# บทที่ 2

## โครงสร้างของระบบ On-line

### 2.1 ระบบไมโครคอมพิวเตอร์แบบ On-line [3]

ระบบไมโครคอมพิวเตอร์แบบ On-line เป็นระบบที่ออกแบบขึ้นเพื่อให้มีสภาพการทำงานที่ผู้ใช้ที่อยู่ห่างไกลติดต่อกับเครื่องคอมพิวเตอร์ได้โดยตรงทาง Terminal สามารถโต้ตอบกับเครื่องคอมพิวเตอร์ตลอดเวลาการทำงาน เพื่อหลีกเลี่ยงการเตรียมข้อมูลในรูปของบัตรหรือเทป โดยข้อมูลจะถูกส่งไปเก็บไว้ใน On-line Storage การติดต่อรับส่งข้อมูลหรือผลลัพธ์แต่ละครั้งมีปริมาณไม่มากนักจะช่วยให้ Turn Around Time ในการใช้เครื่องคอมพิวเตอร์สำหรับงานขนาดเล็กสั้นลง และจะอำนวยความสะดวกในการค้นหาข้อมูลได้อย่างรวดเร็ว ภายใต้การ Update อยู่เสมอ ข้อมูลจึงทันเหตุการณ์ตลอดเวลา

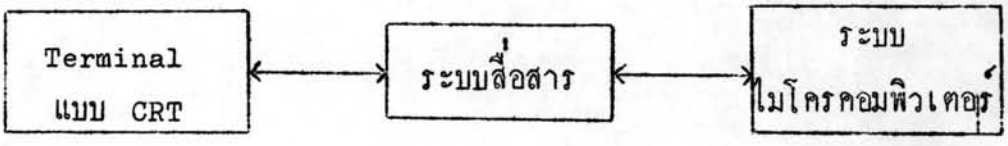
### 2.2 ส่วนประกอบในระบบไมโครคอมพิวเตอร์แบบ On-line

ระบบไมโครคอมพิวเตอร์แบบ On-line ที่ศึกษานี้ประกอบด้วย (ดูรูปที่ 2.1)

(1) ระบบไมโครคอมพิวเตอร์

(2) Terminal แบบ CRT

(3) ระบบสื่อสาร




รูปที่ 2.1 ระบบไมโครคอมพิวเตอร์แบบ On-line

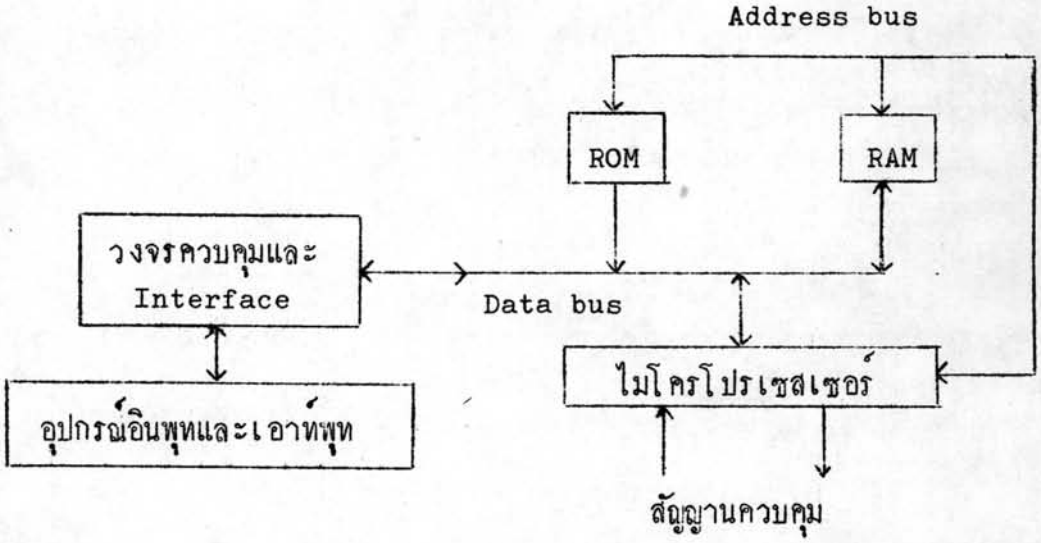
## 2.2.1 ระบบไมโครคอมพิวเตอร์

ระบบไมโครคอมพิวเตอร์ ประกอบด้วยส่วนประกอบสำคัญ 2 ส่วนคือ

1) Hardware

2) Software

Hardware[4] Hardware ของระบบไมโครคอมพิวเตอร์ที่สำคัญ ได้แก่ ไมโครโปรเซสเซอร์ Memory อุปกรณ์อินพุทและเอ้าพุท มีการจัดส่วนประกอบให้ทำงานสัมพันธ์กันตามลักษณะดังในรูปที่ 2.2




รูปที่ 2.2 การจัดส่วนประกอบทาง Hardware ของระบบไมโครคอมพิวเตอร์

Data bus เป็นกลุ่มของสายสัญญานที่ยอมให้ข้อมูลผ่านได้ทั้งสองทางเชื่อมอยู่ระหว่างไมโครโปรเซสเซอร์ ROM RAM และวงจร Interface

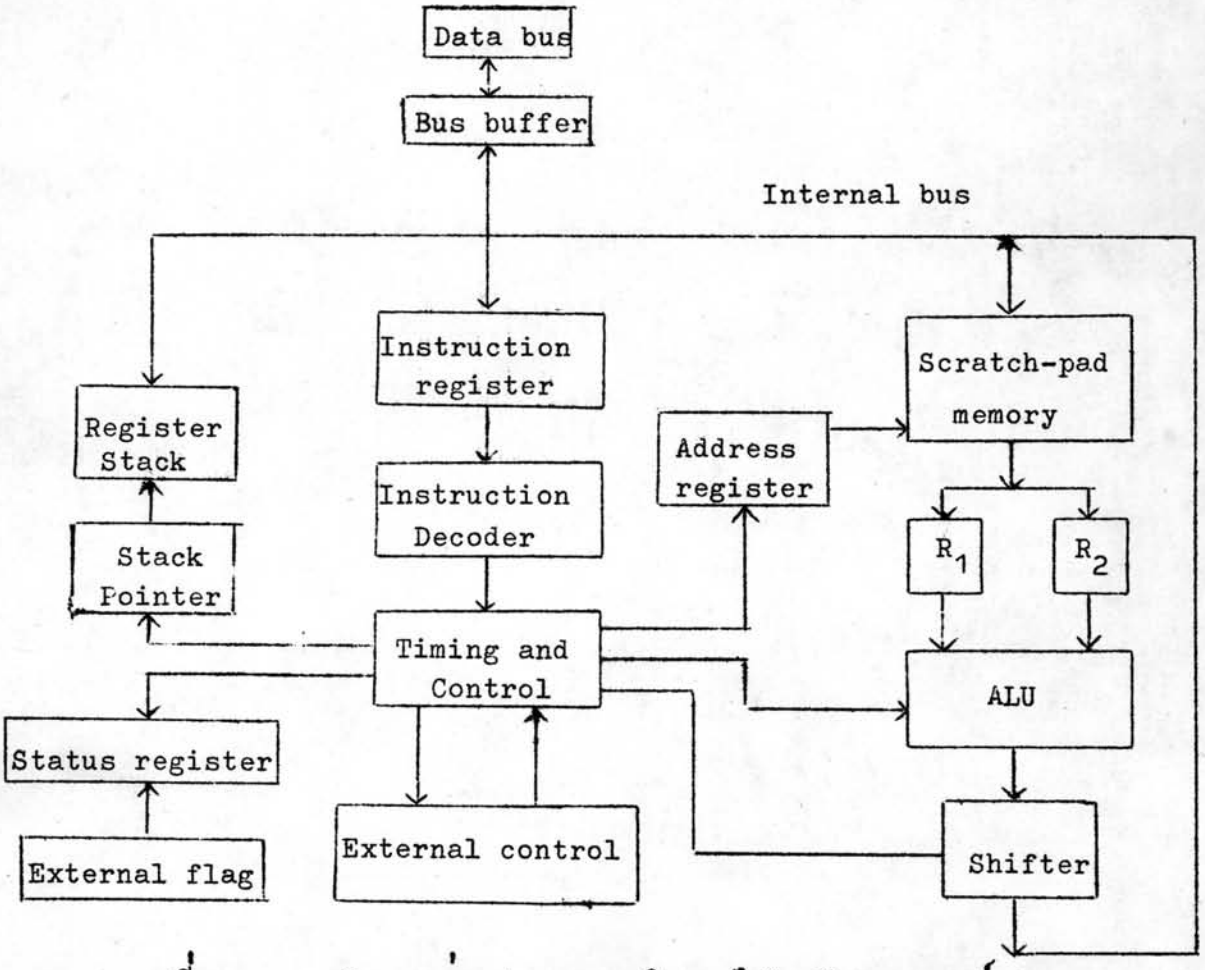
Address bus เป็นกลุ่มของสายสำหรับส่งค่า Address จากไมโครโปรเซสเซอร์ไปยัง ROM และ RAM เพื่อกำหนดตำแหน่งที่จะนำคำสั่งหรือ

ข้อมูลออกมาใช้

ROM (Read Only Memory) ใช้เก็บโปรแกรมสำหรับงานบางอย่างที่ไม่มีการเปลี่ยนแปลง อาจจะเป็น Microprogram ที่ควบคุม Micro-operation ของ CPU

RAM (Random Access Memory) ประกอบด้วย LSI (Large Scale Intergrated Circuit) จำนวนหนึ่งซึ่งใช้สำหรับเก็บโปรแกรมหรือข้อมูลของผู้ใช้จึงมีการเปลี่ยนแปลงอยู่เสมอ ส่วนนี้ก็คือ Main memory ของระบบไมโครคอมพิวเตอร์นั่นเอง

ไมโครโปรเซสเซอร์ (Microprocessor) เป็นส่วนสำคัญที่สุดในระบบไมโครคอมพิวเตอร์ มีส่วนประกอบภายในดังในรูปที่ 2.3




รูปที่ 2.3 แผนผังแสดงส่วนประกอบภายในของไมโครโปรเซสเซอร์

IR (Instruction Register) จะเป็นตัวรับคำสั่งจาก Memory ผ่านทาง Data bus แล้วส่งต่อไป Instruction decoder ทีละคำสั่ง

Instruction Decoder รับคำสั่งจาก IR มาแปลออกเป็นสัญญานควบคุมย่อย เพื่อควบคุมการทำงานของแต่ละส่วนในระบบ

Scratch-pad Memory เป็น Register กลุ่มหนึ่งซึ่งทำหน้าที่เก็บข้อมูลชั่วคราว เป็น Index Register เป็นที่เก็บ Address เป็น Memory buffer หรือเป็นที่เก็บ Program counter สิ่งที่ส่งเข้ามาเก็บเข้าได้ทาง Data bus เวลาส่งออกออกได้ทาง Data bus และ register $R_1$ หรือ $R_2$

ALU (Arithmatic and Logical Unit) ทำหน้าที่เกี่ยวกับการคำนวณและเปรียบเทียบทั้งแบบฐาน 2 และฐาน 10 มี Shifter ช่วยในการคำนวณต่ออยู่ด้วย

Status Register ประกอบด้วย flip-flop อยู่จำนวนหนึ่งเพื่อเก็บ Status เช่น ค่าใน Accumulator เป็นศูนย์ มี Output carry จาก ALU Status จะเป็น "1" เมื่อเกิดเหตุการณ์ที่กำหนด Status จะตรวจสอบได้ด้วยคำสั่งเพื่อตัดสินใจทำตามเงื่อนไขต่อไป

Address Register เป็นตัวกำหนด Address ใน Scratch-pad memory

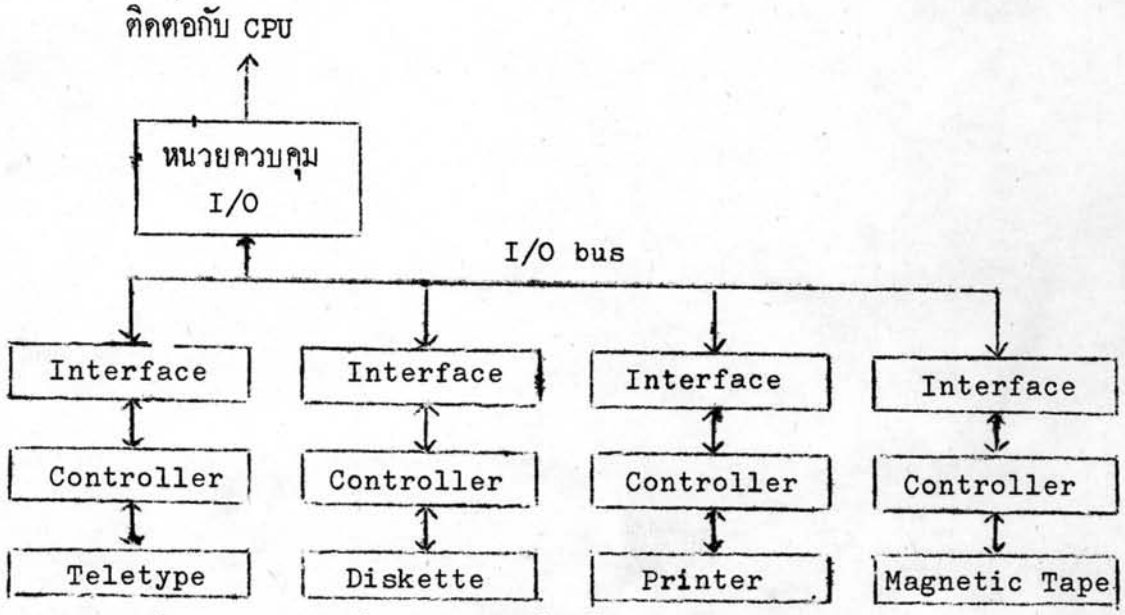
Stack Register เป็นตัวเก็บข้อมูลแบบ LIFO (Last In First Out) จะใช้เมื่อทำงานเกี่ยวกับ Subroutine nesting หรือ

Interrupt เพื่อเก็บค่าใน Scratch-pad memory ไว้ชั่วคราว บางแบบใช้ Memory เป็น Stack ส่วน Stack pointer จะชี้ตำแหน่ง Top ของ Stack

วงจร Timing และวงจรควบคุม จะสร้าง timing และสัญญานควบคุมรายละเอียดของลำดับการทำงานในแต่ละคำสั่ง ควบคุมการรับส่งข้อมูลใน Bus ภายใน ควบคุมการทำงานใน ALU และใน Scratch-pad memory และยังสร้างสัญญานติดกับภายนอกให้รู้ถึง Status ภายใน ระหว่างการ Fetch หน่วยควบคุมจะสร้าง Fetch status signal ที่จะบอกให้ Memory รู้ว่าต้องการคำสั่ง สวนระหว่าง Execute หน่วยควบคุมอาจจะอยู่ใน Memory read หรือ write หรือ I/O status ขึ้นอยู่กับคำสั่งนั้น

Bus buffer ควบคุมโดย Control Status Register เมื่อต้องการข้อมูลจากภายนอก Bus จะอยู่ใน Input mode เมื่อจะส่งข้อมูลออก Bus จะอยู่ใน Output mode

วงจรควบคุมและ Interface เป็นตัวเชื่อมอยู่ระหว่าง CPU กับอุปกรณ์ภายนอก ถ้ามีอุปกรณ์อยู่หลายตัว การจัดส่วนประกอบเพื่อเชื่อมกับอุปกรณ์ภายนอกมีดังต่อไปนี้ (ดูรูปที่ 2.4)




รูปที่ 2.4 การจัดส่วนประกอบเพื่อเชื่อมกับอุปกรณ์ภายนอก

วงจร Interface จะทำหน้าที่แปลคำสั่งที่ได้รับจาก I/O bus เพื่อส่งให้อุปกรณ์แต่ละตัว โดยสร้างสัญญาณป้อนให้กับหน่วยควบคุมของแต่ละอุปกรณ์ และยังทำหน้าที่ Synchronize การไหลของข้อมูลระหว่าง CPU กับอุปกรณ์ภายนอก

หน่วยควบคุม Input และ Output รับผิดชอบเกี่ยวกับ Information ที่อยู่ใน I/O bus โดยได้รับคำสั่งจาก CPU และทำงานตามคำสั่งโดยติดต่อกับอุปกรณ์ภายนอกผ่านวงจร Interface ก่อนเริ่มการติดต่อหน่วยควบคุมจะส่ง Address ของอุปกรณ์ที่จะติดต่อเข้าไปใน I/O bus ก่อน Interface ประจำอุปกรณ์จะ Decode ค่า Address ที่ส่งมา ถ้าเป็นของตัวเองก็จะเชื่อมตัวเองเข้ากับ I/O bus ถ้าไม่ใช่จะปลดตัวเองออก

Input and Output Bus ประกอบด้วยกลุ่มของสายส่งข้อมูล สายส่ง Address ของอุปกรณ์ สายส่ง Function code และสายส่งสภาพของ Status

หน่วยควบคุมอุปกรณ์ภายนอก (Controller) มีหน้าที่ควบคุมอุปกรณ์ภายนอกที่เป็น Electromechanical ตามสัญญาณที่ได้รับจาก Interface โดยที่อุปกรณ์ภายนอกแต่ละตัวจะมีหน่วยควบคุมประจำ ซึ่งวงจรภายในไม่เหมือนกันขึ้นอยู่กับชนิดของอุปกรณ์นั้น

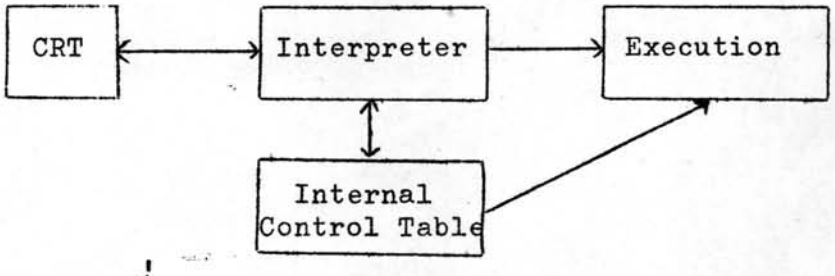
Software[5] Softwareของระบบไมโครคอมพิวเตอร์ที่จำเป็นสำหรับการปฏิบัติงานกับระบบ มีด้วยกัน 2 ส่วน คือ

1) Monitor

2) Interpreter

Monitor เป็นชื่อเรียก Operating System สำหรับเครื่องขนาดเล็ก บางทีก็เรียกว่า Executive Program มีหน้าที่ควบคุมการทำงานของคอมพิวเตอร์ให้มีประสิทธิภาพ ขณะทำงานโปรแกรมชุดนี้จะต้องอยู่ใน Memory ตลอดเวลา ดังนั้นบางเครื่องจึงเก็บโปรแกรมชุดนี้ไว้ใน ROM เมื่อเปิดเครื่องก็ทำงานได้ทันทีไม่ต้องเสียเวลา Load เข้าไป ตัว Monitor เองประกอบด้วย Routine ย่อยๆหลายอัน บาง Routine ใช้สำหรับตรวจดูค่าใน Memory บาง Routine ใช้สำหรับ Test Memory บาง Routine ใช้สำหรับป้อนค่าเข้าไปใน Memory ตาม Address ที่กำหนด บาง Routine ใช้ร่วมกับ Software อื่นๆ หรือเรียก Software ตัวอื่นมาใช้แล้วไปทำงานภายใต้ Software อื่นเช่น Interpreter . สำหรับ Software ที่ใช้ร่วมกับ Monitor ได้แก่ Floppy Disk Controller Firmware ซึ่งจะใช้คำสั่งควบคุม Disk ต้องส่งผ่าน Monitor เข้าไป ปกติในขณะทำงาน Monitor จะเป็นตัวรับผิดชอบในการรับและส่งข้อมูลกับอุปกรณ์ภายนอกด้วย

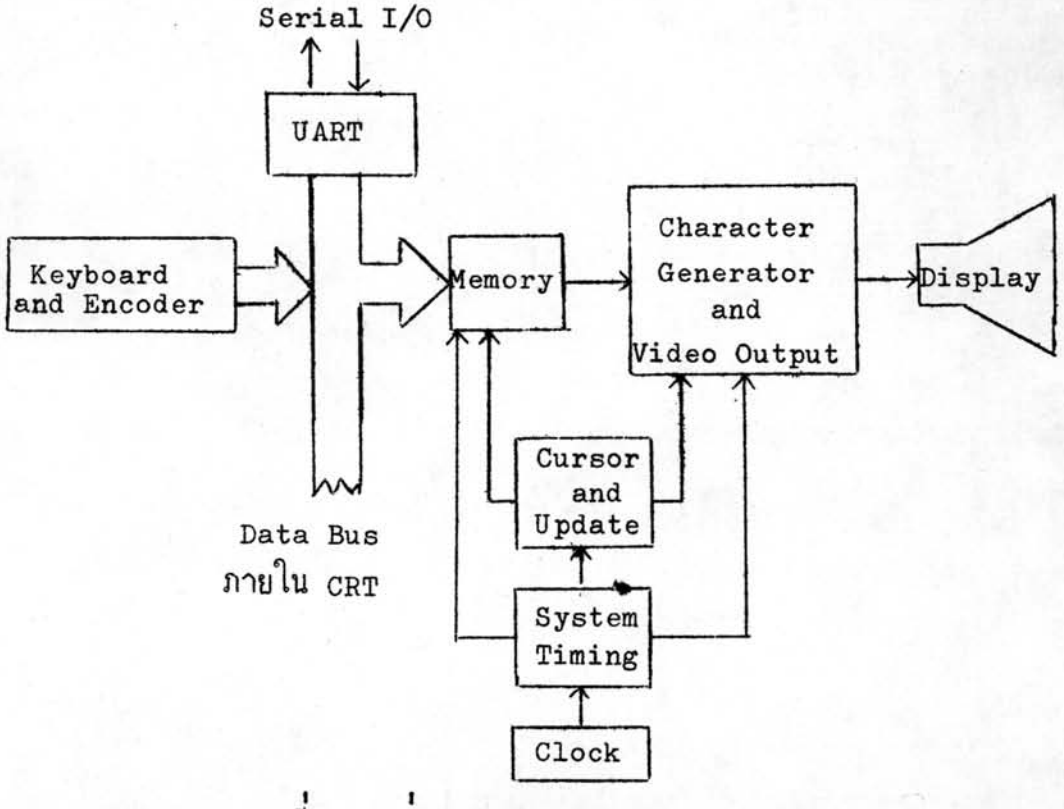
Interpreter เป็นโปรแกรมของระบบ (System-Program) ที่แปลคำสั่งใน High level language แล้ว Execute ทันทีทีละคำสั่งที่ป้อนเข้ามา การทำงานในลักษณะนี้ทำให้ Response เร็ว ในการเปลี่ยนแปลงโปรแกรม การแก้ไขโปรแกรมทำได้รวดเร็วในลักษณะการโต้ตอบกับเครื่อง (Interactive) เมื่อป้อนคำสั่งเข้าทาง Keyboard คำสั่งจะถูกแปลทันที ซึ่งจะทำให้ได้ Solution ในขณะที่ CRT ยังต่อโดยตรง (On-line) กับเครื่องคอมพิวเตอร์ ดังรูปที่ 2.5




รูปที่ 2.5 แผนภาพแสดงการทำงานของ Interpreter

## 2.2.2 Terminal แบบ CRT (Cathode Ray Tube)[6]

Terminal แบบ CRT เป็นอุปกรณ์ที่ใช้ป้อนข้อมูลหรือโปรแกรมเข้าทาง Keyboard เมื่อกดปุ่มบน Keyboard สัญญาณข้อมูลจะถูกส่งไปยังเครื่องคอมพิวเตอร์ด้วยรหัส เช่น รหัส ASCII (American Standard Code for Information-Interchange) นอกจากข้อมูลแล้วบน Keyboard ยังมีปุ่มควบคุมเช่น BREAK CR (Carriage Retern) ซึ่งจะสั่งให้เครื่องคอมพิวเตอร์ทำงานตามคำสั่งได้ ส่วนประกอบของ Terminal แบบ CRT มีดังต่อไปนี้ (ดูรูปที่ 2.6)




รูปที่ 2.6 ส่วนประกอบของ Terminal แบบ CRT

หอสมุดกลาง จุฬาลงกรณมหาวิทยาลัย



Keyboard and Encoder เป็นส่วนที่สร้างรหัสตัวอักษร เช่น รหัส ASCII และสัญญาณควบคุมบางตัวโดยวิธีกดปุ่มบน Keyboard ขณะกดปุ่มรหัสจะส่งเข้า Data Bus เพื่อส่งผ่าน UART ออกไปภายนอก ในขณะเดียวกันรหัสก็จะส่งไปเก็บไว้ที่ Memory ด้วย

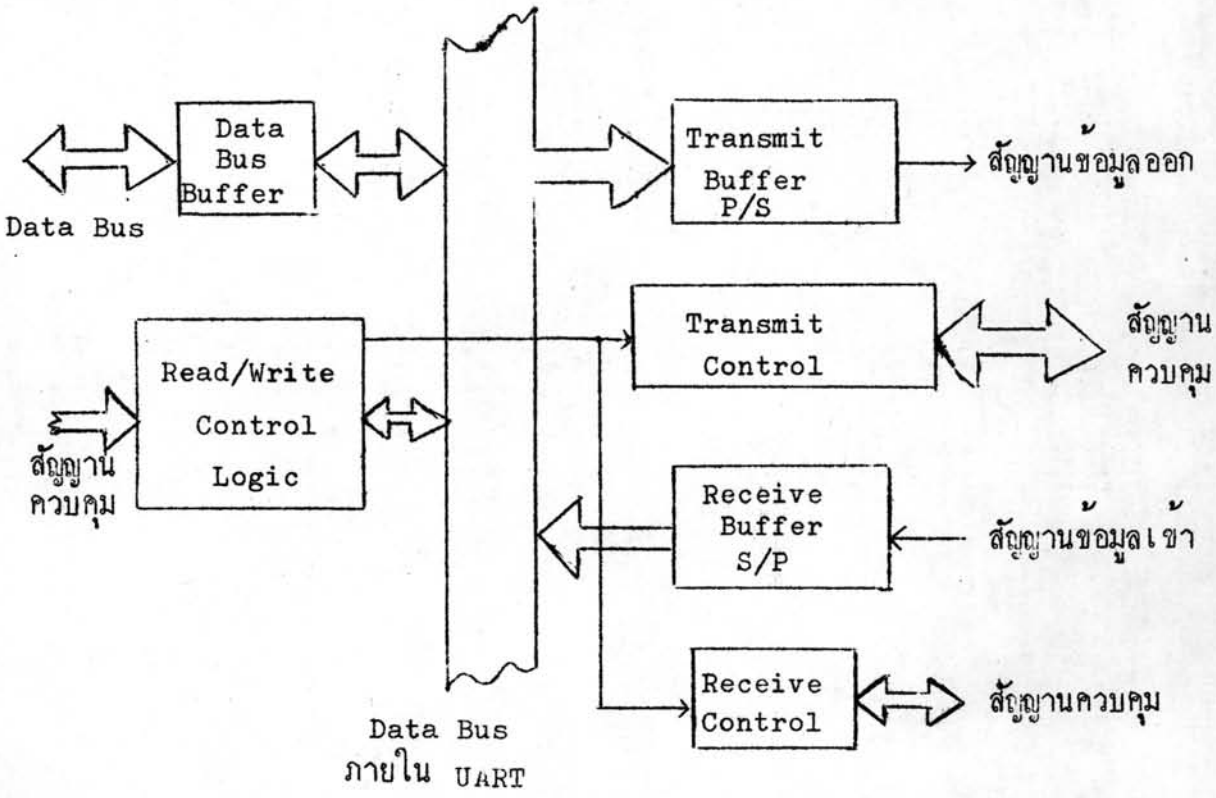
Memory ส่วนนี้ประกอบด้วย RAM จำนวนหนึ่ง ทำหน้าที่เก็บรหัสตัวอักษรที่ป้อนเข้ามาทาง Keyboard หรือรหัสตัวอักษรที่ป้อนเข้ามาจากเครื่องคอมพิวเตอร์ผ่านทาง UART เพื่อเก็บไว้แสดงบนจอภาพ รหัสตัวอักษรที่เก็บไว้จะส่งให้วงจร Character Generator อีกทีหนึ่ง

Character Generator and Vidio Output มีหน้าที่สร้างตัวอักษรให้ปรากฏบนจอภาพ โดยรับรหัสข้อมูลจาก Memory มาเปลี่ยนเป็นกลุ่มของ Dot ที่จะปรากฏเป็นตัวอักษรบนจอ ส่วนนี้ประกอบด้วย ROM ที่เก็บชุดของ Dot ตัวอักษร และวงจรสร้าง Dot ในตำแหน่งที่กำหนดด้วย Cursor บนจอกับวงจรขยายที่จะทำให้จอภาพทำงานได้

System Timing นำ Clock มาสร้าง Timing ให้แก่ทุกส่วนเพื่อกำหนดจุดเริ่มต้นและระยะเวลาทำงานของแต่ละส่วนให้สัมพันธ์กันอย่างถูกต้อง โดยเฉพาะการสร้าง Dot บนจอตามตำแหน่งและเวลาที่กำหนด เพราะการสร้างภาพจะต้อง Refresh ตลอดเวลาเพื่อไม่ให้ตัวอักษรจางหายไป

วงจร Cursor และ Update เป็นวงจรที่รับผิดชอบเกี่ยวกับตำแหน่งที่จะสร้างและเปลี่ยนแปลงตัวอักษรใน Memory โดย Cursor เป็นตัวกำหนดตำแหน่งบนจอซึ่งจะตรงกับตำแหน่งใน Memory ที่จะสร้างหรือแก้ไข การกำหนดตำแหน่งบนจอวงจรนี้จะสร้าง Dot รูปสี่เหลี่ยมเล็กๆไว้ที่ตำแหน่งจะสร้างหรือแก้ไขตัวอักษร

UART (Universal Asynchronous Receiver and Transmitter)[7]
UART มีหน้าที่รับผิดชอบเกี่ยวกับการ Interface สัญญาณข้อมูลแบบ Serial กับสัญญาณข้อมูลแบบ Parallel มีส่วนประกอบดังรูป 2.7




รูป 2.7 การทำงานของ UART

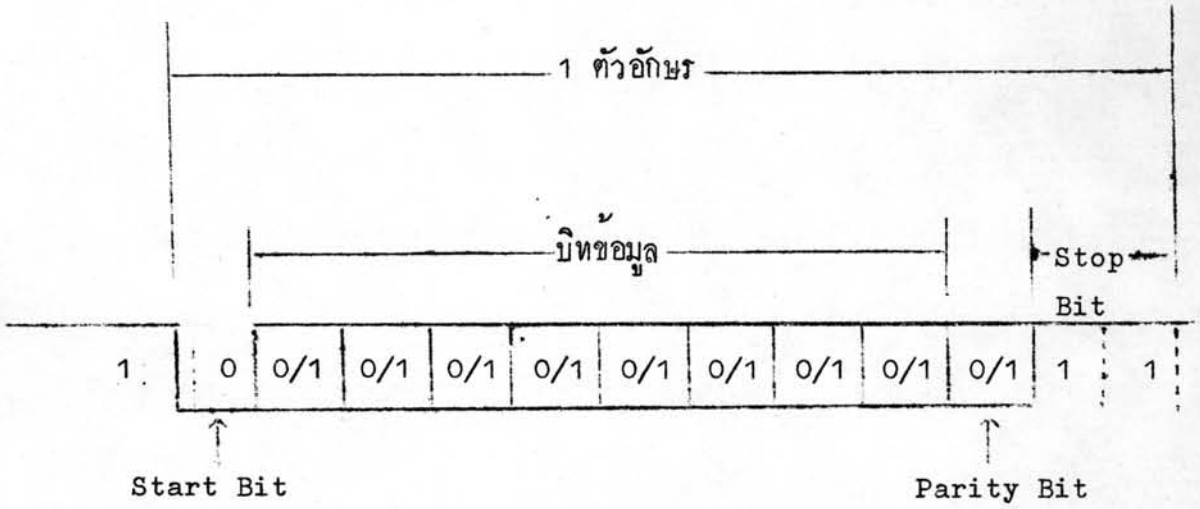
การทำงานของ UART แบ่งออกเป็นสองส่วน ส่วนแรกจะรับข้อมูลแบบ Parallel ผ่านทาง Data Bus Buffer เพื่อเปลี่ยนเป็นแบบ Serial พร้อมกับเพิ่ม Start และ Stop Bit โดย Transmit Buffer แล้วจึงส่งออก ส่วนที่สองจะรับสัญญาณแบบ Serial เพื่อเปลี่ยนเป็น Parallel พร้อมกับตัด Start และ Stop Bit ออก โดย Receive Buffer แล้วส่งไปยัง Data Bus ผ่าน Data Bus Buffer

การทำงานของ UART อยู่ภายใต้การควบคุมของสัญญาณที่ป้อนให้กับ Read/Write Control Logic ซึ่งจะทำงานร่วมกับ Transmit Control และ Receive Control ขึ้นอยู่กับว่าจะส่งข้อมูลเข้าหรือออก

Format[4] การรับส่งข้อมูลของ UART ทางด้าน Serial สำหรับการส่งแบบ Asynchronous เมื่อจะส่ง 1 ตัวอักษรจะต้องประกอบด้วยบิท (Bit) ต่างๆดังนี้ (ดูรูปที่ 2.8)

1) บิท Start 1 บิท
2) ตามด้วยข้อมูล 5-8 บิท แล้วแต่รหัสที่ใช้
3) ตามด้วยบิทตรวจสอบ (Parity Bit) 1 บิทหรือจะไม่มีก็ได้
4) บิท Stop จำนวน 1 หรือ 2 บิท

004885




รูปที่ 2.8 Format การรับส่งข้อมูลของ UART

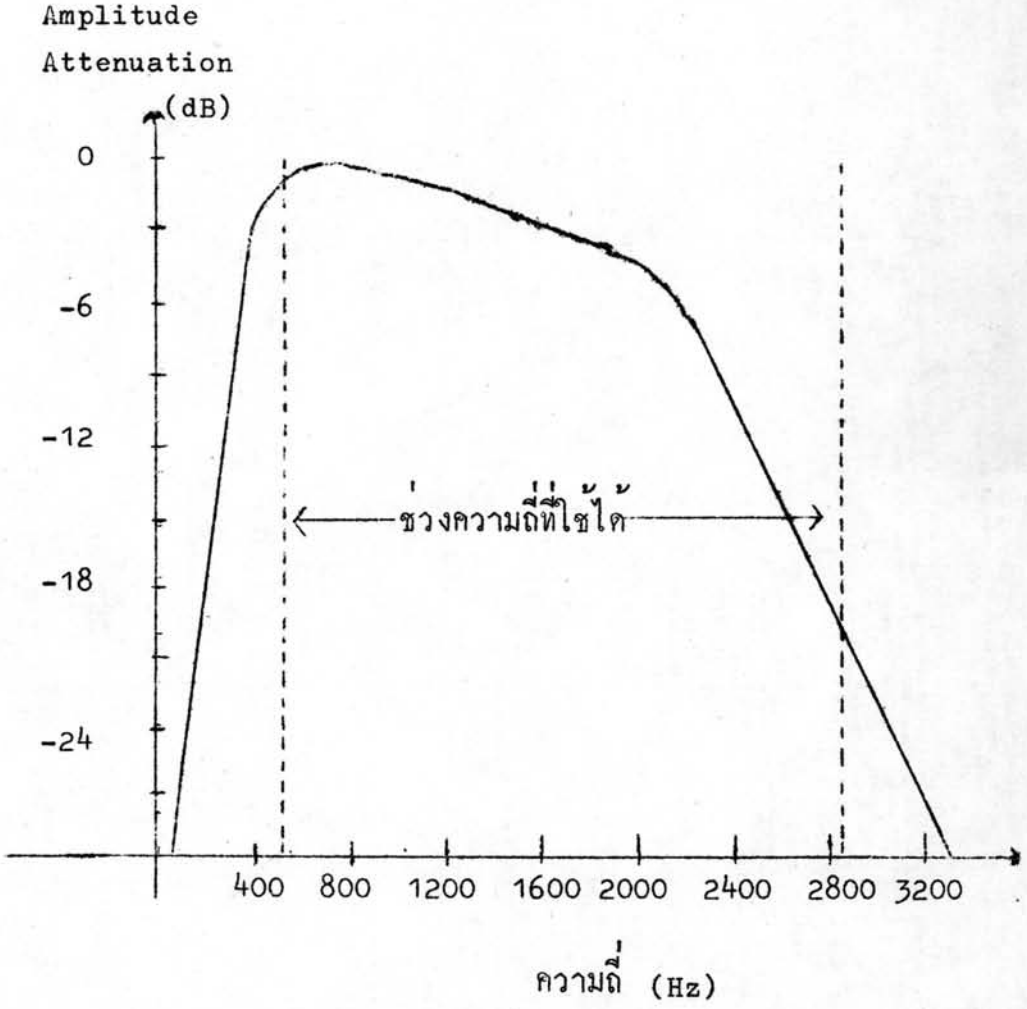
2.2.3 ระบบสื่อสารข้อมูล ระบบสื่อสารข้อมูลโดยปกติจะประกอบด้วย

1) สายโทรศัพท์
2) โมเด็ม

i 17A0A101

## สายโทรศัพท์ (Telephone Line)[8]

สายโทรศัพท์ได้ออกแบบไว้ในย่านความถี่เสียง ยอมให้เฉพาะความถี่ประมาณ 300-3300 Hz ผ่านได้หรือช่วงความถี่ประมาณ 3 KHz ความถี่ที่ต่ำหรือสูงกว่านี้ Amplitude จะลดลงอย่างรวดเร็วเนื่องจากคุณสมบัติด้านความถี่ของสาย เมื่อจะใช้สายโทรศัพท์สำหรับส่งข้อมูลก็ต้องเลือกความถี่ที่จะ Modulate ให้อยู่ภายในช่วง 300-3300 Hz ผลตอบทางความถี่ของสายโทรศัพท์แสดงอยู่ในรูปที่ 2.9




รูปที่ 2.9 ผลตอบทางความถี่ของสายโทรศัพท์

จากข้อกำหนดทางด้านความถี่ที่ยอมให้ช่วงความถี่ผ่านเพียง 3KHz จะมีผลถึง Information หรือ Data bit ส่งผ่านสายโทรศัพท์ จำนวนบิท/วินาทีสูงสุดที่ส่งได้ทางทฤษฎี เป็นไปตามสมการ[9]

$$C = B \log_2(1 + \frac{S}{N}) \quad \text{บิท/วินาที} \qquad (2.1)$$

ในที่นี้ C เป็น Channel Capacity

B เป็น Bandwidth

S เป็น Signal Power

N เป็น Noise Power

โดยถือว่า Channel ถูกกระทบกระเทือนโดย Gaussian Noise เท่านั้นถ้ามี Noise อื่นรบกวนหรือ Distortion จะทำให้รูปสมการเปลี่ยนแปลงไปและค่า C จะลดลง

## โมเด็ม (Modem)

การสื่อสารข้อมูลจะมุ่งความสนใจไปที่ข่ายสายโทรศัพท์ พยายามที่จะใช้ข่ายสายโทรศัพท์เพื่อรับส่งข้อมูลระหว่างคอมพิวเตอร์กับ Terminal เพราะสายโทรศัพท์มักจะติดตั้งอยู่แล้วโดยทั่วไปกับค่าใช้จ่ายเมื่อใช้บริการก็ต่ำ น่าที่จะใช้เป็นสื่อกลางในการติดต่อ แต่เนื่องจากสายโทรศัพท์ได้ออกแบบไว้สำหรับใช้ในย่านความถี่เสียง ยอมให้สัญญาณที่มีความถี่ประมาณ 300-3300 Hz ผ่านได้เท่านั้น หรือช่วงความถี่ประมาณ 3KHz ถ้าส่งสัญญาณข้อมูลซึ่งเป็นสัญญาณ Digital เข้าไปในสายโทรศัพท์โดยตรงสัญญาณจะเสียรูปร่างรวดเร็ว เนื่องจากความถี่ของสัญญาณ Digital สูงกว่าความถี่ของสายโทรศัพท์มากจึงทำให้ส่งข้อมูลไปไม่ได้ไกล

เพื่อให้ส่งสัญญาณข้อมูลไปได้ไกลในสายโทรศัพท์จึงจำเป็นต้องเปลี่ยนรูปของสัญญาณจากสัญญาณ Digital ให้เป็นสัญญาณความถี่ในย่านความถี่เสียงเรียกว่า

Modulation เมื่อสัญญาณส่งไปถึงปลายทางก็จะเปลี่ยนกลับไปเป็นสัญญาน Digital ตามเดิมเรียกว่า Demodulation การส่งและรับข้อมูลโดยทั่วไปอุปกรณ์ที่ทำหน้าที่ Modulation และ Demodulation มักจะอยู่รวมกันเรียกสั้นๆว่า Modem

โมเด็มแบ่งออกเป็นประเภทต่างๆขึ้นอยู่กับวิธี Modulation ดังนี้ [9]

1. On-OFF Keying (OOK) เป็นรูปหนึ่งของ Amplitude Modulation บางทีเรียกว่า Amplitude Shift Keying (ASK) เมื่อนำไปใช้กับการ Modulation แบบ Multilevel ระบบ OOK อธิบายการส่งพัลส์ได้ดังนี้

$$S_{AM}(t) = U_T(t) \cos (Wc\ t+\phi_T), \text{ สำหรับการส่งช่วงความถี่ Mark} \quad (2.2)$$
$$= 0 \quad , \text{ สำหรับการส่งช่วงความถี่ Space}$$

ในที่นี้ $U_T(t)$ เป็น Rectangular pulse

Modulation แบบนี้มักใช้กับการส่งข้อมูลที่มีความเร็วต่ำ

2. Frequency Shift Keying (FSK) เป็นการ Modulate ทาง frequency ด้วย Rectangular pulse โดยมี Amplitude ของ Carrier คงที่ ดังนั้นความถี่ส่งออกจะมี Amplitude คงที่ แต่ความถี่จะเปลี่ยนจากความถี่หนึ่งไปอีกความถี่หนึ่งอย่างรวดเร็วเมื่อระดับของสัญญาน Digital เปลี่ยน เราสามารถอธิบายการส่งพัลส์ได้ดังนี้

$$S_{FM}(t) = A \cos W_1 t \quad , \text{ สำหรับการส่งช่วงความถี่ Mark} \quad (2.3)$$
$$= A \cos W_2 t \quad , \text{ สำหรับการส่งช่วงความถี่ Space}$$

ในที่นี้ $W_1$ และ $W_2$ เป็นความถี่คงที่ของพัลส์ที่เป็น Mark และ Space ตามลำดับ

3. Phase-Shift Keying (PSK) เป็นการ Modulate ทางเฟส (Phase) ด้วย Rectangular พัลส์ โดยมี Amplitude และความถี่คงที่ แต่จะเปลี่ยนเฟสอย่างรวดเร็วเมื่อระดับสัญญาน Digital เปลี่ยน อธิบายการส่งพัลส์ได้ดังนี้

$$S_{PM}(t) = A\cos(W_c t+\phi_1) \quad \text{, สำหรับการส่งช่วงความถี่ Mark} \qquad (2.4)$$

$$= A\cos(W_c t+\phi_2) \quad \text{, สำหรับการส่งช่วงความถี่ Space}$$

ในที่นี้ $\phi_1$ และ $\phi_2$ เป็นเฟสของสัญญาน Mark และ Space ตามต้องการ ในกรณีที่เป็น Biphase PSK $\phi_1$ และ $\phi_2$ โดยมากจะเป็น 0 และ $\pi$