

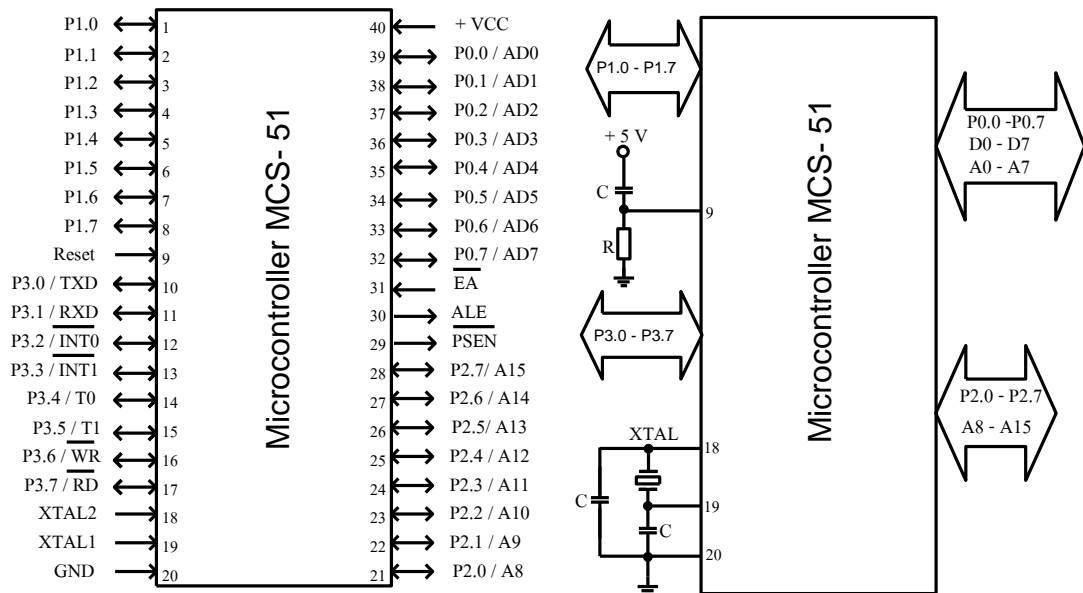
บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์

2.1.1 สถาปัตยกรรมไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล MCS-51 มีอยู่ด้วยกันหลายเบอร์ เช่น 80C31, 80C32, 80C51, 80C52, 89C51, 89C52 เป็นต้น ซึ่งการจัดเรียงขาของไมโครคอนโทรลเลอร์แสดงดังรูปที่ 2.1



รูปที่ 2.1 การจัดเรียงขาของไมโครคอนโทรลเลอร์

จากรูปที่ 2.1 ประกอบด้วยพอร์ตที่ใช้งานอยู่ 4 พอร์ตได้แก่ พอร์ต 0 พอร์ต 1 พอร์ต 2 และพอร์ต 3 มีรายละเอียดแต่ละพอร์ตดังนี้

- พอร์ต 0 (P0.0 –P0.7) เป็นอินพุทหรือเอาต์พุทพอร์ต ถ้ามีการขยายหน่วยความจำภายนอกหรืออินพุท / เอาต์พุทภายนอกจะใช้เป็น Data Bus (D0-D7) และ Address Bus (A0- A7)
- พอร์ต 1 (P1.0-P1.7) เป็นอินพุทหรือเอาต์พุทพอร์ต
- พอร์ต 2 (P2.0-P2.7) เป็นอินพุทหรือเอาต์พุทพอร์ต ถ้ามีการต่อหน่วยความจำภายนอกจะใช้เป็น Address Bus (A8-A15)

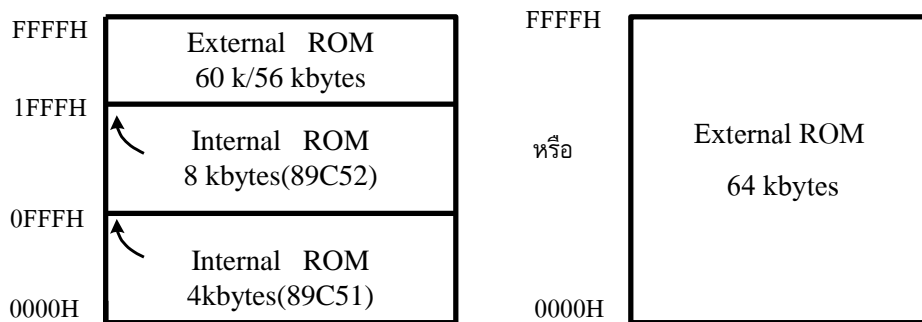
- พอร์ต 3 (P3.0-P3.7) เป็นอินพุทหรือเอาต์พุทพอร์ต ถ้าไม่ใช้ให้เป็นอินพุท/เอาต์พุทก็สามารถทำหน้าที่ตามชื่อหลังได้ดังนี้
- P3.0/TxD (Transmit Data) ใช้เป็นขาเอาต์พุทสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม
- P3.1/RxD (Receive Data) ใช้เป็นขาอินพุทสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม
- P3.2/ $\overline{\text{INT0}}$ (Interrupt 0) รับสัญญาณขัดจังหวะจากภายนอก No. 0
- P3.3/ $\overline{\text{INT1}}$ (Interrupt 1) รับสัญญาณขัดจังหวะจากภายนอก No. 1
- P3.4/T0 (Timer/Counter0) สามารถโปรแกรมได้ว่าจะให้เป็น Timer หรือ Counter ถ้าใช้สัญญาณ Clock จากภายนอกเข้ามาจะเป็น Counter ถ้าใช้สัญญาณ Clock จากภายในจะเป็น Timer
- P3.5/T1 (Timer/Counter1) ทำหน้าที่ทำนองเดียวกับ T0
- P3.6/ $\overline{\text{WR}}$ (Write) ส่งสัญญาณควบคุมการเขียนข้อมูลจาก MCS-51 ไปยังภายนอก
- P3.7/ $\overline{\text{RD}}$ (Read) ส่งสัญญาณควบคุมการอ่านข้อมูลจากภายนอกเข้ามายัง MCS-51
- Reset เป็นขาอินพุท Active High เริ่มต้นทำงานขา Reset จะได้รับลอจิก “1” ซึ่งจะทำให้ PC=0000H, P0=P1=P2=P3=FFH, SP=07H และรีจิสเตอร์ทุกตัวเป็นศูนย์
- XTAL 1 และ XTAL 2 เป็นขาที่ใช้สำหรับต่อกับตัวคริสตอล เพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะ การทำงานของไมโครคอนโทรลเลอร์ใช้ในย่านความถี่ 11-16 MHz
- $\overline{\text{EA}}$ (External Access) เป็นขาอินพุทถ้ารับลอจิก “0” ใช้โปรแกรมจัดระบบจากภายนอก ถ้ารับลอจิก “1” ใช้โปรแกรมจัดระบบจากภายในเช่น MCS-51 เบอร์ AT89C51, AT89C52
- ALE (Address Latch Enable) ขานี้จะส่งสัญญาณที่มีความถี่ 1 / 6 เท่าของสัญญาณนาฬิกาจากคริสตอลมาตลอดเวลา ยกเว้นกรณีที่ต้องการติดต่อกับหน่วยความจำภายนอก ขานี้จะเป็นลอจิก “1” แล้วจะส่ง Address 8 บิต (A0- A7) มาที่ P0.0-P0.7 ช่วงขณะแล้วเปลี่ยนเป็น Data Bus (D0-D7)
- $\overline{\text{PSEN}}$ (Program Store Enable) ขานี้จะส่งสัญญาณ Active Low ออกมา 2 ครั้ง ในแต่ละแมชชีนไซเคิล เมื่อต้องการอ่านโปรแกรมจากภายนอก แต่ถ้าต้องการอ่านโปรแกรมจากภายในจะไม่มีสัญญาณที่ขานี้
- +Vcc เป็นขาที่ใช้สำหรับต่อไฟเพื่อเลี้ยงวงจร DC +5V
- GND (Ground) เป็นขากราวด์สำหรับต่อกับกราวด์ของระบบ DC 0V

2.1.2 การจัดหน่วยความจำ

- Program Memory (ROM) เก็บโปรแกรมจัดระบบงาน
 - Data Memory (RAM) เก็บข้อมูลหรือโปรแกรมที่มีการเปลี่ยนแปลงแก้ไขได้
- เช่น Set Point ต่างๆ

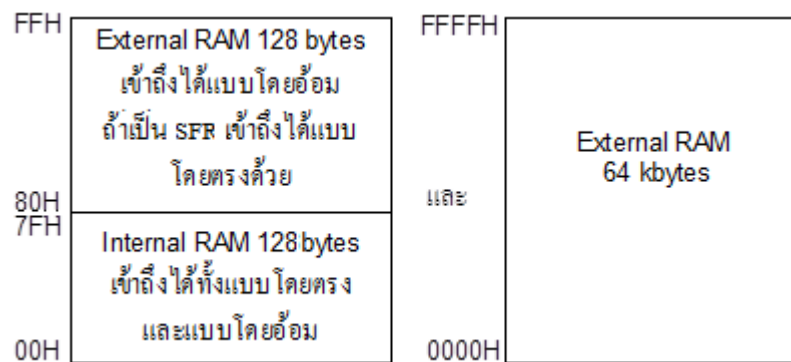
Address ของ Program Memory แสดงดังตารางที่ 2.1

ตารางที่ 2.1 Address ของ Program Memory



Address ของ Data Memory แสดงดังตารางที่ 2.2

ตารางที่ 2.2 Address ของ Data Memory



Address สำหรับการบริการการขัดจังหวะ

- 0003H ← จากอินเทอร์รัปต์ INT0
- 0013H ← จากอินเทอร์รัปต์ INT1
- 0023H ← จากอินเทอร์รัปต์ของการสื่อสารแบบอนุกรม
- 000BH ← จากอินเทอร์รัปต์ของ T0
- 001BH ← จากอินเทอร์รัปต์ของ T1
- 002BH ← จากอินเทอร์รัปต์ของ T2

Internal RAM Address 00H-7FH (128 Bytes) แสดงดังตารางที่ 2.3
ตารางที่ 2.3 Internal RAM Address 00H-7FH (128 Bytes)

	MSB							LSB	
7FH									
30H	ย่านการใช้งานทั่วไป								
2FH	7F	7E	7D	7C	7B	7A	79	78	
2EH	77	76	75	74	73	72	71	70	
2DH	6F	6E	6D	6C	6B	6A	69	68	
2CH	67	66	65	64	63	63	61	60	
2BH	5F	5E	5D	5C	5B	5A	59	58	
2AH	57	56	55	54	53	52	51	50	
29H	4F	4E	4D	4C	4B	4A	49	48	
28H	47	46	45	44	43	42	41	40	
27H	3F	3E	3D	3C	3B	3A	39	38	
26H	37	36	35	34	33	32	31	30	
25H	2F	2E	2D	2C	2B	2A	29	28	
24H	27	26	25	24	23	22	21	20	
23H	1F	1E	1D	1C	1B	1A	19	18	
22H	17	16	15	14	13	12	11	10	
21H	0F	0E	0D	0C	0B	0A	09	08	
20H	07	06	05	04	03	02	01	00	
18H-1FH	Register Bank 3								R0-R7 (8 Bytes)
10H-17H	Register Bank 2								R0-R7 (8 Bytes)
08H-0FH	Register Bank 1								R0-R7 (8 Bytes)
00H-07H	Register Bank 0								R0-R7 (8 Bytes)

ย่าน 20H-2FH ใช้คำสั่งเข้าถึงข้อมูลในระดับบิตได้ทั้งหมด 128 บิต (16 Bytes)

2.1.3 รีจิสเตอร์ (Register) หรือตัวแปร

1 รีจิสเตอร์ใช้งานทั่วไป (R0-R7 Address 00H-07H)

2 รีจิสเตอร์ใช้งานเฉพาะ (SFR Address 80H-FFH)

เมื่อ SFR: Special Function Register

AT89C51 มี SFR 22 ตัว ส่วน AT89C52 AT89SXX มี SFR 28 ตัว แสดงรายละเอียดดังตารางที่ 2.4

ตารางที่ 2.4 รีจิสเตอร์ใช้งานเฉพาะ

Symbol	Name of Register	Address
*A	Accumulator	E0H
*B	B	F0H
SP	Stack Pointer	81H

ตารางที่ 2.4 รีจิสเตอร์ใช้งานเฉพาะ (ต่อ)

Symbol	Name of Register	Address
DPL	Data Pointer Low	82H
DPH	Data Pointer High	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	A0H
*P3	Port 3	B0H
SBUF	Serial Data Buffer	99H
TL0	Timer/Counter Low 0	8AH
TH0	Timer/Counter High 0	8BH
TL1	Timer/Counter Low 1	8CH
TH1	Timer/Counter High 1	8DH
PCON	Power Control	87H
*SCON	Serial Control	98H
*TCON	Timer/Counter Control	88H
TMOD	Timer/Counter Mode Control	89H
*IE	Interrupt Enable Control	A8H
*IP	Interrupt Priority Control	B8H
*PSW	Program Status Word	D0H

ส่วนรีจิสเตอร์ที่แสดงดังตารางที่ 2.5 เป็น Register 6 ตัวที่มีอยู่ใน AT89C52, AT89SXX

ตารางที่ 2.5 รีจิสเตอร์เพิ่มเติมของ AT89C52, AT89SXX

Symbol	Name of Register	Address
TL2	Timer/Counter Low 2	CCH
TH2	Timer/Counter High 2	CDH
RCAP2L	Capture Register Low	CAH
RCAP2H	Capture Register High	CBH
T2CON	Timer/Counter Control 2	C8H
T2MOD	Timer/Counter Mode Control 2	C9H

Note: PC (Program Counter) ไม่มี Address ประจำตัว

* เป็นรีจิสเตอร์ที่เข้าถึงในระดับบิตได้

ตารางชื่อประจำบิตของรีจิสเตอร์ที่เข้าถึงในระดับบิตได้ แสดงดังตารางที่ 2.6

ตารางที่ 2.6 ตารางชื่อประจำบิตของรีจิสเตอร์ที่เข้าถึงในระดับบิตได้

Symbol	ชื่อประจำบิตของรีจิสเตอร์							
Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A	ACC.7	ACC.6	ACC.5	ACC.4	ACC.3	ACC.2	ACC.1	ACC.0
B	B.7	B.6	B.5	B.4	B.3	B.2	B.1	B.0
P0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
P1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
P2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
P3	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
IE	EA	-	ET2	ES	ET1	EX1	ET0	EX0
IP	-	-	PT2	PS	PT1	PX1	PT0	PX0
PSW	C	AC	F0	RS1	RS0	OV	F1	P

คำอธิบาย Special Function Register : SFR

Accumulator : ACC, A

ใช้เป็นตัวตั้งและเก็บผลลัพธ์จากการกระทำทางคณิตศาสตร์และลอจิก หรือเป็นตัวรับ/ส่ง ข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก

Register: B

ใช้งานได้ทั่วไปหรือใช้เป็นตัวคูณหรือตัวหารในคำสั่ง MUL AB, DIV AB

Stack Pointer: SP

ใช้เป็นตัวชี้ตำแหน่งของ Stack ในคำสั่ง PUSH, POP, CALL, RET

Port Register : P0, P1, P2, P3

ใช้เป็น Input/Output Port เพื่อติดต่อกับอุปกรณ์รอบนอกในกรณีที่มีการต่อหน่วยความจำหรือหน่วยอินพุต/เอาต์พุตภายนอกเพิ่มเติม P0 ใช้เป็น A0-A7 และ D0-D7 P2 ใช้เป็น A8-A15 P3.6 เป็น \overline{WR} (Write Data) P3.7 เป็น \overline{RD} (Read Data)

Serial Data Buffer : SBUF

ใช้ในการเก็บข้อมูลที่ส่งออกที่ขา TxD หรือรับข้อมูลเข้ามาจากขา RxD ในการรับ/ส่งข้อมูลแบบอนุกรม

Timer/Counter Register : T0 (TH0, TL0) T1 (TH1, TL1) T2 (TH2, TL2)

ใช้ในการสร้างฐานเวลา จับเวลา หรือนับจำนวนพัลส์มีขนาด 16 บิต

Capture Register: RCAP2 (RCAP2H, RCAP2L)

ใช้งานเมื่อกำหนดให้ T2 ทำงานในโหมดแคปเจอร์ ซึ่งเป็นโหมดที่กำหนดให้ไมโครคอนโทรลเลอร์ทำการตรวจจับการเปลี่ยนแปลงสถานะทางลอจิกที่ขา T2EX เพื่อใช้ประโยชน์ในการวัดคาบเวลา ความถี่ และการเปลี่ยนแปลงของสัญญาณที่ขา T2EX มีขนาด 16 บิต

Control Register มีทั้งหมด 6 ตัว ได้แก่

Power Control : PCON

ใช้ในการกำหนดอัตราการรับ/ส่งข้อมูลแบบอนุกรมและกำหนดการทำงานในโหมดประหยัดพลังงาน

Serial Control : SCON

ใช้ในการควบคุมการทำงานของวงจรสื่อสารแบบอนุกรมภายในไมโครคอนโทรลเลอร์

Timer/Counter Control : TCON, T2CON

ใช้ในการควบคุมการทำงานของวงจร Timer / Counter ภายในไมโครคอนโทรลเลอร์

Timer/Counter Mode Control : TMOD, T2MOD

ใช้ในการกำหนดโหมดหรือลักษณะการทำงานของ Timer/Counter ภายในไมโครคอนโทรลเลอร์

Interrupt Enable Control : IE

ใช้ในการกำหนดลักษณะการตอบสนองการอินเทอร์รัปต์

Interrupt Priority Control : IP

ใช้ในการกำหนดลำดับความสำคัญของการตอบสนองการบริการอินเทอร์รัปต์ว่าจะให้ตอบสนองการเกิดอินเทอร์รัปต์ในลักษณะใดก่อน

Program Counter: PC

ใช้เป็นตัวชี้ตำแหน่งของโปรแกรมที่ CPU จะไปประมวลผลเมื่อ Reset จะทำให้ PC=0000H (16 บิต) แล้วจะเปลี่ยนแปลงอัตโนมัติตามคำสั่งที่ประมวลผล

Program Status Word : PSW

ใช้เป็นตัวแสดงสภาพของผลลัพธ์จากการทำงานทางคณิตศาสตร์หรือลอจิกและยังใช้เป็นตัวเลือกให้ R0-R7 อยู่ใน Register Bank ใดๆ

2.1.4 หลักการเขียนโปรแกรมภาษาแอสเซมบลี

ภาษาคอมพิวเตอร์แบ่งเป็น 2 ระดับ

1. ภาษาระดับสูง (High Level Language) เช่น C, VB
2. ภาษาระดับต่ำ (Low Level Language) ได้แก่ ภาษาเครื่องและภาษาแอสเซมบลี

ภาษาเครื่อง เขียนแทนด้วยรหัสเลขฐานสองหรือฐาน 16 เช่นต้องการให้ค่าของ A=5DH ภาษาเครื่องคือ 745DH หรือ 0111, 0100, 0101, 1101B

ภาษาแอสเซมบลี เขียนแทนด้วยรหัสย่อตามที่บริษัทผู้ผลิตไมโครคอนโทรลเลอร์หรือไมโครโปรเซสเซอร์เบอร์นั้นๆเป็นผู้กำหนดเช่นต้องการให้ค่าของ A=5DH ภาษาแอสเซมบลีคือ MOV A, #5DH

รูปแบบของโปรแกรมภาษาแอสเซมบลี แบ่งเป็น 4 Fields ได้แก่

1 Label	2 Mnemonic	3 Operands	4 Comment
DELAY:	MOV	R1,#50H	;R1=50H
LOOP:	MOV	R2,#60H	;R2=60H
HERE:	DJNZ	R2,HERE	;R2-1 UNTIL=00H
	DJNZ	R1,LOOP	;R1-1 IF≠ 00H GO LOOP
	RET		;IF R1=00H RETURN

1. Label เป็นจุดที่บอกให้รู้ตำแหน่งการกระโดดไปหรือการเรียกโปรแกรมย่อยนั้นๆ การเขียน Label ต้องไม่เว้นวรรคปิดท้ายด้วยโคลอน (:)

2. Mnemonic เป็นรหัสคำสั่ง บอกให้รู้ว่าต้องการทำอะไรเขียนตามข้อกำหนด

3. Operands ตัวกระทำหรือตำแหน่งการกระโดด ถ้ามีมากกว่า 1 ให้คั่นด้วยคอมมา (,) (Mnemonic + Operands = Command, คำสั่ง)

4. Comment คำอธิบายโปรแกรม ต้องขึ้นต้นด้วยเซมิโคลอน (;) มีหรือไม่มี Comment ก็ได้ ซึ่งจะไม่มีผลต่อโปรแกรม

ตารางสัญลักษณ์ของค่าต่างๆในการเขียนโปรแกรมภาษาแอสเซมบลี MCS-51 แสดงดังรูปที่

2.7

ตารางที่ 2.7 สัญลักษณ์ของค่าต่างๆในการเขียนโปรแกรมภาษาแอสเซมบลี MCS-51

สัญลักษณ์	คำอธิบาย
Rn	R0-R7 อ้างอิงข้อมูลที่เป็นค่าของ R0-R7
Direct หรือ Dadd	00H-0FFH อ้างอิงข้อมูลในหน่วยความจำภายในที่กำหนดตำแหน่งขนาด 8 บิต แบบโดยตรง
@Ri	@R0, @R1 อ้างอิงข้อมูลในหน่วยความจำภายในโดยค่าตำแหน่งขนาด 8 บิต กำหนดด้วยค่าของ R0,R1
#data	#00H-#0FFH ข้อมูลขนาด 8 บิต
#data16	#0000H-#0FFFFH ข้อมูลขนาด 16 บิต
Bit	00H-0FFH อ้างอิงข้อมูลที่ละบิตใน Internal RAM,SFR
Addr16(Cadd)	เป็น Label หรือตำแหน่งขนาด 16 บิต อ้างอิงตำแหน่งของหน่วยความจำได้ 64 k (0000H-0FFFFH)
Addr11(Cadd)	เป็น Label หรือตำแหน่งขนาด 11 บิต อ้างอิงตำแหน่งของหน่วยความจำได้ 2 k (0000H-07FFH นับจากจุดที่อยู่)
Rel (Cadd)	เป็น Label หรือตำแหน่งขนาด 8 บิต อ้างอิงตำแหน่งของหน่วยความจำห่างจากจุดที่อยู่ในระยะ 00H-0FFH

2.1.5 การเข้าถึงข้อมูล (Addressing Mode)

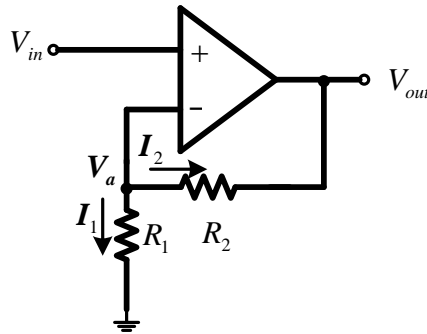
-Direct Addressing	EX.	MOV	A,7FH	;A = (7FH)
-Indirect Addressing	EX.	MOV	A,@R1	;A = (R1)
-Immediate Addressing	EX.	MOV	A,#7FH	;A = 7FH
-Register Addressing	EX.	MOV	A,R1	;A = R1
-Index Addressing	EX.	MOVC	A,@A+DPTR	;A = (A+DPTR)
-Register Specific Instruction	EX.	INC	A	;A ← A+1

Note : (7FH) = ข้อมูลที่อยู่ใน Internal RAM Address 7FH

(R1) = ข้อมูลที่อยู่ใน Internal RAM Address กำหนดโดยค่าของ R1

2.2.1 วงจรขยายสัญญาณแบบไม่กลับเฟส (Non-Inverting Amplifier)

วงจขยายสัญญาณแบบไม่กลับเฟส แสดงดังรูปที่ 2.2 ประกอบด้วยออปแอมป์และตัวต้านทานสองตัวโดยที่ตัวต้านทาน R_1 ต่ออยู่ระหว่างขาอินพุตลบและกราวด์ ส่วนตัวต้านทาน R_2 ต่อป้อนกลับจากเอาต์พุตกลับมายังอินพุตลบ



รูปที่ 2.2 วงจขยายสัญญาณแบบไม่กลับเฟส

สมมติให้ออปแอมป์เป็นอุดมคติ ซึ่งในการหาอัตราขยายของวงจร หาได้โดยใช้กฎกระแสของเคอร์ชอฟฟ์ (Kirchhoff's Current Law : **KCL**) ที่โหนด V_a และใช้คุณสมบัติของ ออปแอมป์คือ กระแสไหลเข้าขาอินพุตของออปแอมป์ทั้งสองขามีค่าเป็นศูนย์ และแรงดันที่ตกคร่อมระหว่างขาอินพุตทั้งสองมีค่าเป็นศูนย์ หรือกล่าวได้ว่าแรงดันขาอินพุตขาบวกเท่ากับแรงดันขาอินพุตลบของออปแอมป์ ดังนั้นที่โหนด V_a จะได้

$$V_a = V_{in}$$

และ

$$I_1 + I_2 = 0$$

$$\frac{V_{in} - 0}{R_1} + \frac{V_{in} - V_{out}}{R_2} = 0$$

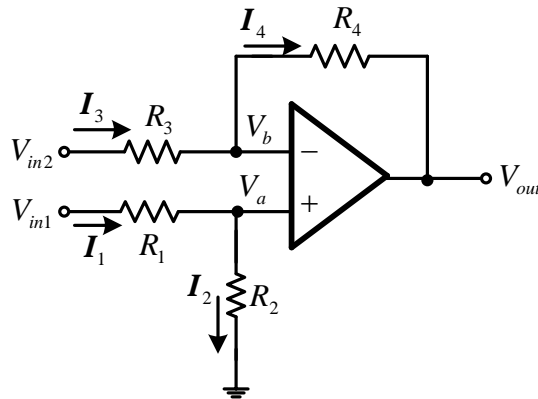
จะได้ค่าแรงดันเอาต์พุตเป็น

$$\boxed{V_{out} = \left(1 + \frac{R_2}{R_1}\right)V_{in}} \quad (2.1)$$

จากสมการ (2.1) จะเห็นว่าสัญญาณเอาต์พุตและสัญญาณอินพุตมีเฟสตรงกัน โดยที่อัตราการขยายขึ้นอยู่กับค่าความต้านทาน R_1 และ R_2

2.2.2 วงจรขยายผลต่างของสัญญาณ (Differential Amplifier)

วงจรรขยายผลต่างของสัญญาณ แสดงดังรูปที่ 2.3 ประกอบด้วยแรงดันอินพุต 2 อินพุตคือ V_{in1} , V_{in2} และตัวต้านทานต่อรวมกันอยู่ 4 ตัว คือ R_1 , R_2 , R_3 , R_4 หากค่าแรงดันเอาท์พุทได้ดังนี้



รูปที่ 2.3 วงจรรขยายผลต่างของสัญญาณ

จากคุณสมบัติของออปแอมป์ทางอุดมคติ จะได้ $V_a = V_b$

ใช้ **KCL** ที่โหนด V_a

จะได้

$$\begin{aligned} I_1 + I_2 &= 0 \\ \frac{V_a - V_{in1}}{R_1} + \frac{V_a - 0}{R_2} &= 0 \end{aligned} \quad (2.2)$$

$$V_a = \left(\frac{R_2}{R_1 + R_2} \right) V_{in1}$$

ใช้ **KCL** ที่โหนด V_b

จะได้

$$\begin{aligned} I_3 + I_4 &= 0 \\ \frac{V_b - V_{in2}}{R_3} + \frac{V_b - V_{out}}{R_4} &= 0 \end{aligned} \quad (2.3)$$

$$V_{out} = \left(\frac{R_3 + R_4}{R_3} \right) V_b - \frac{R_4}{R_3} V_{in2}$$

แทนค่าสมการ (2.2) ลงในสมการ (2.3) จะได้

$$V_{out} = \left(\frac{R_3 + R_4}{R_3} \right) \left(\frac{R_2}{R_1 + R_2} \right) V_{in1} - \frac{R_4}{R_3} V_{in2} \quad (2.4)$$

ถ้ากำหนดให้ ค่าความต้านทานเป็นดังนี้ $R_1 = R_3 = R_x$, $R_2 = R_4 = R_y$ แทนในสมการ (2.4) จะได้

$$V_{out} = \frac{R_y}{R_x} (V_{in1} - V_{in2}) \quad (2.5)$$

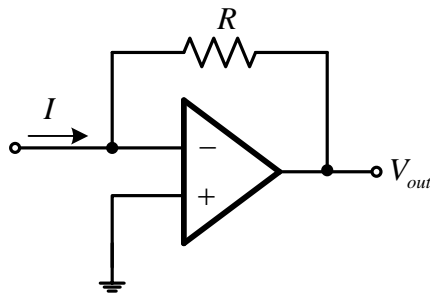
ถ้ากำหนดให้ ค่าความต้านทานมีค่าเท่ากันทุกตัวนั่นคือ $R_1 = R_2 = R_3 = R_4 = R$ แทนในสมการ (2.4) จะได้

$$V_{out} = V_{in1} - V_{in2} \quad (2.6)$$

จากสมการ (2.4), (2.5) และ (2.6) จะเห็นได้ว่าแรงดันเอาต์พุตขึ้นอยู่กับผลต่างของแรงดันอินพุตทั้งสองและค่าความต้านทานที่ต่ออยู่ ดังนั้นในการเลือกใช้สมการใดขึ้นอยู่กับผู้ใช้งานที่จะออกแบบเลือกใช้

2.2.3 วงจรแปลงกระแสเป็นแรงดัน (Current to Voltage Converter)

วงจรแปลงกระแสเป็นแรงดันทำหน้าที่ในการแปลงกระแสทางอินพุตให้ได้เอาต์พุตเป็นแรงดัน ซึ่งมีวงจรพื้นฐานแสดงดังรูปที่ 2.4



รูปที่ 2.4 วงจรแปลงกระแสเป็นแรงดัน

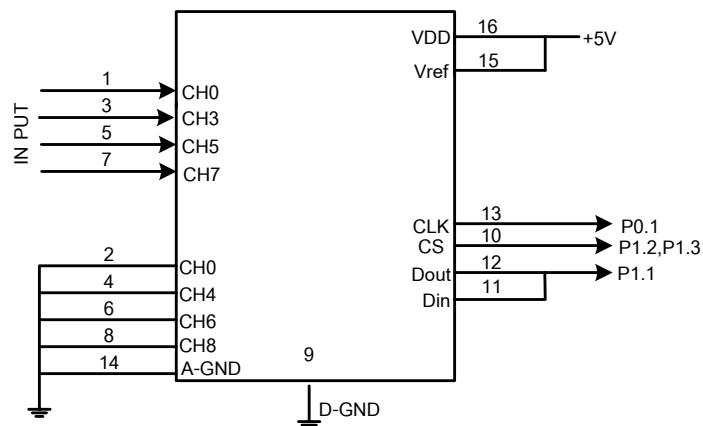
จากรูปที่ 2.4 จะเห็นได้ว่าแรงดันที่ขั้วลบของออปแอมป์จะมีค่าเท่ากับแรงดันที่ขั้วบวกของออปแอมป์ซึ่งมีค่าเป็นศูนย์ จึงทำให้แรงดันที่ขั้วลบของออปแอมป์มีค่าเป็นศูนย์ด้วย จะได้กระแสอินพุต I เท่ากับกระแสที่ไหลผ่านตัวต้านทาน R เขียนสมการได้เป็น

$$V_{out} = -IR \tag{2.7}$$

จากสมการ (2.7) จะได้แรงดันเอาต์พุตจะขึ้นอยู่กับค่าของกระแสอินพุต I คูณกับค่าตัวต้านทาน R ที่ป้อนกลับมายังอินพุต ซึ่งในการใช้งานจะกำหนดให้ค่าความต้านทาน R มีค่าคงที่ จะได้แรงดันเอาต์พุตจะแปรตามกระแสอินพุตที่ป้อน ในวงจรนี้มีข้อจำกัดที่การจ่ายกระแสไปยังเอาต์พุตเกินขีดจำกัดเพราะจะทำให้ออปแอมป์เกิดการอิมิตัว

2.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล (Analog to Digital Converter :ADC)

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลเขียนย่อเป็น ADC เป็นการเปลี่ยนสัญญาณที่เป็นอนาลอกได้แก่สัญญาณที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง เช่นระดับของแรงดัน หรือปริมาณกระแสไฟฟ้าให้กลายเป็นสัญญาณดิจิทัลที่มีอยู่ในรูปเลขฐานสองคือ 0 และ 1 เป็นฟังก์ชันที่ไม่ขึ้นกับเวลา วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล แสดงดังรูปที่ 2.5



รูปที่ 2.5 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล MCP3208

จากรูปที่ 2.5 ประกอบด้วยไอซีเบอร์ ADC MCP3208 ตัวต้านทานและตัวเก็บประจุ ซึ่งไอซีเบอร์ ADC MCP3208 เป็น ADC ที่มีเอาต์พุตเป็น 12 บิต โดยรับสัญญาณอินพุตที่มีแรงดันอยู่ในช่วง 0-5.00 V

การทำงานของวงจร ADC เป็นดังนี้คือ ขา CS ต่อเข้ากับ MCS เบอร์ 89C51 ที่ P1.2 และ P1.3 วงจร ADC ก็จะทำงาน และเมื่อ ADC กำลังทำงานอยู่ แล้วเกิดขัดข้องหรือไม่สามารถทำงานได้ตามปกติ จะต้องทำการรีเซ็ตไอซีทุกครั้ง

หลักการคำนวณหาค่าสัญญาณเอาต์พุตแบบดิจิตอลขณะที่มีอินพุตเป็นสัญญาณอนาลอกใดๆ แสดงดังในตารางที่ 2.8

ตารางที่ 2.8 การคำนวณค่าสัญญาณเอาต์พุตแบบดิจิตอลเมื่ออินพุตเป็นสัญญาณอนาลอก

อินพุตแบบอนาลอก(โวลต์)	อินพุต $\times 819$	เอาต์พุตแบบดิจิตอล
0	0	00H
5.00(V_{ref})	4095	FFFH
4.00	3276	CCCH
3.00	2457	999H
2.00	1638	666H
1.00	819	333H

สูตรที่ใช้ในการคำนวณ นั้นคือ

$$\boxed{Digital\ Output\ (Decimal) = \frac{4095 \times V_{input}}{V_{ref}}} \quad (2.8)$$

เมื่อ Digital Output (Decimal) คือเอาต์พุตแบบดิจิตอลที่อยู่ในรูปของเลขฐานสิบ

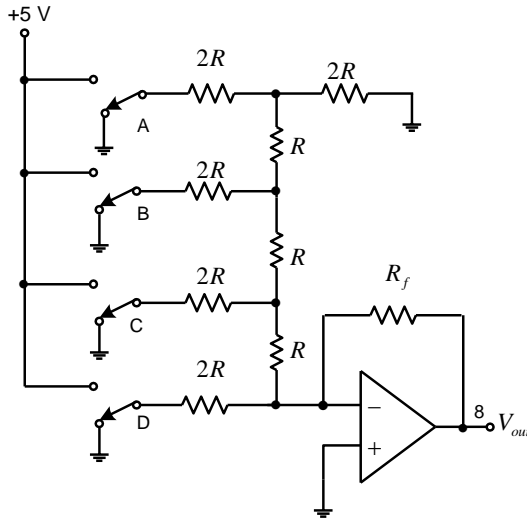
V_{input} คือแรงดันอินพุตที่เป็นสัญญาณอนาลอก

V_{ref} คือแรงดันอ้างอิงมีค่าเป็น 5.00 โวลต์

2.4 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก (Digital to Analog Converter : DAC)

การแปลงกลุ่มของสัญญาณดิจิตอลให้กลายเป็นปริมาณของแรงดันหรือกระแสนั้นจะใช้ วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกหรือเขียนย่อเป็น DAC วงจรพื้นฐานของ DAC จะใช้

ออปแอมป์ เป็นตัวรวบรวมสัญญาณ และ อาศัยเทคนิคใน การต่อวงจรที่ อินพุทของ ออปแอมป์ วงจรนี้เรียกว่าวงจร R-2R Ladder แสดงดังรูปที่ 2.6



รูปที่ 2.6 วงจร R-2R Ladder

จากวงจรในรูปที่ 2.6 เขียนสมการแรงดันเอาต์พุตได้เป็น

$$V_o = \frac{V_{ref}}{16} D \tag{2.9}$$

เมื่อ V_o คือแรงดันเอาต์พุตของวงจรมีหน่วยเป็นโวลต์ V_{ref} คือ แรงดันอ้างอิงที่มีค่า 5.00 โวลต์ และ D เลขฐานสิบที่แปลงมาจากสัญญาณดิจิทัล

ตัวอย่าง เมื่อมีอินพุทป้อนเข้าวงจร R-2R Ladder มีค่าเป็น 0101 โดยที่ $V_{ref} = 2.50$ โวลต์ คำนวณหาค่า V_o ได้เป็นดังนี้

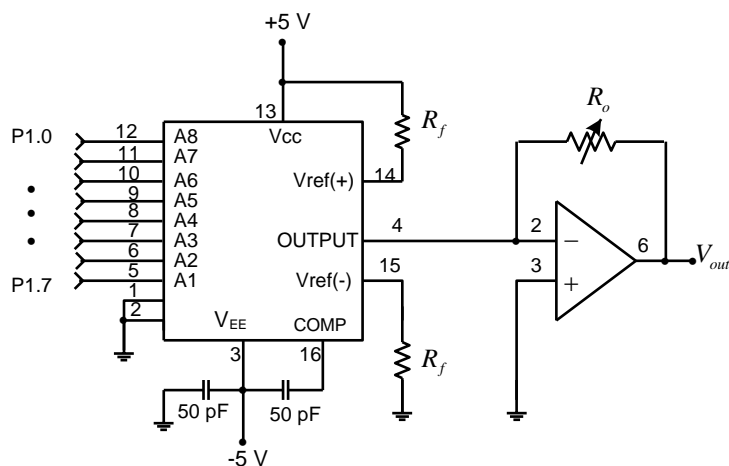
วิธีทำ
$$V_o = \frac{V_{ref}}{16} D$$

เมื่อ $V_{ref} = 2.50$ โวลต์

$$D = 0101 = 5_{10}$$

$$\text{จะได้ } V_o = \frac{2.50}{16} 5 = 0.78125 \text{ โวลต์}$$

วงจรแปลง DAC อีกแบบหนึ่งแสดงดังรูปที่ 2.7 ประกอบด้วย ไอซีเบอร์ MC 1408 และ ออปแอมป์ โดยที่ไอซีเบอร์ MC 1408 เป็นตัวแปลงสัญญาณดิจิทัล 8 บิต ให้เป็นกระแส แล้วใช้ออปแอมป์เบอร์ 358 แปลงกระแสที่ได้เป็นแรงดันที่มีค่าอยู่ในช่วง 0-2.50 โวลต์



รูปที่ 2.7 วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาลอก

จากวงจรในรูปที่ 2.7 คำนวณหาค่าแรงดันเอาต์พุตได้ดังนี้

$$V_o = \frac{V_{cc}}{R_f} \times R_o \times D_o \quad (2.10)$$

เมื่อ V_{cc} คือไฟเลี้ยงวงจรมีค่าเป็น 5 โวลต์

R_f คือตัวต้านทานที่ต่ออยู่ระหว่างขา 13 และ 14 ของไอซี MC 1408

R_o คือตัวต้านทานป้อนกลับของออปแอมป์ และ D_o คือค่าข้อมูล (Data) ขนาด 8 บิต

ตัวอย่าง เมื่อป้อนสัญญาณดิจิทัลให้กับวงจรตามรูปที่มีค่าเป็น 10101010

ค่าความต้านทาน $R_f = 4.7k\Omega$ $R_o = 2.35k\Omega$ $V_{cc} = 5$ โวลต์ จงคำนวณหาค่าแรงดันเอาต์พุต

วิธีทำ จากสมการของแรงดันเอาต์พุตตามสมการ (2.10) จะได้

$$\begin{aligned}
 V_o &= \frac{V_{cc}}{R_f} \times R_o \times \left(\frac{D7}{2} + \frac{D6}{4} + \frac{D5}{8} + \frac{D4}{16} + \frac{D3}{32} + \frac{D2}{64} + \frac{D1}{128} + \frac{D0}{256} \right) \\
 &= \frac{5}{4.7} \times 2.35 \times \left(\frac{1}{2} + \frac{0}{4} + \frac{1}{8} + \frac{0}{16} + \frac{1}{32} + \frac{0}{64} + \frac{1}{128} + \frac{0}{256} \right) \\
 &= 2.55 \times 0.664 \\
 &= 1.6932
 \end{aligned}$$

หลักการคำนวณหาค่าสัญญาณเอาต์พุตแบบอนาลอกขณะที่มีสัญญาณอินพุตเป็นดิจิทัลใดๆ แสดงดังตารางที่ 2.9

ตารางที่ 2.9 การคำนวณค่าสัญญาณเอาต์พุตแบบอนาลอกเมื่ออินพุตเป็นสัญญาณดิจิทัล

อินพุตแบบดิจิทัล	อินพุตที่เป็นเลขฐาน 10	เอาต์พุตแบบอนาลอก(โวลต์)
00H	0	0
FFH	255	$2.55(V_{ref})$
0AH	10	0.1
33H	51	0.51
66H	102	1.02
99H	153	1.53
CCH	204	2.04

สูตรที่ใช้ในการคำนวณ

$$V_{out} = \frac{V_{ref} \times \text{Decimal Input}}{255} \quad (2.11)$$

เมื่อ Decimal Input คืออินพุตที่เป็นเลขฐานสิบ ถ้าแรงดันอ้างอิง $V_{ref} = 2.55$ โวลต์ แต่ละ Step ของสัญญาณอินพุตจะเท่ากับ 0.01 โวลต์