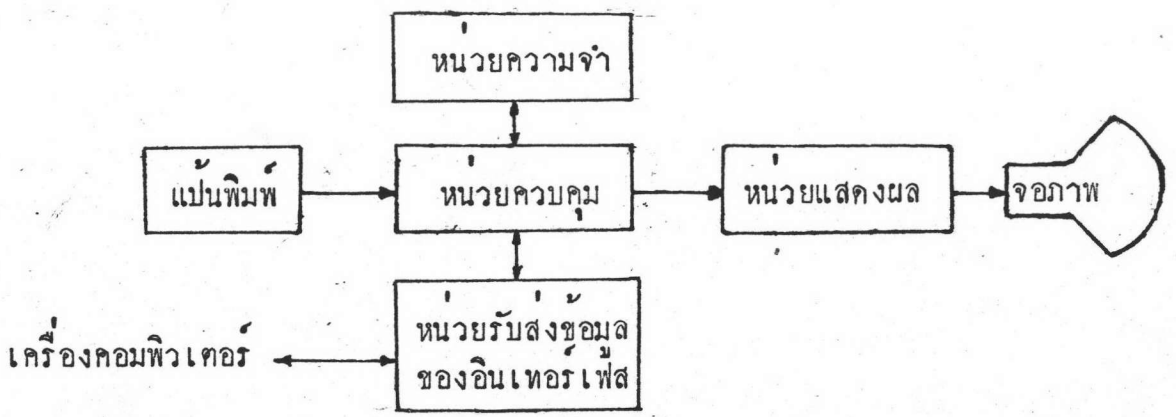
หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

ระบบควบคุมข้อมูลของเทอร์มินอล

2.1 หลักการเบื้องต้นของเทอร์มินอล

เทอร์มินอลเป็นอุปกรณ์ไอโอ (I/O Device) ของเครื่องคอมพิวเตอร์ ใช้สำหรับป้อนข้อมูลหรือโปรแกรมเข้าทางแป้นพิมพ์ เมื่อกดปุ่มบนแป้นพิมพ์ข้อมูลจะถูกส่งไปยังเครื่องคอมพิวเตอร์ด้วยรหัส ASC II (American Standard Code For Information Interchange) นอกจากนี้เทอร์มินอลยังเป็นส่วนแสดงผลของเครื่องคอมพิวเตอร์โดยการรับข้อมูลจากเครื่องคอมพิวเตอร์ เพื่อแสดงผลของข้อมูลด้วยภาพบนหน้าจอภาพ ภายในเทอร์มินอลประกอบด้วยส่วนต่าง ๆ ดังรูปที่ 2.1



รูปที่ 2.1 แสดงส่วนประกอบภายในเทอร์มินอล

2.1.1 แป้นพิมพ์ ทำหน้าที่สร้างรหัสข้อมูลต่าง ๆ ตามอักขระและการควบคุมบนแป้นพิมพ์ เพื่อป้อนให้กับหน่วยควบคุมสำหรับนำเข้าเก็บในหน่วยความจำ หรือส่งให้กับหน่วยรับส่งข้อมูลของอินเทอร์เฟสสำหรับส่งต่อไปยังเครื่องคอมพิวเตอร์

2.1.2 หน่วยความจำ ทำหน้าที่เก็บรหัสข้อมูลที่ป้อนเข้าทางแป้นพิมพ์ หรือรหัสข้อมูลที่ป้อนเข้าทางหน่วยรับส่งข้อมูลของอินเทอร์เฟส เพื่อเก็บไว้ให้หน่วยแสดงผลนำไปแสดงผลบนจอภาพ

2.1.3 หน่วยแสดงผล ทำหน้าที่สร้างตัวอักขระให้ปรากฏบนจอภาพ โดยรับรหัสข้อมูลมาจากหน่วยความจำและเปลี่ยนเป็นสัญญาณภาพตามรหัสข้อมูลที่อยู่ในหน่วยความจำ

2.1.4 หน่วยรับส่งข้อมูลของอินเทอร์เฟส ทำหน้าที่เชื่อมโยงเทอร์มินอลกับเครื่องคอมพิวเตอร์และจัดการเกี่ยวกับการรับส่งข้อมูลระหว่างเทอร์มินอลกับเครื่องคอมพิวเตอร์

2.1.5 หน่วยควบคุม ทำหน้าที่สร้างสัญญาณควบคุมต่าง ๆ ภายในเทอร์มินอลเพื่อควบคุมข้อมูลระหว่างแป้นพิมพ์กับหน่วยความจำและข้อมูลระหว่างหน่วยความจำกับหน่วยรับส่งข้อมูลของอินเทอร์เฟส ตลอดจนการนำข้อมูลภายในหน่วยความจำไปแสดงผล

การวิจัยนี้จะเน้นเฉพาะการออกแบบระบบควบคุมข้อมูล ซึ่งจะทำงานประสานกันระหว่างส่วนวงจรควบคุม ส่วนวงจรหน่วยความจำ ส่วนวงจรแป้นพิมพ์ ส่วนวงจรอินเทอร์เฟส และส่วนวงจรแสดงผล

2.2 หลักการเบื้องต้นของระบบควบคุมข้อมูล

ระบบควบคุมข้อมูล ทำหน้าที่จัดการข้อมูลที่เป็นรหัส ASC II ที่รับจากวงจรแป้นพิมพ์หรือจากวงจรอินเทอร์เฟส เพื่อนำข้อมูลเหล่านี้เข้าเก็บหรือนำออกจากวงจรหน่วยความจำโดยการควบคุมของวงจรควบคุม ข้อมูลที่เป็นรหัส ASC II แบ่งออกเป็น 2 ประเภท ประเภทแรก คือรหัสอักขระที่นำไปแสดงผลบนจอภาพ ข้อมูลประเภทที่สอง คือรหัสควบคุมใช้เพื่อจัดตำแหน่งรหัสอักขระประเภทแรกในหน่วยความจำและเป็นรหัสควบคุมเครื่องคอมพิวเตอร์ตามความต้องการของผู้ใช้

2.3 ลักษณะการทำงานระบบควบคุมข้อมูล

ส่วนประกอบภายในระบบควบคุมข้อมูลประกอบด้วยส่วนต่าง ๆ คือ (ดูรูปที่ 2.2)

2.3.1 วงจรแป้นพิมพ์

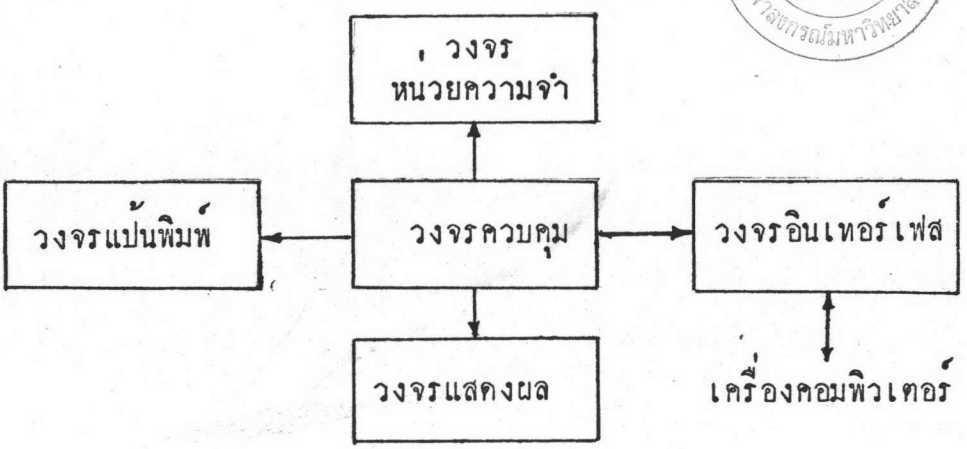
2.3.2 วงจรหน่วยความจำ

2.3.3 วงจรควบคุม

2.3.4 วงจรอินเทอร์เฟส

2.3.5 วงจรแสดงผล

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 2.2 แสดงส่วนประกอบภายในวงจรระบบควบคุมข้อมูล

2.4 การทำงานของวงจรแป้นพิมพ์

วงจรแป้นพิมพ์เป็นส่วนที่สร้างรหัสอักขระและรหัสควบคุม รหัสที่สร้างเป็นรหัส ASC II 7 บิท (Bit) เมื่อกดปุ่มบนแป้นพิมพ์วงจรภายในจะส่งรหัสข้อมูลออกเพื่อป้อนเข้าสู่วงจรควบคุม ปุ่มกดข้อมูลบนแป้นพิมพ์จะมีปุ่มกดอยู่ 2 ประเภทคือ

2.4.1 ปุ่มกดอักขระในภาษาอังกฤษ ได้แก่ปุ่มกดตัวอักษร A ถึง Z ปุ่มกดตัวเลข 0 ถึง 9 และปุ่มกดอักขระพิเศษ เพื่อกดปุ่มบนแป้นพิมพ์สัญญาณรหัส ข้อมูลจะถูกส่งไปยังวงจรหน่วยควบคุม เพื่อให้วงจรควบคุมนำข้อมูลเหล่านั้นเข้าเก็บยังวงจรหน่วยความจำ หรือส่งข้อมูลภายนอกเทอร์มินอลว่าเป็นโมด (Mode) ใด

2.4.2 ปุ่มกดควบคุม ได้แก่ปุ่มกด HOME ปุ่มกด CLEAR ฯลฯ ปุ่มกดควบคุมเหล่านี้จะมีมากน้อยขึ้นอยู่กับชนิดของเทอร์มินอล เมื่อมีการกดปุ่มเหล่านี้สัญญาณรหัสควบคุม จะถูกส่งไปยังวงจรหน่วยควบคุม เพื่อใช้จัดตำแหน่งข้อมูลที่เป็นอักขระในภาษาอังกฤษ ในวงจรหน่วยความจำหรือส่งไปยังวงจรอินเทอร์เฟส ซึ่งขึ้นอยู่กับการรับส่งข้อมูลของเทอร์มินอลเป็นโมดใด

2.5 หน้าที่และการทำงานของปุ่มกดควบคุม

ปุ่มกดควบคุมของเทอร์มินอล ประกอบด้วยปุ่มกดควบคุมต่าง ๆ ดังนี้

- ปุ่มกด CLEAR ใช้ลบข้อมูลที่แสดงบนหน้าจอภาพทั้งหมดโดยตำแหน่งของเคอร์เซอร์ (Cursor) จะถูกย้ายมาที่ตำแหน่งเริ่มต้นของบรรทัดแรก

- ปุ่มกด HOME ใช้สำหรับย้ายตำแหน่งของเคอร์เซอร์ไปตำแหน่งเริ่มต้น ของบรรทัดแรก โดยข้อมูลที่แสดงบนจอภาพจะไม่ถูกลบ

- ปุ่มกด ENTER ใช้สำหรับย้ายตำแหน่งเคอร์เซอร์ตำแหน่งเริ่มต้นของบรรทัด ถ้าตำแหน่งเคอร์เซอร์อยู่ที่ตำแหน่งแรกของบรรทัด กดปุ่ม ENTER จะไม่มีผลถ้าวิธีการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์กับเทอร์มินอลเป็นแบบฟูลดูเพล็กก็จะส่งรหัสของ ENTER ไปยังเครื่องคอมพิวเตอร์

- ปุ่มกด → ใช้สำหรับเลื่อนตำแหน่งเคอร์เซอร์ไปทางขวาของหน้าจอภาพอีกหนึ่งตัวอักษร โดยไม่มีผลกับข้อมูลที่แสดงบนจอภาพ

- ปุ่มกด ← ใช้สำหรับเลื่อนตำแหน่งเคอร์เซอร์ไปทางซ้ายของหน้าจอภาพอีกหนึ่งตัวอักษร โดยไม่มีผลกับข้อมูลที่แสดงบนจอภาพ

- ปุ่มกด ↓ ใช้สำหรับเลื่อนตำแหน่งเคอร์เซอร์ไปยังบรรทัดต่อไปหนึ่งบรรทัดโดยไม่มีผลกับข้อมูลที่แสดงบนจอภาพ

- ปุ่มกด ↑ ใช้สำหรับเลื่อนตำแหน่งเคอร์เซอร์ถอยหลังหนึ่งบรรทัด โดยไม่มีผลกับข้อมูลที่แสดงบนจอภาพ

- ปุ่มกด TAB H ใช้สำหรับส่งรหัส TAB H ไปยังวงจรอินเทอร์เฟส เพื่อส่งไปยังเครื่องคอมพิวเตอร์

- ปุ่มกด ESC ใช้สำหรับส่งรหัส ESC ไปยังวงจรอินเทอร์เฟส เพื่อส่งไปยังเครื่องคอมพิวเตอร์

- ปุ่มกด DEL ใช้สำหรับส่งรหัส DEL ไปยังวงจรอินเทอร์เฟส เพื่อส่งไปยังเครื่องคอมพิวเตอร์

2.6 ตารางรหัส ASC II 7 บิท

ตารางที่ 2.1 แสดงรหัส ASC II ขนาด 7 บิท ที่ใช้ภายในระบบควบคุมข้อมูลของเทอร์มินอลและรหัสของแป้นพิมพ์

Bit Positions 1,2,3,4	Bit Positions 5,6,7							
	000	100	010	110	001	101	011	111
0 0 0 0	NUL	DLE	SPACE	0	@	P		p
1 0 0 0	SOH	DC1	!	1	A	Q	a	q
0 1 0 0	STX	DC2	"	2	B	R	b	r
1 1 0 0	ETX	DC3	#	3	C	S	c	s
0 0 1 0	EOT	DC4	$	4	D	T	d	t
1 0 1 0	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	ACK	SYN	&	6	F	V	f	v
1 1 1 0	BEL	ETB	'	7	G	W	g	w
0 0 0 1	BS	CAN	(	8	H	X	h	x
1 0 0 1	HT	EM	)	9	I	Y	i	y
0 1 0 1	LF	SUB	*	:	J	Z	j	z
1 1 0 1	VT	ESC	+	;	K	[	k	{
0 0 1 1	FF	FS	⌐	<	L	\	l	|
1 0 1 1	CR	GS	-	=	M	]	m	}
0 1 1 1	SO	RS	.	>	N	^	n	~
1 1 1 1	SI	US	/	?	O	-	o	DEL

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

NUL	=	All zeroes	DC1	=	Device control #1
SOH	=	Start of header	DC2	=	Device control #2
STX	=	Start of text	DC3	=	Device control #3
ETX	=	End of text	DC4	=	Device control #4
EOT	=	End of transmission	NAK	=	Negative acknowledgement
ENQ	=	Inquiry	SYN	=	Synchronous idle
ACK	=	Acknowledgement	ETB	=	End transmitted block
BEL	=	Bell	CAN	=	Cancel
BS	=	Back space	EM	=	End of medium
HT	=	Horizontal tab	SUB	=	Start special sequence
LF	=	Line feed	ESC	=	Escape or break
VT	=	Vertical tab	FS	=	File separator
FF	=	Form feed	GS	=	Group separator
CR	=	Carriage return	RS	=	Record separator
SO	=	Shift out	US	=	Unit separator
SI	=	Shift in	DEL	=	Delete
DLE	=	Data link escape			

ตารางที่ 2.1 แสดงรหัส ASC II ขนาด 7 บิท

2.7 การทำงานของวงจรหน่วยความจำ

วงจรหน่วยความจำทำหน้าที่เก็บรหัสอักขระที่ส่งจากวงจรแป้นพิมพ์และวงจรอินเทอร์-เฟสเพื่อนำไปแสดงผลบนจอภาพและเก็บโปรแกรมโมนิเตอร์ (Monitor Program) ซึ่งเป็นโปรแกรมที่ทำหน้าที่ควบคุมการทำงานของวงจรควบคุม วงจรหน่วยความจำ แบ่งออกเป็น 2 ประเภทตามลักษณะการใช้งานคือ รอม (ROM ย่อมาจาก Read Only Memory) และแรม (RAM ย่อมาจาก Rondom Access Memory)

รอมเป็นหน่วยความจำที่มีไว้อ่าน (Read) ข้อมูลได้อย่างเดียว ไม่สามารถป้อนข้อมูลหรือโปรแกรมเข้าไปในหน่วยความจำชนิดนี้ได้ ดังนั้นจึงมักใช้เป็นที่เก็บโปรแกรมโมนิเตอร์ ซึ่งผู้ผลิตจะบรรจุโปรแกรมถาวรไว้ในหน่วยความจำชนิดนี้เลยจากโรงงาน ผู้ใช้ไม่สามารถจะบรรจุโปรแกรมใหม่เข้าไปได้ จึงมีวิวัฒนาการหน่วยความจำนี้ให้โปรแกรมและลบได้ทำให้ใช้สะดวกขึ้น แต่วิธีการลบโปรแกรมจะต้องทำโดยการถอดหน่วยความจำส่วนนี้มาลบด้วยแสงอุลตราไวโอเรด (Ultraviolet) เพื่อลบโปรแกรมที่มีอยู่ก่อนแล้วจึงจะโปรแกรมใหม่ ได้หน่วยความจำชนิดนี้เรียกว่าอีพรอม (EPROM ย่อมาจาก Erasable Programable Read Only Memory) สำหรับการใช้งานจะเป็นหน่วยความจำที่เก็บโปรแกรมโมนิเตอร์ เพราะคุณสมบัติของรอม และอีพรอมเวลาปิดไฟข้อมูลที่เก็บอยู่ภายในจะไม่ถูกลบหายไป โปรแกรมโมนิเตอร์นี้ จะต้องเก็บอยู่ในหน่วยความจำตลอดเวลาเพราะเมื่อเปิดเครื่องสามารถทำงานได้ทันที โดยไม่ต้องเสียเวลาโหลด (Load) เข้าไปใหม่

แรมเป็นหน่วยความจำที่สามารถอ่านและเขียนข้อมูลได้ ถ้ามีการป้อนข้อมูลใหม่ เข้าไปในหน่วยความจำชนิดนี้แล้วก็จะไปลบข้อมูลเดิมที่เก็บอยู่ ดังนั้นจึงใช้เป็นที่เก็บข้อมูล ที่มีการเปลี่ยนแปลงบ่อย ๆ ในหน่วยความจำแรม จึงใช้เก็บข้อมูล ส่วนที่เป็นรหัสอักขระสำหรับนำไปแสดงผลบนจอภาพ คุณสมบัติของแรมเมื่อปิดไฟข้อมูลที่เก็บอยู่ภายในจะถูกลบหายไป แรมที่ใช้กันอยู่มี 2 ชนิดคือ สแตติคแรม (Static RAM) และไดนามิคแรม (Dynamic RAM) การใช้งานส่วนมากใช้สแตติคแรม เพราะง่ายต่อการออกแบบวงจร

อีพรอมที่ใช้กันมากคืออีพรอมเบอร์ 2708 และแรมที่ใช้กันมากคือแรมเบอร์ 2114 ซึ่งมีข้อดีคือใช้ไฟตรง 5 โวลท์เท่านั้น

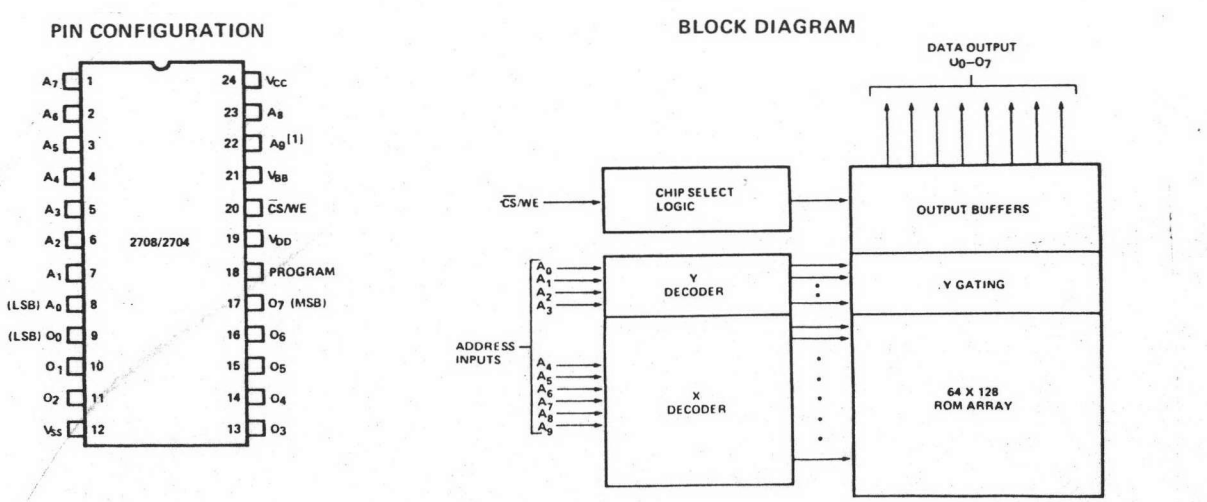
2.7.1 ลักษณะการทำงานของอีพรอมเบอร์ 2708

อีพรอมที่ใช้กันมากคือเบอร์ 2708 มีขนาด 1024 $\times$ 8 บิท (ดูรูป 2.3) เป็นไอซีที่มีขาแอดเดรส (Address) 10 ขาคือขา A_0- A_9 และมีขาข้อมูล 8 ขาคือ O_0- O_7 ซึ่งเป็นขาสำหรับนำข้อมูลเข้าหรือออก ข้อมูลที่จะถูกเขียน (Write) หรืออ่าน (Read) โดยควบคุมด้วยขา $\overline{CS}/WE$ เป็นขาที่ควบคุมสภาวะของอีพรอมว่าอยู่ในสภาวะไทสเตท(Tri-State) หรือไม่ การใช้งานของอีพรอมเบอร์นี้สามารถเลือกใช้ได้ 3 โมด

- โมดแรกคือ ดีซีเลคท์โมด (Deselect Mode) เป็นโมดที่ทำให้ขาข้อมูล O_0- O_7 อยู่ในสภาวะลอย (High Inspedance) โดยไม่คำนึงว่าขาแอดเดรส A_0- A_9 จะเป็นอะไร ทำให้สามารถต่อรวมกับหน่วยความจำอื่น ๆ ได้ หรือในกรณีต้องการเพิ่มขนาดของหน่วยความจำให้มากขึ้น การทำงานในโมดนี้ควบคุมที่ขา $\overline{CS}$/WE ให้มีศักดา V_{IH}

- โมดสองคือโมดอ่าน (Read Mode) เป็นโมดที่ใช้สำหรับการอ่านโปรแกรมออกทางขาข้อมูล O_0- O_7 ตำแหน่งข้อมูลในอีพรอมที่จะอ่านออกมากำหนดด้วยแอดเดรส A_0- A_9 การทำงานในโมดนี้ควบคุมที่ขา $\overline{CS}$/WE ให้มีศักดา V_{IL}

- โมดสามคือโปรแกรมโมด (Program Mode) ในโมดนี้ใช้สำหรับเขียนข้อมูล เข้าอีพรอมโดยผ่านขาข้อมูล O_0- O_7 ตำแหน่งข้อมูลที่จะเขียนลงอีพรอมต้องกำหนดด้วยแอดเดรส A_0- A_9 การเขียนข้อมูลลงอีพรอมจะต้องใช้เครื่องอีพรอมโปรแกรมเมอร์ (EPROM Programer) เท่านั้น สำหรับการเลือกใช้โมดต่าง ๆ ของอีพรอมเบอร์ 2708 ดูจากตารางที่ 2.2 รายละเอียดต่าง ๆ ของอีพรอมเบอร์ 2708 ดูจากภาคผนวก ค



รูปที่ 2.3 แสดงรูปร่างและส่วนประกอบภายในอีพรอมเบอร์ 2708

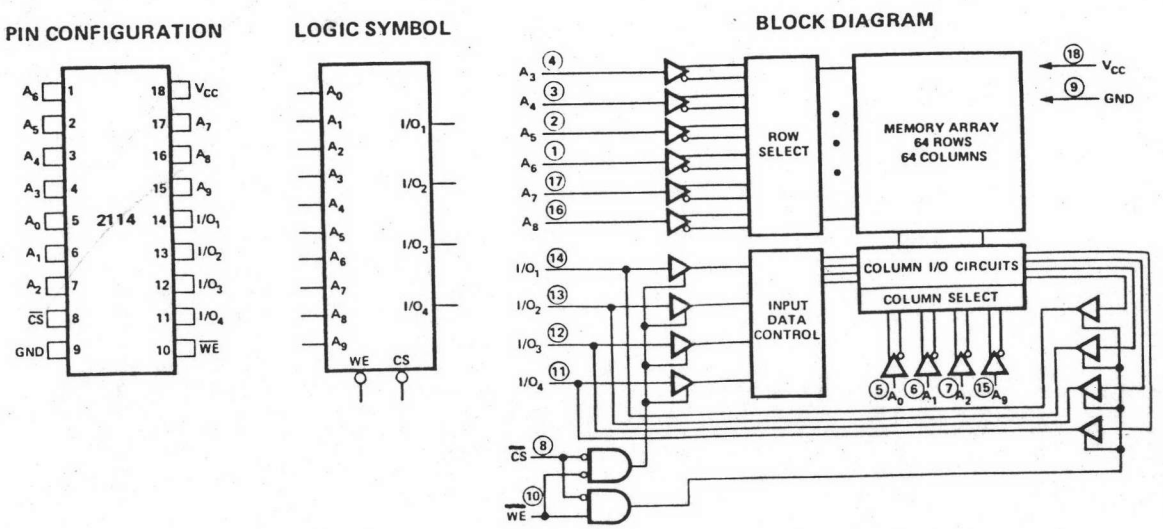
PIN CONNECTION DURING READ OR PROGRAM

	PIN NUMBER							
MODE	DATA I/O 9-11, 13-17	ADDRESS INPUTS 1-8, 22, 23	V_{SS} 12	PROGRAM 18	V_{DD} 19	$\overline{CS}$/WE 20	V_{BB} 21	V_{CC} 24
READ	D_{OUT}	A_{IN}	GND	GND	+12	V_{IL}	-5	+5
DESELECT	HIGH IMPEDANCE	DON'T CARE	GND	GND	+12	V_{IH}	-5	+5
PROGRAM	D_{IN}	A_{IN}	GND	PULSED 26V	+12	V_{IHW}	-5	+5

ตารางที่ 2.2 แสดงตารางการเลือกโมดของอีพรอมเบอร์ 2708

2.7.2 ลักษณะการทำงานของแรมเบอร์ 2114

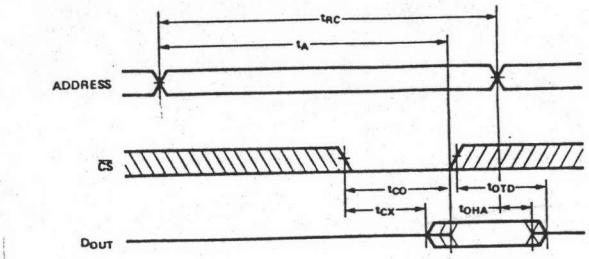
แรมที่นิยมใช้กันมากคือเบอร์ 2114 เป็นแรมชนิดสแตติคแรมขนาด 1024x4 บิท (ดูรูป 2.4) มีขาแอดเดรส 10 ขาคือขา $A_0 - A_9$ เป็นขาสัญญาณที่ใช้กำหนดตำแหน่งของข้อมูลภายในแรมสำหรับการอ่านและเขียนข้อมูล เนื่องจากขาแอดเดรสมี 10 ขา (10 บิท) จึงสามารถกำหนดตำแหน่งภายในแรมได้ถึง 1024 ตำแหน่ง ขาข้อมูลมี 4 ขาคือ I/O_1-I/O_4 เป็นที่รับข้อมูลเข้าและออกจากแรม สำหรับขา $\overline{CS}$ (Chip Select) ทำหน้าที่เอนเอเบิล (Enable) วงจร ประโยชน์ของขาสัญญาณนี้ยังใช้ในกรณีต้องการขยายขนาดหน่วยความจำและขา $\overline{WE}$ เป็นขาควบคุมการอ่านและเขียนข้อมูล รูปที่ 2.5 แสดงรูปสัญญาณการอ่านและเขียนของแรมเบอร์ 2114 รายละเอียดของแรมเบอร์ 2114 ดูจากภาคผนวก ค



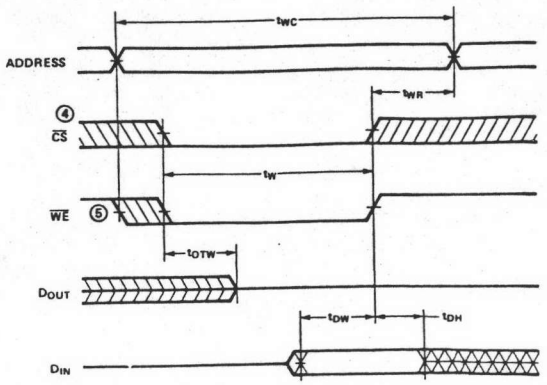
รูปที่ 2.4 แสดงรูปร่างและส่วนประกอบภายในแรมเบอร์ 2114

WAVEFORMS

READ CYCLE③



WRITE CYCLE



รูปที่ 2.5 แสดงรูปสัญญาณอ่านและเขียนของแรมเบอร์ 2114

2.8 การทำงานของวงจรควบคุม

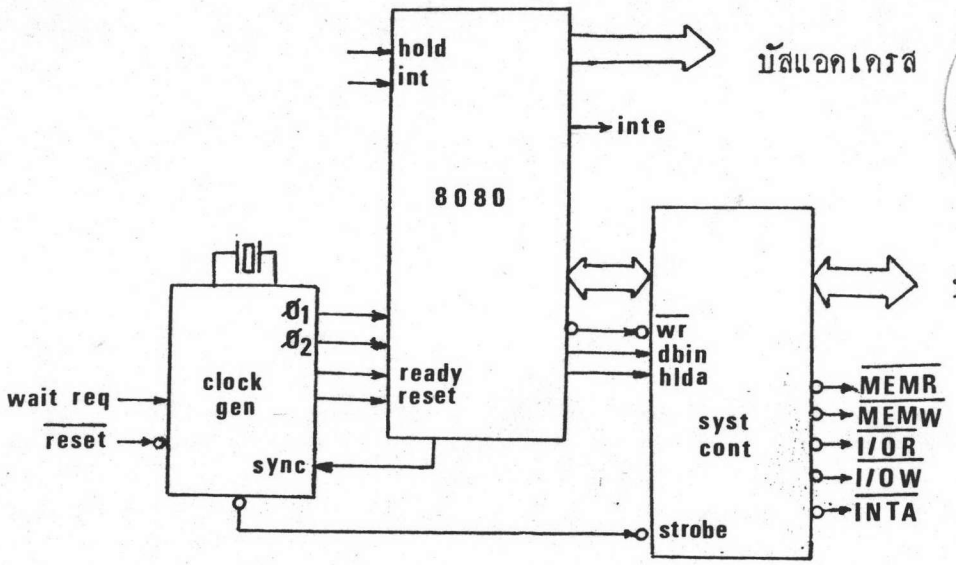
วงจรควบคุมประกอบด้วยส่วนสำคัญ 2 ส่วนคือ ส่วนที่เป็นฮาร์ดแวร์(Hardware) และซอร์ฟแวร์ (Software) ทำหน้าที่สร้างสัญญาณควบคุมทั้งหมดให้ระบบควบคุมข้อมูลของเทอร์มินอล

2.8.1 ฮาร์ดแวร์

ฮาร์ดแวร์ของวงจรควบคุมได้แก่ตัวไมโครโปรเซสเซอร์เป็นส่วนสำคัญที่สุด ในปัจจุบันนี้มีหลายบริษัทได้ผลิตไมโครโปรเซสเซอร์ขึ้น แต่ในขณะนี้ไมโครโปรเซสเซอร์ที่นิยมใช้กันอยู่คือไมโครโปรเซสเซอร์เบอร์ 8080 ถูกสร้างขึ้นครั้งแรกโดยบริษัทอินเทล เป็นไมโครโปรเซสเซอร์ชนิด 8 บิท สาเหตุอีกอันหนึ่งที่นิยมในเบอร์นี้เพราะมีหนังสือจำนวนมาก ที่เขียนเกี่ยวกับไมโครโปรเซสเซอร์เบอร์นี้ แต่ข้อเสียของไมโครโปรเซสเซอร์เบอร์นี้ก็คือไม่สามารถทำงานเพียงตัวเดียวได้ จะต้องมีวงจรสร้างสัญญาณนาฬิกาให้กับไมโครโปรเซสเซอร์และวงจรสร้างสัญญาณควบคุมระบบเพิ่มขึ้นอีก

วงจรควบคุมประกอบด้วย (ดูรูปที่ 2.6)

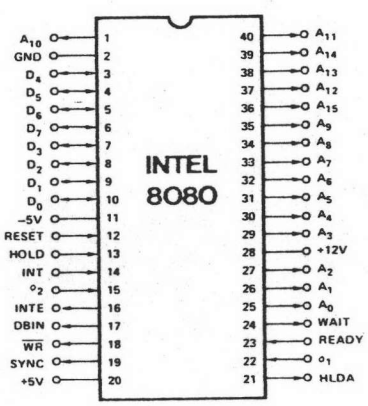
1. ไมโครโปรเซสเซอร์เบอร์ 8080
2. วงจรสร้างสัญญาณนาฬิกา (Clock Generator)
3. วงจรขับสัญญาณบัสข้อมูล 2 ทิศทางและวงจรสร้างสัญญาณควบคุมระบบ(Bi-Directional Bus Driver And System Control)



รูปที่ 2.6 แสดงส่วนประกอบภายในวงจรควบคุม

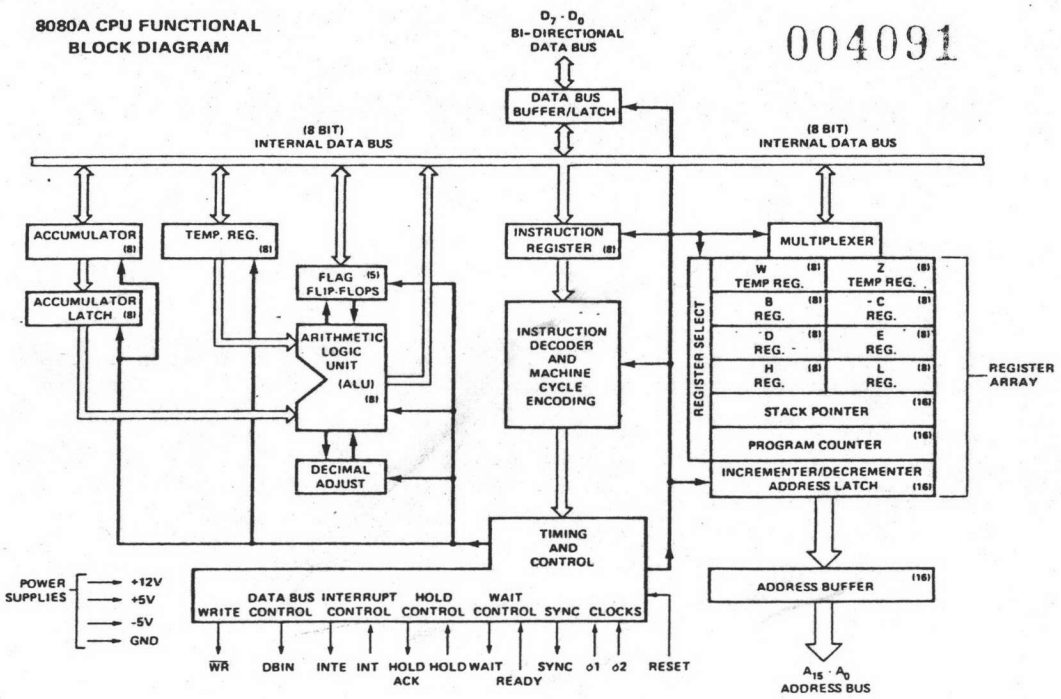
2.8.2 การทำงานของไมโครโปรเซสเซอร์เบอร์ 8080

ไมโครโปรเซสเซอร์เบอร์ 8080 ชนิด 8 บิท ที่ผลิตออกสู่ตลาดเป็นตัวแรกสุด ใช้เป็นซีพียู (CPU ย่อมาจาก Central Processing Unit) ในระบบไมโครคอมพิวเตอร์ ปัจจุบันเป็นที่นิยมใช้กันแพร่หลาย มีการประยุกต์ใช้งานในด้านควบคุมมากที่สุด ไมโครโปรเซสเซอร์เบอร์ 8080 เป็นไอซี 40 ขา ดังแสดงในรูปที่ 2.7



รูปที่ 2.7 แสดงรูปร่างไมโครโปรเซสเซอร์เบอร์ 8080

ลักษณะโครงสร้างภายในไมโครโปรเซสเซอร์เบอร์ 8080 แสดงในรูปที่ 2.8 ประกอบด้วยส่วนต่าง ๆ คือ



รูปที่ 2.8 แสดงโครงสร้างภายในไมโครโปรเซสเซอร์เบอร์ 8080

- อินสตรัคชันรีจิสเตอร์ (Instruction Register) เป็นรีจิสเตอร์ที่รับข้อมูลที่เป็นออปโคด (Op-Code) จากหน่วยความจำผ่านดาต้าบัฟเฟอร์แลทช์ (Data Buffer/Latch) แล้วส่งต่อไปให้อินสตรัคชันดีโคดและแมชินไซเกิ้ลเอนโคดดิ้ง (Instruction Decode And Machine Cycle Encoding)

- ดาต้าบัสบัฟเฟอร์แลทช์ เป็นรีจิสเตอร์ชนิดไทสเตด (Tri-State) ที่ใช้แยกข้อมูลภายในของไมโครโปรเซสเซอร์ออกจากบัสข้อมูลภายนอก

- อินสตรัคชันดีโคดและแมชินไซเกิ้ลเอนโคดดิ้ง ทำหน้าที่รับข้อที่เป็นออบโคดจากอินสตรัคชันรีจิสเตอร์มาแปลออกเป็นสัญญาณควบคุมย่อย เพื่อควบคุมการทำงานภายในไมโครโปรเซสเซอร์

- โปรแกรมเคาน์เตอร์ (Program Counter) เป็นรีจิสเตอร์สำหรับเก็บข้อมูลที่เป็นแอดเดรสของหน่วยความจำ เพื่อบอกตำแหน่งของคำสั่งต่อไป

- สแตคพอยเตอร์ (Stack Pointer) ใช้สำหรับเก็บข้อมูลที่เป็นแอดเดรสของหน่วยความจำ เพื่อบอกตำแหน่งยอดของสแตค โดยใช้เนื้อที่ส่วนหนึ่งของหน่วยความจำเป็นสแตค โดยสแตคพอยเตอร์เป็นตัวชี้ตำแหน่งยอดของสแตค สแตคที่ใช้เป็นแบบ เข้าก่อนออกทีหลัง (Fist in Last Out)

- เอแอลยู (ALU ย่อมาจาก Arithematic Logical Unit)ทำหน้าที่เกี่ยวกับการคำนวณ เปรียบเทียบข้อมูล ปฏิบัติการทางลอจิก

- แฟลก (Flag) เป็นรีจิสเตอร์ที่แสดงสภาวะการทำงานของเอแอลยู ในคำสั่งคำนวณ คำสั่งเปรียบเทียบข้อมูลและการปฏิบัติทางลอจิกประกอบด้วยแฟลก 5 ตัวคือ แฟลกศูนย์ (Zero Flag) แฟลกตัวทด (Carry Flag) แฟลกเครื่องหมาย (Sign Flag) แฟลกตัวช่วยทด (Auxilary Flag) และแฟลกพาริตี้ (Parity Flag)

- รีจิสเตอร์ชั่วคราว (Temporary Register) เป็นรีจิสเตอร์ใช้สำหรับช่วยการคำนวณของเอแอลยู ไม่สามารถเรียกใช้ด้วยคำสั่ง

- กลุ่มรีจิสเตอร์ (Register Array) เป็นกลุ่มรีจิสเตอร์ภายในทำหน้าที่เก็บข้อมูลชั่วคราว ซึ่งสามารถเรียกใช้ด้วยคำสั่ง ประกอบด้วยรีจิสเตอร์ A,B,C,D,E,Z,W สำหรับรีจิสเตอร์ Z,W ไม่สามารถเรียกใช้ด้วยคำสั่งสำหรับการเคลื่อนย้ายข้อมูล ภายในรีจิสเตอร์แต่ละตัวเกิดขึ้นโดยมีรีจิสเตอร์ซีเลคท์ (Register Select) และมัลติเพล็กเซอร์ (Mulitiplexer) เป็นตัวเลือกโดยผ่านการควบคุมจากไทม์มิงและคอนโทรล (Timing And Control)

2.8.3 สัญญาณต่าง ๆ ของไมโครโปรเซสเซอร์เบอร์ 8080

- บัสแอดเดรส (Address Bus) คือ A_0 - A_{15} เป็นกลุ่มสัญญาณขนาด 16 บิท แบบไทสเตค เพื่อส่งออกไปกำหนดตำแหน่งในหน่วยความจำหรือตำแหน่งไอโอเนื่องจากบัสแอดเดรสมี 16 บิท ทำให้สามารถกำหนดตำแหน่งในหน่วยความจำได้ถึง 64 กิโลไบท์

- บัสข้อมูล (Data Bus) คือ D_0 - D_7 เป็นกลุ่มสัญญาณขนาด 8 บิท 2 ทิศทาง (Bi-Directional) ใช้สำหรับติดต่อรับส่งข้อมูลจากหน่วยความจำหรือไอโอ

- SYNC เป็นสัญญาณที่ส่งออกเพื่อแสดงถึงจุดเริ่มต้นของทุก ๆ แมชินไซเกิ้ล (Machine Cycle)

- DBIN เป็นสัญญาณที่ส่งออกเพื่อแสดงว่าขณะนี้บัสข้อมูลกำลังอยู่ในช่วงรับข้อมูลเข้าในไมโครโปรเซสเซอร์ ใช้เป็นสัญญาณเปิดรับข้อมูลจากหน่วยความจำ หรือไอโอเข้าสู่บัสข้อมูล

- READY เป็นสัญญาณที่แจ้งให้ไมโครโปรเซสเซอร์รู้ว่าขณะนี้ข้อมูลที่ต้องการจากหน่วยความจำหรือไอโออยู่บนบัสข้อมูลเรียบร้อยแล้ว

- $\overline{WR}$ เป็นสัญญาณที่ใช้ควบคุมการรับส่งข้อมูลไปเก็บในหน่วยความจำหรือส่งไปยังไอโอ ข้อมูลจะต้องคงที่ช่วงขณะที่ไมโครโปรเซสเซอร์ส่งสัญญาณนี้ออกมา

- WAIT เป็นสัญญาณที่ซีพียูส่งออกมาแสดงถึงสภาวะของไมโครโปรเซสเซอร์ว่าตอนนี้อยู่ในช่วงรอคอย

- HOLD เป็นสัญญาณที่วงจรภายนอกส่งเข้าเพื่อขอใช้บัสข้อมูลและบัสแอดเดรสในเวลาที่ต้องการเคลื่อนย้ายข้อมูลระหว่างหน่วยความจำกับไอโอ ถ้าไมโครโปรเซสเซอร์ยอมรับสัญญาณ HOLD ก็จะส่งสัญญาณ HLDA ออกมาพร้อมทั้งปลดปล่อยบัสข้อมูล บัสทั้งสองจะอยู่ในสภาพลอย (High Impedance)

- HLDA เป็นสัญญาณที่ไมโครโปรเซสเซอร์ส่งออกมาแสดงถึงการยอมรับต่อสัญญาณ HOLD

- INTE เป็นสัญญาณเพื่อแสดงการยอมรับของไมโครโปรเซสเซอร์ว่า ยอมรับสัญญาณอินเทอร์รัพท์ (interrupt)

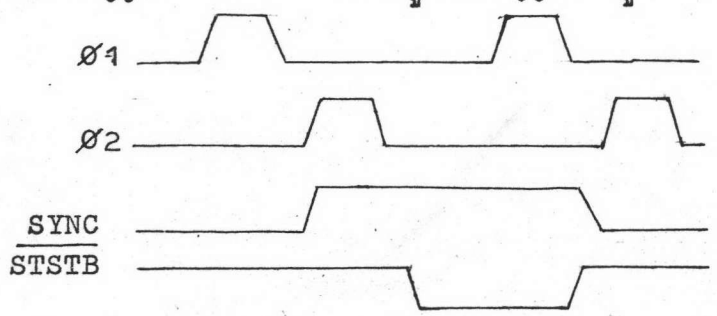
- INT เป็นสัญญาณที่วงจรภายนอกส่งเข้าเพื่อของอินเทอร์รัพท์จากไมโครโปรเซสเซอร์

- RESET สัญญาณรีเซสทำหน้าที่เคลียร์ (Clear) โปรแกรมเคาน์เตอร์ แต่ไม่มีผลต่อแฟลกต่าง ๆ หรือรีจิสเตอร์ภายในไมโครโปรเซสเซอร์

- ϕ_1 และ ϕ_2 เป็นสัญญาณนาฬิกาที่รับจากวงจรภายนอกชนิด 2 เฟส เพื่อกำหนดความเร็วในการปฏิบัติงานของไมโครโปรเซสเซอร์ แต่ความถี่สูงสุดของ ϕ_1 และ ϕ_2 ที่ใช้กับไมโครโปรเซสเซอร์เบอร์ 8080 เพียง 2 เมกกะเฮิทซ์เท่านั้น

2.8.4 วงจรสร้างสัญญาณนาฬิกา

ทำหน้าที่สร้างสัญญาณนาฬิกาชนิด 2 เฟสคือ ϕ_1 และ ϕ_2 ที่มีระดับสัญญาณตั้งแต่ 0.6 โวลท์ ถึง 11 โวลท์ ซึ่งมีความถี่ไม่เกินกว่า 2 เมกกะเฮิทซ์ เพื่อป้อนให้กับไมโครโปรเซสเซอร์เบอร์ 8080 และรับสัญญาณ SYNC จากไมโครโปรเซสเซอร์ เบอร์ 8080 เพื่อสร้างสัญญาณ $\overline{\text{STSTB}}$ โดยมีรูปร่างสัญญาณดังรูปที่ 2.9

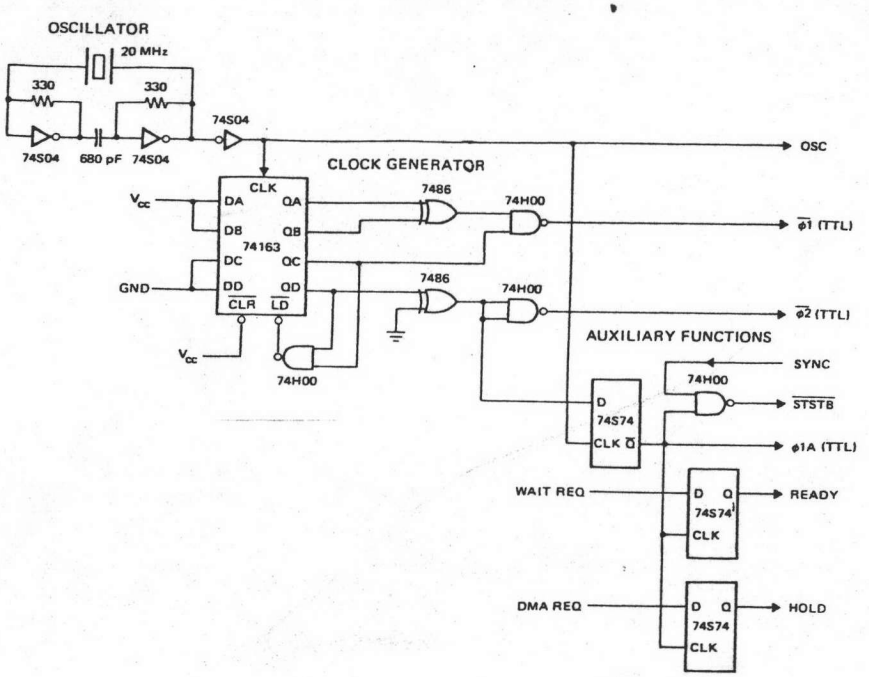


หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

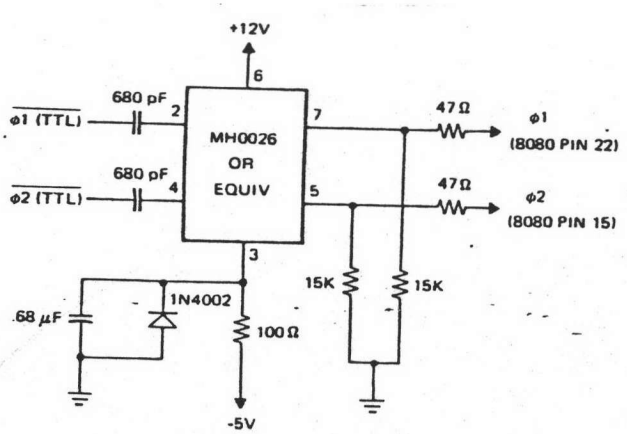
รูปที่ 2.9 แสดงรูปร่างสัญญาณที่สร้างโดยวงจรสัญญาณนาฬิกา

วงจรสร้างสัญญาณนาฬิกาสำหรับไมโครโปรเซสเซอร์เบอร์ 8080 มีวิธีการออกแบบวงจรในส่วนนี้อยู่หลายแบบเช่น

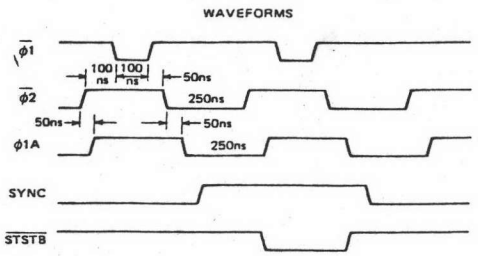
แบบที่ 1 เป็นวงจรที่ออกแบบโดยใช้เกทพื้นฐาน (Basic Gate) ดังรูปที่ 2.10 ประกอบด้วยวงจรสร้างความถี่สัญญาณนาฬิกา โดยใช้คริสตรอลที่มีความถี่ 20 เมกกะเฮทซ์เป็นตัวผลิตสัญญาณนาฬิกาทำให้สัญญาณ OSC มีความถี่ขาออกเท่ากับ 20 เมกกะเฮิทซ์ และสัญญาณ $\overline{\phi}_1$ (TTL) และ $\overline{\phi}_2$ (TTL) เป็นสัญญาณนาฬิกาที่มีความถี่เท่ากับ 2 เมกกะเฮิทซ์ ซึ่งเกิดจากการหารความถี่ 20 เมกกะเฮิทซ์ ด้วยวงจรเคาน์เตอร์เบอร์ 74163 ระดับสัญญาณ $\overline{\phi}_1$ (TTL) และ $\overline{\phi}_2$ (TTL) ยังอยู่ในระดับสัญญาณ TTL เท่านั้น แต่ระดับสัญญาณ ϕ_1 และ ϕ_2 ของไมโครโปรเซสเซอร์ 8080 ต้องการสัญญาณในระดับ 0.6 โวลท์ ถึง 11 โวลท์ เพราะฉะนั้นจะต้องมีการขับสัญญาณ ϕ_1 และ ϕ_2 ให้มีระดับสูงขึ้น โดยใช้วงจรขับสัญญาณระดับสูง (High level Driver) ดังรูปที่ 2.11 ภายในวงจรในรูปที่ 2.10 ยังมีวงจรรับสัญญาณ SYNC จากไมโครโปรเซสเซอร์เพื่อนำมาสร้างสัญญาณ$\overline{\text{STSTB}}$สำหรับสัญญาณϕ_{1A}จะมีความถี่เท่ากับสัญญาณ $\overline{\phi}_2$ ดังรูปที่ 2.12 นอกจากนี้ยังมีวงจรรับสัญญาณออกภายนอกคือสัญญาณ WAIT และ DMA เพื่อสร้างสัญญาณ READY และสัญญาณ HOLD ให้แก่ไมโครโปรเซสเซอร์



รูปที่ 2.10 แสดงวงจรสร้างสัญญาณนาฬิกาด้วยเกทพื้นฐาน



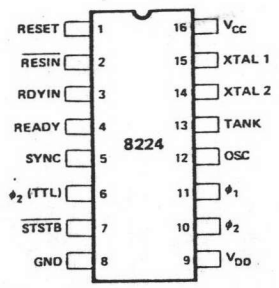
รูปที่ 2.11 แสดงวงจรขับสัญญาณระดับสูง



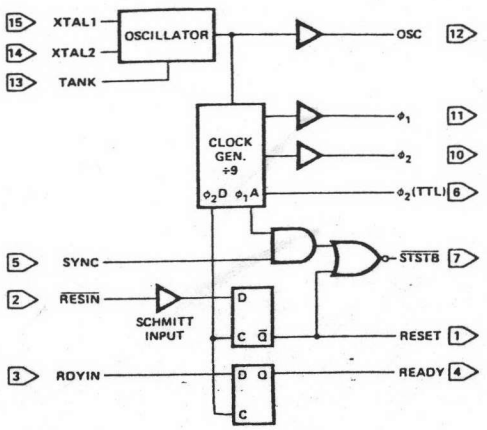
รูปที่ 2.12 แสดงรูปสัญญาณนาฬิกาจากวงจรเกทพื้นฐาน

แบบที่ 2 เพื่อความสะดวก บริษัทอินเทลได้ออกแบบไอซีชนิด LSI(Large Scale Intergrated Circuit) เบอร์ 8224 ขึ้น สำหรับทำหน้าที่สร้างสัญญาณนาฬิกาและขับสัญญาณ Clock Generator And Driver) เพื่อใช้ร่วมกับไมโครโปรเซสเซอร์เบอร์ 8080 จึงทำให้ขนาดของวงจรในส่วนนี้ลดลง ดูรูปที่ 2.13 แสดงรูปร่างและส่วนประกอบภายในไอซีเบอร์ 8224 เป็นไอซีขนาด 16 ขา ไอซีเบอร์ 8224 จะทำหน้าที่ผลิตสัญญาณนาฬิกาชนิด 2 เฟสคือ ϕ_1 และ ϕ_2 ที่มีระดับสัญญาณตั้งแต่ 0.6 โวลท์ ถึง 11 โวลท์ป้อนให้กับไมโครโปรเซสเซอร์ โดยมีรูปร่างสัญญาณดังรูปที่ 2.14 ความถี่สัญญาณนาฬิกาจะถูกควบคุมด้วยคริสตอลอีกทีหนึ่ง ภายในไอซีเบอร์ 8224 มีวงจรหารความถี่คริสตอลลง 9 เท่า ก่อนที่เป็นสัญญาณนาฬิกา ϕ_1 และ ϕ_2 ขนาดความถี่สัญญาณนาฬิกา ϕ_1 และ ϕ_2 ที่ใช้กับไมโครโปรเซสเซอร์ 8080 ได้สูงสุดเพียง 2 เมกกะเฮิทซ์ ดังนั้นการพิจารณากำหนดสัญญาณนาฬิกาให้แก่ไมโครโปรเซสเซอร์เบอร์ 8080 จะใช้คริสตอลได้ไม่เกินกว่า 18 เมกกะเฮิทซ์ ขาสัญญาณต่าง ๆ ของไอซีเบอร์ 8224 ทำหน้าที่ดังนี้

PIN CONFIGURATION

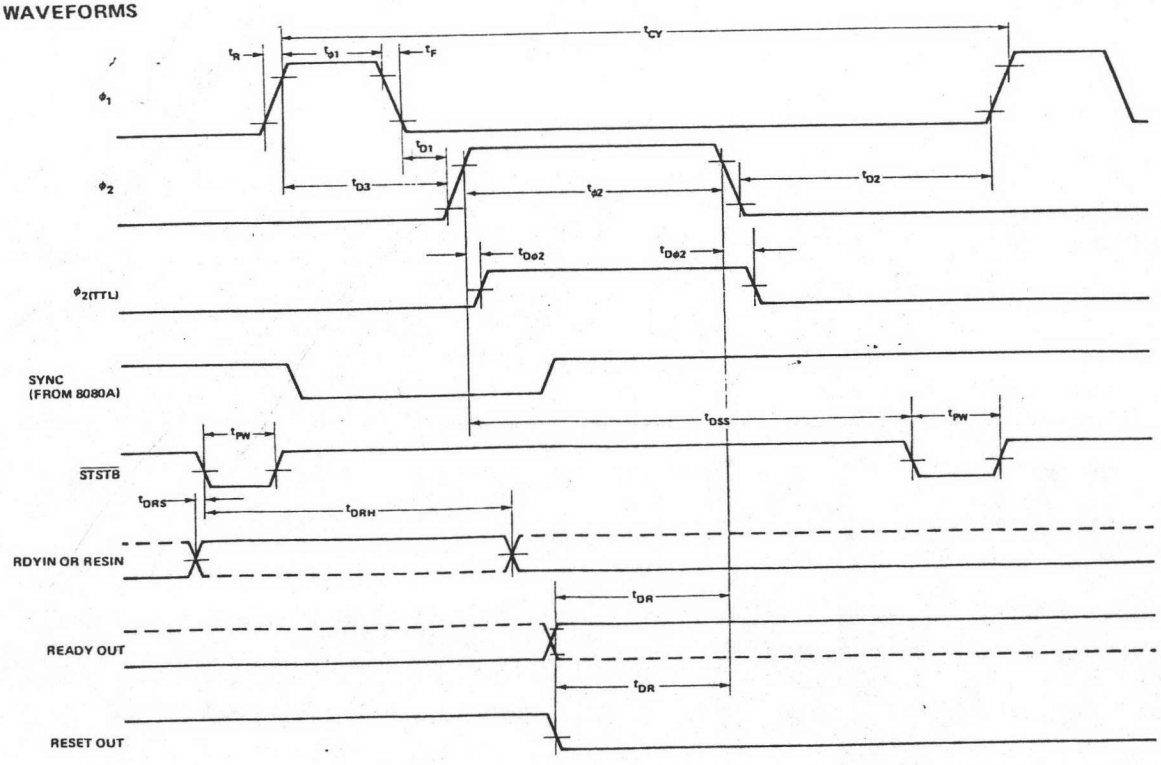


BLOCK DIAGRAM



รูปที่ 2.13 แสดงรูปร่างและส่วนประกอบภายในไอซีเบอร์ 8224

WAVEFORMS



รูปที่ 2.14 แสดงรูปสัญญาณต่าง ๆ ของไอซีเบอร์ 8224

- XTAL 1 และ XTAL 2 เป็นขาที่ใช้ต่อกับคริสตอล โดยคริสตอลจะต้องมีความถี่เป็น 9 เท่าของสัญญาณนาฬิกาที่ใช้จริง

- TANK เป็นขาที่ใช้ต่อกับวงจร LC เมื่อใช้คริสตอลชนิดที่เรียกว่า โอเวอร์โทนคริสตอล (Overtone Crystal)

- OSC เป็นสัญญาณขาออกของสัญญาณนาฬิกาที่มีความถี่เท่ากับคริสตอล

- ϕ_2 (TTL) เป็นขาสัญญาณออกของสัญญาณนาฬิกามีรูปร่างเดียวกับ ϕ_2 แต่อยู่ในระดับสัญญาณ TTL

- ϕ_1, ϕ_2 เป็นขาออกของสัญญาณนาฬิกาชนิด 2 เฟสสำหรับไมโครโปรเซสเซอร์

- SYNC เป็นขาสัญญาณเข้าต่อตรงมาจากขาสัญญาณ SYNC ของไมโครโปรเซสเซอร์เบอร์ 8080 สำหรับผลิตสัญญาณ $\overline{\text{STSTB}}$

- $\overline{\text{RESIN}}$ เป็นสัญญาณเข้าเพื่อควบคุมการรีเซสของไมโครโปรเซสเซอร์

- RESEN เป็นขาสัญญาณออกเพื่อใช้รีเซสไมโครโปรเซสเซอร์

- RDYIN เป็นขาที่ใช้ต่อกับวงจรควบคุมความเร็วของการปฏิบัติคำสั่งไมโครโปรเซสเซอร์

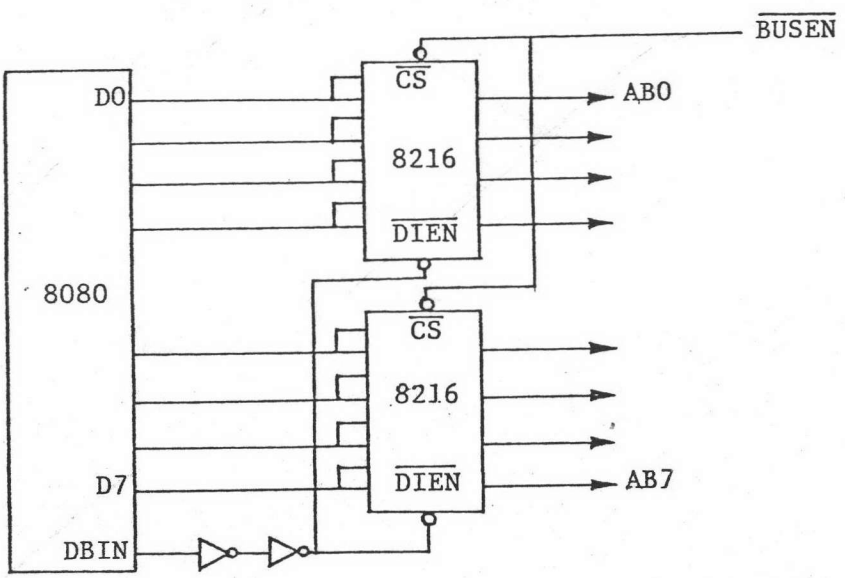
- $\overline{\text{STSTB}}$ เป็นขาสัญญาณออกที่ใช้สำหรับแลทช์ (latch) สัญญาณสแตตัส จากบัสข้อมูลไปเก็บในสแตตัสแลทช์ ซึ่งเกิดทุก ๆ ช่วงแรกของแมชินไซเกิ้ล

2.8.5 วงจรขับสัญญาณบัสข้อมูลสองทิศทางและวงจรสร้างสัญญาณควบคุมระบบ

หน่วยความจำและไอโอสามารถติดต่อกับไมโครโปรเซสเซอร์ได้ทางบัสข้อมูลเป็น 2 ทิศทาง แต่ระดับสัญญาณบัสข้อมูลที่ออกมาจากไมโครโปรเซสเซอร์อยู่มีระดับต่ำ จึงไม่เพียงพอกับการต่อหน่วยความจำและไอโอจำนวนมาก จำเป็นต้องขับสัญญาณบัสข้อมูลให้มีระดับสูงขึ้นเพื่อต่อกับหน่วยความจำและไอโอจำนวนมาก สำหรับวงจรสร้างสัญญาณควบคุมระบบทำหน้าที่สร้างสัญญาณควบคุมภายนอกให้แก่ไมโครโปรเซสเซอร์ตามสัญญาณสแตตัส ที่บอกสภาวะของไมโครโปรเซสเซอร์ที่ถูกส่งออกมาทางบัสข้อมูลในช่วงแรกของทุก ๆ แมชินไซเกิ้ล

การออกแบบวงจรขับสัญญาณข้อมูล 2 ทิศทางและวงจรสร้างสัญญาณควบคุมระบบมีหลายแบบเช่น

แบบที่ 1 การออกแบบวงจรขับสัญญาณบัสข้อมูล 2 ทิศทางโดยใช้ไอซี 8216 เป็นไอซีที่ออกแบบขึ้นมาเพื่อเป็นวงจรขับสัญญาณบัสข้อมูล 2 ทิศทาง ขนาด 4 บิท ซึ่งให้ระดับสัญญาณขาออกสูงถึง 3.65 โวลท์ 50 มิลิแอมป์ จากรูป 2.15 แสดงวงจรขับสัญญาณบัสข้อมูลโดยใช้ไอซีเบอร์ 8216 สองตัวต่อกับบัสข้อมูลขาออกของไมโครโปรเซสเซอร์ ขาสัญญาณที่ใช้ควบคุมการทำงานของไอซี 8216 มี $\overline{CS}$ (Chip Select) และ $\overline{DIEN}$ (Direction Enable) โดย $\overline{CS}$ ต่อเข้ากับ $\overline{BUSEN}$ เพื่อควบคุมสภาวะลอยของบัสข้อมูลในขณะทำ DMA (Direct Memory Access) และขาสัญญาณ DBIN จากไมโครโปรเซสเซอร์ต่อเข้าขาสัญญาณ $\overline{DIEN}$ เพื่อควบคุมทิศทางการเคลื่อนย้ายข้อมูลระหว่างไมโครโปรเซสเซอร์กับหน่วยความจำหรือไอโอ

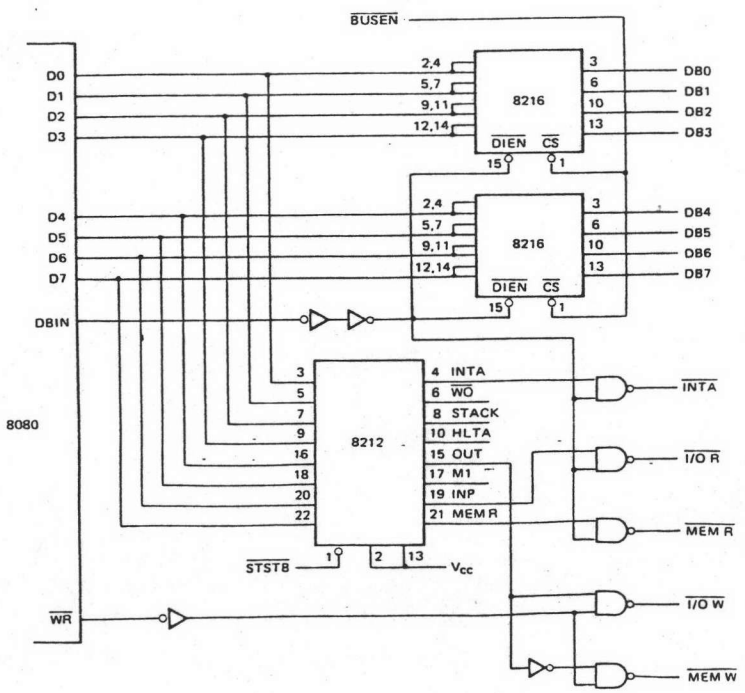


รูปที่ 2.15 แสดงการต่อไอซีเบอร์ 8216 กับไมโครโปรเซสเซอร์

การออกแบบวงจรสร้างสัญญาณควบคุมระบบใช้ไอซีเบอร์ 8212 ซึ่งเป็นวงจรแลทช์ทำหน้าที่แลทช์สัญญาณสแตตัสได้เพื่อประกอบเป็นสัญญาณควบคุมอีกทีหนึ่ง รูปที่ 2.16 แสดงวงจรสร้างสัญญาณควบคุมระบบโดยใช้ไอซีเบอร์ 8212 ซึ่งต่อกับบัสข้อมูลในช่วงแรกของทุก ๆ แมชินไซเกิ้ลในจังหวะเดียวกับสัญญาณ $\overline{STSTB}$ จากรูปวงจรจะเห็นได้ว่าสัญญาณควบคุมเบื้องต้นที่เกิดจากวงจรมีอยู่ 5 สัญญาณคือ

1. สัญญาณอ่านข้อมูลจากหน่วยความจำ ($\overline{MEMR}$)
2. สัญญาณเขียนข้อมูลไปยังหน่วยความจำ ($\overline{MEMW}$)
3. สัญญาณอ่านข้อมูลจากไอโอ ($\overline{I/OR}$)
4. สัญญาณเขียนข้อมูลไปยังหน่วยความจำ ($\overline{I/OW}$)
5. สัญญาณตอบสนองอินเทอร์รัพท์ ($\overline{INTA}$)

โดยสัญญาณทั้ง 5 เกิดจากการรวมกัน (Combination) ของสัญญาณ DBIN สัญญาณ $\overline{WR}$ และสัญญาณ INTA กับสแตตัสบิท (Status Bit)



รูปที่ 2.16 แสดงวงจรสร้างสัญญาณควบคุมระบบ

แบบที่ 2 บริษัทอินเทลได้ออกแบบไอซีเบอร์ 8228 เพื่อทำหน้าที่ขับสัญญาณบัสข้อมูล และสร้างสัญญาณควบคุมระบบให้กับไมโครโปรเซสเซอร์เบอร์ 8080 สัญญาณควบคุมระบบ ใช้ควบคุมหน่วยความจำและอุปกรณ์ไอโอต่าง ๆ รูปที่ 2.17 แสดงส่วนประกอบภายในไอซีเบอร์ 8228 ประกอบด้วยวงจรขับสัญญาณบัสข้อมูล 2 ทิศทาง (Bi-Direction Bus Drive) วงจรสแตตัสแลทช์ (Status Latch) และวงจรเกทติ้งอาร์เรย์ (Gating Array)

วงจรขับสัญญาณบัสข้อมูล 2 ทิศทาง ใช้บัฟเฟอร์ (Buffer) ระหว่างบัสข้อมูลของไมโครโปรเซสเซอร์กับบัสข้อมูลของหน่วยความจำและไอโอ เพราะบัสข้อมูลของไมโครโปรเซสเซอร์มีขนาดของสัญญาณบัสข้อมูลเพียง 3.3 โวลท์ 1.9 มิลิแอมป์เท่านั้น ไม่เพียงพอที่จะนำไปต่อกับวงจรอื่น ๆ โดยตรง จึงจำเป็นจะต้องขับสัญญาณบัสข้อมูลก่อน โดยใช้วงจรขับสัญญาณบัสข้อมูล 2 ทิศทาง ทำให้สัญญาณบัสข้อมูลเพิ่ม 4.75 โวลท์ 10 มิลิแอมป์ ซึ่งเพียงพอกับการนำไปต่อกับหน่วยความจำที่มีขนาดใหญ่หรือไอโอจำนวนมาก

วงจรสแตตัสแลทช์ ทำหน้าที่บอกสภาวะของไมโครโปรเซสเซอร์แต่ละแมชินไซเกิ้ลว่าต้องการติดต่อกับวงจรภายนอกส่วนใดบ้าง โดยส่งสภาวะออกมาทางบัสข้อมูลของ ไมโครโปรเซสเซอร์ในช่วงเวลาที่ไม่มีการใช้บัสข้อมูล เพื่อเก็บในวงจรสแตตัสแลทช์โดยการควบคุมของสัญญาณ $\overline{\text{STSTB}}$ เอาท์พุทของวงจรสแตตัสแลทช์จะถูกต่อไปยังวงจรเกทติ้งอาร์เรย์ ซึ่งเป็นส่วนที่ใช้สร้างสัญญาณควบคุมระบบ

วงจรเกทติ้งอาเรย์ ทำหน้าที่สร้างสัญญาณควบคุมระบบให้ไมโครโปรเซสเซอร์โดยสัญญาณควบคุมระบบแบ่งออกเป็น 2 สัญญาณคือ สัญญาณควบคุมการอ่าน และสัญญาณควบคุมการเขียน

สัญญาณควบคุมการอ่านคือ $\overline{\text{MEMR}}$, $\overline{\text{I/OR}}$, $\overline{\text{INTA}}$ เกิดจากการรวมของสแตตัสบิทกับสัญญาณ DBIN ของไมโครโปรเซสเซอร์

สัญญาณควบคุมการเขียนคือ สัญญาณ $\overline{\text{MEMW}}$, $\overline{\text{I/OW}}$ เกิดจากการรวมของสแตตัสบิทกับสัญญาณ $\overline{\text{WR}}$ ของไมโครโปรเซสเซอร์

หน้าที่ต่าง ๆ ของสัญญาณควบคุมประกอบด้วย

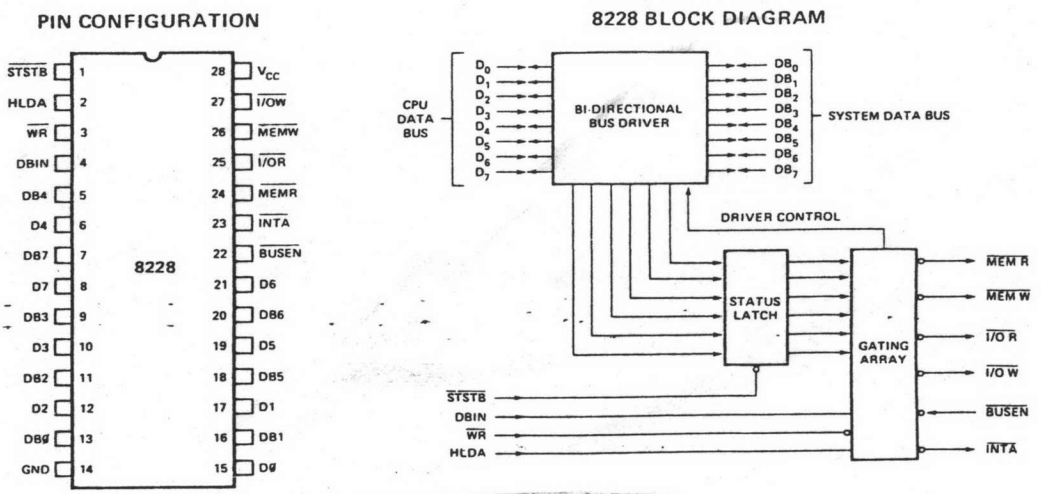
$\overline{\text{INTA}}$ เป็นสัญญาณเพื่อบอกการยอมรับการอินเทอร์รัพท์จากวงจรภายนอก

$\overline{\text{MEMR}}$ เป็นสัญญาณเพื่อบอกว่าต้องการอ่านข้อมูลมาจากหน่วยความจำ

$\overline{\text{MEMW}}$ เป็นสัญญาณเพื่อบอกว่าต้องการเขียนข้อมูลลงหน่วยความจำ

$\overline{\text{I/OR}}$ เป็นสัญญาณควบคุมการรับข้อมูลมาจากไอโอ

$\overline{\text{I/OW}}$ เป็นสัญญาณควบคุมการส่งข้อมูลไปยังไอโอ



รูปที่ 2.17 แสดงรูปร่างและส่วนประกอบภายในไอซีเบอร์ 8228

2.8.6 ซอร์ฟแวร์

ซอร์ฟแวร์เป็นโปรแกรมโมนิเตอร์ที่เก็บไว้อยู่ในวงจรหน่วยความจำ ส่วนอีพรอมใช้สำหรับควบคุมการปฏิบัติงานเกี่ยวกับข้อมูลภายในเทอร์มินอลโดยวงจรควบคุมจะสร้างสัญญาณควบคุมวงจรต่าง ๆ ภายในเทอร์มินอล ตามลำดับคำสั่งในโปรแกรมโมนิเตอร์ ภายในโปรแกรมโมนิเตอร์จะประกอบด้วยโปรแกรมย่อยต่าง ๆ ที่ทำหน้าที่ควบคุมคือ

- โปรแกรมควบคุมข้อมูลจากแป้นพิมพ์
- โปรแกรมควบคุมการจัดตำแหน่งภายในหน่วยความจำ
- โปรแกรมควบคุมตำแหน่งเคอร์เซอร์
- โปรแกรมควบคุมการรับส่งข้อมูลของอินเทอร์เฟสกับวงจรหน่วยความจำ
- โปรแกรมของวงจรแสดงผล

จะเห็นได้ว่าถ้ามีการแก้ไขการทำงานของเทอร์มินอล เพียงแต่แก้ไขโปรแกรมโมนิเตอร์เท่านั้น

2.9 การทำงานของวงจรอินเทอร์เฟส

ทำหน้าที่เชื่อมโยงเทอร์มินอลกับเครื่องคอมพิวเตอร์และจัดการเกี่ยวกับ การรับส่งข้อมูลระหว่างเทอร์มินอลกับเครื่องคอมพิวเตอร์ การรับส่งข้อมูลระหว่างเทอร์มินอลกับเครื่องคอมพิวเตอร์มี 2 แบบคือ แบบฟูลล์ดูเพล็กและแบบฮาล์ดเพล็ก การส่งข้อมูลจากเทอร์มินอล ไปยังเครื่องคอมพิวเตอร์มี 3 วิธีคือ

1. แคเร็คเตอร์โมด (Character Mode) เป็นการส่งข้อมูลทีละหนึ่งตัวอักขระของเทอร์มินอลให้แก่เครื่องคอมพิวเตอร์ ข้อมูลจากแป้นพิมพ์จะถูกส่งไปยังวงจรอินเทอร์เฟสเพื่อส่งไปยังเครื่องคอมพิวเตอร์โดยไม่นำข้อมูลนั้นเก็บเข้าวงจรหน่วยความจำเพื่อแสดงผล เมื่อเครื่องคอมพิวเตอร์ได้รับข้อมูลแล้วก็จะส่งข้อมูลตัวเดิมกลับมายังเทอร์มินอล เพื่อนำเข้าเก็บในหน่วยความจำสำหรับไว้แสดงผลบนจอภาพ วิธีนี้จะต้องใช้สำหรับการรับส่งข้อมูลแบบฟูลล์เพล็ก

2. ไลน์โมด (Line Mode) เป็นการส่งข้อมูลทีละบรรทัดของเทอร์มินอล เช่น ถ้าในหนึ่งบรรทัดจุอักขระได้ 80 ตัว ก็จะส่งได้สูงสุดครั้งละ 80 ตัว ข้อดีของการส่งวิธีนี้คือเตรียมข้อมูลภายในเทอร์มินอลได้ก่อน โดยสามารถแก้ไขข้อมูลให้เรียบร้อยก่อนที่จะส่งไปยังเครื่องคอมพิวเตอร์ ซึ่งข้อมูลจากแป้นพิมพ์จะนำเก็บในหน่วยความจำเทอร์มินอลก่อน แล้วจึงส่งไปยังเครื่องคอมพิวเตอร์สำหรับวิธีนี้ใช้กับการรับส่งข้อมูลแบบฮาล์ดูเพล็ก

3. บล็อกโมด (Block Mode) เป็นการส่งข้อมูลทีละชุดข้อมูล ซึ่งชุดข้อมูลอาจจะเป็นข้อมูลที่ปรากฏบนจอภาพของเทอร์มินอลทีละหลาย ๆ บรรทัด ข้อดีของการส่งข้อมูลวิธีนี้เหมือนกับไลน์โมด

2.10 การทำงานของวงจรแสดงผล

วงจรแสดงผลทำหน้าที่สร้างอักขระต่าง ๆ ให้ปรากฏบนจอภาพ โดยรับรหัสข้อมูลจากวงจรหน่วยความจำมาเปลี่ยนเป็นสัญญาณภาพ จำนวนอักขระที่แสดงบนจอภาพ มีจำนวนต่างกันขึ้นอยู่กับการจัดจำนวนอักขระบนหน้าจอของวงจรแสดงผล โดยทั่วไปแล้วการจัดจำนวนอักขระบนหน้าจออาจจะจัดให้มีขนาด 24 บรรทัดและในหนึ่งบรรทัดมีจำนวนอักขระ 80 ตัว หรืออาจ

จัดให้มีขนาด 16 บรรทัดและในหนึ่งบรรทัดมีจำนวนอักขระ 64 ตัว ก็ขึ้นอยู่กับการออกแบบวงจร ในส่วนนี้ ภาพที่ปรากฏบนจอจะมีเคอร์เซอร์ปรากฏอยู่ด้วย โดยลักษณะของเคอร์เซอร์ มีรูปร่างเป็นสี่เหลี่ยมเล็ก ๆ ซึ่งเคอร์เซอร์เป็นตัวกำหนดตำแหน่งบนจอ ซึ่งตรงกับตำแหน่งในหน่วยความจำ เพื่อใช้บอกตำแหน่งสำหรับการเปลี่ยนแปลงแก้ไขข้อมูลในหน่วยความจำ