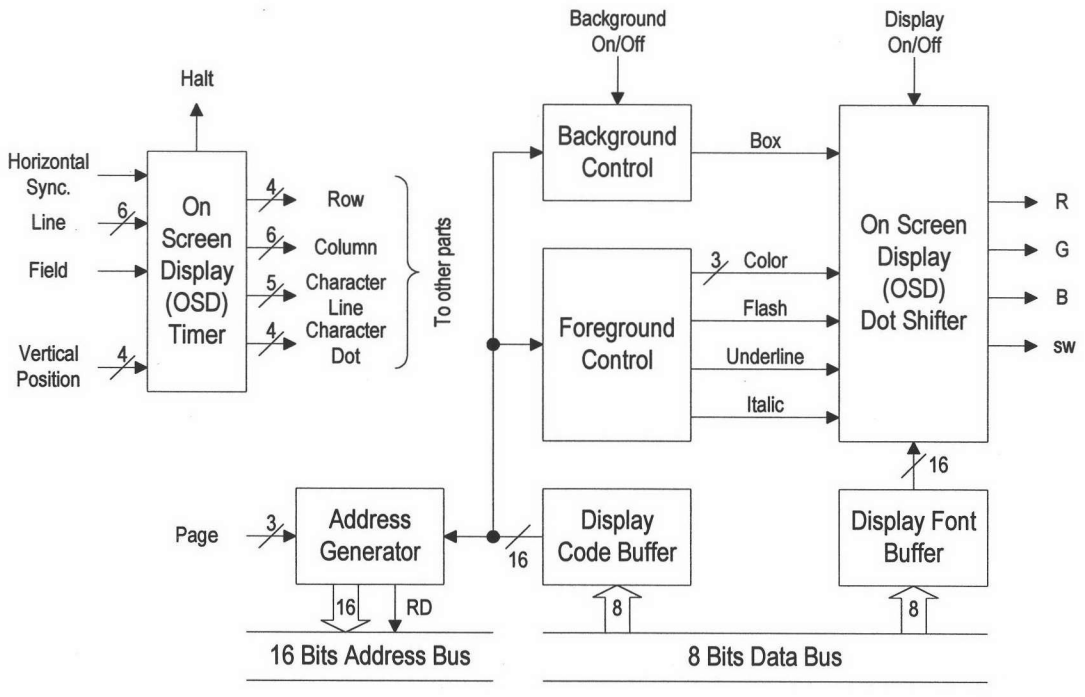
บทที่ 5

ตัวกำเนิดการแสดงผลบนหน้าจอ

5.1 โครงสร้างภายในของตัวกำเนิดการแสดงผลบนหน้าจอ

ตัวกำเนิดการแสดงผลบนหน้าจอ (On Screen Display Generator) เป็นส่วนประกอบในตัวประมวลผลคำบรรยายภาพไทย-อังกฤษแบบซ่อนได้ (Thai-English Closed Caption Processor) ที่มีความสำคัญ เพราะเป็นส่วนที่สร้างสัญญาณภาพของตัวอักษร ที่ประกอบเป็นคำบรรยายภาพออกมา โครงสร้างภายในแสดงไว้ในรูปที่ 5.1 แต่ละส่วนมีหน้าที่ดังต่อไปนี้



รูปที่ 5.1 โครงสร้างภายในของตัวกำเนิดการแสดงผลบนหน้าจอ

1. *ตัวจับเวลาการแสดงผลบนหน้าจอ* (OSD Timer) รับสัญญาณซิงก์แนวตั้ง (Horizontal Sync.), เส้น (Line) และฟิลด์ (Field) มาจากตัวประมวลผลซิงก์ (Sync. Processor) และตำแหน่งแนวตั้ง (Vertical Position) มาจากรีจิสเตอร์ควบคุมการแสดงผลบนหน้าจอ (OSD Control Register) ที่กล่าวถึงในบทที่ 4 เพื่อนำมากำหนดจังหวะการทำงานของส่วนอื่น ๆ สัญญาณออกที่สำคัญ ได้แก่

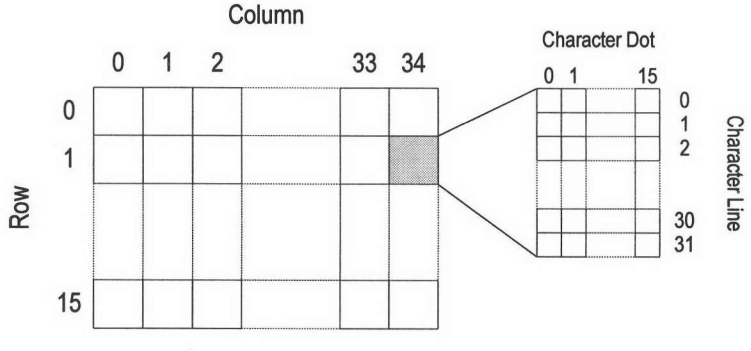
- *หยุด* (Halt) เป็นสัญญาณที่ส่งให้หน่วยประมวลผลกลาง (Central Processing Unit) ที่จะกล่าวถึงในบทที่ 6 เมื่อสัญญาณนี้มีค่า '1' เป็นการบอกหน่วยประมวลผลกลางว่าตัวกำเนิดการแสดงผลบนหน้าจอจะทำงาน และขอใช้บัสตำแหน่งที่อยู่ (Address Bus) กับบัสข้อมูล (Data Bus) เมื่อหน่วยประมวลผลกลางได้รับสัญญาณนี้แล้ว จะหยุดการทำงานชั่วขณะหลังจากที่ทำคำสั่ง (Instruction) ปัจจุบันเรียบร้อย และปล่อยบัสทั้งสองให้ตัวกำเนิดการแสดงผลบนหน้าจอใช้งาน

- *บรรทัด* (Row) เป็นสัญญาณขนาด 4 บิต ที่เกิดจากการจับเวลาในแนวตั้งของสัญญาณภาพรวม เพื่อบอกว่าตรงกับบรรทัดคำบรรยายภาพที่เท่าใด มีค่าตั้งแต่ 0 ถึง 15 ดังแสดงในรูปที่ 5.2

- *สดมภ์* (Column) เป็นสัญญาณขนาด 6 บิต ที่เกิดจากการจับเวลาในแนวนอนของสัญญาณภาพรวม เพื่อบอกว่าตรงกับสดมภ์ที่เท่าใดในบรรทัดคำบรรยายภาพนั้น มีค่าตั้งแต่ 0 ถึง 34 ดังแสดงในรูปที่ 5.2

- *เส้นตัวอักษร* (Character Line) เป็นสัญญาณขนาด 5 บิต ที่เกิดจากการจับเวลาในแนวตั้ง เพื่อบอกว่าตรงกับเส้นแนวนอนที่เท่าใดของตัวอักษรในบรรทัดนั้น มีค่าตั้งแต่ 0 ถึง 31 ดังแสดงในรูปที่ 5.2

- *จุดตัวอักษร* (Character Dot) เป็นสัญญาณขนาด 4 บิต ที่เกิดจากการจับเวลาในแนวนอน เพื่อบอกว่าตรงกับจุดที่เท่าใดบนเส้นแนวนอนของตัวอักษรนั้น มีค่าตั้งแต่ 0 ถึง 15 ดังแสดงในรูปที่ 5.2



รูปที่ 5.2 สัญญาณออกของตัวจับเวลาการแสดงผลบนหน้าจอ

สัญญาณบรรทัด, สดมภ์, เส้นตัวอักษร และจุดตัวอักษร จะถูกป้อนให้แก่ส่วนอื่นที่เหลืออีก 6 ส่วนในตัวกำเนิดการแสดงผลบนหน้าจอ เพื่อกำหนดจังหวะการทำงานให้สอดคล้องกัน

2. *ตัวกำเนิดตำแหน่งที่อยู่* (Address Generator) ทำหน้าที่สร้างตำแหน่งที่อยู่ (Address) ขนาด 16 บิต ส่งไปยังบัสตำแหน่งที่อยู่ (Address Bus) เพื่ออ่านข้อมูลที่จำเป็นสำหรับการแสดงผลบนหน้าจอต่อไปนี้เข้ามา

- *รหัสแสดงผล* (Display Code) เป็นรหัสขนาด 16 บิต ที่ใช้แทนตัวอักษรที่จะแสดงผล หรืออาจเป็นรหัสที่กำหนดสี และลักษณะของตัวอักษรถัดไป ตำแหน่งที่อยู่ของรหัสแสดงผลนี้ คำนวณจากบรรทัด (Row) กับสดมภ์ (Column) ที่มาจากตัวจับเวลาการแสดงผลบนหน้าจอ และหน้า (Page) ที่มาจากรีจิสเตอร์ควบคุมการแสดงผลบนหน้าจอ (OSD Control Register) รายละเอียดของรหัสนี้จะกล่าวถึงในหัวข้อที่ 5.2
- *รูปแบบอักขระแสดงผล* (Display Font) ประกอบด้วยรูปแบบอักขระของพยัญชนะ, สระบน, สระล่าง และวรรณยุกต์ ตำแหน่งที่อยู่ของรูปแบบอักขระนี้ คำนวณจากเส้นตัวอักษร (Character Line) ที่มาจากตัวจับเวลาการแสดงผลบนหน้าจอ และรหัสแสดงผลที่อ่านมาก่อนหน้า

3 . *ที่พักรหัสแสดงผล* (Display Code Buffer) ใช้เก็บรหัสแสดงผลขนาด 16 บิต ซึ่งอ่านจากหน่วยความจำเข้าถึงแบบสุ่ม (Random Access Memory) เข้ามาทางบัสข้อมูล (Data Bus) ขนาด 8 บิต 2 ครั้งรวมเป็น 16 บิต

4. *ที่พักรูปแบบอักขระแสดงผล* (Display Font Buffer) ใช้เก็บรูปแบบอักขระแสดงผลขนาด 16 บิต ซึ่งอ่านจากหน่วยความจำอ่านอย่างเดียว (Read Only Memory) เข้ามาทางบัสข้อมูล (Data Bus) ขนาด 8 บิต 4 ครั้ง โดย 2 ครั้งแรกเป็นรูปแบบอักขระของพยัญชนะ ส่วน 2 ครั้งหลังเป็นของสระบน, สระล่าง และวรรณยุกต์ นำมาซ้อนกันเป็นรูปแบบอักขระเดียวด้วยการ OR

5. *ส่วนควบคุมพื้นหน้า* (Foreground Control) ทำหน้าที่ควบคุมตัวเลื่อนจุดการแสดงผลบนหน้าจอ (OSD Dot Shifter) เพื่อให้ตัวอักษรที่แสดงในสดมภ์ (Column) ถัดไปมีสี และลักษณะที่ถูกต้อง สัญญาณออกที่ไปควบคุม ได้แก่

- สี (Color) เป็นสัญญาณที่ควบคุมสีของตัวอักษรที่จะปรากฏ มีขนาด 3 บิต แต่ละบิตแทนแม่สีแต่ละสี อันได้แก่ แดง, เขียว และน้ำเงิน ถ้าบิตประจำแม่สีใดเป็น '1' แสดงว่าตัวอักษรที่ปรากฏจะมีสีนั้นอยู่ด้วย
- กะพริบ (Flash) เป็นสัญญาณที่ควบคุมการกะพริบของตัวอักษรที่จะปรากฏ ถ้ามีค่า '1' หมายความว่าตัวอักษรถัดไปจะต้องถูกทำให้กะพริบ

- ขีดเส้นใต้ (Underline) เป็นสัญญาณที่ควบคุมการขีดเส้นใต้ของตัวอักษรที่จะปรากฏ ถ้ามีค่า '1' หมายความว่าตัวอักษรถัดไปจะต้องถูกขีดเส้นใต้
- เอียง (Italic) เป็นสัญญาณที่ควบคุมความเอียงของตัวอักษรที่จะปรากฏ ถ้ามีค่า '1' หมายความว่าตัวอักษรถัดไปจะต้องถูกแสดงเป็นตัวเอียง

สัญญาณออกนี้จะเปลี่ยนแปลงเมื่อรหัสแสดงผล (Display Code) เป็นรหัสที่สั่งให้เปลี่ยนสีและลักษณะตัวอักษร

6. *ส่วนควบคุมพื้นหลัง* (Background Control) ทำหน้าที่ควบคุมตัวเลื่อนจุดการแสดงผลบนหน้าจอ (OSD Dot Shifter) เพื่อให้การแสดงพื้นหลังของคำบรรยายภาพถูกต้อง สัญญาณควบคุมที่ส่งออกไป คือ กรอบ (Box) ถ้ามีค่าเป็น '1' หมายความว่าตัวอักษรถัดไปจะมีพื้นหลังสีดำ ถ้าเป็น '0' จะมีพื้นหลังเป็นภาพ สัญญาณกรอบขึ้นกับสัญญาณเปิด-ปิดพื้นหลัง (Background On/Off) ที่ต่อตรงมาจากสวิตช์บนแผงควบคุม (Control Panel) ถ้าเป็น '0' จะทำให้สัญญาณกรอบเป็น '0' ตลอด แต่ถ้าเป็น '1' สัญญาณกรอบจะขึ้นกับรหัสแสดงผลดังจะกล่าวต่อไปในหัวข้อที่ 5.2

7. *ตัวเลื่อนจุดการแสดงผลบนหน้าจอ* (OSD Dot Shifter) เป็นส่วนที่รับรูปแบบอักขระ (Font) เข้ามาทำการเลื่อนออกไป เพื่อกำเนิดสัญญาณแม่สี (R, G และ B) กับสัญญาณสั่งให้แทรกสัญญาณภาพ (SW) ส่งให้แก่ตัวเข้าจังหวะซ้อนทับภาพ (Video Overlay Synchronizer) ทั้งนี้การทำงานจะขึ้นกับสัญญาณเปิด-ปิดการแสดงผล (Display On/Off) ที่ต่อตรงมาจากสวิตช์บนแผงควบคุม ถ้าสัญญาณนี้เป็น '0' สัญญาณสั่งให้แทรกสัญญาณภาพจะเป็น '1' ตลอดคือไม่ให้แทรกภาพ คำบรรยายภาพก็จะไม่ปรากฏบนจอ แต่หากเป็น '1' สัญญาณออกจะเป็นไปตามรูปแบบอักขระและการควบคุมของส่วนควบคุมพื้นหน้า กับส่วนควบคุมพื้นหลัง

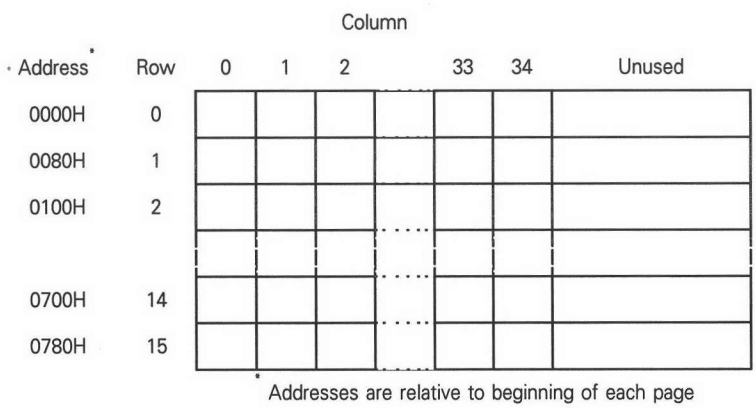
5.2 การจัดเก็บ และความหมายของรหัสแสดงผล

ข้อมูลคำบรรยายภาพที่ถอดรหัสแล้วโดยหน่วยประมวลผลกลาง (Central Processing Unit) จะถูกเก็บในหน่วยความจำเข้าถึงแบบสุ่ม (Random Access Memory) ในรูปแบบของรหัสแสดงผล (Display Code) โดยแต่ละช่องสัญญาณข้อมูลจะใช้เนื้อที่ในการเก็บรหัสแสดงผล 2 หน้า ตรงกับแนวความคิดเรื่องหน่วยความจำแสดงผล (Displayed Memory) และหน่วยความจำไม่แสดงผล (Non Displayed Memory) ของเครื่องถอดรหัส ตามที่อธิบายไว้ในบทที่ 2 รวมทั้ง 4 ช่องสัญญาณข้อมูลจะใช้เนื้อที่ในการเก็บรหัสแสดงผลทั้งสิ้น 8 หน้า แต่ละหน้าแบ่งเป็นส่วนต่าง ๆ ดังนี้

- *บรรทัด* (Row) ใน 1 หน้าที่ใช้เก็บรหัสคำบรรยายภาพแบ่งเป็น 16 บรรทัด เพื่อให้สามารถรองรับการแสดงผลในโมดข้อความที่ต้องแสดงตัวอักษรได้สูงสุด 15 บรรทัด ตามที่อธิบายในบทที่ 2 เหตุที่มีจำนวนบรรทัดมากกว่าที่ระบุในข้อกำหนด ก็เพราะการทำเป็น 16 บรรทัดจะออกแบบได้สะดวกกว่า 15 บรรทัด

- *สดมภ์* (Column) ในแต่ละบรรทัดจะมี 35 สดมภ์ ใช้ในการแสดงคำบรรยายภาพ 32 สดมภ์ และแสดงช่องว่างที่ช่วยให้อ่านง่ายอีก 2 สดมภ์ รวมเป็น 34 สดมภ์ อีก 1 สดมภ์ใช้เก็บรหัสแสดงผลที่กำหนดสี และลักษณะตัวอักษรเริ่มต้นในแต่ละบรรทัด ทำหน้าที่แทนรหัสตำแหน่งเบื้องต้น (Preamble Address Code) ที่กล่าวถึงในบทที่ 2

เนื่องจากรหัสแสดงผลที่เก็บในแต่ละสดมภ์มีขนาด 16 บิต หรือ 2 ไบต์ ในแต่ละบรรทัดจึงต้องใช้ที่เก็บรหัสแสดงผลทั้งหมด 70 ไบต์ เพื่อความสะดวกในการออกแบบจึงใช้เนื้อที่ในการเก็บรหัสแสดงผลแต่ละบรรทัดจำนวน 128 ไบต์ แต่ใช้งานจริง ๆ เพียง 70 ไบต์ อีก 58 ไบต์ที่เหลือจะไม่ใช้งาน รวมทั้งหน้าจะใช้เนื้อที่ในการเก็บรหัสแสดงผลเท่ากับ 2,048 ไบต์ โครงสร้างการจัดเก็บรหัสแสดงผลเทียบกับจุดเริ่มต้นของแต่ละหน้า แสดงดังในรูปที่ 5.3



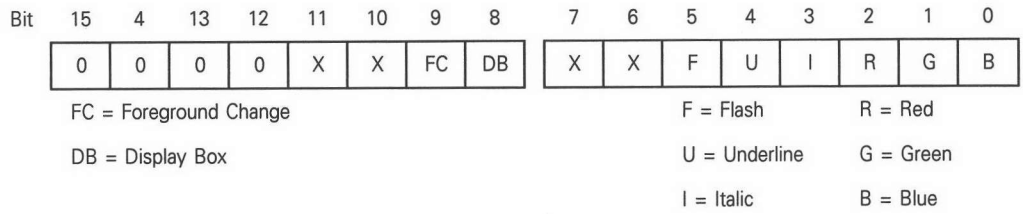
รูปที่ 5.3 โครงสร้างการจัดเก็บรหัสแสดงผลเทียบกับจุดเริ่มต้นของแต่ละหน้า

จากโครงสร้างในรูปที่ 5.3 ตำแหน่งที่อยู่ (Address) ของรหัสแสดงผลจึงคำนวณได้ดังนี้

ตำแหน่งที่อยู่ = เลขบรรทัด * 128 + เลขสดมภ์ * 2 + ตำแหน่งที่อยู่เริ่มต้นของแต่ละหน้า

รหัสแสดงผลที่เก็บในแต่ละสดมภ์มีขนาด 2 ไบต์ ใช้แทนตัวอักษรที่จะปรากฏในสดมภ์นั้น หรือเป็นรหัสแสดงผลที่กำหนดสี และลักษณะของตัวอักษรตัวถัดไป การจำแนกว่ารหัสแสดงผลที่อ่านเข้ามาใช้แทนตัวอักษรที่จะปรากฏ หรือเป็นรหัสที่กำหนดสี และลักษณะของตัวอักษรถัดไป สังเกตได้ดังนี้

- ในกรณีที่รหัสแสดงผลไบต์ที่ 1 มีค่าอยู่ระหว่าง 00H-0FH รหัสแสดงผลทั้ง 2 ไบต์ จะเป็นรหัสที่กำหนดสี และลักษณะของตัวอักษรถัดไป ในกรณีนี้ข้อมูลทั้ง 2 ไบต์จะมีรูปแบบดังแสดงในรูปที่ 5.4 โดยแต่ละบิตมีความหมายดังนี้



รูปที่ 5.4 รูปแบบของรหัสแสดงผลเมื่อเป็นรหัสที่กำหนดสี และลักษณะของตัวอักษรถัดไป

  - บิตที่ 9 (Foreground Change) หากบิตนี้มีค่าเป็น '1' ส่วนควบคุมพื้นหน้า (Foreground Control) จะนำรหัสแสดงผลบิตที่ 5 ถึง 0 ไปแทนที่สัญญาณออกเดิม แต่หากบิตนี้เป็น '0' สัญญาณออกของส่วนควบคุมพื้นหน้าจะยังคงเดิม
  - บิตที่ 8 (Display Box) หากบิตนี้มีค่าเป็น '1' หมายความว่าให้แสดงพื้นหลังสีดำแทนที่รหัสแสดงผลในสดมภ์นี้ ส่วนควบคุมพื้นหลัง (Background Control) จะให้สัญญาณกรอบ (Box) เป็น '1' แต่ถ้าบิตนี้เป็น '0' หมายความว่าให้แสดงภาพเป็นพื้นหลังแทน ส่วนควบคุมพื้นหลังจะให้สัญญาณกรอบเป็น '0'
  - บิตที่ 5 ถึง 3 เป็นบิตที่กำหนดลักษณะตัวอักษร ได้แก่ ตัวกะพริบ (Flash), ตัวขีดเส้นใต้ (Underline) และตัวเอียง (Italic) รหัสแสดงผลทั้ง 3 บิตนี้จะถูกนำไปแทนที่สัญญาณออกทั้ง 3 ที่ตรงกันของส่วนควบคุมพื้นหน้า เมื่อรหัสแสดงผลบิตที่ 9 เป็น '1'
  - บิตที่ 2 ถึง 0 เป็นบิตที่กำหนดสีของตัวอักษร รหัสแสดงผลทั้ง 3 บิตนี้จะถูกนำไปแทนที่สัญญาณสี (Color) ของส่วนควบคุมพื้นหน้า เมื่อรหัสแสดงผลบิตที่ 9 เป็น '1'

- ในกรณีที่รหัสแสดงผลไบต์ที่ 1 มีค่าตั้งแต่ 10H-FFH รหัสแสดงผลทั้ง 2 ไบต์นี้ จะเป็นรหัสที่แทนตัวอักษร โดยรหัสแสดงผลไบต์ที่ 1 ถ้ามีค่าอยู่ในช่วง 20H ถึง 7FH จะแทนตัวอักษรภาษาอังกฤษตามรหัสแอสกี (ASCII) แต่ถ้ามีค่าอยู่ในช่วง 80H ถึง FFH จะแทนตัวอักษรภาษาไทยตามรหัสสมอ. สำหรับรหัสแสดงผลไบต์ที่ 2 จะเป็นรหัสที่แทน สระบน, สระล่าง และวรรณยุกต์ในภาษาไทยเพื่อประหยัดเนื้อที่ในการเก็บ โดยแบ่งเป็น 3 เขต (Field) ได้แก่ วรรณยุกต์ (Tone Mark), สระบน (Upper Vowel) และสระล่าง (Lower Vowel) ดังแสดงในรูปที่ 5.5 แต่ละเขตมีความหมายดังนี้

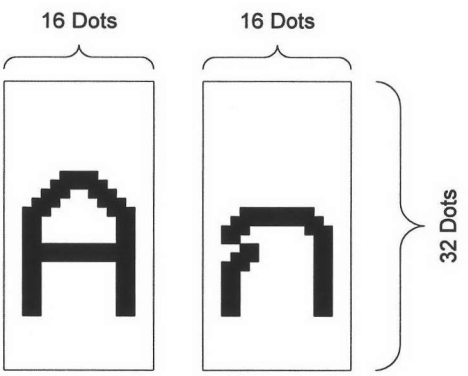
- วรรณยุกต์ (Tone Mark) เป็นเขตที่เก็บรหัสแสดงผลที่แทนวรรณยุกต์เป็นส่วนใหญ่ รหัสแต่ละค่าจะแทนวรรณยุกต์ใด ดูได้จากรูปที่ 5.5
- สระบน (Upper Vowel) เป็นเขตที่เก็บรหัสแสดงผลที่แทนสระบนเป็นส่วนใหญ่ รหัสแต่ละค่าจะแทนสระบนใด ดูได้จากรูปที่ 5.5
- สระล่าง (Lower Vowel) เป็นเขตที่เก็บรหัสแสดงผลที่แทนสระล่างเป็นส่วนใหญ่ รหัสแต่ละค่าจะแทนสระล่างใด ดูได้จากรูปที่ 5.5

Bit	Tone Mark				Upper Vowel				Lower Vowel		
	7	6	5		4	3	2		1	0	
	0	0	0	ไม่มีวรรณยุกต์	0	0	0	ไม่มีสระบน	0	0	ไม่มีสระล่าง
	0	0	1	ไม้เอก	0	0	1	ไม้หันอากาศ	0	1	สระอุ
	0	1	0	ไม้โท	0	1	0	สระอิ	1	0	สระอู
	0	1	1	ไม้ตรี	0	1	1	สระอี	1	1	จุดล่าง
	1	0	0	ไม้จัตวา	1	0	0	สระอึ			
	1	0	1	การันต์	1	0	1	สระอือ			
	1	1	0	ไม่ใช้งาน	1	1	0	ไม้ไต่คู้			
	1	1	1	ไม่ใช้งาน	1	1	1	สระอำ			

รูปที่ 5.5 ความหมายของเขตทั้ง 3 ในรหัสแสดงผลไบต์ที่ 2 ที่แทนสระบน, สระล่าง และวรรณยุกต์

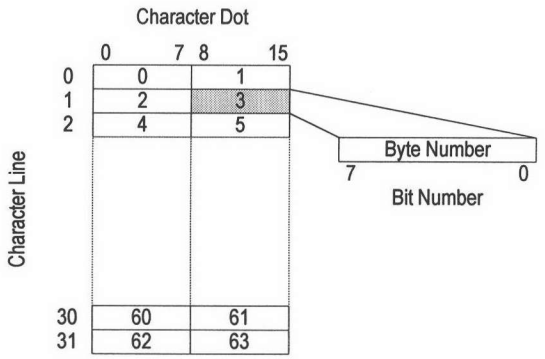
5.3 การจัดเก็บรูปแบบอักขระ

รูปแบบอักขระที่เก็บไว้ในหน่วยความจำอ่านอย่างเดียว (Read Only Memory) แต่ละตัวจะมีขนาดสูง 32 จุด กว้าง 16 จุด ทั้งภาษาไทย และภาษาอังกฤษ ดังตัวอย่างในรูปที่ 5.6



รูปที่ 5.6 ตัวอย่างรูปแบบอักขระ

จุดแต่ละจุดในรูปแบบอักขระแต่ละตัวจะใช้เนื้อที่ในการเก็บ 1 บิต แต่ละเส้นตัวอักษร (Character Line) ซึ่งมี 16 จุดตัวอักษร (Character Dot) จึงใช้เนื้อที่ในการเก็บ 16 บิต หรือ 2 ไบต์ รูปแบบอักขระแต่ละตัวมี 32 เส้นตัวอักษร รวมเนื้อที่ในการเก็บรูปแบบอักขระแต่ละตัวเท่ากับ 64 ไบต์ โดยจะเก็บเรียงต่อเนื่องกันไปตั้งแต่ไบต์ที่ 0 ถึงไบต์ที่ 63 แต่ละไบต์จะตรงกับตำแหน่งดังแสดงในรูปที่ 5.7



รูปที่ 5.7 การจัดเรียงรูปแบบอักขระไบต์ต่าง ๆ เป็นรูปของตัวอักษร

จากโครงสร้างในรูปที่ 5.7 ตำแหน่งที่อยู่ (Address) ของรูปแบบอักขระจึงคำนวณได้ดังนี้

ตำแหน่งที่อยู่ = รหัสแสดงผล * 64 + เส้นตัวอักษร * 2 + ตำแหน่งที่อยู่ที่เริ่มเก็บรูปแบบอักขระ

รูปแบบอักขระที่เก็บในหน่วยความจำอ่านอย่างเดียวมีทั้งสิ้นรวม 256 รูปแบบ สำหรับภาษาอังกฤษ 128 รูปแบบ ซึ่งตรงกับรหัสแอสกี (ASCII) เมื่อรหัสแสดงผลมีค่าอยู่ระหว่าง 00H ถึง 7FH และสำหรับภาษาไทยอีก 128 รูปแบบ ซึ่งตรงกับรหัสสมอ. เมื่อรหัสแสดงผลมีค่าอยู่ระหว่าง 80H ถึง FFH ดังรูปที่ 5.8

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00H																
10H																
20H		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
30H	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
40H	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
50H	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
60H	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
70H	p	q	r	s	t	u	v	w	x	y	z	{	l	}	~	

รูปที่ 5.8 ตารางแสดงรูปแบบอักขระเทียบกับรหัสตัวอักษร

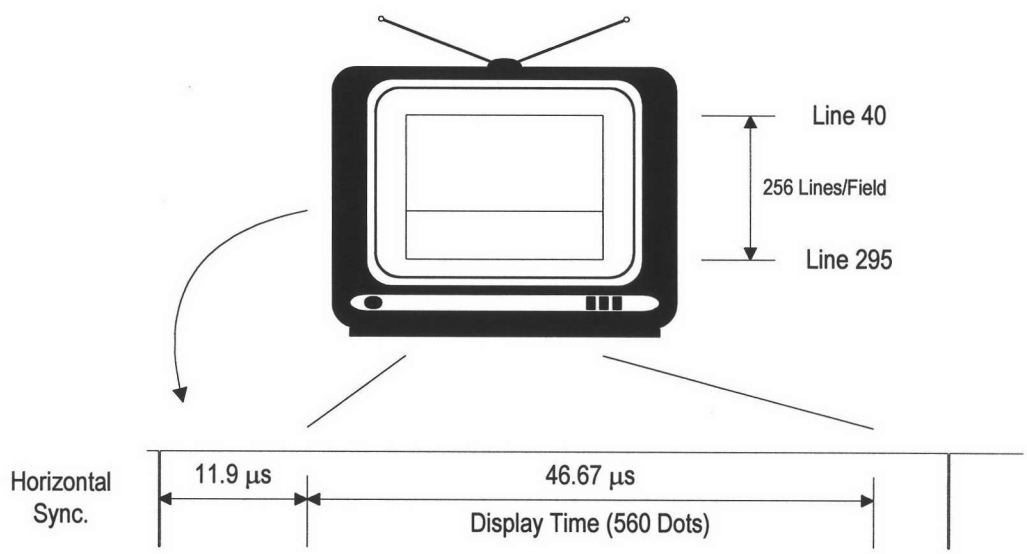
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
80H	◌่	◌้	◌๊	◌๋	◌์	◌ั	◌ิ	◌ี	◌ึ	◌ื	◌ั	◌ิ	◌ี	◌ื	◌ั	◌ิ
90H	◌ี	◌ื	◌ั	◌ิ	◌ี											
A0H		ก	ข	ฃ	ค	ฅ	ฆ	ง	จ	ฉ	ช	ซ	ฌ	ญ	ฎ	ฏ
B0H	ฐ	ฑ	ฒ	ณ	ด	ต	ถ	ท	ธ	น	บ	ป	ผ	ฝ	พ	ฟ
C0H	ภ	ม	ย	ร	ฤ	ล	ฦ	ว	ศ	ษ	ส	ห	ฬ	อ	ฮ	ฯ
D0H	ะ	◌ั	า	◌ำ	◌ิ	◌ี	◌ึ	◌ื	◌ุ	◌ู	◌ฺ					฿
E0H	เ	แ	โ	ใ	ไ	ๅ	ๆ	◌็	◌่	◌้	◌๊	◌๋	◌์	◌		
F0H	๐	๑	๒	๓	๔	๕	๖	๗	๘	๙			◌๋	◌์	◌	◌

รูปที่ 5.8 (ต่อ) ตารางแสดงรูปแบบอักขระเทียบกับรหัสตัวอักษร

สังเกตว่ามีการเก็บรูปแบบอักขระที่มีการผสมกันระหว่างสระบน กับวรรณยุกต์เอาไว้แล้ว ทั้งนี้เพื่อความสะดวกในการออกแบบ โดยรหัสแสดงผลไบต์ที่ 2 ในเขต (Field) วรรณยุกต์ (Tone Mark) กับสระบน (Upper Vowel) จะนำมาต่อกันเป็น 6 บิต แล้วใช้ตารางเปิดดู (Look Up Table) เลือกรูปแบบอักขระที่ตรงกันออกมา

5.4 จังหวะการทำงานของตัวกำเนิดการแสดงผลบนหน้าจอ

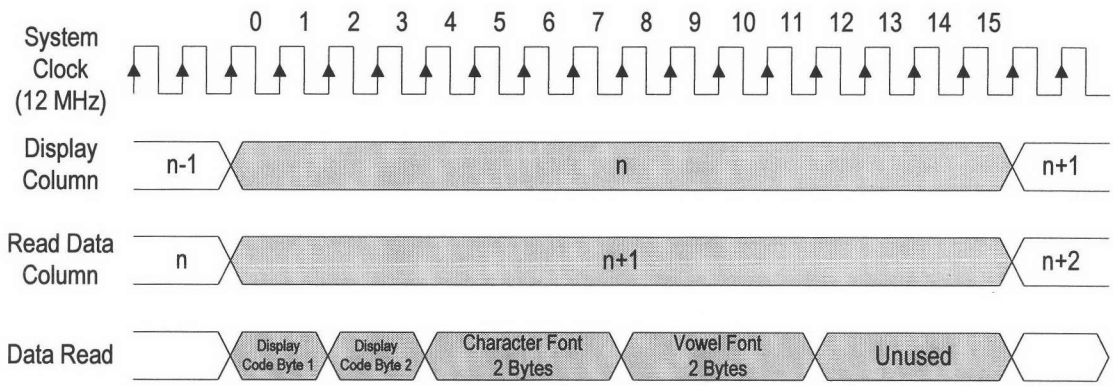
จังหวะในการแสดงผลตัวอักษรให้อยู่กลางจอภาพพอดี แสดงได้ดังรูปที่ 5.9 โดยตำแหน่งการแสดงผลตัวอักษรในแนวตั้งจะเริ่มที่เส้น (Line) ที่ 40 และจบที่เส้นที่ 295 ของแต่ละฟีลด์ (Field) ซึ่งจะตรงกับกลางจอทางแนวตั้ง ส่วนตำแหน่งในแนวนอนจะเริ่มห่างจากขอบขาลงของสัญญาณ



รูปที่ 5.9 จังหวะการแสดงผลให้อยู่กลางหน้าจอ

ซิงก์แนวนอน (Horizontal Sync.) 11.9 ไมโครวินาที และแสดงจุดต่อกันไปจำนวน 560 จุด สำหรับ 35 สดมภ์ (Column) แต่ละจุดใช้ 1 สัญญาณนาฬิกา (12 เมกะเฮิรตซ์) จึงแสดงจุดต่อกันนาน 46.67 ไมโครวินาที หรือหยุดที่ 58.57 ไมโครวินาที ซึ่งจะตรงกลางจอทางแนวนอน

การแสดงตัวอักษรในแต่ละสดมภ์ (Column) จะต้องอ่านรหัสแสดงผล (Display Code) และรูปแบบอักขระ (Font) เข้ามาก่อนที่จะถึงเวลาแสดงผลจริง ๆ เนื่องจากว่าตัวอักษรที่แสดงจะอยู่ต่อกัน จึงต้องอ่านข้อมูลทั้งสองนี้เข้ามาให้เสร็จในเวลา 16 สัญญาณนาฬิกาที่กำลังแสดงตัวอักษรในสดมภ์ก่อนหน้า ซึ่งมีจังหวะการอ่านข้อมูลเข้ามาแสดงดังรูปที่ 5.10 ดังนี้



รูปที่ 5.10 จังหวะการอ่านรหัสแสดงผล และรูปแบบอักขระเพื่อแสดงตัวอักษรในแต่ละสดมภ์

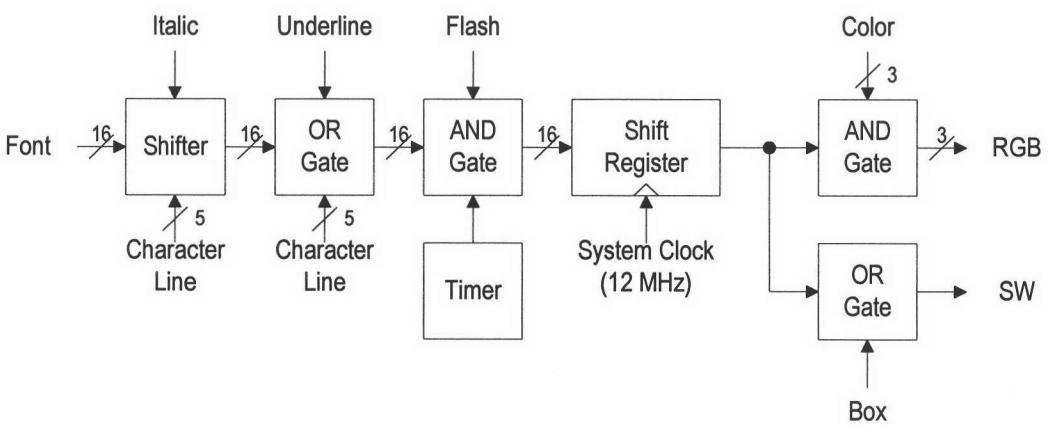
- สัญญาณนาฬิกาที่ 0 และ 1 อ่านรหัสแสดงผลไบต์ที่ 1 จากหน่วยความจำเข้าถึงแบบสุ่ม (Random Access Memory) เก็บไว้ในที่พักรหัสแสดงผล (Display Code Buffer)
- สัญญาณนาฬิกาที่ 2 และ 3 อ่านรหัสแสดงผลไบต์ที่ 2 จากหน่วยความจำเข้าถึงแบบสุ่ม เก็บไว้ในที่พักรหัสแสดงผล
- สัญญาณนาฬิกาที่ 4 ถึง 7 อ่านรูปแบบอักขระของพยัญชนะจำนวน 2 ไบต์ จากหน่วยความจำอ่านอย่างเดียว (Read Only Memory) เก็บไว้ในที่พักรูปแบบอักขระ (Display Font Buffer)
- สัญญาณนาฬิกาที่ 8 ถึง 11 อ่านรูปแบบอักขระของวรรณยุกต์กับสระบน หรือรูปแบบอักขระของสระล่างจำนวน 2 ไบต์ จากหน่วยความจำอ่านอย่างเดียว เก็บไว้ในที่พักรูปแบบอักขระ โดยนำไปซ้อนกับรูปแบบอักขระของพยัญชนะที่อ่านมาก่อนด้วยการ OR กันเป็นรูปแบบอักขระเดียว
- สัญญาณนาฬิกาที่ 12 ถึง 15 ไม่ได้ใช้งานบัส (Bus) เพื่ออ่านข้อมูลใด

สาเหตุที่ต้องใช้เวลาถึง 2 คาบสัญญาณนาฬิกาต่อการอ่านข้อมูลเข้ามา 1 ไบต์ เนื่องจากว่าหน่วยความจำเข้าถึงแบบสุ่ม (Random Access Memory) ที่ใช้งานมีช่วงเวลาเข้าถึง (Access Time) นาน 100 นาโนวินาที และหน่วยความจำอ่านอย่างเดียว (Read Only Memory) ที่ใช้งานมีช่วงเวลาเข้าถึงนาน 120 นาโนวินาที ซึ่งนานกว่า 1 คาบสัญญาณนาฬิการะบบ (System Clock) ความถี่ 12 เมกะเฮิรตซ์ ที่นานเพียง 83.33 นาโนวินาที การอ่านข้อมูลแต่ละไบต์จึงต้องใช้เวลา 2 คาบสัญญาณนาฬิกาที่นาน 166.67 วินาที เพื่อให้หน่วยความจำทั้งสองทำงานทัน

หลังจากที่อ่านรูปแบบอักขระเข้ามาครบแล้ว ตัวเลื่อนจุดการแสดงผลบนหน้าจอ (OSD Dot Shifter) จะนำรูปแบบอักขระที่อ่านเข้ามาได้เลื่อน (Shift) ออกไปทีละจุด โดยเริ่มเลื่อนจุดแรกที่สัญญาณนาฬิกาที่ 16 ซึ่งตรงกับสัญญาณนาฬิกาที่ 0 ของการแสดงตัวอักษรถัดไป

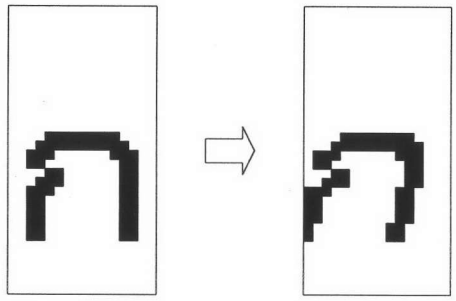
5.5 การแสดงตัวอักษรให้มีสี และลักษณะต่าง ๆ

ภายในตัวเลื่อนจุดการแสดงผลบนหน้าจอ (OSD Dot Shifter) มีวงจรที่ทำหน้าที่ปรับรูปแบบอักขระ (Font) ที่จะแสดงผล เพื่อให้แสดงอักขระที่มีลักษณะพิเศษได้ดังรูปที่ 5.11 ซึ่งวิธีการปรับรูปแบบอักขระอธิบายได้ดังนี้



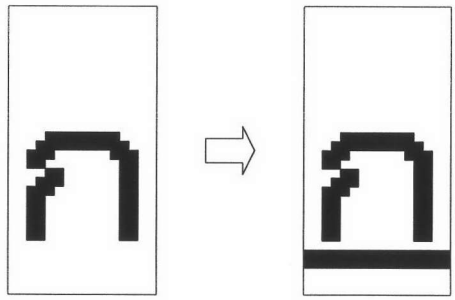
รูปที่ 5.11 แผนภาพวงจรภายในตัวเลื่อนจุดการแสดงผลบนหน้าจอ

1. *ตัวเอียง* ทำได้โดยการเลื่อนจุดของรูปแบบอักขระที่อ่านเข้ามา 1 จุดแนวนอนทุก ๆ 4 จุดแนวตั้ง โดยจะเลื่อนไปทางขวาเมื่ออยู่ครึ่งบนของตัวอักษร และเลื่อนไปทางซ้ายเมื่ออยู่ครึ่งล่างของตัวอักษร สัญญาณที่ควบคุมทิศทางการเลื่อน คือ เส้นตัวอักษร (Character Line) ตัวอย่างการปรับรูปแบบอักขระให้เป็นตัวเอียงแสดงไว้ในรูปที่ 5.12



รูปที่ 5.12 การปรับรูปแบบอักขระให้เป็นตัวเอียง

2. *ตัวขีดเส้นใต้* ทำได้โดยการตรวจเส้นตัวอักษรว่าเป็นเส้นที่ 27 กับ 28 หรือไม่ ถ้าใช่ก็จะเพิ่มจุดลงไปในรูปแบบอักขระให้เต็ม ตัวอย่างการปรับรูปแบบอักขระให้เป็นตัวขีดเส้นใต้แสดงดังรูปที่ 5.13



รูปที่ 5.13 การปรับรูปแบบอักขระให้เป็นตัวขีดเส้นใต้

3. *ตัวกะพริบ* ทำได้โดยการใช้ตัวจับเวลาเป็นตัวคอยปิดเปิดเกตให้รูปแบบอักขระผ่าน เมื่อเกตเปิด รูปแบบอักขระจะผ่านออกไปตามปกติตัวอักษรก็จะปรากฏ เมื่อเกตปิด รูปแบบอักขระจะถูกกั้น ทำให้ตัวอักษรหายไป

4. *สี* ทำได้โดยนำบิตที่ได้จากรีจิสเตอร์เลื่อนข้อมูล (Shift Register) ที่เลื่อนรูปแบบอักขระออกมาทีละบิต มาทำการ AND กับสี (Color) ถ้าบิตของรูปแบบอักขระเป็น '1' สัญญาณออก RGB ก็จะมีค่าตามสี แต่ถ้าบิตของรูปแบบอักขระเป็น '0' สัญญาณออก RGB ก็จะเป็น '0' ด้วย คือ สีดำนั่นเอง