หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
การออกแบบระบบควบคุมข้อมูลของเทอร์มินอล

3.1 จุดมุ่งหมายของการออกแบบระบบควบคุมข้อมูลของเทอร์มินอล

การออกแบบระบบควบคุมข้อมูลของเทอร์มินอล เพื่อควบคุมข้อมูลระหว่างแป้นพิมพ์กับหน่วยความจำและการรับส่งข้อมูลระหว่างหน่วยความจำกับอินเทอร์เฟส ทำได้โดยการออกแบบการทำงานของไมโครโปรเซสเซอร์ การออกแบบระบบควบคุมข้อมูลของเทอร์มินอลแบ่งเป็น 2 ส่วนดังนี้

1. วงจรควบคุมภายใน ซึ่งประกอบด้วย
   - ไมโครโปรเซสเซอร์
   - หน่วยความจำ
   - วงจรเชื่อมโยงแป้นพิมพ์เข้ากับไมโครโปรเซสเซอร์
2. โปรแกรมควบคุมการทำงานของไมโครโปรเซสเซอร์ ซึ่งเรียกว่าโปรแกรมโมนิเตอร์ ทำหน้าที่ควบคุมการย้ายรหัสข้อมูลจากแป้นพิมพ์เข้าสู่หน่วยความจำหรือไปยังอินเทอร์เฟส และควบคุมการรับข้อมูลจากอินเทอร์เฟสเข้าสู่หน่วยความจำ ตลอดจนควบคุมการทำงานของหน่วยแสดงผล

3.2 การออกแบบวงจรหน่วยควบคุม

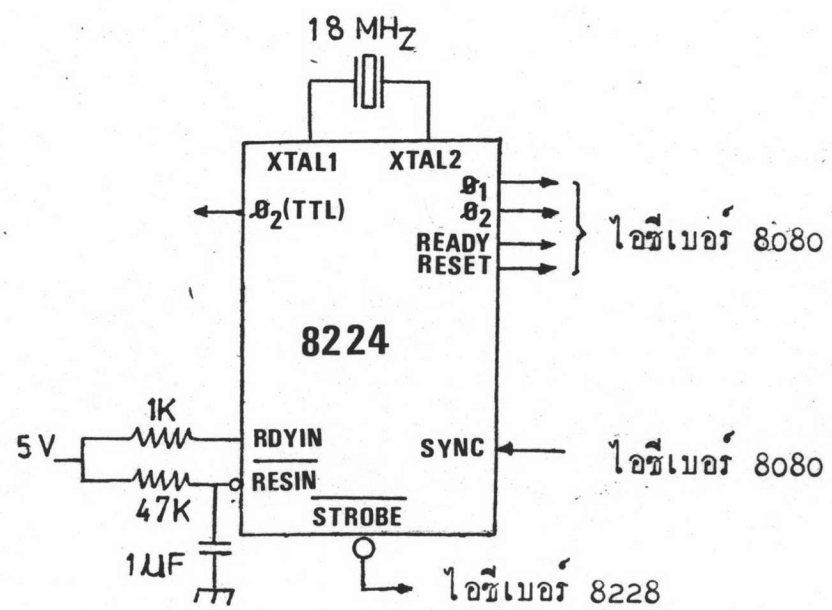
มีวัตถุประสงค์เพื่อการออกแบบวงจรส่วนสร้างสัญญาณควบคุมต่าง ๆ ภายในเทอร์มินอล โดยผู้วิจัยได้เลือกไมโครโปรเซสเซอร์เบอร์ 8080 เป็นตัวควบคุมหลัก ดังนั้นจำเป็นต้องมีวงจรภายนอกอีก 2 วงจรต่อประกอบกันเพื่อให้ไมโครโปรเซสเซอร์ทำงานได้โดยสมบูรณ์ได้แก่

- วงจรสร้างสัญญาณนาฬิกา
- วงจรสร้างสัญญาณควบคุมระบบ

และผู้วิจัยได้เลือกไอซีเบอร์ 8224 ทำหน้าที่สร้างสัญญาณนาฬิกาและไอซีเบอร์ 8228 ทำหน้าที่สร้างสัญญาณควบคุมระบบให้กับไมโครโปรเซสเซอร์ 8080 ดังจะได้กล่าวโดยละเอียดต่อไป

3.2.1 การออกแบบวงจรสร้างสัญญาณนาฬิกา

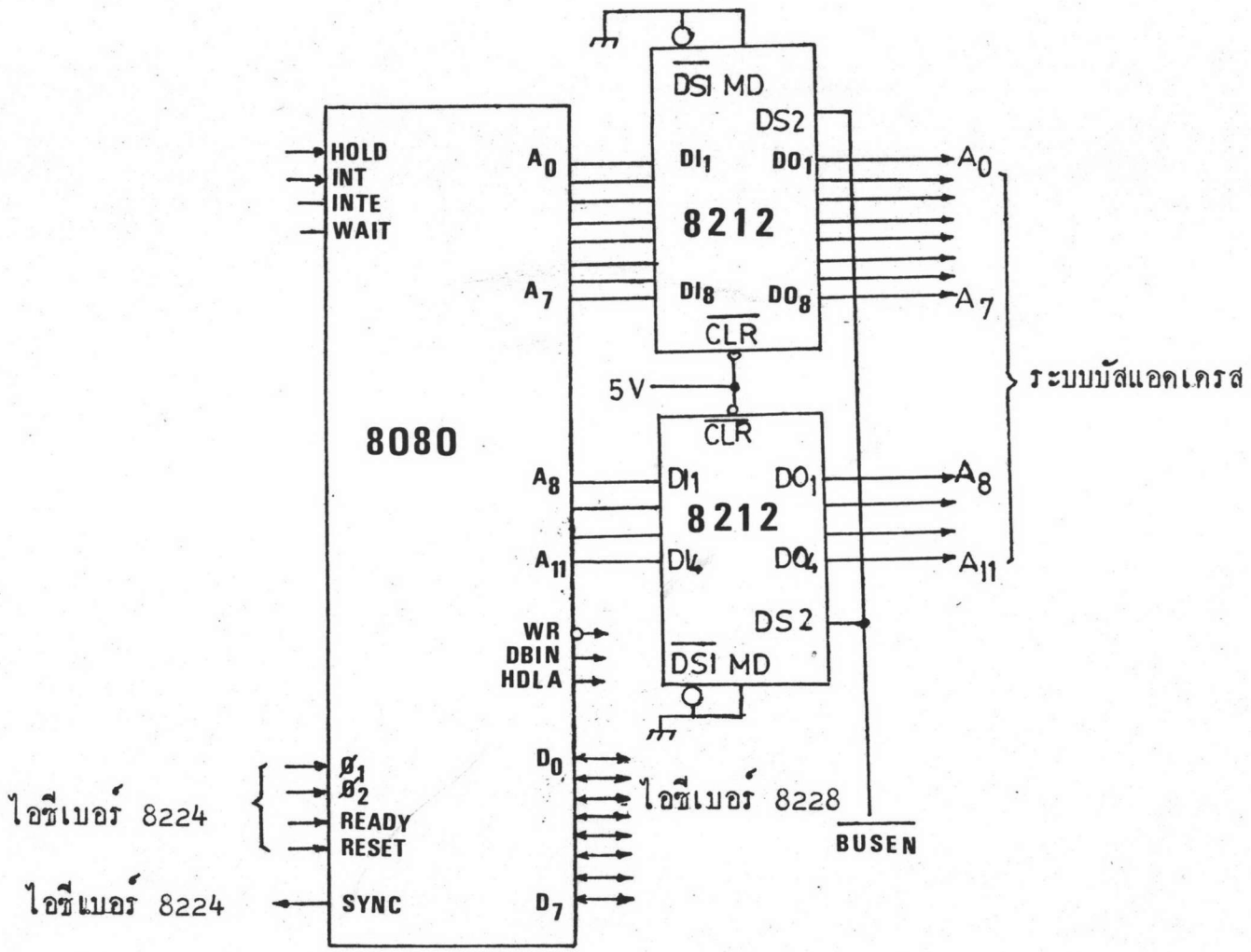
การออกแบบโดยใช้ไอซีเบอร์ 8224 ดังรูปที่ 3.1 ทำหน้าที่สร้างสัญญาณนาฬิกาและขับสัญญาณนาฬิกาชนิด 2 เฟสคือ ϕ_1 และ ϕ_2 ป้อนให้กับไมโครโปรเซสเซอร์เบอร์ 8080 มีระดับสัญญาณอยู่ในช่วง 0.6 ถึง 11 โวลท์ สัญญาณความถี่นาฬิกา ϕ_1 และ ϕ_2 สร้างขึ้นจากความถี่ 18 เมกกะเฮิทซ์ควบคุมโดยคริสตอลที่ต่ออยู่ขา XTAL1 กับขา XTAL2 ของวงจรซึ่งจะได้ความถี่ของสัญญาณนาฬิกา ϕ_1 และ ϕ_2 เท่ากับ 2 เมกกะเฮิทซ์ และ ϕ_2 (TTL) เป็นสัญญาณนาฬิกาที่มีความถี่เท่ากับ 2 เมกกะเฮิทซ์ แต่ระดับสัญญาณอยู่ในระดับ 0-5 โวลท์ โดยมีเฟสเดียวกับ ϕ_2 สำหรับขา $\overline{\text{RESIN}}$ ต่อความต้านทาน 47K กับตัวประจุ 1μF เพื่อสร้างสัญญาณเพาร์เวอร์อัพรีเซส (Power up Reset) ให้กับไมโครโปรเซสเซอร์และขา RDYIN ต่อความต้านทาน 1K กับไฟ 5 โวลท์ เพื่อทำให้ขา RDYIN เป็นลอจิก "1" เพื่อไม่ให้ไมโครโปรเซสเซอร์เสียเวลาคอยข้อมูลเพราะหน่วยความจำและไอโอสามารถทำงานได้เร็วกว่าไมโครโปรเซสเซอร์ นอกจากนี้ยังมีขาสัญญาณ READY และ RESET จะถูกต่อตรงไปยังไมโครโปรเซสเซอร์ ขาสัญญาณ SYNC ซึ่งต่อไปยังขาสัญญาณ SYNC ของไมโครโปรเซสเซอร์ เพื่อรับสัญญาณ SYNC จากไมโครโปรเซสเซอร์มาเพื่อสร้างสัญญาณ $\overline{\text{STROBE}}$ ให้กับไอซีเบอร์ 8228 เพื่อแลทช์แตตัสจากบัสข้อมูลทุก ๆ แมชินไซเกิ้ล



รูปที่ 3.1 แสดงวงจรสร้างสัญญาณนาฬิกาของไอซีเบอร์ 8224

3.2.2 การออกแบบวงจรไมโครโปรเซสเซอร์

วงจรไมโครโปรเซสเซอร์ทำหน้าที่ปฏิบัติงานตามคำสั่งของโปรแกรมควบคุมที่เก็บไว้ในอีพรอม โดยสร้างสัญญาณควบคุมวงจรภายนอกตามคำสั่งของโปรแกรมควบคุม วงจรไมโครโปรเซสเซอร์ประกอบด้วยไอซีเบอร์ 8080 เป็นไมโครโปรเซสเซอร์และไอซีเบอร์ 8212 สองตัวทำหน้าที่ขับสัญญาณบัสแอดเดรส ดูรูปที่ 3.2

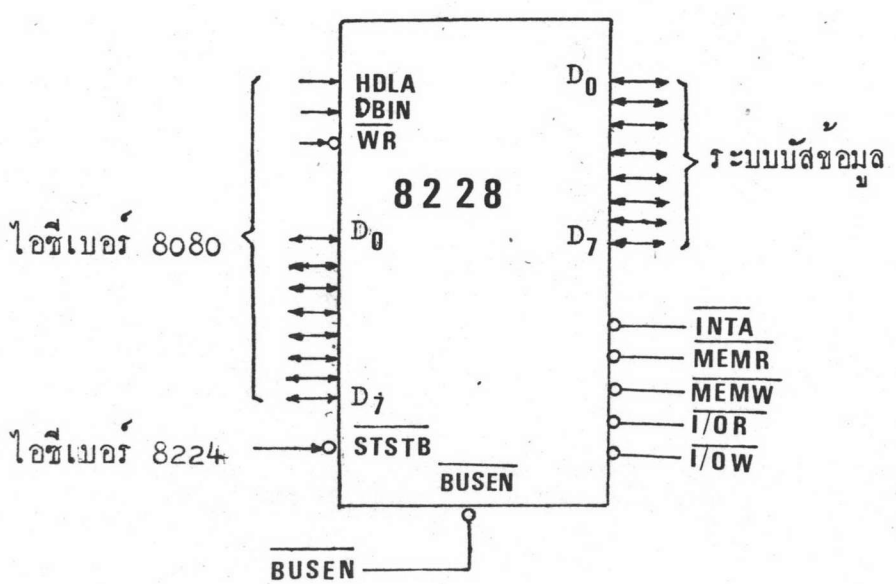


รูปที่ 3.2 วงจรส่วนไมโครโปรเซสเซอร์และวงจรขับสัญญาณบัสแอดเดรส

วงจรนี้ใช้บังคับแอดเดรส $A_0 - A_{11}$ เท่านั้น ซึ่งเพียงพอกับความต้องการ เพราะขนาดของหน่วยความจำที่ออกแบบมีขนาด 3 กิโลไบท์ บัสแอดเดรส $A_0 - A_{11}$ จะต้องผ่านวงจรขับสัญญาณบัสแอดเดรส เพื่อให้สามารถขับสัญญาณได้สูงขึ้น โดยใช้ไอซีเบอร์ 8212 จะให้แรงขับสัญญาณถึง 3.65 โวลท์ 50 มิลิแอมป์สำหรับสัญญาณ $\overline{\text{BUSEN}}$ รับมาจากวงจรภายนอกในกรณีใช้ DMA (Direct Memory Access) เพื่อใช้ควบคุมสภาวะของบัสแอดเดรสให้เป็นวงจรเปิดเวลาทำ DMA ในวงจรไมโครโปรเซสเซอร์เบอร์ 8080 มีขาสัญญาณ INT เพื่อรับสัญญาณจากวงจรภายนอกที่ส่งมาขออินเทอร์รัพท์และมีขาสัญญาณ HOLD เพื่อรับสัญญาณจากวงจรภายนอกที่ส่งมาขอใช้บัสแอดเดรสและบัสข้อมูล ส่วนสัญญาณที่ส่งออกจากไมโครโปรเซสเซอร์ 8080 มีขาสัญญาณ SYNC ส่งไปยังไอซีเบอร์ 8224 สัญญาณ $\overline{\text{WR}}$ สัญญาณ DBIN และสัญญาณ HDLA ถูกส่งไปยังไอซีเบอร์ 8228 พร้อมกับบัสข้อมูล D_0-D_7

3.2.3 การออกแบบวงจรสร้างสัญญาณควบคุมระบบ

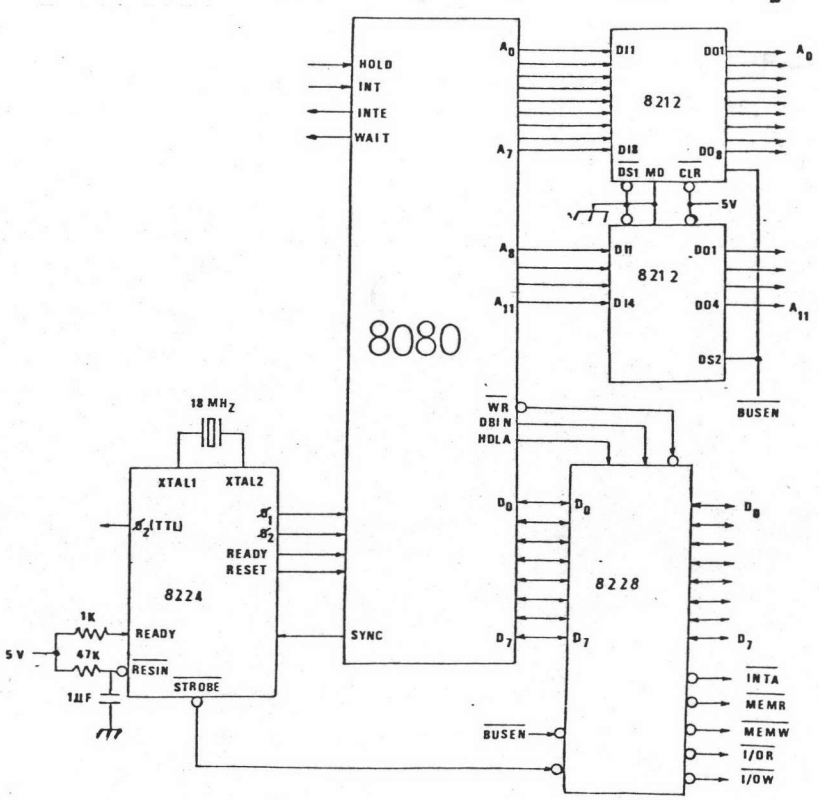
วงจรสร้างสัญญาณการควบคุมระบบทำหน้าที่สร้างสัญญาณควบคุมต่าง ๆ จากไมโครโปรเซสเซอร์และขับสัญญาณบัสข้อมูลโดยใช้ไอซีเบอร์ 8228 ในการออกแบบดังรูปที่ 3.3



รูปที่ 3.3 แสดงวงจรสร้างสัญญาณควบคุมและขับสัญญาณบัสข้อมูล

ลักษณะบัสข้อมูลที่ต่อเข้าไอซีเบอร์ 8228 เป็นแบบ 2 ทิศทาง (Bi-Direction Bus) บัสข้อมูลจากไมโครโปรเซสเซอร์ก่อนจะถูกส่งเป็นระบบบัสข้อมูลจะขับสัญญาณบัสข้อมูลก่อนโดยวงจรภายในไอซีเบอร์ 8228 สำหรับสัญญาณควบคุมที่เกิดจากไอซีเบอร์ 8228 คือสัญญาณ $\overline{\text{INTA}}$, $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{I/OR}}$ และ $\overline{\text{I/OW}}$ เกิดจากการส่งสแตตัสมาจากไมโครโปรเซสเซอร์ทางบัสข้อมูลในขณะบัสข้อมูลว่างในจังหวะเดียวกับสัญญาณ $\overline{\text{STSTB}}$ มาจากไอซีเบอร์ 8224 เพื่อทำการแลทช์สัญญาณควบคุมไว้ สำหรับขาสัญญาณ $\overline{\text{BUSEN}}$ เป็นสัญญาณควบคุมจากวงจรภายนอก ในกรณีเมื่อใช้ DMA ควบคุมสภาวะของบัสข้อมูลให้เป็นวงจรเปิดเวลาทำ DMA สัญญาณที่ส่งออกจากไมโครโปรเซสเซอร์มาควบคุมไอซีเบอร์ 8228 คือสัญญาณ HLDA เป็นสัญญาณที่ส่งมาเพื่อบอกการยอมรับต่อสัญญาณ HOLD สัญญาณ $\overline{\text{WR}}$ เป็นสัญญาณควบคุมที่บอกว่าต้องการส่งข้อมูลจากไมโครโปรเซสเซอร์ไปเก็บในหน่วยความจำหรือส่งไปยังไอโอ สำหรับสัญญาณ DBIN เป็นสัญญาณควบคุมทิศทางของบัสข้อมูลว่าขณะนี้จะรับข้อมูลจากบัสข้อมูลเข้าสู่ไมโครโปรเซสเซอร์หรือส่งข้อมูลภายในไมโครโปรเซสเซอร์ออกบัสข้อมูล

วงจรในรูปที่ 3.4 เป็นวงจรของไมโครโปรเซสเซอร์ที่ออกแบบสมบูรณ์โดยใช้ไอซีเบอร์ 8224 ทำหน้าที่สร้างสัญญาณนาฬิกาและขับสัญญาณไอซีเบอร์ 8080 เป็นไมโครโปรเซสเซอร์ และไอซีเบอร์ 8228 ทำหน้าที่สร้างสัญญาณควบคุมระบบและขับสัญญาณบัสข้อมูล



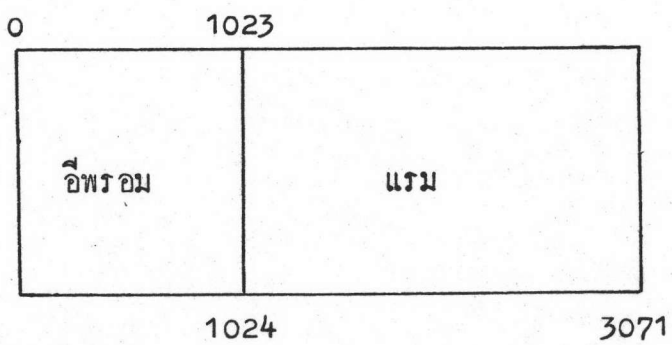
รูปที่ 3.4 แสดงวงจรหน่วยควบคุมของไมโครโปรเซสเซอร์

3.3 การออกแบบหน่วยความจำ

การออกแบบหน่วยความจำเพื่อใช้เก็บข้อมูลที่จะนำไปแสดงผลบนหน้าจอภาพของเทอร์มินอล และสำหรับเก็บโปรแกรมโมนิเตอร์ ในการออกแบบวงจรหน่วยความจำที่ใช้เก็บข้อมูลสำหรับแสดงผลใช้ไอซีเบอร์ 2114 ซึ่งเป็นแรมขนาด 1024 × 4 บิท และหน่วยความจำส่วนที่เก็บโปรแกรม ใช้ไอซีเบอร์ 2708 ซึ่งเป็นอีพรอมขนาด 1024 × 8 บิท

สิ่งที่พิจารณาในการออกแบบขนาดของหน่วยความจำชนิดแรมที่เก็บข้อมูลแสดงผลนั้น ผู้วิจัยได้พิจารณาจากจำนวนอักขระทั้งหมดที่แสดงผลบนหน้าจอภาพ ซึ่งการออกแบบหน่วยความจำ ได้ออกแบบสำหรับเทอร์มินอลที่แสดงผลข้อมูลบนจอภาพขนาด 16 บรรทัดและใน 1 บรรทัด บรรจุอักขระได้ 64 ตัว ดังนั้นจึงมีอักขระทั้งหมด 1024 ตัว อักขระแต่ละตัวที่นำเข้าเก็บในหน่วยความจำ ใช้พื้นที่ในหน่วยความจำตัวละ 1 ไบท์ จึงทำให้ขนาดแรมในส่วนนี้มีขนาด 1024 ไบท์ (1 กิโลไบท์) แต่จะต้องเตรียมพื้นที่แรมไว้สำหรับทำสแตคอีก 20 ไบท์ ผู้วิจัยจึงได้ออกแบบหน่วยความจำแรมทั้งหมด 2048 ไบท์ (2 กิโลไบท์) โดยใช้ไอซีเบอร์ 2114 จำนวน 4 ตัว สำหรับขนาดหน่วยความจำอีพรอมได้เตรียมไว้ขนาด 1 กิโลไบท์ ซึ่งเพียงพอกับโปรแกรมโมนิเตอร์

การจัดเรียงตำแหน่งของหน่วยความจำแรมและอีพรอมแสดงในรูปที่ 3.5 โดยให้อีพรอมอยู่ตั้งแต่ตำแหน่ง 0 ถึง 1023 ซึ่งเป็นตำแหน่งเริ่มต้นของหน่วยความจำและตามด้วยตำแหน่งของแรมตั้งแต่ตำแหน่ง 1024 ถึง 3071 ในการจัดเก็บอักขระในหน่วยความจำ จะเริ่มเก็บตั้งแต่ตำแหน่ง 1024 ซึ่งเป็นตำแหน่งเริ่มต้นของแรมจนถึงตำแหน่ง 2047



รูปที่ 3.5 แสดงการจัดเรียงหน่วยความจำชนิดอีพรอมและแรม

รูปที่ 3.6 แสดงวงจรของหน่วยความจำประกอบด้วยอีพรอมเบอร์ 2708 ที่ต่อขาแอดเดรส $A_0 - A_9$ เข้ากับสายแอดเดรส $A_0 - A_9$ ของระบบบัสแอดเดรส ซึ่งสายแอดเดรสมีทั้งหมด 10 สาย สามารถชี้ตำแหน่งในอีพรอมได้ 1 กิโลไบท์และขาข้อมูล O_0-O_7 ต่อเข้ากับสายข้อมูล $D_0 - D_7$ ของระบบบัสข้อมูล เพื่อรับข้อมูลจากการอ่านและขาสัญญาณ $\overline{CS}$ จะถูกต่อตรงเข้าวงจรถอดรหัส (Decoder) เบอร์ 74155 สำหรับควบคุมการอ่านข้อมูลจากอีพรอม

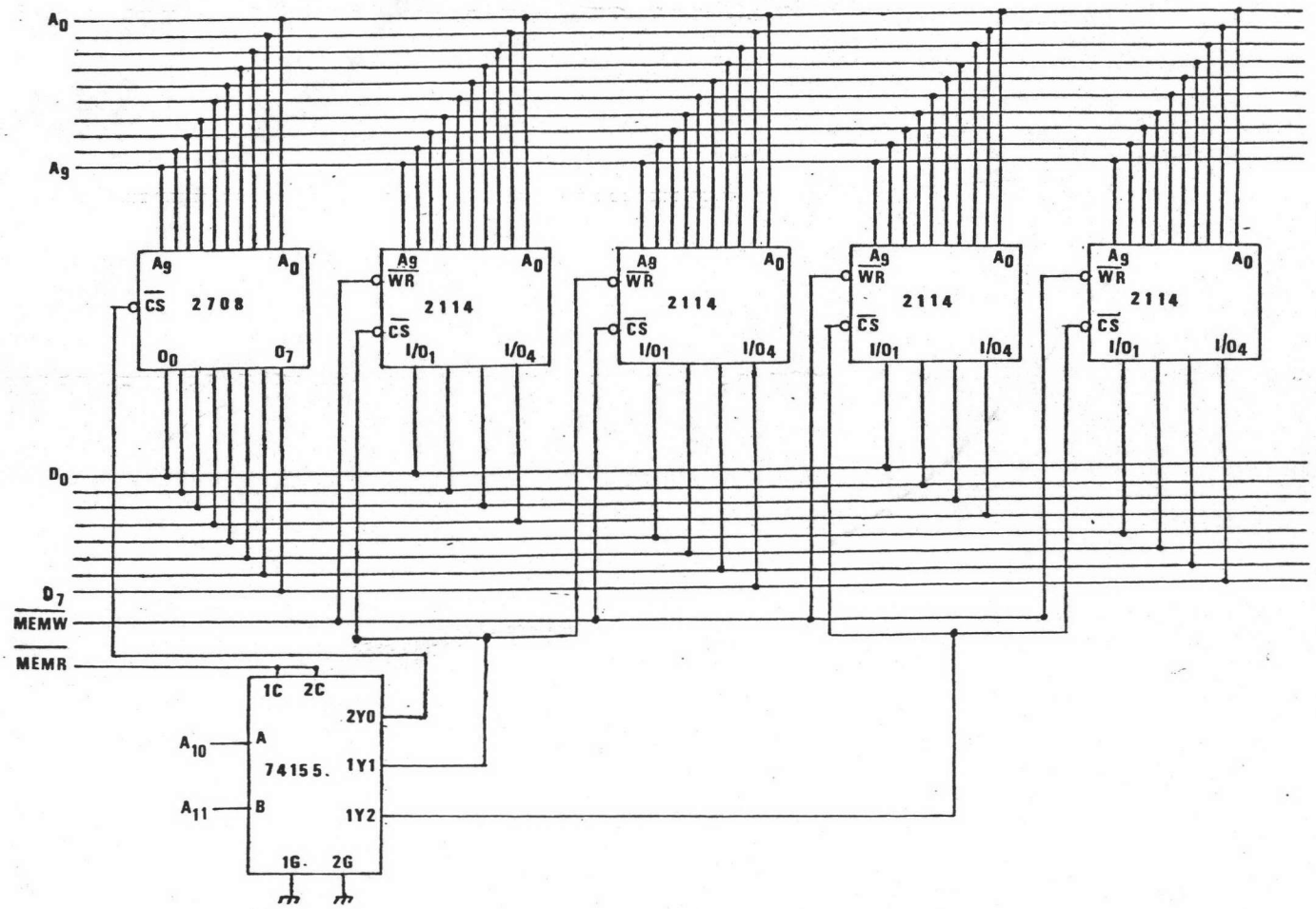
แรมเบอร์ 2114 มีทั้งหมด 4 ตัว ใช้แรม 2 ตัวต่อหน่วยความจำขนาด 1024 X 8 บิท เพราะแรมแต่ละตัวมีขนาดเพียง 1024 X 4 บิทเท่านั้น ขาแอดเดรส $A_0 - A_9$ ของแรมทุกตัวถูกต่อเข้ากับสายแอดเดรส $A_0 - A_9$ ของระบบบัสแอดเดรสและขาข้อมูล I/O_1-I/O_4 ของคู่แรมจะถูกต่อดังนี้ ถ้าขา $I/O_1 - I/O_4$ ของแรมตัวแรกต่อเข้ากับสายข้อมูล D_0 ถึง D_3 แรมตัวที่สองต้องต่อขา $I/O_1 - I/O_4$ เข้าสายข้อมูล $D_4 - D_7$ ขาสัญญาณ $\overline{CS}$ ทำหน้าที่เป็นเอนเนเบิ้ล การทำเอนเนเบิ้ลทำเป็นคู่โดยการควบคุมจากวงจรถอดรหัสเบอร์ 74155 ขาสัญญาณ $\overline{WE}$ เป็นขาสัญญาณควบคุมการอ่านและเขียนข้อมูลในหน่วยความจำนี้ต้องต่อกับสัญญาณ $\overline{MEMW}$ จากไมโครโปรเซสเซอร์

วงจรถอดรหัสเบอร์ 74155 ทำหน้าที่ถอดรหัสสัญญาณแอดเดรส A_{10}, A_{11} และ $\overline{MEMR}$ เพื่อเลือกใช้หน่วยความจำแรมและอีพรอม สำหรับสัญญาณ $\overline{MEMR}$ สงมาจากไมโครโปรเซสเซอร์ เพื่อใช้ควบคุมการอ่านข้อมูลจากอีพรอมเท่านั้น ตารางที่ 3.1 แสดงการเลือกใช้แรมและอีพรอมจากการควบคุมของสัญญาณ A_{10}, A_{11} และ $\overline{MEMR}$

A_{11}	A_{10}	$\overline{MEMR}$	ตำแหน่งในหน่วยความจำ
0	0	0	เริ่มจาก 0 ถึง 1023 (2 Y 0)
0	1	1	เริ่มจาก 1024 ถึง 2047 (1 Y 1)
1	0	1	เริ่มจาก 2048 ถึง 3071 (1 Y 2)

ตารางที่ 3.1 แสดงการเลือกใช้อีพรอมและแรม

i 17016427



รูปที่ 3.6 แสดงวงจรหน่วยความจำ

3.4 การออกแบบวงจรเชื่อมโยงระหว่างแป้นพิมพ์กับไมโครโปรเซสเซอร์

การออกแบบวงจรในส่วนนี้เพื่อนำรหัสข้อมูลจากแป้นพิมพ์เข้าสู่ไมโครโปรเซสเซอร์ทางบัสข้อมูล แป้นพิมพ์ที่ใช้เป็นของบริษัท NEC มีปุ่มกดข้อมูลบนแป้นพิมพ์แบ่งออกเป็น 2 ส่วนคือ ส่วนแรกได้แก่ปุ่มกดตัวเลข (Numeric Key) และส่วนที่สองได้แก่ปุ่มกดที่มีตัวเลขปนอยู่กับตัวอักษร (Alpha-Numeric Key) ในการกดปุ่มข้อมูลจะมีสัญญาณรหัสข้อมูลที่เป็นรหัส ASC II 8 บิท และสัญญาณ STB (Strobe) ออกมาทุกครั้ง โดยสัญญาณ STB จะออกมาในจังหวะเดียวกับรหัสข้อมูล

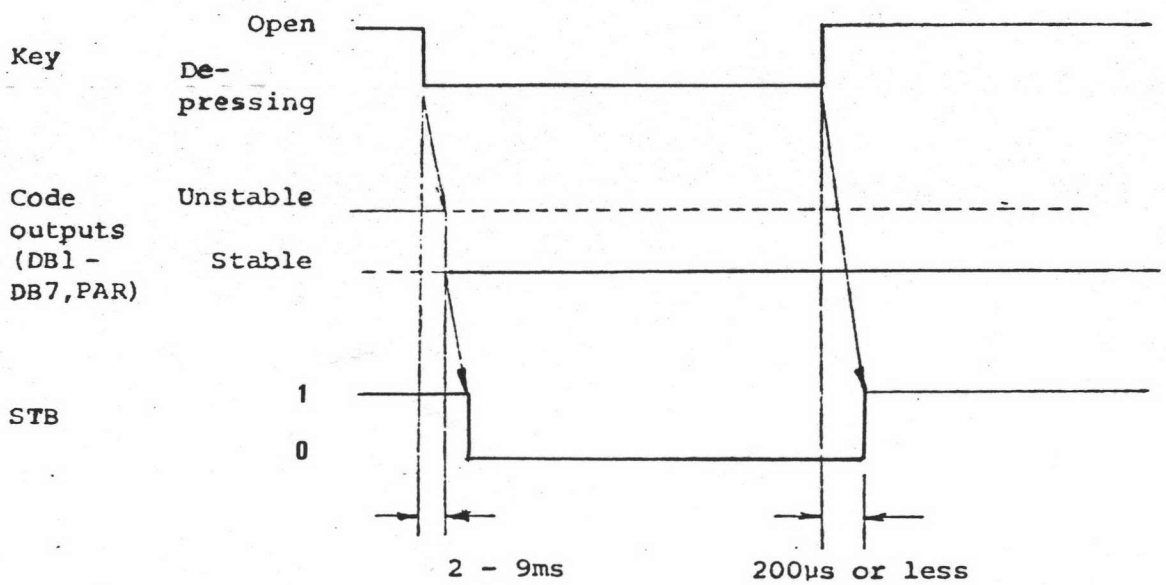
รหัสข้อมูลที่เกิดจากแป้นพิมพ์เป็นรหัส ASC II 8 บิท (ดูจากตารางที่ 3.2) ประกอบด้วยบิทข้อมูล 7 บิทคือ D_{B1} - D_{B7} และอีก 1 บิทเป็น (PAR (Even Parity Bit) จากตารางที่ 3.2 รหัสสัญญาณที่เกิดจากแป้นพิมพ์จะมีรหัสเป็นคอมพลีเมนท์(Complement) ของรหัส ASC II สำหรับปุ่มกดควบคุมที่มีอยู่บนแป้นพิมพ์ได้แก่

- BTAB	- HTAB
- HEX	- SPACE
- CHKOFF	- DEL
- ⟶	- BS
- ⟵	- NL
- SKIP	- LC
- CTL	- UC
- CAN	- ESC

Signal / Symbol	DB1	DB2	DB3	DB4	DB5	DB6	DB7	Signal / Symbol	DB1	DB2	DB3	DB4	DB5	DB6	DB7
ESC	1	1	0	1	1	0	0	@	0	0	0	0	0	0	1
BTAB	0	0	0	0	0	0	0	[	1	1	0	1	1	0	1
HEX	1	0	0	0	0	0	0	NL	0	1	0	1	0	0	0
CHK OFF	0	1	0	0	0	0	0	BS	0	0	0	1	0	0	0
←	1	1	1	0	1	0	0	UC	1	1	1	1	0	0	0
→	1	1	0	0	0	0	0	A	1	0	0	0	0	0	1
SKIP	0	0	1	0	0	0	0	S	1	1	0	0	1	0	1
1	1	0	0	0	1	1	0	D	0	0	1	0	0	0	1
2	0	1	0	0	1	1	0	F	0	1	1	0	0	0	1
3	1	1	0	0	1	1	0	G	1	1	1	0	0	0	1
4	0	0	1	0	1	1	0	H	0	0	0	1	0	0	1
5	1	0	1	0	1	1	0	J	0	1	0	1	0	0	1
6	0	1	1	0	1	1	0	K	1	1	0	1	0	0	1
7	1	1	1	0	1	1	0	L	0	0	1	1	0	0	1
8	0	0	0	1	1	1	0	;	1	1	0	1	1	1	0
9	1	0	0	1	1	1	0	:	0	1	0	1	1	1	0
0	0	0	0	0	1	1	0	]	1	0	1	1	1	0	1
-	1	0	1	1	0	1	0	LC	1	1	1	1	0	0	0
^	0	1	1	1	1	0	1	Z	0	1	0	1	1	0	1
¥	0	0	1	1	1	0	1	X	0	0	0	1	1	0	1
DEL	1	1	1	1	1	1	1	C	1	1	0	0	0	0	1
CAN	0	0	0	1	1	0	0	V	0	1	1	0	1	0	1
Q	1	0	0	0	1	0	1	B	0	1	0	0	0	0	1
W	1	1	1	0	1	0	1	N	0	1	1	1	0	0	1
E	1	0	1	0	0	0	1	M	1	0	1	1	0	0	1
R	0	1	0	0	1	0	1	,	0	0	1	1	0	1	0
T	0	0	1	0	1	0	1	.	0	1	1	1	0	1	0
Y	1	0	0	1	1	1	1	/	1	1	1	1	0	1	0
U	1	0	1	0	1	0	1	HTAB	1	0	0	1	0	0	0
I	1	0	0	1	0	0	1	(space)	0	0	0	0	0	1	0
O	1	1	1	1	0	0	1								
P	0	0	0	0	1	0	1								

ตารางที่ 3.2 แสดงตารางรหัส ASC II ของแป้นพิมพ์ NEC

เมื่อมีการกดปุ่มบนแป้นพิมพ์จะมีสัญญาณ STB ออกมาพร้อมกับรหัสข้อมูล โดยสัญญาณจะมีการเปลี่ยนแปลงทุกครั้งที่มีการกดปุ่มข้อมูล รูปที่ 3.7 แสดงความสัมพันธ์ของการกดปุ่มข้อมูลบนแป้นพิมพ์กับการเกิดสัญญาณรหัสข้อมูลและสัญญาณ STB



รูปที่ 3.7 แสดงความสัมพันธ์ของการกดปุ่มแป้นพิมพ์กับสัญญาณรหัสข้อมูลและสัญญาณ STB

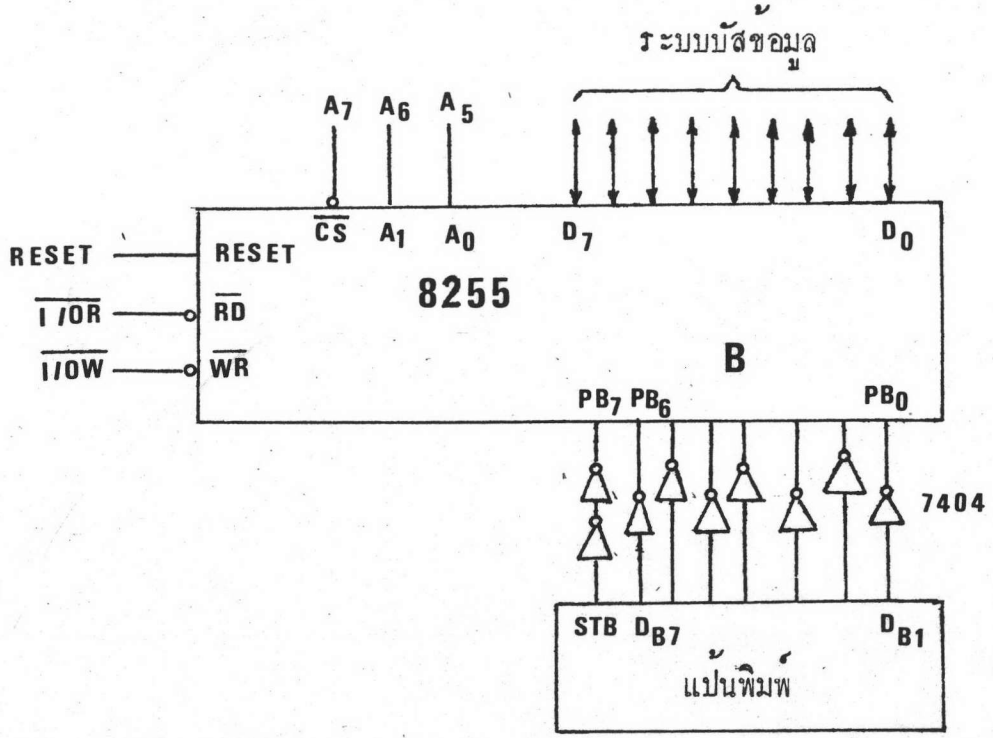
จากรูปที่ 3.7 เมื่อมีการกดปุ่มข้อมูล สัญญาณ STB จะมีการเปลี่ยนแปลงจากลอจิก "1" เป็นลอจิก "0" ถ้ายังไม่ปล่อยมือจากปุ่มกดข้อมูล สัญญาณ STB ก็ยังคงเป็นลอจิก "0" อยู่จนกว่าจะปล่อยมือจากปุ่มกดสัญญาณ STB จึงจะเปลี่ยนจากลอจิก "0" กลับเป็นลอจิก "1" ในการกดปุ่มข้อมูลแต่ละครั้ง สัญญาณรหัสข้อมูลจะถูกแลทช์ไว้ที่วงจรเอาท์พุทของแป้นพิมพ์ เช่น กดปุ่ม A รหัส 01000001 จะถูกแลทช์ไว้จนกว่าจะกดปุ่มข้อมูลใหม่ สัญญาณรหัสข้อมูลที่เกิดขึ้นจะถูกแลทช์แทนรหัส 01000001

รูปที่ 3.8 เป็นวงจรการต่อแป้นพิมพ์ของบริษัท NEC เข้าสู่ระบบบัสข้อมูลโดยใช้ไอซีเบอร์ 8255 ซึ่งเป็นโปรแกรมเมเบิ้ลเพอริเฟอรัลอินเทอร์เฟส (Programmable Peripheral Interface) ที่ถูกออกแบบให้เป็นไอโอพอร์ท (I/O Port) มีทั้งหมด 3 พอร์ทคือ พอร์ท A, B และ C แต่ละพอร์ทมีขนาด 8 บิท การกำหนดพอร์ทของไอซีเบอร์ 8255 ให้เป็นอินพุทหรือเอาท์พุทนั้นขึ้นอยู่กับโปรแกรมเริ่มต้น (Initial Program) ที่ให้กับไอซีเบอร์ 8255 ในครั้งแรก ในการออกแบบนี้ ผู้วิจัยได้กำหนดพอร์ท A เป็นเอาท์พุท พอร์ท B เป็นอินพุทและพอร์ท C เป็นทั้งอินพุทและเอาท์พุท ดังนั้นไอซีเบอร์ 8255 จึงเป็นตัวเชื่อมต่อระหว่างแป้นพิมพ์กับระบบบัสข้อมูลโดยสายข้อมูล D_{B1} - D_{B7} จากแป้นพิมพ์ทั้ง 7 สาย ก่อนที่จะต่อเข้าขา PB_0 - PB_6 ของพอร์ท B จะต้องผ่านอินเวอร์เตอร์เกท (Inverter Gate) เสียก่อน เพราะรหัส ASC II ที่เกิดขึ้นจากแป้นพิมพ์เป็นรหัสที่อินเวอร์ท (Invert) กับรหัส ASC II มาตรฐาน สำหรับสัญญาณ STB ของแป้นพิมพ์ก่อนต่อเข้าขา PB_7 ของไอซีเบอร์ 8255 จะต้องผ่านอินเวอร์เตอร์เกท 2 ตัว เพราะเอาท์พุทของสัญญาณ STB จากแป้นพิมพ์เป็นโอเพลคอลเลคเตอร์ (Open Collector) จึงทำให้ระดับสัญญาณ STB ในขณะไม่กดปุ่มบนแป้นพิมพ์จะเป็นลอจิก "1" เมื่อมีการกดปุ่มบนแป้นพิมพ์สัญญาณ STB จะเปลี่ยนเป็นลอจิก "0" ดังนั้นสัญญาณ STB ใช้เป็นที่บอกสภาวะการกดปุ่มข้อมูล ขาสัญญาณ $\overline{RD}$ และ $\overline{WR}$ ต่อกับสัญญาณ $\overline{I/OR}$ และ $\overline{I/OW}$ ของไอซีเบอร์ 8228 เพื่อควบคุมการอ่านและเขียนข้อมูลระหว่างไอซีเบอร์ 8255 กับระบบบัสข้อมูล โดยขา A_0 และ A_1 และ $\overline{CS}$ ที่ต่อกับแอดเดรสบัส A_5, A_6 และ A_7 ใช้ควบคุมตำแหน่งของพอร์ทตามตารางที่ 3.3 ขาสัญญาณ RESET ต่อโดยตรงมาจากไอซีเบอร์ 8224 ที่ขา RESET เพื่อต้องการให้ไอซีเบอร์ 8255 ทำรีเซสพร้อมกับระบบทั้งหมด การทำรีเซสของไอซีเบอร์ 8255 ทุกครั้งจะต้องโหลดโปรแกรมเริ่มต้นให้ใหม่ทุกครั้ง สำหรับขาข้อมูล D_0 - D_7 จะถูกต่อเข้าระบบบัสข้อมูลเพื่อใช้รับข้อมูล

8255 BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A → DATA BUS
0	1	0	1	0	PORT B → DATA BUS
1	0	0	1	0	PORT C → DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS → PORT A
0	1	1	0	0	DATA BUS → PORT B
1	0	1	0	0	DATA BUS → PORT C
1	1	1	0	0	DATA BUS → CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS → 3-STATE
1	1	0	1	0	ILLEGAL CONDITION

ตารางที่ 3.3 แสดงวิธีการเลือกพอร์ทไอซีเบอร์ 8255



รูปที่ 3.8 แสดงวงจรเชื่อมต่อแป้นพิมพ์กับระบบบัสข้อมูล

3.5 โปรแกรมโมนิเตอร์

การเขียนโปรแกรมโมนิเตอร์ต้องมีความสัมพันธ์กับการออกแบบวงจรภายในเทอร์มินอล เพราะโปรแกรมโมนิเตอร์เป็นโปรแกรมที่ใช้ควบคุมการทำงานของไมโครโปรเซสเซอร์ซึ่งมีหน้าที่บังคับการทำงานของวงจรต่าง ๆ ภายในเทอร์มินอลตามลำดับคำสั่งของโปรแกรมโมนิเตอร์ แผนผังทั้งหมดของโปรแกรมโมนิเตอร์แสดงในรูปที่ 3.9 ซึ่งประกอบด้วยโปรแกรมควบคุม 3 ส่วนคือ

1. โปรแกรมของหน่วยแสดงผล เป็นโปรแกรมที่เขียนขึ้นเพื่อนำข้อมูลที่เก็บในหน่วยความจำไปแสดงผลบนหน้าจอภาพ โปรแกรมส่วนนี้เป็นผลงานวิจัยของนายสำนวน หิรัญวงษ์ ในหัวข้อวิจัย "การพัฒนาเครื่องรับโทรทัศน์ให้เป็นหน่วยแสดงผล" โดยโปรแกรมของหน่วยแสดงผลจะดึงข้อมูลที่เก็บอยู่ในหน่วยความจำตั้งแต่ตำแหน่ง 1024 ถึง 2047 ไปแสดงผลบนหน้าจอภาพเทอร์มินอล จากแผนผังโปรแกรมโมนิเตอร์ในรูปที่ 3.9 ถ้ามีเครื่องหมาย * อยู่ในแผนผังใดก็จะเป็นแผนผังของหน่วยแสดงผล

2. โปรแกรมในส่วนรับส่งข้อมูลของอินเทอร์เฟส เป็นโปรแกรมที่เขียนขึ้น เพื่อนำข้อมูลจากวงจรภายนอกเข้าเก็บในหน่วยความจำและนำข้อมูลจากหน่วยความจำ ไปยังวงจรภายนอกของเทอร์มินอล ซึ่งโปรแกรมส่วนนี้เป็นผลงานวิจัยของนายเพ็ชรรัตน์ อารีรักษ์ ในหัวข้อวิจัย "การออกแบบสร้างวงจรรับส่งข้อมูลของเทอร์มินอล" จากแผนผังโปรแกรมโมนิเตอร์ในรูปที่ 3.9 ถ้ามีเครื่องหมาย ** อยู่ในแผนผังใดก็จะเป็นแผนผังในส่วนรับส่งข้อมูลของอินเทอร์-เฟส กับวงจรภายนอก

3. โปรแกรมระบบควบคุมข้อมูลของเทอร์มินอล เป็นโปรแกรมที่ผู้วิจัย ที่ผู้วิจัยเขียนขึ้นสำหรับควบคุมข้อมูลระหว่างแป้นพิมพ์กับหน่วยความจำและระหว่างหน่วยความจำกับอินเทอร์เฟสในแผนผังโปรแกรมโมนิเตอร์ในรูปที่ 3.9 ถ้าไม่มีเครื่องหมาย * แสดงอยู่ภายใน ก็จะเป็นแผนผังในส่วนโปรแกรมระบบควบคุมข้อมูลของเทอร์มินอล ซึ่งมีคุณสมบัติดังนี้

- โปรแกรมนี้เขียนขึ้นสำหรับการรับส่งข้อมูลของเทอร์มินอลแบบฟูลล์เพล็ก โดยวิธีแคแร็คเตอร์โมดเท่านั้น

- การออกแบบปุ่มกดควบคุมของเทอร์มินอล ผู้วิจัยได้ออกแบบปุ่มควบคุมตามหัวข้อ 2.5 ในบทที่ 2 จึงจำเป็นจะต้องดัดแปลงปุ่มควบคุมบนแป้นพิมพ์ของบริษัท NEC ให้เหมาะสมกับงานวิจัย เพราะแป้นพิมพ์ของบริษัท NEC ที่นำมาใช้งานวิจัยนี้ไม่ได้ออกแบบ ไว้สำหรับเครื่องเทอร์มินอลโดยทั่วไป จำเป็นต้องมีการดัดแปลงการใช้งานของปุ่มกดควบคุมโดยแก้ไขการทำงานของโปรแกรมระบบควบคุมข้อมูลบางตัวเช่น

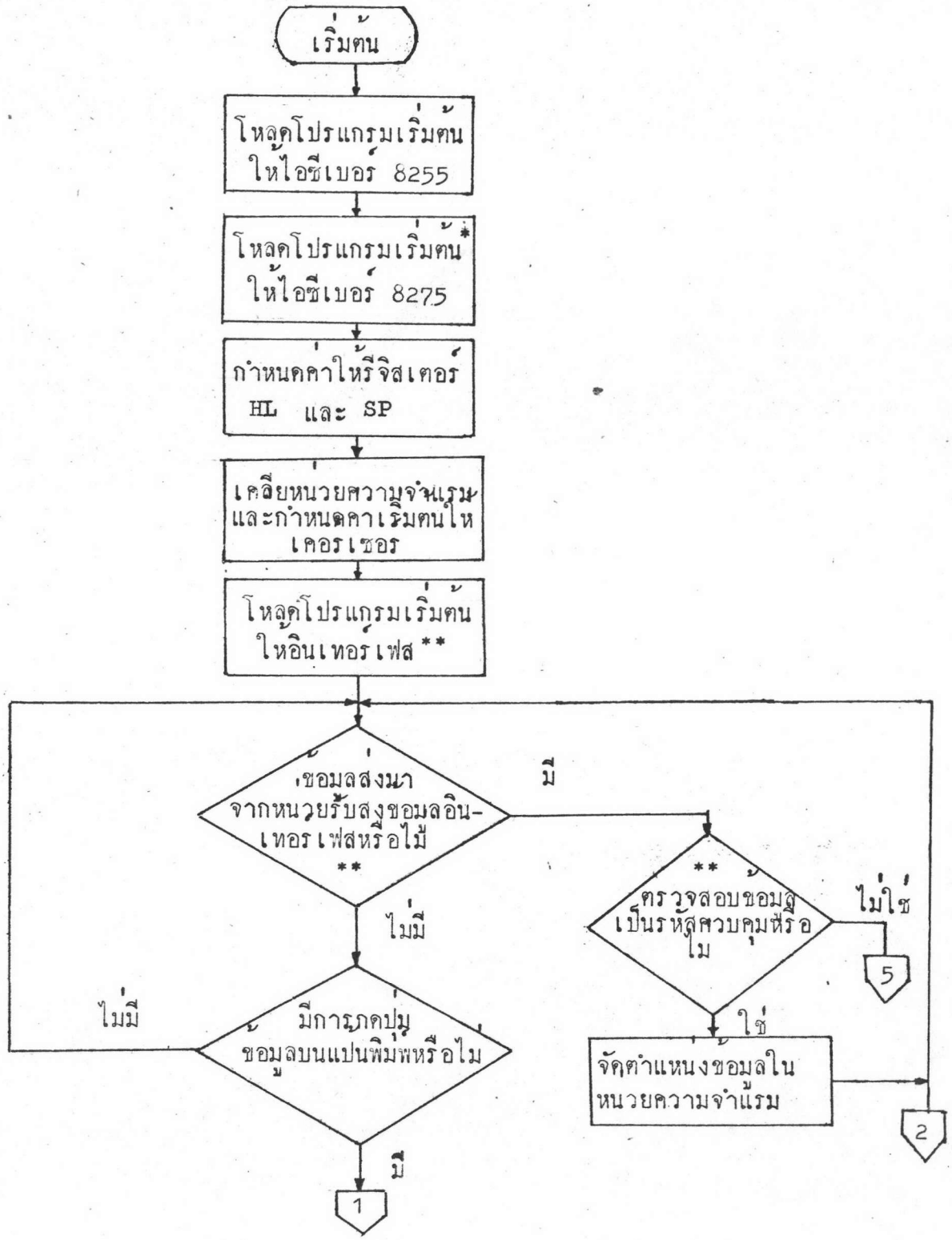
ปุ่มกด	HOME	ใช้ปุ่มกด	CAN	แทน
ปุ่มกด	↑	ใช้ปุ่มกด	HEX	แทน
ปุ่มกด	↓	ใช้ปุ่มกด	CHK OFF	แทน
ปุ่มกด	CLEAR	ใช้ปุ่มกด	SKIP	แทน
ปุ่มกด	ENTER	ใช้ปุ่มกด	BS	แทน

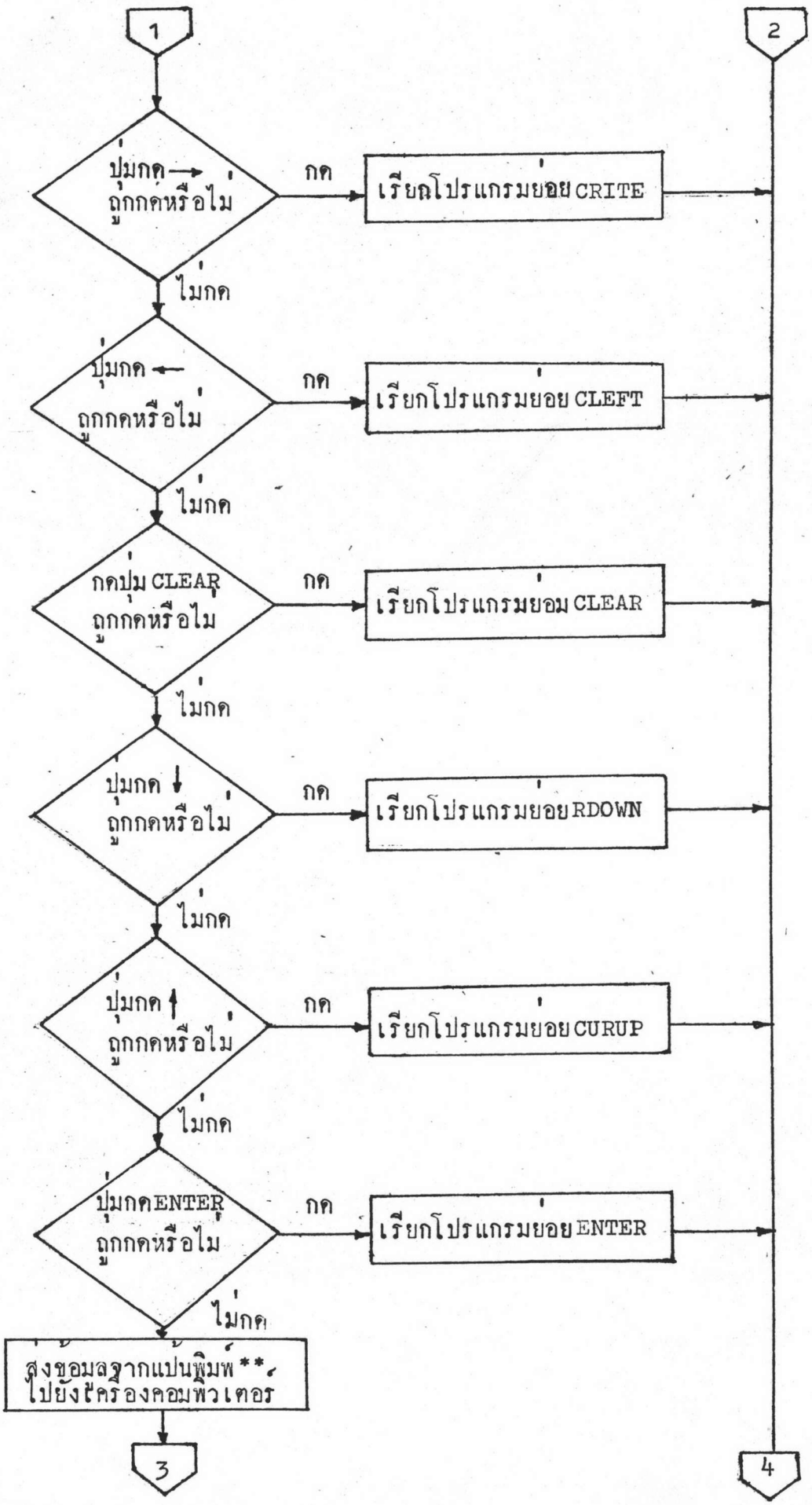
- โปรแกรมระบบควบคุมข้อมูลใช้หน่วยความจำตั้งแต่ตำแหน่ง 1024 ถึง 2047 ส่วนการใช้รีจิสเตอร์บางตัวภายในไมโครโปรเซสเซอร์เพื่อควบคุมการแสดงผลมีดังนี้

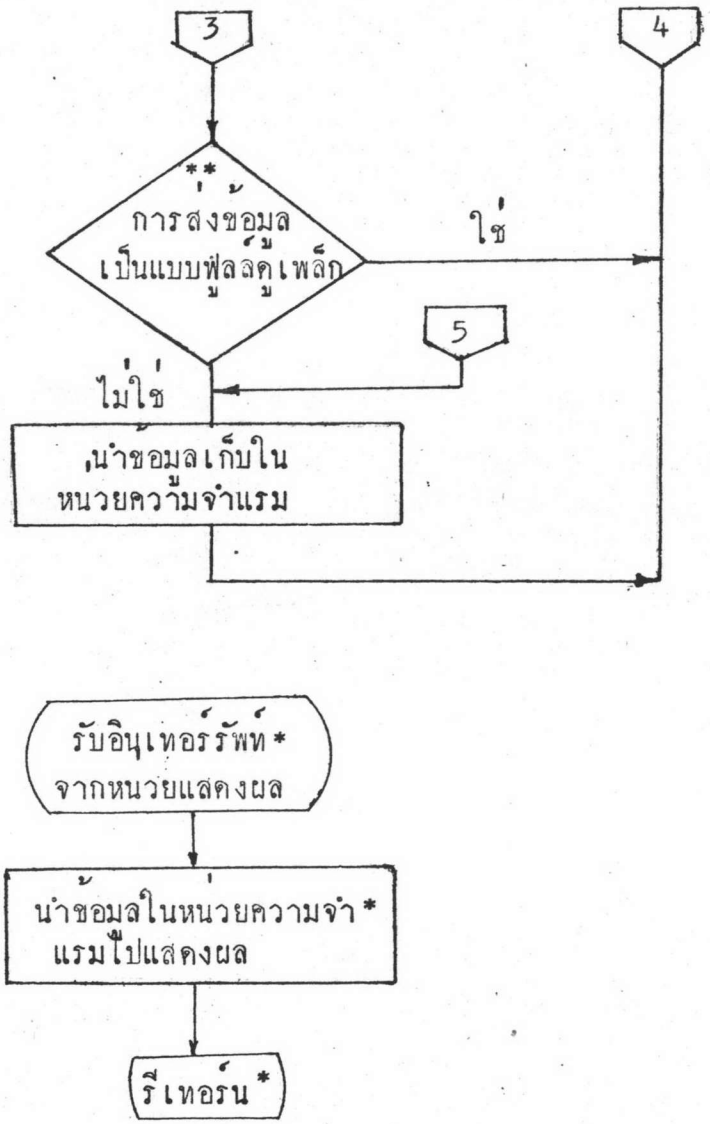
รีจิสเตอร์	ลักษณะการใช้งาน
B	ใช้เก็บค่าตำแหน่งของเคอร์เซอร์ที่แสดงบนจอภาพทางด้านโรว์ (Row)
C	ใช้เก็บค่าตำแหน่งของเคอร์เซอร์ที่แสดงบนจอภาพทางด้านคอลัมน์ (Column)
HL	แทนตำแหน่งในหน่วยความจำซึ่งเป็นที่เก็บข้อมูล สำหรับแสดงผล

- โปรแกรมระบบควบคุมข้อมูลสำหรับหน่วยแสดงผลที่มีขนาด 16 บรรทัด และในแต่ละบรรทัดบรรจุอักขระได้ 64 ตัว โดยหน่วยแสดงผลจะนำข้อมูลที่เก็บในหน่วยความจำ ตั้งแต่ตำแหน่ง 1024 ถึง 2047 ไปแสดงผล ซึ่งข้อมูลที่เก็บในหน่วยความจำตำแหน่ง 1024 จะใช้เป็นตำแหน่งเริ่มต้นเพื่อให้หน่วยแสดงผลอ่านข้อมูลนำไปแสดงผลบนหน้าจอภาพเป็นตัวแรกของบรรทัดแรก โดยหน่วยแสดงผลจะอ่านข้อมูลในหน่วยความจำ 64 ไบท์ต่อการแสดงผลบนหน้าจอภาพใน 1 บรรทัด

3.6 แผนผังโปรแกรมโมนิเตอร์และโปรแกรมควบคุมข้อมูล




1
2
ปุ่มกด → ถูกกดหรือไม่
กด
เรียกโปรแกรมย่อย CRITE
ไม่กด
ปุ่มกด ← ถูกกดหรือไม่
กด
เรียกโปรแกรมย่อย CLEFT
ไม่กด
กดปุ่ม CLEAR ถูกกดหรือไม
กด
เรียกโปรแกรมย่อม CLEAR
ไม่กด
ปุ่มกด ↓ ถูกกดหรือไม่
กด
เรียกโปรแกรมย่อย RDOWN
ไม่กด
ปุ่มกด ↑ ถูกกดหรือไม่
กด
เรียกโปรแกรมย่อย CURUP
ไม่กด
ปุ่มกด ENTER ถูกกดหรือไม
กด
เรียกโปรแกรมย่อย ENTER
ไม่กด
ส่งข้อมูลจากแป้นพิมพ์ ** ไปยังเครื่องคอมพิวเตอร์
3
4



รูปที่ 3.9 แสดงแผนผังโปรแกรมโมนิเตอร์

```
;
;          MONITOR  CRT  PROGRAM
;
           ORG      0
           MVI      A, 8AH
           OUT      ME3
           LXI      H,0400H
           LXI      SP,STACK
;
;          POWER  UP
;          CLEAR MEMORY & HOME CURSOR
;
LOPP       MVI      A,20H
           MOV      M,A
           MOV      A,H
           CPI      0BH
           JZ       CMMHP
CMMLP      INX      H
           JMP      LOPP
CMMHP      MOV      A,L
           CPI      0FFH
           JZ       CHOM1
           JMP      CMMLP
CHOM1      LXI      H,0400H
           LXI      B,0000H
```

```
;
;           MONITOR  START
;
INPUT       CALL     INKEY
            CPI      0FH
            JZ       INPUT
            CPI      03H          ; CURSOR RIGHT
            JZ       CRT
            CPI      17H          ; CURSOR LEFT
            JZ       CLEF
            CPI      04H          ; CLEAR MEMORY
            JZ       CLR
            CPI      18H          ; HOME
            JZ       CHOM
            CPI      02H          ; CURSOR DOWN
            JZ       CROLD
            CPI      01H          ; CURSOR UP
            JZ       CUP
            CPI      08H          ; ENTER
            JZ       ENT
            MOV      M,A          ; DATA KB TO MM
            CALL     CRITE
            JMP      INPUT
```

```
;
;           CALL SUBROUTINE
;
CRT         CALL      CRITE
            JMP       INPUT
CLEF        CALL      CLEFT
            JMP       INPUT
CLR         CALL      CLEAR
            JMP       INPUT
CHOM        CALL      HOME
            JMP       INPUT
CROLD       CALL      RDOWN
            JMP       INPUT
CUP         CALL      CURUP
            JMP       INPUT
ENT         CALL      ENTER
            JMP       INPUT
;
;           INKEY
;
INKEY       IN        ME1
            ORA       A
LOPKB       JP        INKEY
            IN        ME1
            ORA       A
            JM        LOPKB
            IN        ME1
            RET
```

```
;
;           CURSOR UP
;
CURUP       LXI     D,0040H
            MOV     A,B
            CPI     00H
            JZ      UPCR
            DCR     B
            MOV     A,L
            SBB     E
            MOV     L,A
            MOV     A,H
            SBB     D
            MOV     H,A
            JMP     UPRET
UPCR        LXI     H,07C0H
            MOV     A,C
            ADD     L
            MOV     L,A
            MVI     B,0FH
UPRET       RET
```

```
;
;           CURSOR DOWN
;
RDOWN       LXI     D,0040H
            MOV     A,B
            CPI     0FH
            JZ      ROBIN
            INR     B
            MOV     A,L
            ADD     E
            MOV     L,A
            MOV     A,H
            ADC     D
            MOV     H,A
DORET       RET
ROBIN       CALL    ROLLUP
            LXI     H,07C0H
            MOV     A,C
            ADD     L
            MOV     L,A
            MVI     B,0FH
            JMP     DORET
```

```
;
;           CURSOR LEFT
;
CLEFT       MOV     A,B
            CPI     OOH
            JZ      CTEST
            MOV     A,C
            CPI     OOH
            JZ      CBCR
CDCR        DCR     C
            DCX     H
            JMP     CLRET
CTEST       MOV     A,C
            CPI     OOH
            JZ      CBEGIN
            JMP     CDCR
CBEGIN      MVI     B,OFH
            MVI     C,3FH
            LXI     H,07FFH
            JMP     CLRET
CBCR        DCR     B
            MVI     C,3FH
            DCX     H
CLRET       RET
```

```
;
;           MOVE CURSOR RIGHT
;
CRITE       MOV     A,B
            CPI     OFH
            JZ      CREND
            MOV     A,C
            CPI     3FH
            JZ      CRINR
CRNEXT      INR     C
            INX     H
            JMP     CRRET
CREND       MOV     A,C
            CPI     3FH
            JZ      CRLRP
            JMP     CRNEXT
CRINR       INX     H
            INR     B
            MVI     C,OOH
            JMP     CRRET
CRLRP       CALL    ROLLUP
            MVI     B,OFH
            MVI     C,OOH
            LXI     H,07COH
CRRET       RET
```

```
;
;           CLEAR MEMORY
;
CLEAR       LXI     H,0400H
LOP         MVI     A,20H
            MOV     M,A
            MOV     A,H
            CPI     0BH
CMML        JZ      CMMH
            INX     H
            JMP     LOP
CMMH        MOV     A,L
            CPI     0FFH
            JZ      BEGIN
            JMP     CMML
BEGIN       LXI     H,0400H
            LXI     B,0000H
            RET
;
;           HOME
;
HOME        LXI     B,0000H
            LXI     H,0400H
            RET
;
;           ENTER
;
ENTER       MOV     A,L
            SUB     C
            MOV     L,A
            MVI     C,00H
            RET
```

```
;
;           SCROLL   MODE
;
ROLLUP      LXI      H,0440H
            LXI      D,0400H
LOPUP       MOV      A,H
            CPI      0BH
            JZ       LOPRET
            MOV      A,M
            XCHG
            MOV      M,A
            INX      H
            INX      D
            XCHG
            JMP      LOPUP
LOPRET      RET
STACK       EQU      0BFFH
ME1         EQU      00H
ME3         EQU      20H
            END
```