline{RAS}$ low, it must not be aborted prior to satisfying the minimum $\overline{RAS}$ and $\overline{CAS}$ pulse widths. In addition, a new cycle must not begin until the minimum $\overline{RAS}$ precharge time, t_{RP}, has been satisfied. Once a cycle begins, internal clocks and other circuits within the KM41464A begin a complex sequence of events. If the sequence is broken by violating minimum timing requirements, loss of data integrity can occur.

Read

A read cycle is achieved by maintaining the write enable input ($\overline{W}$) high during a $\overline{RAS}$/$\overline{CAS}$ cycle. The four outputs of the KM41464A remains in the HI-Z state until valid data appears at the output. The KM41464A has common data I/O pins. For this reason an output enable control input ($\overline{OE}$) has been provided so the output buffer can be precisely controlled. For data to appear at the outputs, $\overline{OE}$ must be low for the period of time defined by t_{OEA} and t_{OEZ}. If $\overline{CAS}$ goes low before t_{RCD}(max), the access time to valid data is specified by t_{RAC}. If $\overline{CAS}$ goes low after t_{RCD}(max), the access time is measured from $\overline{CAS}$ and is specified by t_{CAC}. In order to achieve the minimum access time, t_{RAC}(min), it is necessary to bring $\overline{CAS}$ low before t_{RCD}(max).

Write

The KM41464A can perform early write, and read-modify-write cycles. The difference between these cycles is in the state of data-out and is determined by the timing relationship between $\overline{W}$, $\overline{OE}$ and $\overline{CAS}$. In any type of write cycle, Data-in must be valid at or before the falling edge of $\overline{W}$ or $\overline{CAS}$, whichever is later.

Early Write: An early write cycle is performed by bringing $\overline{W}$ low before $\overline{CAS}$. The 4-bit wide data at the data input pins is written into the addressed memory cells. Throughout the early write cycle the outputs remain in the Hi-Z state regardless of the state of the $\overline{OE}$ input.

Read-Modify-Write: In this cycle, valid data from the addressed cell appears at the outputs before and during the time that data is being written into the same cell locations. This cycle is achieved by bringing $\overline{W}$ low after $\overline{CAS}$ and meeting the data sheet read-modify-write timing requirements. The output enable input ($\overline{OE}$) must be low during the time defined by t_{OEA} and t_{OEZ} for data to appear at the outputs. If t_{CWD} and t_{RWD} are not met the output may contain invalid data. Conforming to the $\overline{OE}$ timing requirements prevents bus contention on the KM41464's DQ pins.

Data Output

The KM41464A has tri-state output buffers which are controlled by $\overline{CAS}$ and $\overline{OE}$. When either $\overline{CAS}$ or $\overline{OE}$ is high (V_{IH}), the output are in the high impedance (Hi-Z) state. In any cycle in which valid data appears at the outputs, the outputs first remains in the Hi-Z state until the data is valid and then the valid data appears at the outputs. The valid data remains at the outputs until $\overline{CAS}$ or $\overline{OE}$ returns high. This is true even if a new $\overline{RAS}$ cycle occurs (as in hidden refresh). Each of the KM41464A operating cycles are listed below after the corresponding output state produced by the cycle.

Valid Output Data: Read, Read-Modify-Write, Hidden Refresh, Page Mode Read, Page Mode Read-Modify-Write.

Hi-Z Output State: Early Write, $\overline{RAS}$-only Refresh, Page Mode write, $\overline{CAS}$-only cycle.

Indeterminate Output State: Delayed Write (t_{CWD} or t_{RWD} are not met)

Refresh

The data in the KM41464A is stored on a tiny capacitor within each memory cell. Due to leakage, the data will leak off after a period of time. To maintain data integrity it is necessary to refresh each of the rows every 4 ms. There are several ways to accomplish this.

$\overline{RAS}$-Only Refresh: This is the most common method

KM41464A OPERATION (Continued)

for performing refresh. It is performed by strobing in a row address with $\overline{RAS}$ while $\overline{CAS}$ remains high. This must be performed on each of the 256 row addresses (A_0-A_7) every 4ms.

$\overline{CAS}$-Before-$\overline{RAS}$ Refresh: The KM41464A has $\overline{CAS}$-before-$\overline{RAS}$ on-chip refreshing capability that eliminates the need for external refresh addresses. If $\overline{CAS}$ is held low for the specified set up time (t_{CSR}) before $\overline{RAS}$ goes low, the on-chip refresh circuitry is enabled. An internal refresh operation automatically occurs and the on-chip refresh address counter is internally incremented in preparation for the next $\overline{CAS}$-before-$\overline{RAS}$ refresh cycle.

Hidden Refresh: A hidden refresh cycle may be performed while maintaining the latest valid data at the output by extending the $\overline{CAS}$ active time and cycling $\overline{RAS}$. The KM41464A hidden refresh cycle is actually a $\overline{CAS}$-before-$\overline{RAS}$ refresh cycle within an extended read cycle. The refresh row address is provided by the on-chip refresh address counter. This eliminates the need for the external row address that is required in hidden refresh cycles by DRAMS that do not have $\overline{CAS}$-before-$\overline{RAS}$ refresh capability.

Other Refresh Methods: It is also possible to refresh the KM41464A by using read, write or read-modify-write cycles. Whenever a row is accessed, all the cells in that row are automatically refreshed. There are certain applications in which it might be advantageous to perform refresh in this manner but in general $\overline{RAS}$-only or $\overline{CAS}$-before-$\overline{RAS}$ refresh are the preferred methods.

Page Mode

Page mode memory cycles provide faster access and lower power dissipaton than normal memory cycles. In page mode, it is possible to perform read, write or read-modify-write cycles. As long as the applicable timing requirements are observed, it is possible to mix these cycles in any order. A page mode cycle begins with a normal cycle. While $\overline{RAS}$ is kept low to maintain the row address, $\overline{CAS}$ is cycled to strobe in additional column addresses. This eliminates the time required to set up and strobe sequential row addresses for the same page.

Power-up

If $\overline{RAS}$=V_{SS} during power-up the KM41464A might begin an active cycle. This condition results in higher than necessary current demands from the power supply during power-up. It is recommended that $\overline{RAS}$ and $\overline{CAS}$ track with V_{CC} during power-up or be held at a valid V_{IH} in order to minimize the power-up current.

An initial pause of 100μsec is required after power-up followed by 8 initialization cycles before proper device operation is assured. Eight initialization cycles are also required after any 4 msec period in which there are no $\overline{RAS}$ cycles. An initialization cycle is any cycle in which $\overline{RAS}$ is cycled.

Termination

The lines from the TTL driver circuits to the KM41464A inputs act like unterminated transmission lines resulting in significant positive and negative overshoots at the inputs. To minimize overshoot it is advisable to terminate the input lines and to keep them as short as possible. Although either series or parallel termination may be used, series termination is generally recommended since it is simple and draws no additional power. It consists of a resistor in series with the input line placed close to the KM41464A input pin. The optimum value depends on the board layout. It must be determined experimentally and is usually in the range of 20 to 40 ohms.

Board Layout

It is important to lay out the power and ground lines on memory boards in such a way that switching transient effects are minimized. The recommended methods are gridded power and ground lines or separate power and ground planes. The power and ground lines act like transmission lines to the high frequency transients generated by DRAMS. The impedance is minimized if all the power supply traces to all the DRAMS run both horizontally and vertically and are connected at each intersection or better yet if power and ground planes are used.

Address and control lines should be as short as possible to avoid skew. In boards with many DRAMS these lines should fan out from a central point like a fork or comb rather than being connected in a serpentine pattern. Also the control logic should be centrally located on large memory boards to facilitate the shortest possible address and control lines to all the DRAMs.

Decoupling

The importance of proper decoupling cannot be over emphasized. Excessive transient noise or voltage droop on the V_{CC} line can cause loss of data integrity (soft errors). The total combined voltage changes over time in the V_{CC} to V_{SS} voltage (measured at the device pins) should not exceed 500mV.

A high frequency 0.3μF ceramic decoupling capacitor should be connected between the V_{CC} and ground pins of each KM41464A using the shortest possible traces.

KM41464A OPERATION (Continued)

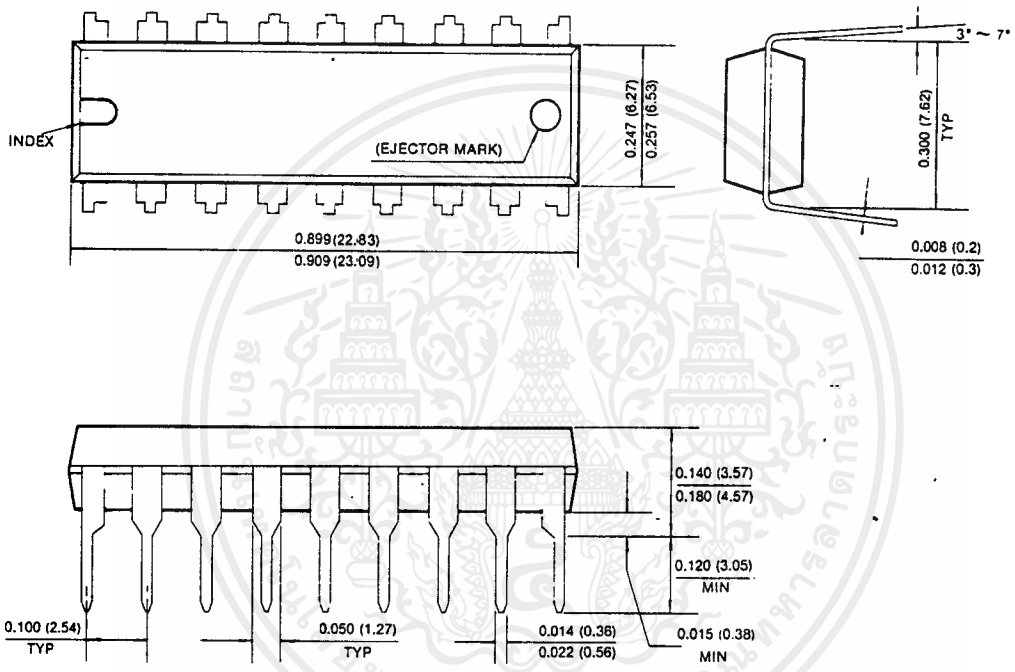
These capacitors act as a low impedance shunt for the high frequency switching transients generated by the KM41464A and they supply much of the current used by the KM41464A during cycling.

In addition, a large tantalum capacitor with a value of 47μF to 100μF should be used for bulk decoupling to recharge the 0.3μF capacitors between cycles, thereby reducing power line droop. The bulk decoupling capacitor should be placed near the point where the power traces meet the power grid or power plane. Even better results may be achieved by distributing more than one tantalum capacitor around the memory array.

PACKAGE DIMENSIONS

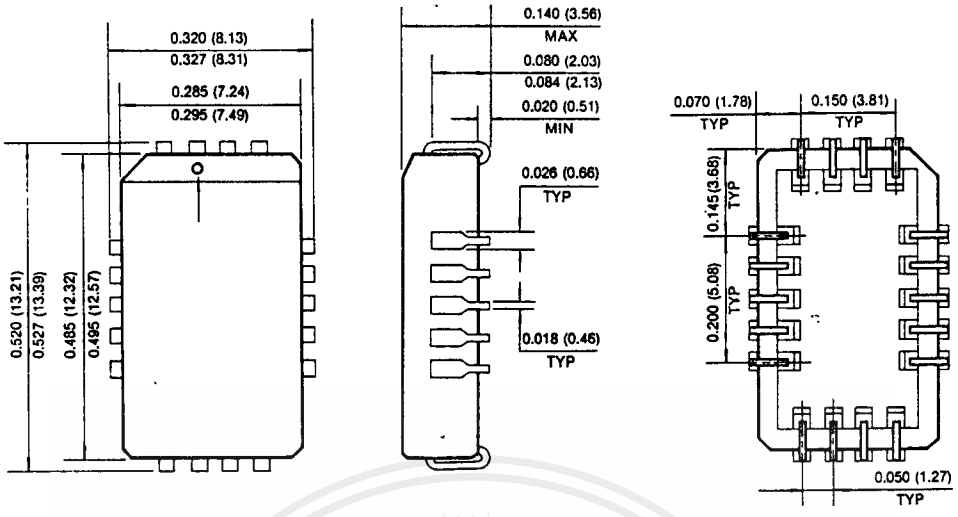
18-LEAD PLASTIC DUAL IN-LINE PACKAGE

Units: Inches (millimeters)



PACKAGE DIMENSIONS (Continued)

18-PIN PLASTIC LEADED CHIP CARRIER



20-PIN PLASTIC ZIGZAG-IN-LINE PACKAGE