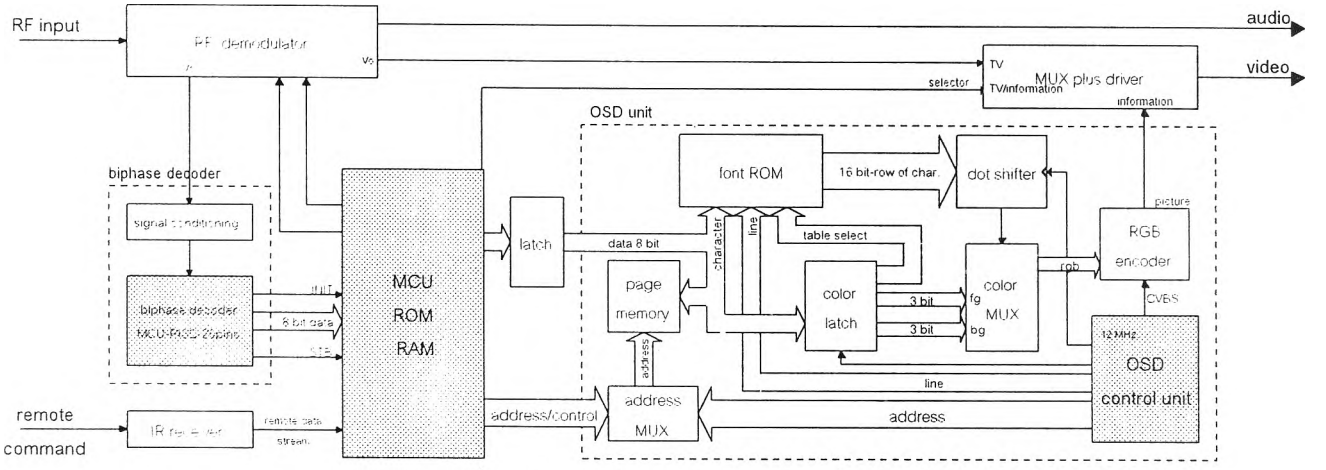
บทที่ 5
เครื่องรับ

5.1 โครงสร้างฮาร์ดแวร์ของเครื่องรับ



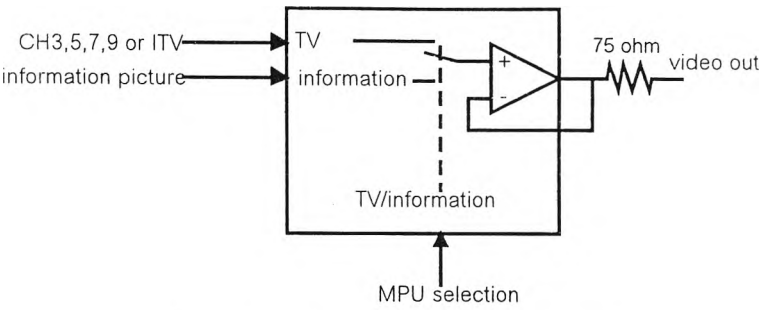
รูปที่ 5.1 โครงสร้างฮาร์ดแวร์ของเครื่องรับ

โครงสร้างฮาร์ดแวร์ของเครื่องรับประกอบด้วย

1. หน่วยประมวลผลกลาง ทำหน้าที่ควบคุมการทำงานหลักและประสานงานกับส่วนต่างๆ ประกอบด้วย

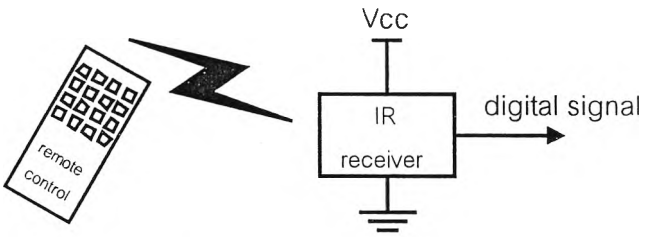
- ไมโครคอนโทรลเลอร์ความเร็วสูง ใช้ไมโครคอนโทรลเลอร์ตระกูล 8051 เบอร์ DS80C320 ของบริษัท Dallas Semiconductor ทำงานที่ความถี่สัญญาณนาฬิกา 24 เมกะเฮิรตซ์ จำนวน 1 ตัว
- หน่วยความจำหลัก (ROM) ขนาด 64 กิโลไบต์ ใช้ไอซีเบอร์ M27C256A จำนวน 1 ตัว
- หน่วยความจำชั่วคราว (RAM) ขนาด 16 กิโลไบต์ ใช้ไอซีเบอร์ UT6264PC-70LL จำนวน 2 ตัว

2. หน่วยควบคุมสัญญาณภาพขาออก ทำหน้าที่สลับภาพระหว่างโมดปกติกับโมดแสดงผลข้อมูล ไอซีมีออป-แอมป์ในตัวด้วย ดังรูปที่ 5.2 ใช้ไอซีเบอร์ MAX454 จำนวน 1 ตัว โดยออป-แอมป์ทำหน้าที่ขยายขับสัญญาณทางด้านขาออก ที่มีแบนด์วิดท์สูงเพียงพอสำหรับสัญญาณโทรทัศน์ และกำหนดความต้านทานขาออกเท่ากับ 75 โอห์ม



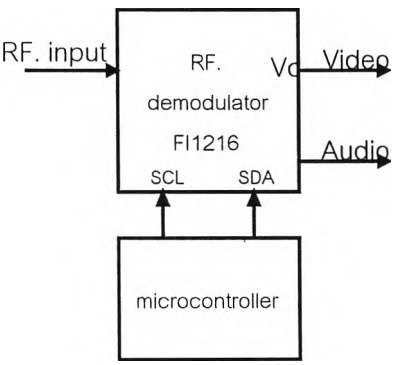
รูปที่ 5.2 หน่วยควบคุมสัญญาณภาพขาออก

3. ส่วนถอดรหัสพัลส์ ใช้วงจรดีมอดูเลตสัญญาณอินฟราเรด ทำหน้าที่แปลงรหัสสัญญาณพัลส์จากเครื่องควบคุมระยะไกลให้เป็นสัญญาณดิจิตอลส่งให้หน่วยประมวลผลประมวลกลางดังรูปที่ 5.3



รูปที่ 5.3 วงจรถอดรหัสพัลส์

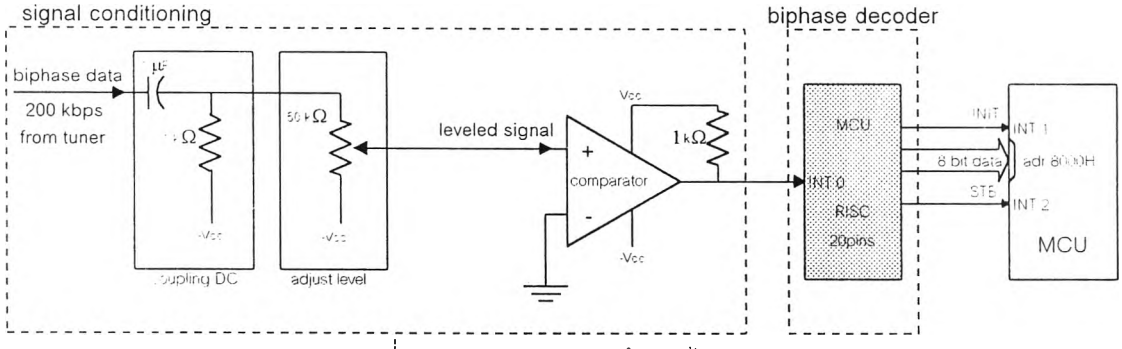
4. วงจรดีมอดูเลตสัญญาณวิทยุ ทำหน้าที่แปลงสัญญาณวิทยุของโทรทัศน์ ให้เป็นสัญญาณภาพและเสียงเบสแบนด์ ควบคุมด้วยการติดต่อแบบไอสแควร์ซี ใช้ไอสแควร์ซีจูนเนอร์เบอร์ FI1216/H/IEC ของบริษัท ฟิลิปส์ จำนวน 1 ตัว ดังรูปที่ 5.4



รูปที่ 5.4 การควบคุมไอสแควร์ซีจูนเนอร์

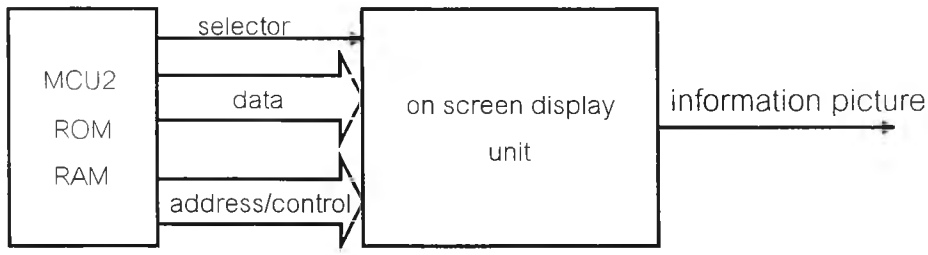
5. หน่วยถอดรหัสไบเฟส ทำหน้าที่ถอดรหัสแบบไบเฟสและแปลงข้อมูลแบบอนุกรมให้เป็นข้อมูลแบบขนาน ส่งข้อมูลให้หน่วยประมวลผลกลาง ดังรูปที่ 5.5 ประกอบด้วย

- ไมโครคอนโทรลเลอร์ขนาดเล็ก ทำหน้าที่ถอดรหัสข้อมูลไบเฟส ใช้ไมโครคอนโทรลเลอร์ความเร็วสูงสถาปัตยกรรมแบบ RISC ของบริษัทเอทเมล เบอร์ AT90S2313-10PC จำนวน 1 ตัว ทำงานที่ความถี่สัญญาณนาฬิกา 10 เมกกะเฮิรตซ์ ประมวลผลคำสั่งได้ 10 ล้านคำสั่งในหนึ่งวินาที
- ส่วนปรับสภาพสัญญาณ ใช้ปรับสภาพสัญญาณข้อมูลไบเฟสให้อยู่ในช่วงระดับแรงดันตามมาตรฐาน TTL ใช้ไอซี comparator เบอร์ LM319 จำนวน 1 ตัว



รูปที่ 5.5 หน่วยถอดรหัสข้อมูลไบเฟส

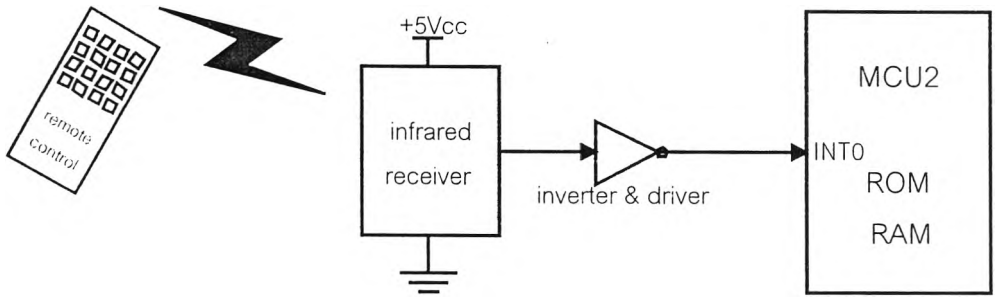
6. ส่วนแสดงผลภาพ ทำหน้าที่แสดงผลภาพข้อมูล โดยทำงานอย่างอิสระกับส่วนอื่น กล่าวคือ จะสร้างสัญญาณภาพโทรทัศน์อยู่ตลอดเวลา ด้วยหน่วยควบคุมการแสดงผล ดังรูปที่ 5.6 ประกอบด้วยหน่วยความจำภาพ ใช้สำหรับเก็บข้อมูลการแสดงผลภาพ ฮาร์ดแวร์สำหรับควบคุมการแสดงผลและส่วนสังเคราะห์ภาพ รายละเอียดขอกล่าวในบทที่ 6



รูปที่ 5.6 หน่วยแสดงผลภาพ

5.2 การทำงานของเครื่องรับ

5.2.1 การถอดรหัสคำสั่งจากเครื่องควบคุมระยะไกล



รูปที่ 5.7 การขับสัญญาณเครื่องควบคุมระยะไกล

สัญญาณารหัสพัลส์จากเครื่องควบคุมระยะไกลถูกดีมอดูเลตด้วยอุปกรณ์รับอินฟราเรด จากนั้นจะถูกขับด้วยอินเวอร์เตอร์ ดังรูปที่ 5.7 เพื่อป้อนเข้าขาอินเทอร์รัพท์ศูนย์ของไมโครคอนโทรลเลอร์

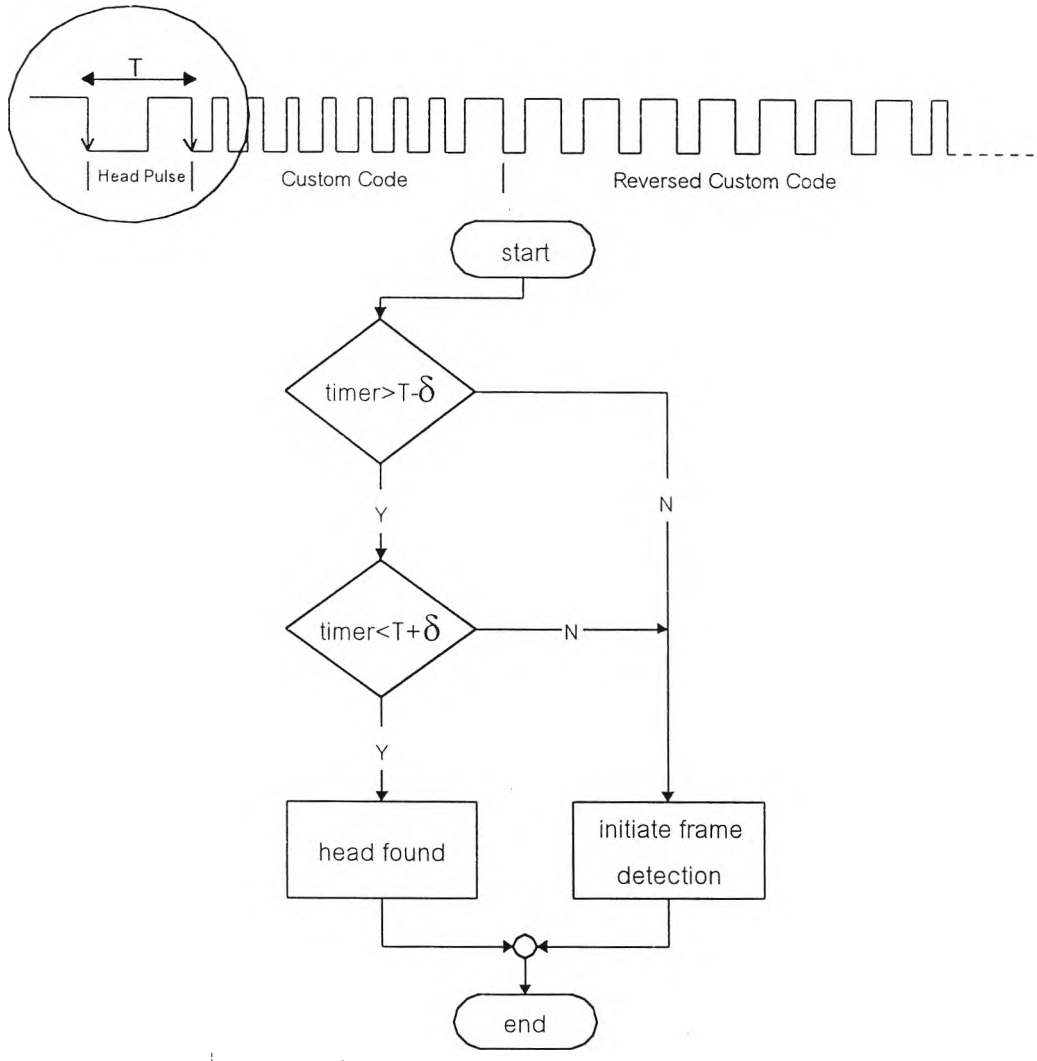
โปรแกรมอินเทอร์รัพท์ศูนย์ใช้วงจรนับภายในจับเวลา เพื่อหาความต่างของคาบเวลาทำให้ทราบสถานะบิตข้อมูล การถอดรหัสคำสั่งเริ่มจากการตรวจสอบส่วนหัวเฟรม หากพบก็จะอ่านข้อมูลเข้ามาทีละบิตโดยอาศัยการตรวจสอบความกว้างของระดับสัญญาณจนครบ 4 ไบต์ ประกอบด้วย Custom code, Data code, Reversed custom code และ Reversed data code

ข้อมูล Custom code กับ Data code จะถูกนำมาตรวจสอบด้วยผลคอมพลีเมนต์ของ Reversed custom code และ Reversed data code หากไม่พบความผิดพลาดก็จะตรวจสอบ Custom code ของเครื่องควบคุมระยะไกล หากถูกต้องจะนำ Data code ไปเก็บไว้ในรีจิสเตอร์และเซตบิตสถานะควบคุมเพื่อให้ทราบว่ามีคำสั่งใหม่เข้ามา เมื่อโปรแกรมหลักมาตรวจสอบพบ ก็จะเคลียร์บิตและนำคำสั่งไปประมวลผล

โปรแกรมอินเทอร์รัพท์ศูนย์ที่ใช้ถอดรหัสคำสั่งจากเครื่องควบคุมระยะไกล จะเก็บสถานะการรับข้อมูลไว้ในรีจิสเตอร์ควบคุมเพื่อช่วยประมวลผล เนื่องจากสัญญาณอินพุตมีเพียงสัญญาณเดียว โครงสร้างของโปรแกรมแสดงด้วยภาษาบรรยายโปรแกรมระดับที่ 2 ได้ดังนี้

```
MODULE: INTERRUPT 0
        STOP TIMER
        IF DETECT_FRAME THEN
                IF HEAD_FOUND THEN
                        DETECT BODY
                ELSE
                        DETECT HEAD
                END IF
        ELSE
                CLEAR HEAD_FOUND
                INITIATE TIMER
        END IF
        IF BITCOUNT>=32 THEN
                IF VERIFY CUSTOMER DATA =TRUE THEN
                        SET REMOTE_COMMAND
                ELSE
                        CLEAR REMOTE_COMMAND
                END IF
        ELSE
                START TIMER
        END IF
END:
```

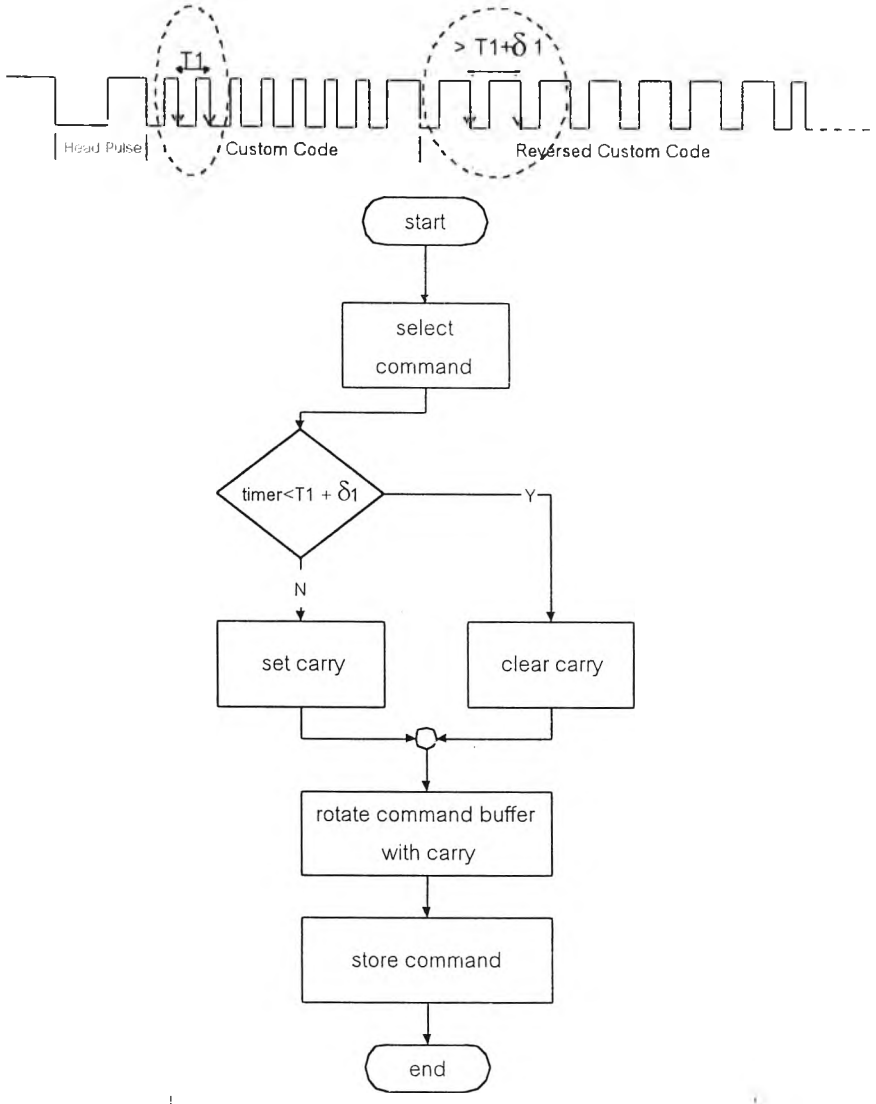
โครงสร้างโปรแกรมถอดรหัสคำสั่งจากเครื่องควบคุมระยะไกล ดังรูปที่ 5.8 และ รูปที่ 5.9



รูปที่ 5.8 โครงสร้างโปรแกรมสำหรับตรวจจับส่วนหัวเฟรม

การตรวจจับส่วนหัวเฟรม ใช้การตัดสินคาบเวลาดังรูปที่ 5.8 โดยคาบเวลาจะต้องมากกว่า T-δ และ น้อยกว่า T+δ จึงจะถือว่าเป็นส่วนหัวจริง โดย δ คือระยะ offset เท่ากับ 0.1T และ T เท่ากับ 13.5 มิลลิวินาที

จากรูปที่ 5.9 โปรแกรมจะตรวจสอบความกว้างของคาบเวลาเพื่อตัดสินความหมายของบิต โดยช่วงเวลาของบิตค่า "0" จะต้องไม่เกิน T1+δ1 โดย T1 เท่ากับ 1.12 มิลลิวินาที T2 เท่ากับ 2.26 มิลลิวินาที และ δ1 เท่ากับ 0.1 T1 จากนั้นเลื่อนบิต carry ไปใส่ในรีจิสเตอร์ ทำเช่นนี้จนครบ 8 บิต ก็จะได้ข้อมูลหนึ่งไบต์ รีจิสเตอร์เก็บข้อมูลทั้ง 4 ตัวจะถูกเลือกจากรีจิสเตอร์ควบคุม



รูปที่ 5.9 โครงสร้างโปรแกรมส่วนตรวจจับข้อมูลไบต์คำสั่ง

5.2.2 การควบคุมการทำงานของจูนเนอร์ผ่านทางสัญญาณควบคุมแบบไอสแควร์ซี

หน่วยประมวลผลกลางควบคุมไอสแควร์ซีจูนเนอร์ ด้วยข้อมูลคำสั่งควบคุมที่ถูกเก็บไว้ในหน่วยความจำถาวร ยกเว้นไบต์ CB1 ดังตารางที่ 5.1

	ADDRESS	DIV1	DIV2	CB1	CB2
CH 3	C0H	05H	E2H	CEH (08H)	A0H
CH4	C0H	06H	4FH	CEH (08H)	A0H
CH5	C0H	0DH	62H	CEH (08H)	90H
CH7	C0H	0EH	3AH	CEH (08H)	90H
CH9	C0H	0FH	19H	CEH (08H)	90H
CH11	C0H	10H	02H	CEH (08H)	90H
ITV	C0H	20H	1BH	CEH (08H)	90H

ตารางที่ 5.1 ตารางแสดงพารามิเตอร์สำหรับควบคุมไอสแควร์ซีจูนเนอร์

หมายเหตุ: ไบต์ CB1 ถูกตัดสินจากโปรแกรมควบคุม โดยมีค่าเท่ากับ 08H สำหรับการจูนช่องแบนด์ต่ำกว่าหรือเท่าเดิม และ CEH ใช้สำหรับการจูนช่องแบนด์สูงกว่า

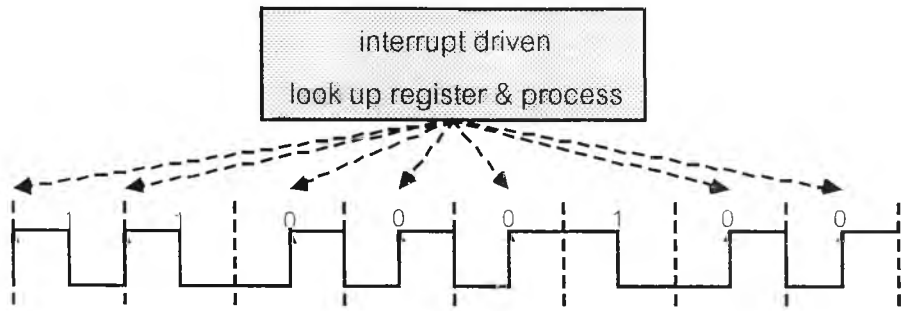
DIV1 และ DIV2 เป็นพารามิเตอร์เพื่อใช้จูนช่อง CB1 และCB2 สำหรับควบคุมกลไกการ จูนดังกล่าวไว้ในบทที่ 2 การสร้างสัญญาณไอสแควร์ซีเพื่อจูนใช้วิธีเขียนโปรแกรมจำลองรูปแบบขึ้น โดยใช้ขาสัญญาณของไมโครคอนโทรลเลอร์สองขา และมีการตรวจสอบบิตตอบรับในการส่งข้อมูลแต่ละไบต์ด้วย

การทำงานของโปรแกรม เริ่มจากโปรแกรมหลัก ตรวจสอบรีจิสเตอร์ภายในและพบว่ามีการเปลี่ยนช่องก็จะอ่านพารามิเตอร์จูนช่องจากตาราง โดยอ้างอิงช่องโทรทัศน์จากรีจิสเตอร์ภายใน จากนั้นจะส่งคำสั่งควบคุมให้กับโปรแกรมย่อยที่ติดต่อกับจูนเนอร์แบบไอสแควร์ซี การทำงานของโปรแกรมการจูนช่องโทรทัศน์ด้วยการเชื่อมต่อแบบไอสแควร์ซีอธิบายด้วยภาษาบรรยายโปรแกรมระดับที่ 2 อยู่ในภาคผนวก ข.

5.2.3 การถอดรหัสข้อมูลไบเฟส

การถอดรหัสข้อมูลไบเฟสใช้ไมโครคอนโทรลเลอร์ขนาดเล็กความเร็วสูงสถาปัตยกรรมแบบ

RISC สามารถทำงาน 10 ล้านคำสั่งภายในหนึ่งวินาทีมาใช้ โปรแกรมใช้หลักการอินเทอร์รัพท์ โดยโปรแกรมอินเทอร์รัพท์จะทำงานร่วมกับรีจิสเตอร์บอกสถานะการรับข้อมูล ดังรูปที่ 5.10 เพื่อให้ทราบสถานะของการถอดรหัสและแปลงข้อมูลไบเฟสให้เป็นข้อมูลแบบขนาน



รูปที่ 5.10 โครงสร้างโปรแกรมแบบ Interrupt-driven

การถอดรหัสใช้ฟังก์ชันของค่าบิตก่อนหน้าและเวลาที่ระยะตรวจสอบต่างๆ ในการตัดสินบิตข้อมูล

Detected bit = f (t, previous bit)

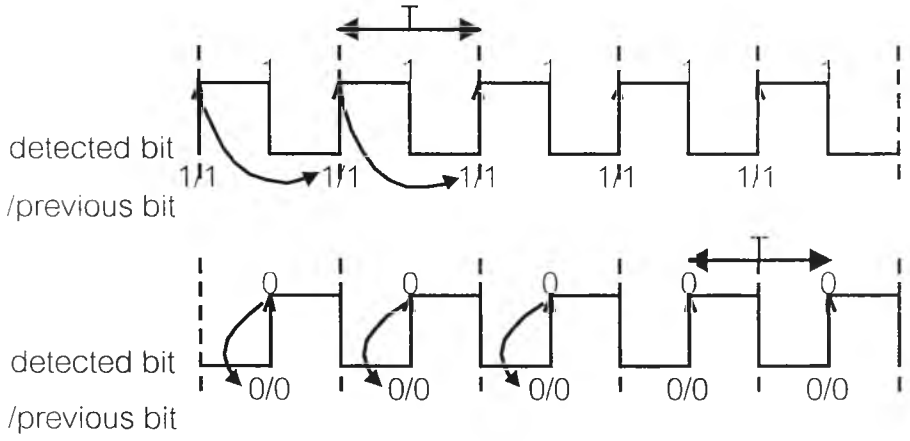
โดย t คือ เวลาที่จับโดย timer ของ CPU

จากวิธีการรับข้อมูลในรูปที่ 5.10 ระยะตรวจสอบข้อมูลสามารถแบ่งออกเป็น 3 แบบ ตามลักษณะการวางตัวของบิตหนึ่งและศูนย์ ดังนี้

1. ระยะตรวจสอบเท่ากับ T

ดังรูปที่ 5.11 การวางตัวของข้อมูลที่มีความซ้ำกันได้แก่ ศูนย์ หรือ หนึ่ง อย่างต่อเนื่อง ทำให้จุดตรวจสอบห่างกันคงที่เท่ากับ T หากพบว่าระยะห่างของจุดตรวจสอบไม่เกิน $T+\delta$ โปรแกรมก็จะตรวจสอบว่าบิตก่อนหน้านี้เป็น 1 หรือ 0 จากบิตสถานะ จึงจะตัดสินบิตนั้นๆได้ โดย δ เท่ากับ 0.1 T

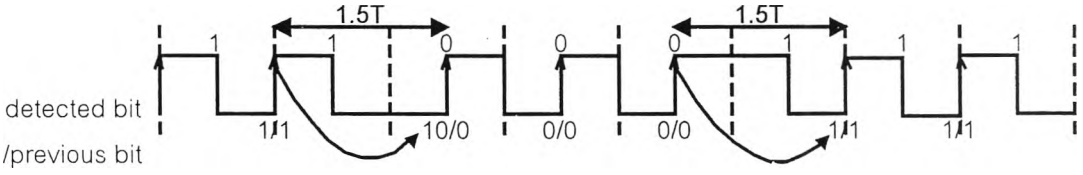
การตัดสินบิตที่มีค่า "1" จะเป็นการตัดสินบิตค่า "1" ที่อยู่ก่อนหน้า และการตัดสินบิตที่มีค่า "0" จะเป็นการตัดสินบิต ณ ตำแหน่งนั้น



รูปที่ 5.11 ข้อมูลไบเฟสที่มีระยะตรวจสอบเท่ากับ T

2. ระยะตรวจสอบเท่ากับ 1.5T

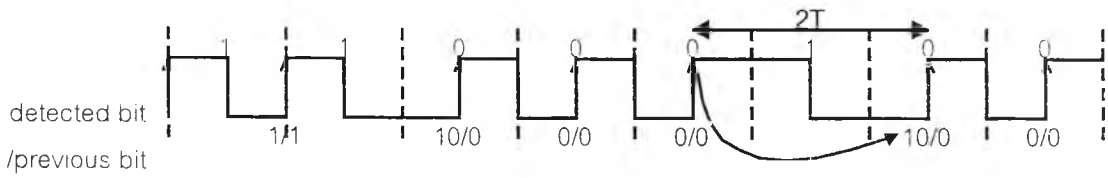
การที่ข้อมูลเปลี่ยนแปลงจาก 1 เป็น 0 หรือ จาก 0 เป็น 1 ระยะห่างของจุดตรวจสอบจะเท่ากับ 1.5 T เมื่อพบว่าระยะห่างของจุดตรวจสอบมีค่าอยู่ในช่วงระหว่าง 1.5T-δ กับ 1.5T+δ โดยที่ δ = 0.1T กรณีค่าบิตก่อนหน้าเป็น "1" จะถอดรหัสข้อมูลเป็น 1 และ 0 พร้อมกันสองบิต และเปลี่ยนค่าบิตก่อนหน้าเป็น "0"



รูปที่ 5.12 ลักษณะข้อมูลไบเฟสที่มีระยะตรวจสอบเท่ากับ 1.5T

กรณีที่บิตเก็บค่าบิตก่อนหน้าเป็น "0" จะตัดสินบิตเป็น "1" รวมทั้งเปลี่ยนค่าบิตก่อนหน้าเป็น "1" ด้วย

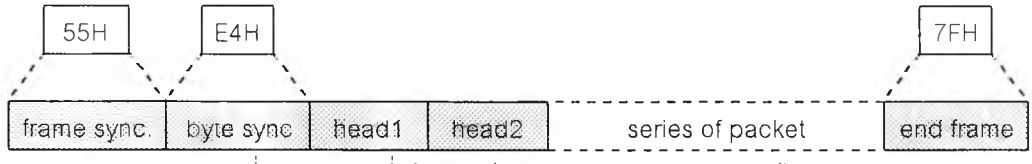
3. ระยะตรวจสอบเท่ากับ 2T



รูปที่ 5.13 ลักษณะข้อมูลไบเฟสที่มีระยะตรวจสอบเท่ากับ 2T

กรณีนี้ เป็นลักษณะเฉพาะของกรณีที่ 2 กล่าวคือ ในกรณีที่บิตข้อมูลเปลี่ยนกลับไปมาระหว่าง "0" เป็น "1" และเป็น "0" ดังรูปที่ 5.13 โดยมีบิตค่าหนึ่งเพียงบิตเดียวนั้น จะเกิดกรณีที่เวลาระหว่างจุดตรวจสอบห่างกันเท่ากับ 2T ขึ้น เมื่อโปรแกรมพบว่าระยะห่างจุดตรวจสอบอยู่ระหว่าง $2T-\delta$ กับ $2T+\delta$ โดย $\delta = 0.1T$ ก็จะตรวจสอบค่าบิตก่อนหน้าว่าเป็น "0" หรือไม่ หากพบว่าเป็น "1" ก็จะเริ่มตั้งต้นกระบวนการรับข้อมูลใหม่ทั้งหมดเนื่องจากลักษณะการถอดรหัสผิดจากโครงสร้างของข้อมูลที่กำหนดไว้ แต่หากเป็น "0" จะถอดรหัสเป็นค่า 1 และ 0 โดยให้ค่าบิตก่อนหน้า เป็น "0" ตามเดิม

ในช่วงว่างระหว่างเฟรม เครื่องส่งจะส่งสัญญาณที่มีค่า "1" ออกมาอย่างสม่ำเสมอ ซึ่งทำให้เครื่องรับสามารถตรวจจับการติดต่อของเครื่องส่งได้อยู่ตลอดเวลา จึงประยุกต์การนับเวลาของจุดตรวจสอบมาใช้ โดยกำหนดให้ค่าเวลา $2T+\delta$ โดย $\delta = 0.1\ T$ จะไม่ทำให้วงจรนับเวลาเกิดล้น (overflow) ดังนั้นหากเกิดการล้น (overflow) ขึ้นแสดงว่าจุดตรวจสอบที่ได้รับไม่ปกติ อาจเกิดปัญหาทางด้านส่งได้ จึงดับสัญญาณไฟที่แสดงสถานะ "รับ" ที่เครื่องรับ



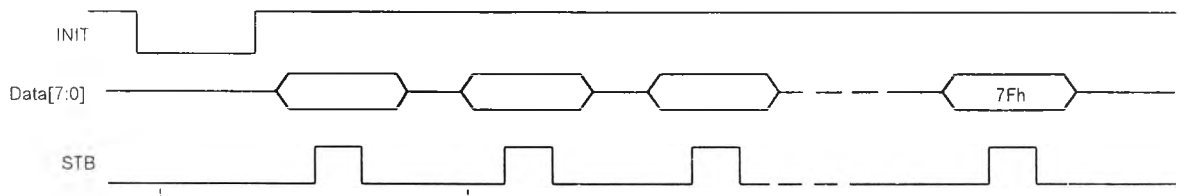
รูปที่ 5.14 ส่วนที่ใช้ซิงโครไนซ์ระดับเฟรมและระดับไบต์

จากข้อกำหนดการส่งข้อมูล เฟรมข้อมูลประกอบด้วยเฟรมซิงโครไนซ์สำหรับการซิงโครไนซ์ระดับเฟรมซึ่งมีค่าเท่ากับ 55H ข้อมูลสำหรับการซิงโครไนซ์ระดับไบต์มีค่า E4H และส่วนหัวของข้อมูล

ซึ่งทำหน้าที่ระบุชนิด รายละเอียดของเฟรมรวมไปถึงรหัสเครื่องปลายทาง แล้วจึงตามด้วยตัวข้อมูล และปิดท้ายด้วยรหัสสิ้นสุดเฟรม 7FH ดังรูปที่ 5.14

โปรแกรมจะคอยดักเฟรมซิงโครไนซ์อยู่เสมอ เมื่อพบก็จะดักข้อมูลสำหรับซิงโครไนซ์ระดับไบต์ เพื่อให้ทราบถึงจุดอ้างอิงของบิตเริ่มต้นของไบต์ จากนั้นจึงจะเข้าสู่สภาวะถอดรหัสข้อมูล โดยจะถอดรหัสทีละบิตจนครบ 1 ไบต์ แล้วจึงแปลงเป็นข้อมูลแบขนานและส่งให้กับหน่วยประมวลผลกลาง ตั้งแต่ไบต์ head1, head2 ไปจนถึงรหัสสิ้นสุดเฟรม (7FH) การทำงานของโปรแกรมส่วนการถอดรหัสไบเฟสด้วยภาษาบรรยายโปรแกรมระดับที่ 2 แสดงในภาคผนวก ข.

โปรแกรมจะส่งสัญญาณ INIT และ STB ด้วย โดยสัญญาณ INIT เป็นสัญญาณบอกสถานะเริ่มต้นการส่งข้อมูลให้หน่วยประมวลผลกลาง และสัญญาณ STB สำหรับบอกตำแหน่งข้อมูลบนสัญญาณข้อมูล ดังรูปที่ 5.15



รูปที่ 5.15 การส่งข้อมูลไบเฟสที่ผ่านการถอดรหัสแล้วให้กับหน่วยประมวลผลกลาง

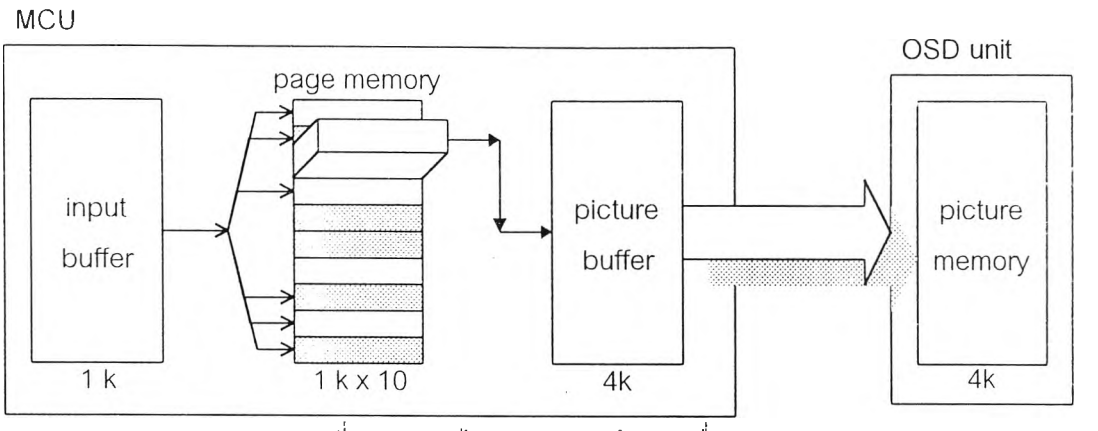
การส่งข้อมูลเริ่มจากสัญญาณ INIT หน่วยประมวลผลกลางก็จะเตรียมตัวรับข้อมูล เมื่อสัญญาณมี STB แสดงตำแหน่งข้อมูลบนบัสข้อมูล หน่วยประมวลผลกลางจะอ่านข้อมูลเข้ามาไว้ในหน่วยความจำชั่วคราวครั้งละหนึ่งไบต์ จนถึงไบต์สุดท้ายของเฟรมซึ่งได้แก่รหัส 7FH ก็จะเลิกรับข้อมูลและกลับมาอยู่ในสถานะพร้อมรับเฟรมใหม่ ข้อมูลที่อ่านเข้ามาจะถูกเก็บไว้ในหน่วยความจำบัฟเฟอร์ก่อนที่จะถูกนำไปประมวลผลต่อไป

5.2.4 การเก็บและเคลื่อนไหวของข้อมูล

หน่วยประมวลผลกลางแบ่งพื้นที่หน่วยความจำชั่วคราวเป็น 3 ส่วน ได้แก่

1. หน่วยความจำสำรองสำหรับรับข้อมูล

ทำหน้าที่พักข้อมูลที่รับจากส่วนถอดรหัสข้อมูลไบเฟส



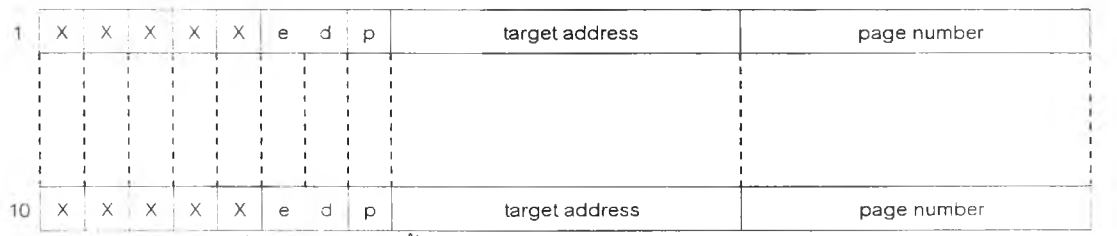
รูปที่ 5.16 การไหลของเฟรมข้อมูลเพื่อแสดงผลภาพ

2. หน่วยความจำสำหรับเก็บเฟรมข้อมูล

page memory	address	page memory	address
1	0800H-0BFFH	6	1C00H-1FFFH
2	0C00H-0FFFH	7	2000H-23FFH
3	1000H-13FFH	8	2400H-27FFH
4	1400H-17FFH	9	2800H-2BFFH
5	1800H-1BFFH	10	2C00H-2FFFH

ตารางที่ 5.2 address ของหน่วยความจำที่จองไว้สำหรับเก็บเฟรมข้อมูล

จากตารางที่ 5.2 แสดง address ของหน่วยความจำที่เก็บเฟรมข้อมูลสำหรับการแสดงผลภาพ โดยสามารถเก็บได้ถึง 10 เฟรมซึ่งแทนหน้าข้อมูล 10 หน้า ตารางเนื้อหาสำหรับเก็บพารามิเตอร์ที่เกี่ยวกับการแสดงผล ดังรูปที่ 5.17



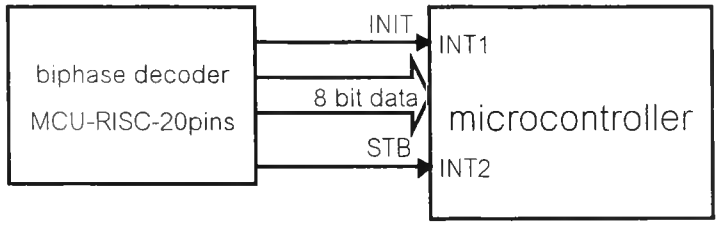
รูปที่ 5.17 ตารางเนื้อหาหน่วยความจำสำหรับเก็บเฟรมข้อมูล

- ไบต์ที่หนึ่ง ทำหน้าที่เก็บสถานะของการแสดงผลภาพ
  - บิต e, "1" แทนหน่วยความจำมีข้อมูล, "0" แทนหน่วยความจำไม่มีข้อมูล
  - บิต d, "1" แทนเฟรมข้อมูลกำลังแสดงผลภาพอยู่, "0" แทนเฟรมข้อมูลไม่ได้ถูกแสดงผล
  - บิต p, "1" แทนเฟรมข้อมูลแบบสาธารณะ, "0" แทนเฟรมข้อมูลทั่วไป
- ไบต์ที่สอง เป็นตำแหน่งเริ่มต้นของหน่วยความจำเฟรมข้อมูล โดยเก็บค่า addressไบต์สูงเท่านั้น เพราะไบต์ต่ำมีค่า 00H
- ไบต์ที่สาม ทำหน้าที่เก็บเลขที่หน้าสำหรับแสดงผล โดยให้หน้าที่ 1-3 เป็นข้อมูลส่วนตัว และหน้าที่ 4 – 6 เป็นข้อมูลสาธารณะ

3. หน่วยความจำสำรองสำหรับการแสดงผลภาพ

ทำหน้าที่เก็บข้อมูลสำหรับแสดงผลภาพที่ผ่านการประมวลผลแล้ว ก่อนที่จะเขียนลงไปที่หน่วยความจำภาพของส่วนแสดงผลภาพทั้งหมดพร้อมกันเพื่อลดเวลาที่ใช้เขียนข้อมูลซึ่งมีผลกระทบต่อการเห็นภาพกระพริบให้น้อยที่สุด

การเคลื่อนไหวของข้อมูลเริ่มจากการรับข้อมูล จากส่วนถอดรหัสแบบไบเฟส โดยสัญญาณบอกสถานะเริ่มต้นบอกให้หน่วยประมวลผลกลางอีนาเบิลการอินเทอร์รัพท์ของอินเทอร์รัพท์สอง ซึ่งมีโปรแกรมย่อยคอยอ่านข้อมูลนำไปเก็บไว้ในหน่วยความจำสำรองสำหรับข้อมูล เมื่อเต็มจะแก้ไขรีจิสเตอร์ให้มีความหมายว่า "เต็ม"



รูปที่ 5.18 การติดต่อส่งข้อมูลระหว่างส่วนถอดรหัสกับหน่วยประมวลผลกลาง

โปรแกรมสำหรับรับข้อมูลจากส่วนถอดรหัสไบเฟส เขียนแทนด้วยภาษาบรรยายโปรแกรมระดับที่ 2 ได้ดังนี้

```
MODULE: INTERRUPT 1
        ENABLE INTERRUPT 2
        LOAD DESTINATION POINTER TO INPUT BUFFER
END:
MODULE: INTERRUPT 2
        IF NOT INPUT_BUFFER_FULL THEN
                READ DATA
                STORE TO INPUT BUFFER
                INCREASE POINTER
                IF DATA = 7FH THEN
                        DISABLE INTERRUPT 2
                        SET INPUT_BUFFER_FULL
                END IF
        ELSE
                DISABLE INTERRUPT 2
        END IF
END:
```

ในส่วนของโปรแกรมหลัก เมื่อโปรแกรมหลักตรวจรีจิสเตอร์ควบคุมพบว่าหน่วยความจำสำรองสำหรับเก็บข้อมูลเต็ม ก็จะตรวจสอบตารางเนื้อหาว่ามีหน้าข้อมูลดังกล่าวถูกเก็บอยู่หรือไม่ หากพบว่ามีจะเขียนข้อมูลลงในบล็อกข้อมูลเดิมนั้น หรือหากไม่พบก็จะเลือกบล็อกข้อมูลที่ว่างเก็บข้อมูล ในกรณีข้อมูลเต็มทั้ง 10 หน้าก็จะเลือก บล็อกเก่าที่สุด โดยอาศัยหลักการคิววงกลมในการเขียนข้อมูลใหม่ลงไปแทนที่

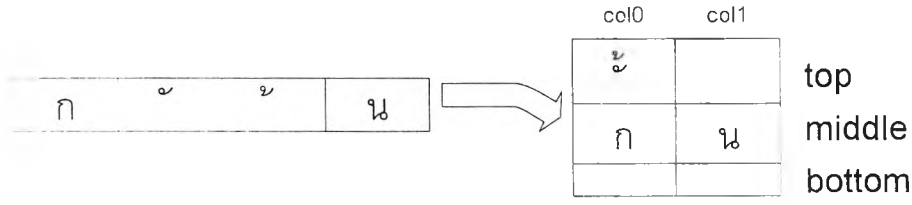
ขั้นตอนการอ่านข้อมูลไปเก็บในหน่วยความจำเฟรมข้อมูล จะตรวจสอบการเข้ารหัสพาริตีคี่ด้วย ถ้าถูกต้องจะถอดรหัสโดยเปลี่ยนพาริตีบิตเป็นศูนย์ หากไม่ถูกต้องจะเขียนรหัสซึ่งแปลเป็นตัวอักขระ "ว่าง" ลงไปแทน เพื่อให้โปรแกรมประมวลผลเป็นค่า "ช่องว่าง" แทนค่าที่ผิดไป การทำงานของโปรแกรมหลักส่วนย้ายข้อมูลจากหน่วยความจำสำรองไปเก็บในหน่วความจำเฟรมข้อมูลด้วยภาษาบรรยายโปรแกรมระดับที่ 2 แสดงในภาคผนวก ข.

เมื่อมีคำสั่งจากเครื่องควบคุมระยะไกลในโมดแสดงภาพข้อมูล โปรแกรมหลักจะอ่านตารางเนื้อหาเพื่อหาหน้าข้อมูล หากพบก็ประมวลผลและบันทึกลงในหน่วยความจำสำรอง เมื่อประมวลผลครบ 1 หน้าจะเขียนไปที่หน่วยความจำภาพพร้อมกัน หรือหากไม่พบก็จะแสดงหน้า "คอย" การทำงานของโปรแกรมหลักตามคำสั่งที่อ่านจากรีจิสเตอร์สถานะของอุปกรณ์ควบคุมระยะไกล อธิบายด้วยภาษาบรรยายโปรแกรมระดับที่ 2 แสดงในภาคผนวก ข.

การแสดงผลภาพโทรทัศน์ระบบ PAL จะแสดง 25 ภาพต่อ 1 วินาที หรือ 1 ภาพใช้เวลา 40 มิลลิวินาที จากการคำนวณ พบว่าเวลาที่ใช้เขียนข้อมูลทั้งหมดประมาณ 10 มิลลิวินาที หรือเท่ากับหนึ่งในสี่ของเวลาที่ใช้แสดงผลหนึ่งภาพ จะเห็นว่าเวลาที่รบกวนการแสดงผลภาพมีน้อยมาก

5.2.5 การประมวลผลเฟรมข้อมูลเพื่อแสดงผลภาพ

จากทฤษฎีเบื้องต้นที่ได้กล่าวไว้ในบทที่ 2 ในการประมวลผลเฟรมข้อมูลเพื่อแสดงผลภาพโดยหนึ่งตัวอักขระถูกแบ่งเป็น 3 ระดับ ดังรูปที่ 5.19



รูปที่ 5.19 การประมวลผลตัวอักขระเพื่อแสดงผล

เมื่อมีการเรียกหน้าที่ต้องการ โปรแกรมจะดูตารางเนื้อหาหน่วยความจำเฟรมข้อมูลว่ามีหน้าที่ต้องการอยู่หรือไม่ หากพบว่ามี ก็จะประมวลผลเพื่อแสดงผลภาพ การทำงานของโปรแกรมหลักส่วนประมวลผลการแสดงผลภาพด้วยภาษาบรรยายโปรแกรมระดับที่ 2 แสดงในภาคผนวก ข.

การประมวลผลทำทีละหนึ่งบรรทัดซึ่งถูกแทนด้วยหนึ่งแพ็กเกต โดยใช้หัวแพ็กเกตและสัญลักษณ์สิ้นสุดเฟรมแทนการสิ้นสุดแพ็กเกต ดังตารางที่ 5.3 การประมวลผลแต่ละบรรทัด จะอ่านข้อมูลทีละไบต์ หากพบสัญลักษณ์สิ้นสุดแพ็กเกตก่อน ก็จะเขียนตัวอักขระช่องว่างจนครบ 34 ช่อง หากข้อมูลที่อ่านเข้ามาเป็นอักขระสำหรับแสดงผล ก็จะประมวลผลโดยยึดตารางอ้างอิงตามสถานะที่กำลัง

ประมวลผลอยู่ เช่น หากขณะนี้รีจิสเตอร์สถานะบอกว่ากำลังประมวลผลภาษาไทย ก็จะเข้าสู่โปรแกรมย่อยของส่วนประมวลผลภาษาไทย เป็นต้น หรือหากพบว่าข้อมูลเป็นรหัสแอสกีควบคุม ก็จะเปลี่ยนแปลงรีจิสเตอร์สถานะ ซึ่งได้แก่ การควบคุมตารางอ้างอิง สีพื้น สีตัวอักษร ตัวเอียง และตัวขีดเส้นใต้

Packet #n	Terminator	Packet #n	Terminator
1	Pk2	9	Pk10
2	Pk3	10	pk11
3	Pk4	11	Pk12
4	Pk5	12	Pk13
5	Pk6	13	Pk14
6	Pk7	14	Pk15
7	Pk8	15	Pk16
8	Pk9	16	End of frame

ตารางที่ 5.3 ตารางแสดงสัญลักษณ์สิ้นสุดแพ็กเกตของแต่ละแพ็กเกต

1. การประมวลผลอักขระภาษาอังกฤษ

การแสดงผลภาษาอังกฤษจะใช้พื้นที่แสดงผลระดับกลางเท่านั้น โดยระดับบนและระดับล่างจะเขียนอักขระว่างไว้ แต่ในกรณีที่ต้องแสดงผลตัวอักษรที่มีการขีดเส้นใต้ ก็จะใส่สัญลักษณ์ ขีดเส้นใต้ตรงพื้นที่การแสดงผลช่วงล่าง การทำงานของโปรแกรมสำหรับประมวลผลการแสดงผลภาษาอังกฤษเขียนด้วยภาษาบรรยายโปรแกรมระดับที่ 2 ได้ดังนี้

```
MODULE: ENGLISH COMPILE
        WRITE SPACE TO THE TOP OF SPACE
        WRITE CONTROL BYTE TO THE TOP OF SPACE
        WRITE DATA TO THE MIDDLE OF SPACE
        WRITE CONTROL BYTE TO THE MIDDLE OF SPACE
        IF NOT UNDERLINE THEN
```

```
            WRITE SPACE TO THE BOTTOM OF SPACE
        ELSE
            WRITE UNDERLINE TO THE BOTTOM OF SPACE
        END IF
        WRITE CONTROL BYTE TO THE BOTTOM OF SPACE
END:
```

2. การประมวลผลอักขระกราฟฟิก

ในกรณีการแสดงผลของอักขระการฟฟิก จะคล้ายกับการประมวลผลอักขระภาษาอังกฤษ แต่จะเขียน Code byte ที่พื้นที่การแสดงผลทั้ง 3 ระดับ อธิบายด้วยภาษาบรรยายโปรแกรมระดับที่ 2 ได้ดังนี้

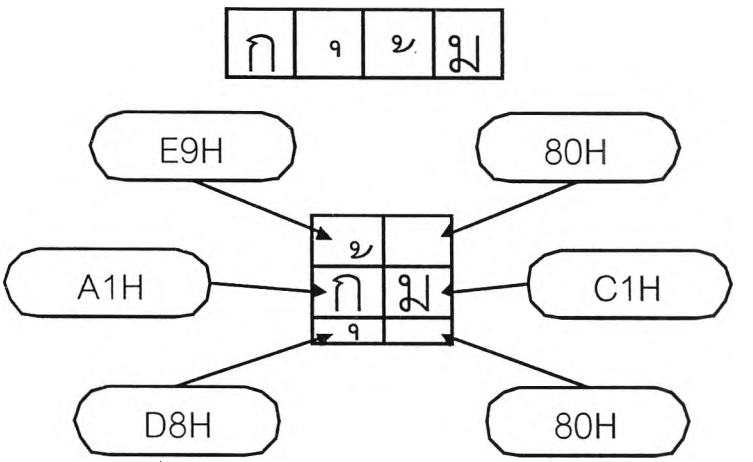
```
MODULE: GRAPHIC COMPILE
        FOR INDEX=1 TO 3
            WRITE CODE BYTE
            WRITE CONTROL BYTE
        NEXT
END:
```

3. การประมวลผลอักขระภาษาไทย

การแสดงผลภาษาไทย ต้องตรวจสอบพยัญชนะและสระประเภทที่วางตัวในพื้นที่การแสดงผลระดับกลาง ส่วนสระและวรรณยุกต์ประเภทที่วางตัวในพื้นที่การแสดงผลระดับบน และ สระและวรรณยุกต์ประเภทที่วางตัวในพื้นที่การแสดงผลระดับล่าง เพื่อให้การประมวลผลเลือกรหัสแสดงผลได้อย่างถูกต้อง จึงแบ่งกลุ่มตัวอักษรโดยยึดหลักการใช้พื้นที่สำหรับการแสดงผล ดังนี้

- กลุ่มที่ 1 ใช้พื้นที่การแสดงผลระดับกลางและล่าง ได้แก่ ญ, ฎ, ฏ, ฐ, ฤ, ฦ, ๅ,ๆ
- กลุ่มที่ 2 ใช้พื้นที่การแสดงผลระดับกลางและบน โดยระดับบนเป็นลักษณะหางตรง ได้แก่ ป, ฟ, ฝ, และ ฬ

- กลุ่มที่ 3 ใช้พื้นที่การแสดงผลระดับกลางและบนเช่นเดียวกันกับกลุ่มที่ 2 แต่ระดับบนเป็นลักษณะหางเอียง ได้แก่ ช, ซ, ส และ ศ
- กลุ่มที่ 4 ใช้พื้นที่การแสดงผลระดับกลางและบนเช่นเดียวกันกับกลุ่มที่ 2 และ 3 แต่ระดับบนเป็นลักษณะเฉพาะของแต่ละตัว ได้แก่ โ, ใ และ ไ
- กลุ่มที่ 5 ใช้พื้นที่การแสดงผลระดับบนแต่ต้องใช้ร่วมกับการแสดงผลกลุ่มอื่นๆด้วย แต่ไม่ใช่วรรณยุกต์ ได้แก่ ไม้หันอากาศ, สระอิ, อี, อึ, อือ และสระอำ
- กลุ่มที่ 6 ใช้พื้นที่การแสดงผลระดับบนแต่ต้องใช้ร่วมกับการแสดงผลกลุ่มอื่นๆด้วย ได้แก่ วรรณยุกต์ เอก โท ตรี จัตวา ไม้ไต่คู้ การันต์ และ ตัวกลม
- กลุ่มที่ 7 พยัญชนะที่เหลือทั้งหมด ซึ่งได้แก่พยัญชนะที่สามารถแสดงผล โดยอาศัยเพียงพื้นที่การแสดงผลระดับกลางเท่านั้น เช่น ม, ก, พ, ร, น, ย, ห, แ ฯลฯ

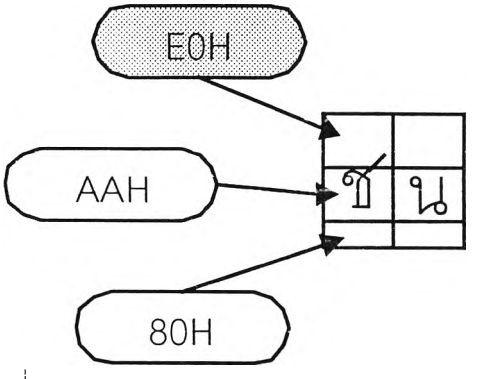


รูปที่ 5.20 ตัวอย่างการประมวลผลคำว่า "กุ้ม"

หนึ่งบรรทัดมี 34 ตัวอักษรหรือแทนด้วย 34 ช่อง การประมวลผลจะทำไปทีละช่อง โดยแต่ละช่องประกอบด้วยพื้นที่การแสดงผลระดับบน, ระดับกลางและระดับล่าง จากรูปที่ 5.20 เมื่อการประมวลผลพบพยัญชนะ ก. ไก่ ก็จะใส่รหัส A1H ที่พื้นที่การแสดงผลระดับกลาง ตัวถัดไปสระอุ จะใส่รหัส D8H ที่พื้นที่การแสดงผลระดับล่าง และไม้โทจะใส่รหัส E9H ที่พื้นที่การแสดงผลระดับบน เมื่อพบพยัญชนะ ม. ม้า ซึ่งเป็นพยัญชนะฐาน ก็จะเริ่มใส่ช่องการแสดงผลช่องต่อไป โดยใส่รหัส C1H ที่พื้นที่การแสดงผลระดับกลาง และรหัส 80H ซึ่งไม่แสดงผลใดๆที่พื้นที่การแสดงผลระดับบนและระดับล่าง รหัสการแสดงผลอ้างอิงจากตารางอักขระ ซึ่งอยู่ในหน่วยความอักขระ (font ROM) ของส่วนแสดงผล ตามตารางที่ 5.4

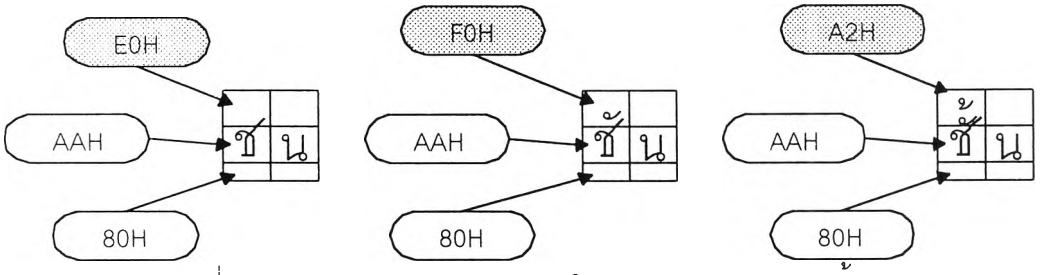
ในกรณีพยัญชนะที่ต้องอาศัยพื้นที่สำหรับแสดงผลระดับบน ดังเช่นคำว่า ชน ซึ่งมีพยัญชนะ ช. ช้าง อยู่ด้วย ดังรูปที่ 5.21 การแสดงผลระดับบนซึ่งเป็นส่วนหางของ ช. ช้างจึงต้องเลือกรหัสการแสดงผลลักษณะหางที่สอดคล้องกัน โดยหางของพยัญชนะแบ่งออกเป็นสองกลุ่มได้แก่ กลุ่มที่ 2 ประกอบด้วย ป, ฟ, ฝ และ ฬ ซึ่งเป็นพวกหางแบบตรง กับ กลุ่มที่ 3 ประกอบด้วย ช, ซ, ส และ ศ ซึ่งเป็นพวกหางเอียง

ตารางที่ 5.4 ตารางอักขระตัวตรงสำหรับการแสดงผลตัวอักษรแบบตัวตรง



รูปที่ 5.21 ตัวอย่างการประมวลผลคำว่า "ชน"

การประมวลผล เมื่อพบพยัญชนะฐาน จะทำการเลือกรหัสทั้งระดับบน กลางและล่าง สำหรับพยัญชนะนั้น เช่น น.หนู จะใส่ช่องว่างที่พื้นที่แสดงผลระดับบนและล่าง แต่ ป.ปลา จะใส่หางในพื้นที่แสดงผลระดับบนด้วย กรณีพบว่ามี สระหรือวรรณยุกต์ตามมา ก็จะย้อนเคอร์เซอร์กลับไปใส่รหัสที่สอดคล้องกัน ดังรูปที่ 5.22 การประมวลผลสำหรับพื้นที่การแสดงผลระดับบนนี้ ใช้หลักการอ้างอิงแบบพลวัตรจากตารางอักขระ โดยการแบ่งกลุ่มพยัญชนะ สระ และวรรณยุกต์ เพื่อจับกลุ่มและชี้ไปยังตารางอักขระ เพื่อเรียกรหัสการแสดงผลระดับบนที่ตรงกับกลุ่มนั้นออกมา ยกตัวอย่าง คำว่า ชั้น เมื่อการประมวลผลพบว่าเป็นพยัญชนะฐานกลุ่ม 3 ซึ่งเป็นหางแบบเอียง บิต Y1 และ Y0 ในรูปที่ 5.23 จะมีค่า 1 และ 0 ตามลำดับ เมื่อพบไม้หันอากาศตัวถัดไป บิต M2 M1 และ M0 จะมีค่า 0 0 1 ตามลำดับ เช่นกันเมื่อพบวรรณยุกต์ไม้โท บิต D2 D1 และ D0 ก็จะมีค่า 0 1 1 ตามลำดับ ซึ่งนำมาเรียงกัน จะได้ค่า 8BH ซึ่งในตารางอักขระพื้นที่การแสดงผลระดับบน จะเป็นรหัส A2H ซึ่งตรงกับการแสดงผลไม้หันอากาศพร้อมกันกับไม้โท และเป็นของตัวพยัญชนะแบบที่ 3 คือมีหางแบบเอียงด้วย



รูปที่ 5.22 ตัวอย่างการประมวลผลคำว่า ชน และ ชัน และ ชั้น

ส่วนการประมวลผลพื้นที่การแสดงผลส่วนล่างนั้นมีเพียงสระอุและสระอูของกลุ่มที่ 1 เท่านั้น อาศัยการตรวจสอบภายในโปรแกรมประมวลผล การทำงานของโปรแกรมสำหรับประมวลผลการแสดงผลภาษาไทยด้วย ภาษาบรรยายโปรแกรมระดับที่ 2 แสดงในภาคผนวก ข.

dynamic pointer	Y1	Y0	M2	M1	M0	D2	D1	D0

Y1Y0	description
00	normal
01	ˋ
10	ˊ
11	reserved

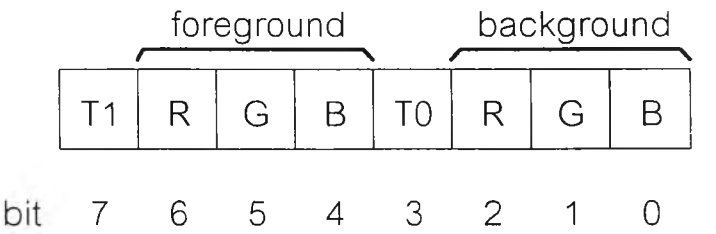
M2M1M0	description
000	normal
001	๑
010	๒
011	๓
100	๔
101	๕
110	ๆ
111	reserved

D2D1D0	description
000	normal
001	๘
010	๑
011	๒
100	๗
101	+
110	๙
111	๐

รูปที่ 5.23 รีจิสเตอร์ชี้ข้อมูลเพื่อช่วยในการประมวลผลการแสดงผลของพื้นที่การแสดงผลระดับบน

4. การประมวลผลรหัสควบคุม

รหัสควบคุมทำหน้าที่ควบคุมการแสดงผลทั้ง ชนิดของภาษา ตัวอักษรหรือการฟฟิก การแสดงผลสี โดยมีรีจิสเตอร์ซึ่งเก็บข้อมูลสีและตารางอ้างอิง ดังรูปที่ 5.24 ทำหน้าที่เก็บสถานะการแสดงผล ณ ตำแหน่งนั้นไว้ โดยจะเปลี่ยนแปลงเมื่อพบรหัสควบคุม ข้อมูลในรีจิสเตอร์จะถูกเขียนในส่วนของ control byte ของแต่ละพื้นที่การแสดงผลระดับบน กลาง และ ล่าง เพื่อให้ส่วนแสดงผลนำไปแสดงผล



รูปที่ 5.24 ไบต์ควบคุมสีและตารางอักขระอ้างอิง

รีจิสเตอร์ควบคุมสีและตารางอักขระ จะควบคุม 3 ลักษณะได้แก่

- ควบคุมสีของตัวอักษร ได้แก่ บิตที่ 6, 5 และ 4 โดยข้อมูล 3 บิตนี้จะถูกส่งป้อนเข้าไปในกลไกการแสดงผลของส่วนแสดงผล จะให้ค่าสีของตัวอักษร

- ควบคุมสีของพื้นหลัง ได้แก่ บิตที่ 2, 1 และ 0 เช่นกันเดียวกัน ข้อมูล 3 บิตนี้จะถูกส่งป้อนเข้าไปในกลไกลการแสดงผลของส่วนแสดงผล จะให้ค่าสีพื้นของตัวอักษรออกมา
- ควบคุมตารางที่ใช้อ้างอิงการแสดงผล ได้แก่ บิตที่ 7 และ 3 หรือ T1 และ T0 โดยสองบิตนี้จะเลือกตารางการแสดงผล แบบภาษาตัวตรง ตัวเอียง หรือ การแสดงผลแบบกราฟฟิก ดังตารางที่ 5.4, ตารางที่ 5.5 และ ตารางที่ 5.6

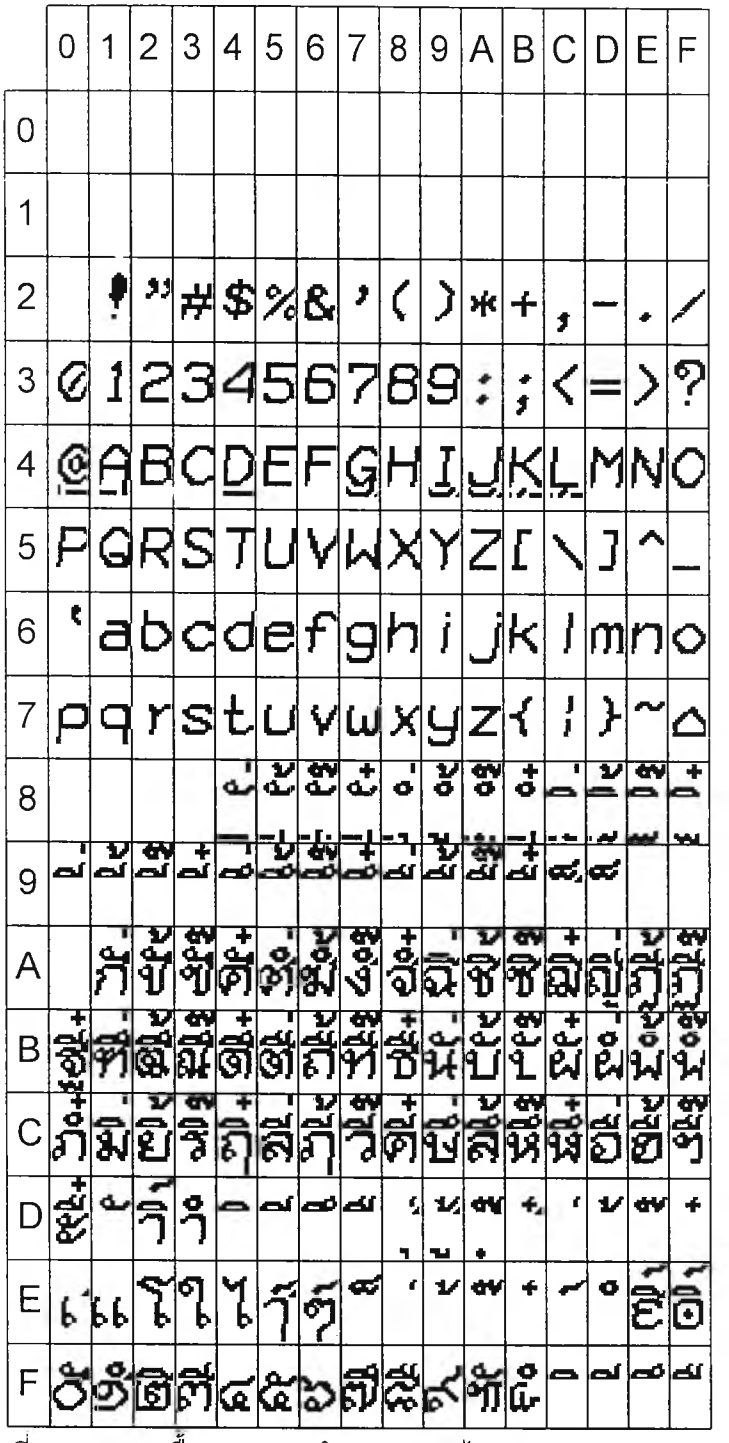
รหัสควบคุมถูกแบ่งเป็น 2 ประเภทตามลักษณะการใช้งาน ได้แก่

1. เปลี่ยนการแสดงผลหลังพบรหัสควบคุม ได้แก่ รหัสสลับภาษา และรหัสสลับการแสดงผลกราฟฟิกและตัวอักษร กลุ่มนี้จะไปแก้ไขข้อมูล T1 และ T0 ในไบต์ควบคุมสีและตารางอักขระอ้างอิง
2. เปลี่ยนการแสดงผลภายหลังพบรหัสควบคุมแต่ต้องอ่านพารามิเตอร์ที่ตามหลังด้วย ได้แก่ รหัสเปลี่ยนสีพื้น และรหัสเปลี่ยนสีตัวอักษร โดยทั้งสองอักขระควบคุมจะต้องอ่านรหัสควบคุมที่ตามมาอีกหนึ่งไบต์ เพื่อให้ทราบสีที่ต้องการจะเปลี่ยน ซึ่งได้แก่ รหัสสี ดำ, แดง, เหลือง, เขียว, น้ำเงิน,ฟ้า, ม่วง และ ขาว กลุ่มนี้จะไปแก้ไขข้อมูลสำหรับการแสดงผลสี ทั้งสีพื้นและสีตัวอักษรของไบต์ควบคุมสีและตารางอักขระอ้างอิง
3. เปลี่ยนการแสดงผลเฉพาะในขอบเขตที่กำหนด ได้แก่ รหัสควบคุมการแสดงผลแบบเอียง และการแสดงผลขีดเส้นใต้ โดยจะมีรหัสเริ่มและรหัสจบสำหรับทั้งสองแบบ ทำให้ทราบจุดเริ่มและจุดสิ้นสุดการแสดงผลทั้งแบบตัวเอียงและขีดเส้นใต้ โดยรหัสควบคุมการแสดงผลแบบเอียงจะไปเลือกตารางการแสดงผลแบบเอียง ดังตารางที่ 5.6 ด้วยการเปลี่ยนพารามิเตอร์ T1 และ T0 ส่วนรหัสควบคุมการแสดงผลขีดเส้นใต้ แต่จะไปแก้ไขบิตเก็บสถานะการแสดงผลแบบเอียงและขีดเส้นใต้ โดยให้เป็น 1 และ 0 แล้วแต่กรณี ซึ่งทำให้โปรแกรมประมวลผลการแสดงผลเลือกที่จะใช้รหัสสำหรับการแสดงผลพื้นที่ระดับล่างให้มีเส้นใต้หรือไม่มีเส้นขีดข้างใต้

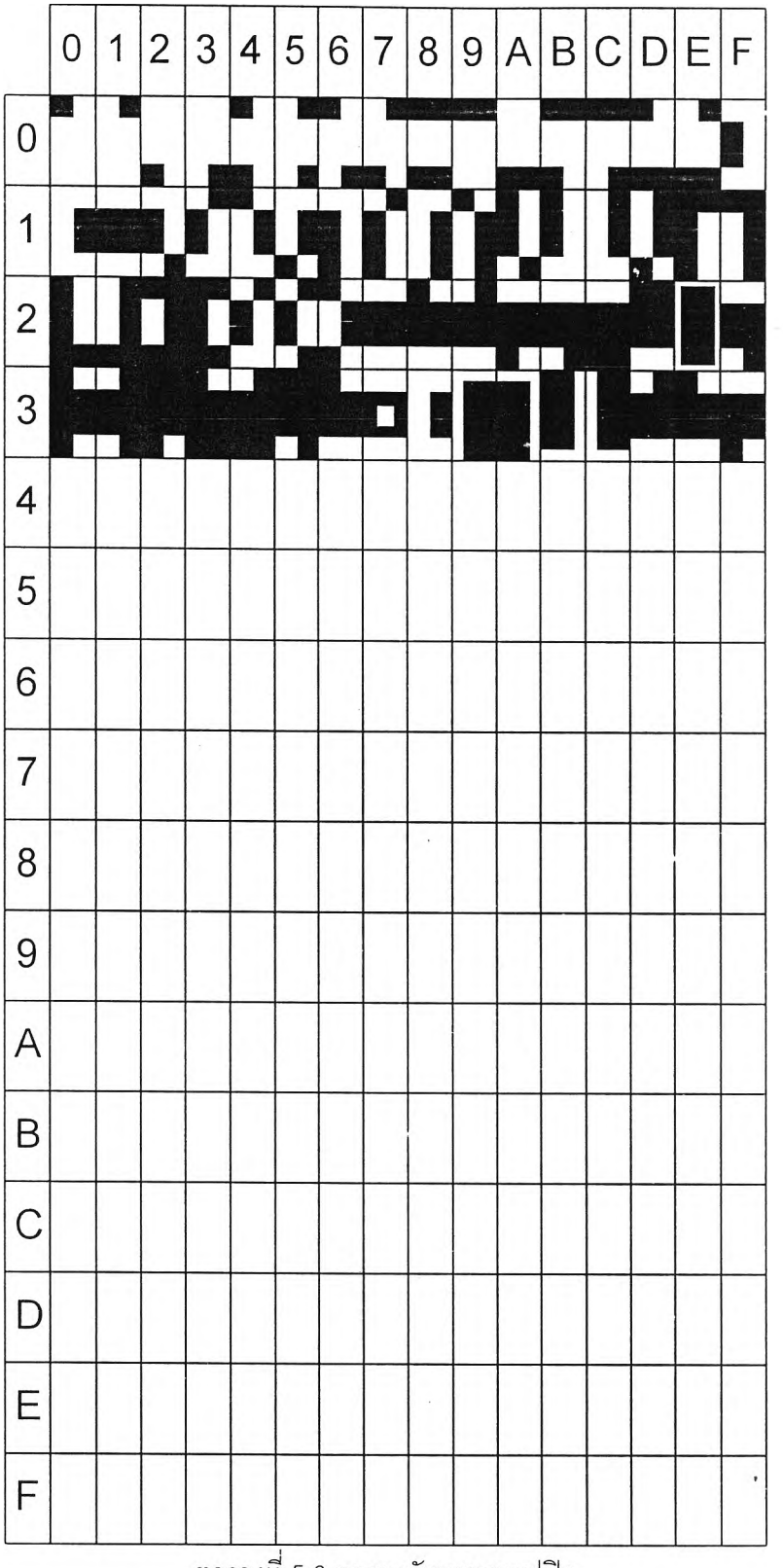
การทำงานของโปรแกรมประมวลผลอักขระควบคุมอธิบายด้วย ภาษาบรรยายโปรแกรมระดับที่ 2 ได้ดังนี้

```
MODULE: UPDATE CONTROL BYTE
        SELECT CASE CONTROL CHARACTER
        CASE LANGUAGE TOGGLE
                CHANGE REFERENCED TABLE BY MODIFY T1 AND T0 BIT
        CASE CHAR./GRAPHIC TOGGLE
                CHANGE REFERENCED TABLE BY MODIFY T1 AND T0 BIT
```

```
        CASE CHANGE BACKGROUND
                CHANGE BACKGROUND BY MODIFY BG_R BG_G AND BG_B BIT
        CASE CHANGE FOREGROUND
                CHANGE FOREGROUND BY MODIFY FG_R FG_G AND FG_B BIT
        CASE START UNDERLINE
                SET UNDERLINE
        CASE STOP UNDERLINE
                CLEAR UNDERLINE
        CASE START ITALIC
                CHANGE REFERENCED TABLE BY MODIFY T1 AND T0 BIT  CASE
        STOP ITALIC
                CHANGE REFERENCED TABLE BY MODIFY T1 AND T0 BIT
        END SELECT
END:
```



ตารางที่ 5.5 ตารางเนื้อหาอักขระสำหรับภาษาไทยและอังกฤษแบบตัวเอียง



ตารางที่ 5.6 ตารางอักขระกราฟฟิก

5.2.6 การทำงานของหน่วยประมวลผลกลาง

หน่วยประมวลผลกลางทำหน้าที่ควบคุมและประสานงานกับฮาร์ดแวร์ที่ทำงานโดยรอบ ได้แก่ การควบคุมด้วยคำสั่งแบบไอสแควร์ซี การรับข้อมูลจากหน่วยถอดรหัสข้อมูลแบบไบเฟส การถอดรหัสคำสั่งของเครื่องควบคุมระยะไกล อีกทั้งยังควบคุมการแสดงผลภาพด้วยการสวิตช์ระหว่างสัญญาณภาพโทรทัศน์ในโมดปกติ และสัญญาณภาพข้อมูลในโมดการแสดงผลข้อมูลผ่านทางอุปกรณ์มัลติเพลกเซอร์

โปรแกรมทำงานของโปรแกรมแบบ Lookup register driven กล่าวคือ โดยจะวนอ่านบิตของรีจิสเตอร์ควบคุม เพื่อกระโดดไปทำงานในส่วนต่างๆ จึงทำให้สามารถทำงานได้หลายอย่างพร้อมกัน และแยกออกจากส่วนอินพุตอื่นๆที่มีมาในรูปของโปรแกรมอินเทอร์รัพท์ภายนอก

ไมโครคอนโทรลเลอร์ของหน่วยประมวลผลกลางทราบว่าต้องทำงานอะไรจากรีจิสเตอร์ควบคุมต่างๆ แต่ละตัวทำหน้าที่ต่างกันไป โดยรีจิสเตอร์เหล่านี้ถูกกำหนดสถานะจากอินพุต ได้แก่ อินพุตจากเครื่องควบคุมระยะไกล ส่วนถอดรหัสข้อมูลไบเฟส และโปรแกรมประมวลผลการแสดงผล การทำงานของโปรแกรมหลักของเครื่องรับด้วยภาษาบรรยายโปรแกรมระดับที่ 2 แสดงในภาคผนวก ข.