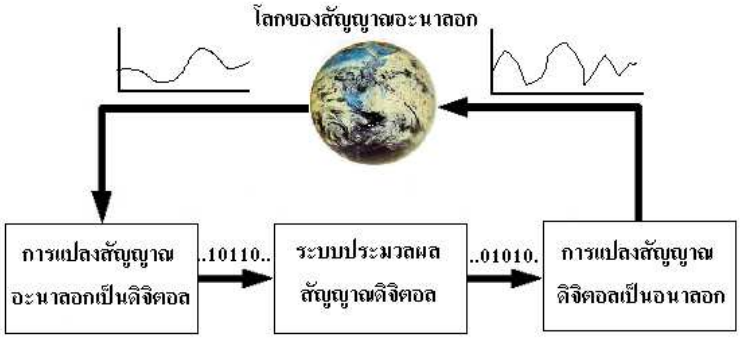
14. การเชื่อมต่อกับสัญญาณอนาลอก

14.1 บทนำ

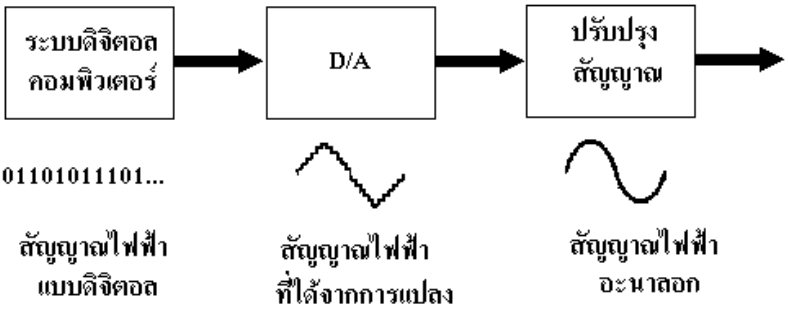


รูปที่ 14.1 แสดงขบวนการแปลงสัญญาณระหว่าอะนาลอกกับดิจิตอล

โดยทั่วๆไปสัญญาณที่ปรากฏอยู่ในทางธรรมชาติจะเป็นสัญญาณที่มีความต่อเนื่อง และมีขนาดได้หลายขนาดไม่จำกัด สัญญาณลักษณะนี้เรียกว่า “สัญญาณอนาลอก” ถ้าต้องการนำสัญญาณอะนาลอกนี้มาประมวลผลด้วยระบบดิจิตอล จะต้องมีการเปลี่ยนสัญญาณนี้ให้เป็นสัญญาณดิจิตอลเสียก่อน ระบบนี้เรียกว่า การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (Analog to Digital conversion หรือ A/D) และในทางตรงข้ามกัน เมื่อประมวลผลได้แล้วสัญญาณที่ได้ยังเป็นสัญญาณดิจิตอลอยู่ ถ้าต้องการนำออกสู่โลกภายนอกแบบสัญญาณอะนาลอก ก็ต้องทำการแปลงสัญญาณให้เป็นสัญญาณอะนาลอกด้วยระบบแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก (Digital to Analog Conversion หรือ D/A)

14.2 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก (Digital-to-Analog Converter หรือ D/A)

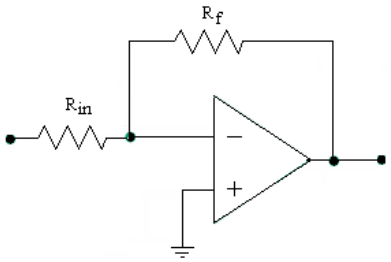
หมายถึงวงจรที่เปลี่ยนข้อมูลทางดิจิตอลให้เป็นค่าอนาลอก ค่าอนาลอกนี้อาจเป็นกระแสไฟฟ้าหรือแรงดันไฟฟ้าก็ได้



รูปที่ 14.2 แสดงขบวนการแปลงสัญญาณจากดิจิตอลเป็นอะนาลอก

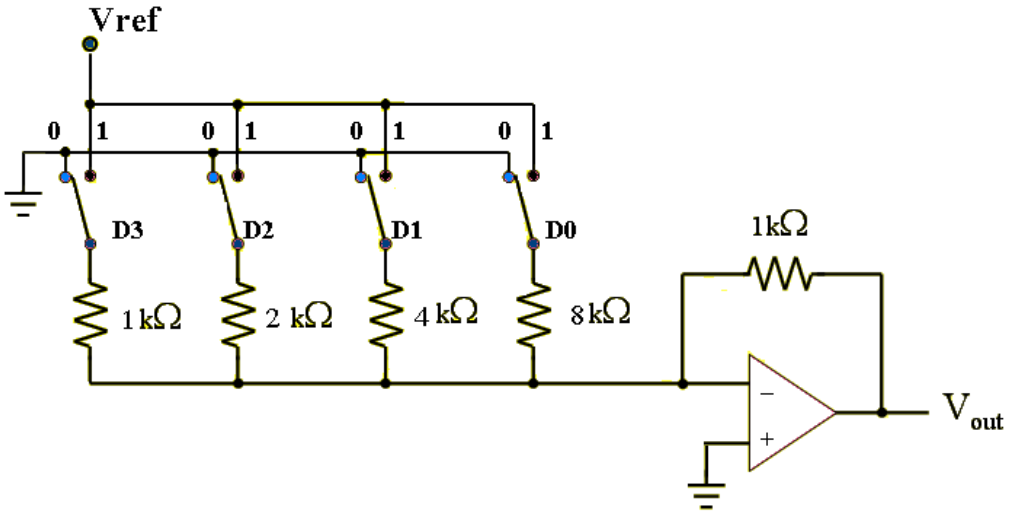
เทคนิคการแปลงสัญญาณดิจิตอลเป็นสัญญาณอะนาลอกมีหลายวิธีเช่น Binary Weighted DAC และ R/2R Ladder การทำงานของแต่ละวิธีมีรายละเอียดดังนี้

14.2.1 Binary Weighted DAC



รูปที่ 14.3 แสดงวงจรออปแอมป์แบบกลับสัญญาณ (Inverting Amp.)

คุณสมบัติของวงจรประเภทนี้ได้แก่ ความต้านทานอินพุทสูง อัตราขยายแรงดันมีค่าเท่ากับ $-R_f/R_{in}$



รูปที่ 14.4 วงจร Binary Weighted DAC

ตามรูปที่ 14.4 จะเห็นได้ว่าแรงดันเอาท์พุทต่ำสุดเกิดเมื่อสวิตซ์ D3- D0 อยู่ตำแหน่ง ‘0’

Vout = 0 โวลท์

ถ้า D3- D0 อยู่ตำแหน่ง ‘1’ จะได้แรงดันเอาท์พุทสูงสุด

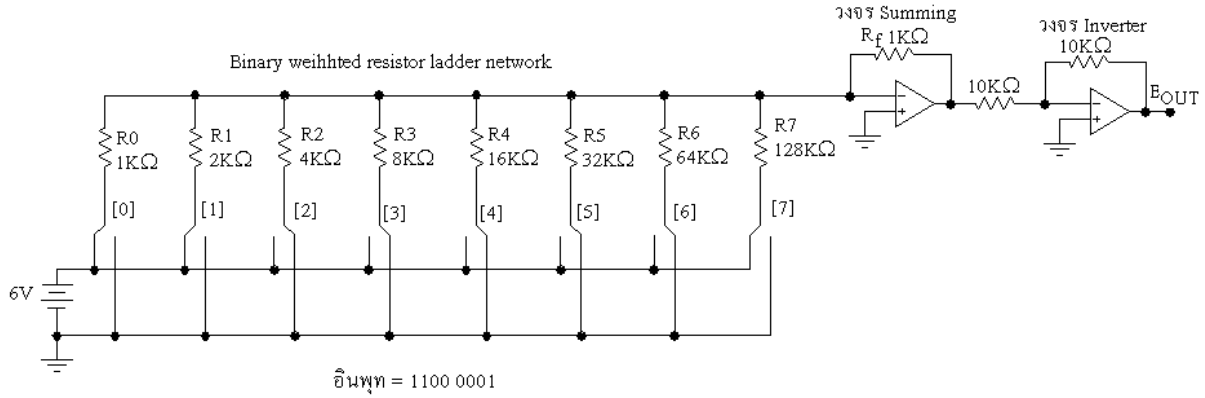
Vout = - Vref x Rf x(1/1k + 1/2k + 1/4k + 1/8k)

Vout = -Vref x 1 x(1/1 + 1/2 + 1/4 + 1/8)

Vout = -Vref x (1 + .5 + .25 + .125)

Vout = -1.875 x Vref

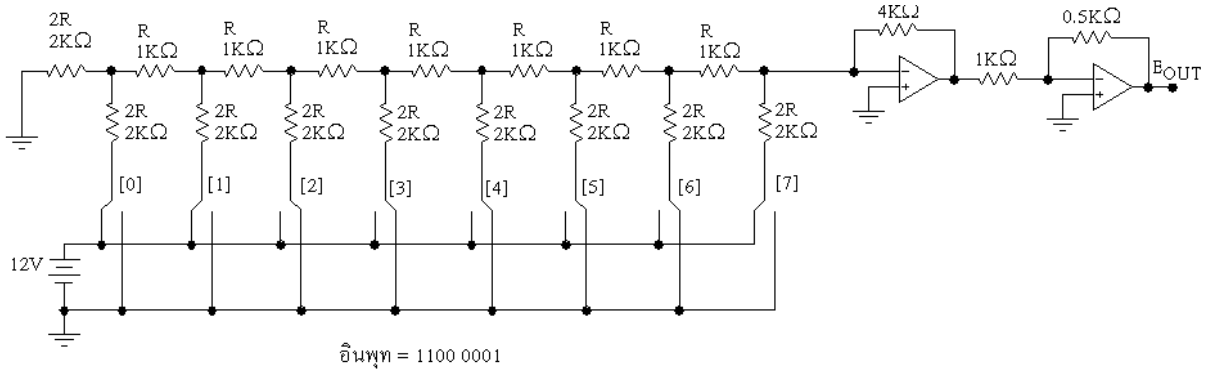
เช่นถ้า Vref = +5V Vout เมื่ออินพุทเป็น ‘1’ หมดจะได้ -9.375 โวลท์ แต่การทำวงจรให้มีจำนวนบิตมากขึ้นต้องใช้ความต้านทานขนาดใหญ่ขึ้น ตัวอย่าง จงคำนวณหาค่า Eout



รูปที่ 14.5 ตัวอย่างวงจร DAC ขนาด 8 บิต

14.2.2 R/2R Ladder

เป็นวิธีที่ใช้กันทั่วๆไปในการผลิตเป็นวงจรรวม เพราะว่า ใช้ค่าความต้านทานต่ำและใช้เพียง 2 ค่าคือ 1R กับ 2R เช่น 1 K กับ 2K ตัวอย่างวงจรมีลักษณะตามรูปที่ 14.6 ซึ่งสามารถวิเคราะห์วงจรโดยใช้ Thevenin’s theorem



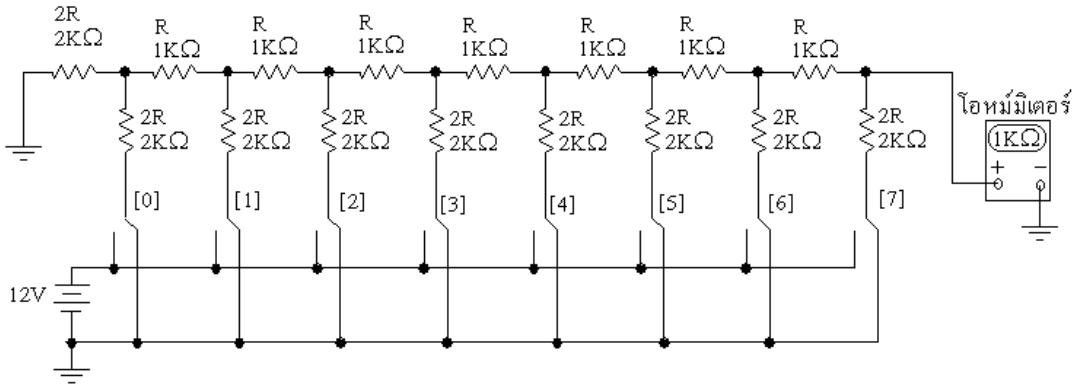
รูปที่ 14.6 ตัวอย่างวงจร DAC แบบ R/2R Ladder ขนาด 8 บิต

การวิเคราะห์วงจรโดยใช้ทฤษฎี Thevenin

หา Rth ของ Thevenin

ที่ตำแหน่งบิตต่างๆ สามารถหา Rth ได้ดังนี้

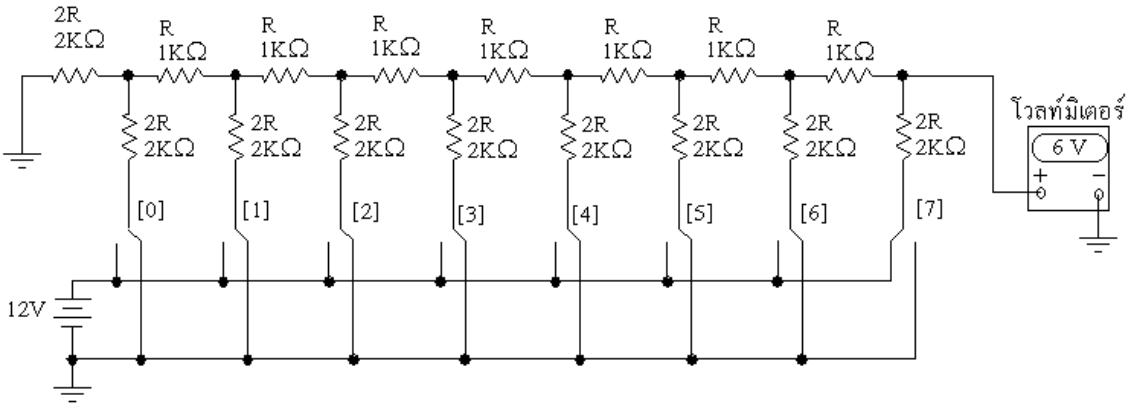
บิต 0 Rth = 1 k, บิต 1 Rth = 1 k,บิต 2 Rth = 1 k,บิต 3 Rth = 1 k,บิต 4 Rth = 1 k,บิต 5 Rth = 1 k, บิต 6 Rth = 1 k และ บิต 7 Rth = 1 k



รูปที่ 14.7 วงจร Thevenin equivalent เพื่อหา Rth

หา Vth ของ Thevenin

ที่ตำแหน่งบิตต่างๆ สามารถหา Vth ได้ดังนี้



รูปที่ 14.8 วงจร Thevenin equivalent เพื่อหา Vth เมื่อบิต 7 เป็น '1'

บิต 0 Vth = 46.875 mV, บิต 1 Vth = 93.75 mV, บิต 2 Vth = 0.1875 V, บิต 3 Vth = 0.375 V,
บิต 4 Vth = 0.75 V, บิต 5 Vth = 1.5 V, บิต 6 Vth = 3V และ บิต 7 Vth = 6V

14.2.3 เทอมที่เกี่ยวกับ DAC

Resolution

เป็นขนาดที่เล็กที่สุดที่ D/A สามารถสร้างได้ ขนาดนี้ขึ้นอยู่กับจำนวนบิตของ D/A หรือคำนวณได้จาก

$$V_{res} = \left(\frac{V_{REF}}{2^n} \right)$$

จำนวนระดับของสัญญาณอะนาลอก

$$\text{Analog Levels} = 2^n$$

ค่าความเที่ยงตรง(Accuracy)

เป็นการเปรียบเทียบค่าจริงกับค่าที่คำนวณได้ ปกติจะกำหนดเป็นเปอร์เซนต์ของค่าเอาท์พุทเมื่อเต็มสเกล (Full scale)

ความเร็ว (Speed)

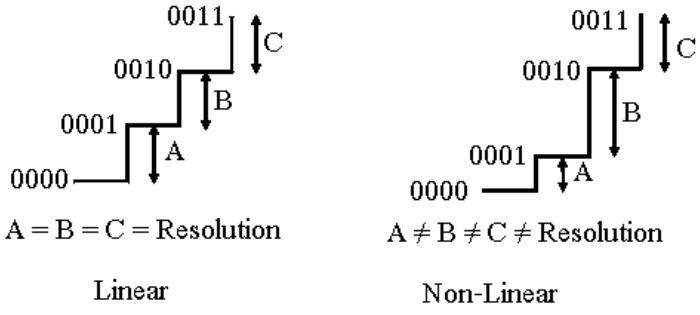
หมายถึงความเร็วของ Output settling time ซึ่งเป็นเวลาที่ต้องใช้เพื่อสร้างค่าเอาท์พุทเมื่ออินพุทมีการเปลี่ยนแปลง

การผิดพลาดใน DAC

DAC ที่ดีต้องให้ค่าเอาท์พุทออกมาตรงตามค่าอินพุท เมื่ออินพุทให้ค่ามากขึ้น สัญญาณเอาท์พุทก็ต้องมีค่ามากขึ้น และการเพิ่มขึ้นของอินพุทแต่ละขั้นก็ต้องให้ค่าเอาท์พุทเพิ่มขึ้นในแต่ละขั้นเท่าๆกันด้วย แต่ถึงกระนั้น ในทางเป็นจริง DAC ก็มีการผิดเพี้ยนเช่นกัน การผิดเพี้ยนใน DAC มี 2 แบบใหญ่ๆ คือ Non-linear distortion และ Non-monotonic distortion

Non-Linear Distortion:

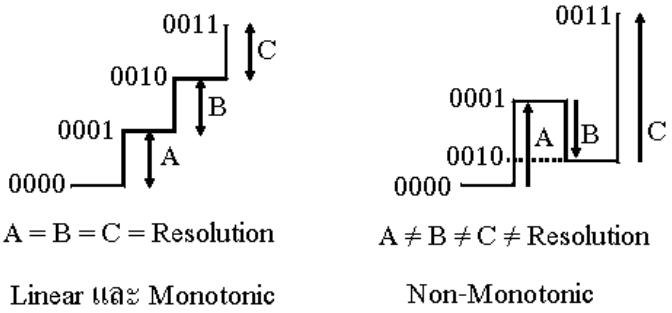
ลักษณะการผิดเพี้ยนแบบนี้จะให้ค่าเอาท์พุทของแต่ละสเต็ปไม่เท่ากัน



รูปที่ 14.9 การผิดเพี้ยนแบบ Non-linear

Non-Monotonic Distortion

ลักษณะการผิดเพี้ยนแบบนี้จะให้ค่าเอาท์พุทของแต่ละสเต็ปไม่เท่ากัน และบางครั้งให้ค่าเอาท์พุทลดลงทั้งๆที่อินพุทมีค่าเพิ่มขึ้น



รูปที่ 14.10 การผิดพลาดแบบ Monotonic

ตัวอย่าง DAC ขนาด 2 บิต มีแรงดันอ้างอิง 8 โวลท์และค่าความเที่ยงตรง @ ± 0.2% จงหาค่า resolution และค่าความเที่ยงตรงในเทอมของแรงดัน

Resolution = $1/2^2 = 1/4 = 25\%$ หรือ Resolution = $(1/4)(8V) = 2V$

Accuracy = $(\pm 0.2\%)(8V) = \pm 16mV$

สรุปการคำนวณ

$$V_{OUT} = V_{ref}^{-} + X_{10}\left(\frac{V_{ref}^{+} - V_{ref}^{-}}{2^n}\right)$$

$$V_{OUT} = V_{\min} + X_{10}\left(\frac{V_{\max} - V_{\min}}{2^n - 1}\right)$$

$$\text{Re}solution = \left(\frac{V_{ref}^{+} - V_{ref}^{-}}{2^n}\right)$$

$$\text{Re}solution = \left(\frac{V_{\max} - V_{\min}}{2^n - 1}\right)$$

$$V_{OUT} = V_{ref}^{-} + X_{10}\text{Re}solution$$

$$V_{OUT} = V_{\min} + X_{10}\text{Re}solution$$

V_{OUT} = Output Voltage

V_{ref}^{-} = Voltage Reference -

V_{ref}^{+} = Voltage Reference +

$V_{\min}$ = Minimum output voltage

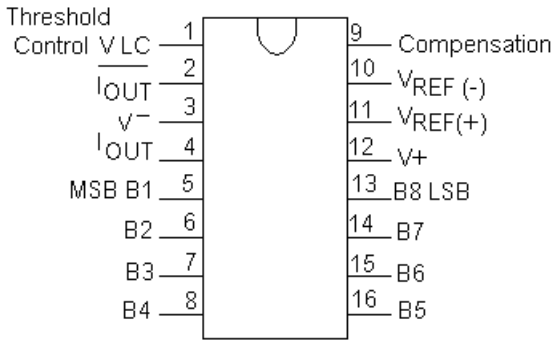
$V_{\max}$ = Maximum output voltage

X_{10} = Input value base 10

n = จำนวนบิต

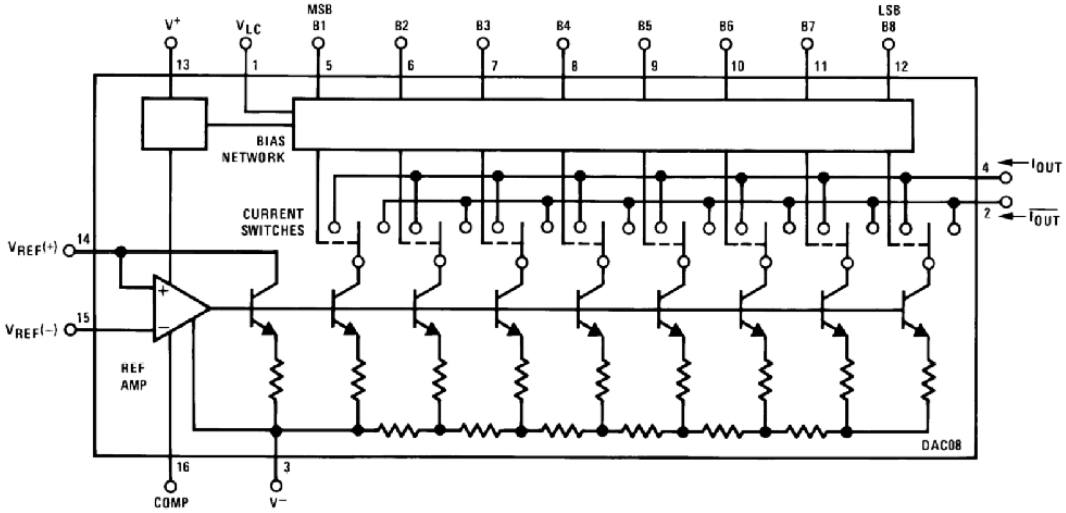
14.2.4 DAC0800

DAC 0800 เป็นวงจรร่วมที่ทำหน้าที่เป็น DAC มีอินพุท 8 บิต ให้เอาท์พุทเป็นกระแส ตำแหน่งขาสัญญาณเป็นตามรูป 4.11 และ บล็อกไดอะแกรมแสดงอยู่ในรูปที่ 14.12



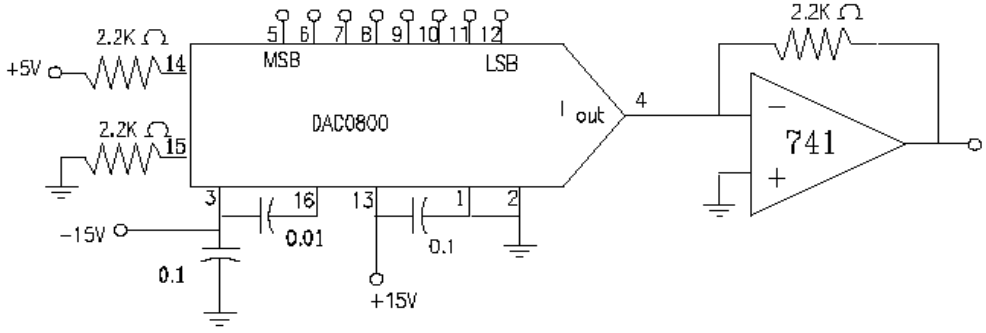
รูปที่ 14.11 ลักษณะตัวถังและขาสัญญาณของ DAC0800

ขาสัญญาณ	ชื่อ	ความหมาย
1	GND	Ground
2	IOUT’	Output
4.	IOUT	Output’
5	B1 → MSB	Input
	: : :	
12	B8 → LSB	Input
14	VREF(+)	Reference voltage for output
15	VREF(-)	Reference voltage for output



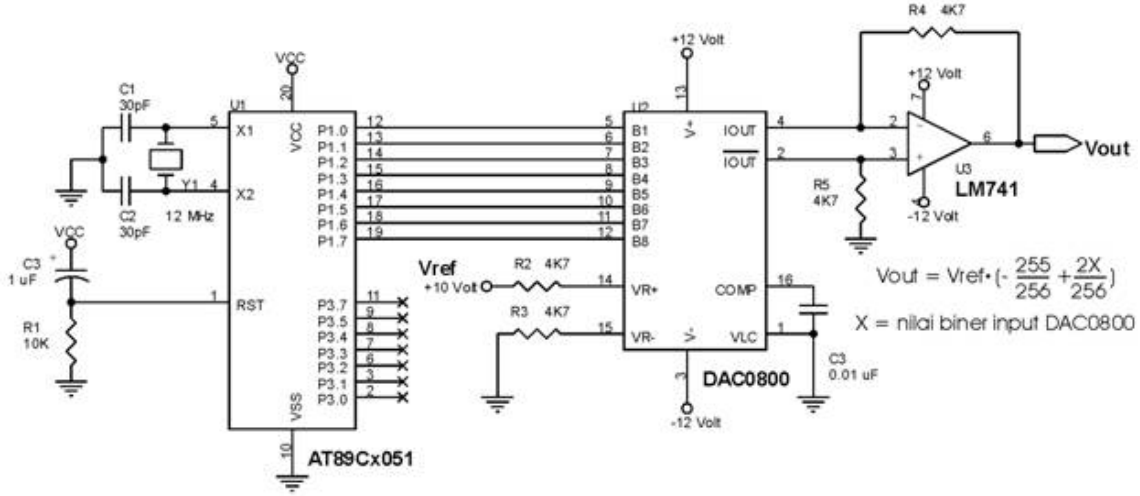
รูปที่ 14.12 ไดอะแกรมของ DAC0800

ตัวอย่างการต่อวงจรเพื่อให้เอาท์พุทเป็นแรงดัน



รูปที่ 14.13 การใช้งาน DAC0800

ตัวอย่างการต่อ AT89C2051 กับ DAC0800

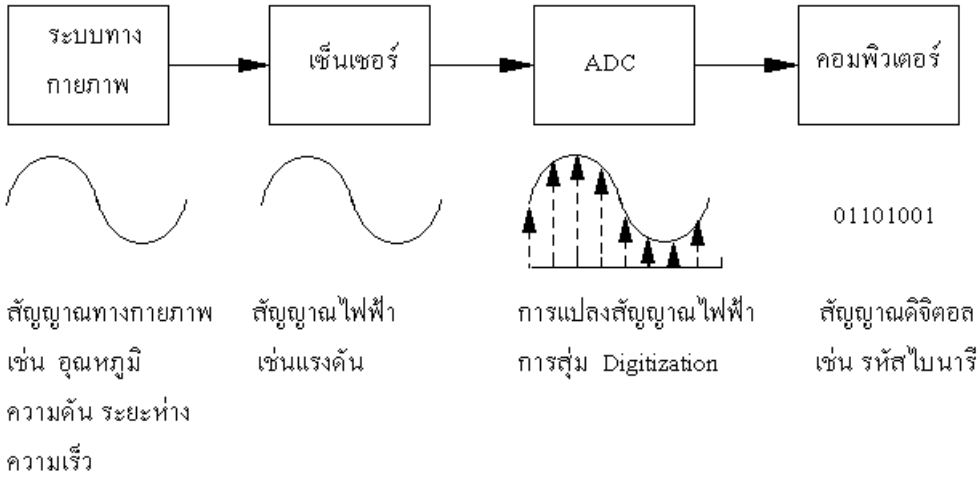


รูปที่ 14.14 การใช้งาน DAC0800 กับ AT89C2051

14.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (Analog-to-Digital Converter หรือ A/D หรือ A2D)

สัญญาณในทางธรรมชาติทั่วๆไปเป็นสัญญาณที่มีความต่อเนื่องมีขนาดได้ไม่จำกัด การที่จะนำสัญญาณเหล่านี้มาประมวลผลด้วยระบบดิจิตอลคอมพิวเตอร์โดยตรงนั้นย่อมไม่สามารถทำได้ อีกประการหนึ่งสัญญาณเหล่านี้บางครั้งก็ไม่ได้อยู่ในรูปของสัญญาณไฟฟ้า ดังนั้นการนำมาประมวลผลด้วยระบบดิจิตอลต้องมีกระบวนการแปลงเสียก่อน ตามรูปที่ 14.15 แสดงถึงขั้นตอนการนำสัญญาณในทางธรรมชาติมาใช้งานในระบบดิจิตอล เริ่มต้นสัญญาณในทางธรรมชาติเช่นอุณหภูมิซึ่งเป็นพลังงานความร้อนต้องถูกแปลงให้เป็นพลังงานไฟฟ้าเสียก่อนด้วยด้วยอุปกรณ์ที่เรียกว่าเซ็นเซอร์ หรือทรานสดิวเซอร์ สัญญาณเอาท์พุทที่ได้จากเซ็นเซอร์นี้ยังคงเป็นสัญญาณที่มีความต่อเนื่อง หรือที่เรียกกันทั่วๆไปว่าสัญญาณอนาลอก

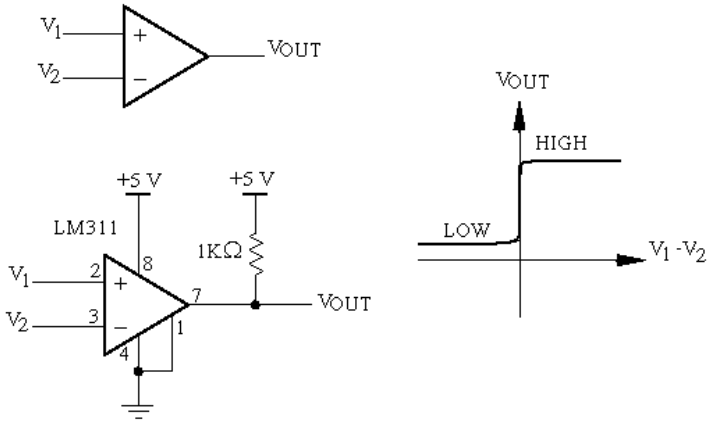
สัญญาณนี้ต้องถูกแปลงให้เป็นสัญญาณดิจิตอลด้วยวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล หรือที่เรียกว่า ADC หลังจากนี้จึงได้เป็นสัญญาไฟฟ้าที่อยู่ในรูปของสัญญาณดิจิตอลที่พร้อมจะนำไปประมวลผลในระบบดิจิตอลได้



รูปที่ 14.15 ลักษณะการใช้งาน ADC

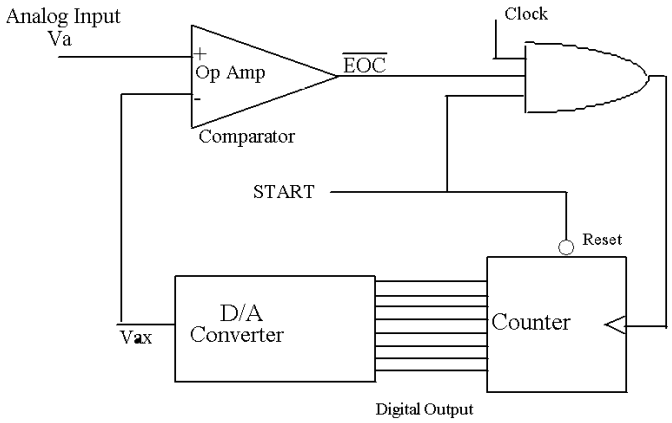
การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลนั้นมีอยู่หลายวิธี เช่นตัวอย่างวงจรแปลสัญญาณอนาลอกเป็นสัญญาณดิจิตอลแบบง่ายๆ โดยการใช้วงจรAnalog comparator ตามรูปที่ 14.16 ซึ่งให้เอาท์พุทออกมาขนาด 1 บิต นอกจากนี้ยังมีวิธีอื่นๆอีกเช่น

- Digital Ramp ADC (Counter method)
- Successive Approximation
- Parallel Comparator (“Flash”)
- Dual Slope



รูปที่ 14.16 ลักษณะของ ADC ขนาด 1 บิต

14.3.1 Digital Ramp ADC (Counter method)

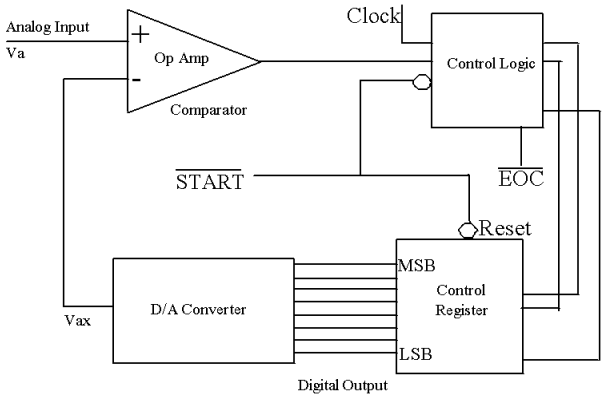


รูปที่ 14.17 ลักษณะของ ADC แบบ Counter methgod

บล็อกไดอะแกรมของวิธีนี้แสดงอยู่ในรูปที่ 14.17 โดยมีการทำงานดังนี้

1) เมื่อเริ่มสั่งให้แปลงสัญญาณ START = 0 จะเป็นการรีเซ็ตวงจรนับ (Counter) ให้ได้เป็น 0 ทุก บิต หลังจากนี้ START จะต้องเป็น 1 เพื่อเริ่มขบวนการทำงาน
2) ค่าจากวงจรนับจะถูกแปลงให้เป็นค่าอนาลอก Vax โดย DAC
3) ค่า Vax จะถูกนำไปเปรียบเทียบกับ ค่าแรงดันอินพุท Va โดยตัวเปรียบเทียบ (Comparator)
   - ถ้า Va > Vax เอาท์พุท $\overline{EOC}$ = 1 ทำให้ผลของการ AND ระหว่าง START $\overline{EOC}$ และ Clock ได้เป็นสัญญาณ Clock ป้อนเข้าวงจรนับ วงจรนับก็จะนับค่าขึ้นไปเรื่อยๆ ขบวนการก็จะเป็นไปตาม ขั้นที่ 2 และ 3 จนกว่า Va น้อยกว่าหรือเท่ากับ Vax
   - ถ้า Va < หรือ = Vax เอาท์พุท $\overline{EOC}$ = 0 วงจรนับจะหยุดนับ เป็นการเสร็จสิ้นการทำงาน

14.3.2 Successive Approximation



รูปที่ 14.18 ลักษณะของ ADC แบบ Successive Approximation

บล็อกไดอะแกรมของวิธีนี้แสดงอยู่ในรูปที่ 14.18 โดยมีการทำงานดังนี้

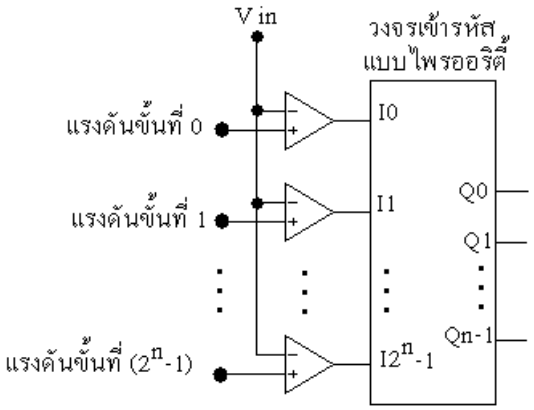
1) เมื่อเริ่มสั่งให้แปลงสัญญาณ START = 0 จะเป็นการรีเซ็ต Control logic และ Control Register ให้เป็น 0 ทุกบิต พร้อมทั้งกำหนดตำแหน่งบิตที่ต้องเซ็ตหรือรีเซ็ตเป็นบิตที่ 7 หลังจากนี้ START จะต้องเป็น 1 เพื่อเริ่มขบวนการทำงาน
2) ค่าจากวงจรนับจะถูกแปลงให้เป็นค่าอนาลอก Vax โดย DAC
3) ค่า Vax จะถูกนำไปเปรียบเทียบกับ ค่าแรงดันอินพุท Va โดยตัวเปรียบเทียบ (Comparator)
   - ถ้า Va > Vax เอาท์พุทของตัวเปรียบเทียบจะเป็น 1 และ $\overline{EOC}$ = 1 เมื่อสัญญาณ Clock เข้ามา Control logic จะเซ็ตบิตที่ระบุให้เป็น 1 พร้อมทั้งเลื่อนการระบุบิตลงมา 1 ตำแหน่ง แล้วกลับไปเปรียบเทียบใหม่
   - ถ้า Va < Vax เอาท์พุท $\overline{EOC}$ = 1 เมื่อสัญญาณ Clock เข้ามา Control logic จะรีเซ็ตบิตที่ระบุให้เป็น 0 พร้อมทั้งเลื่อนการระบุบิตลงมา 1 ตำแหน่ง แล้วกลับไปเปรียบเทียบใหม่
   - การทำงานจะวนอยู่ในข้อ 3 นี้จนกว่าจะทำครบทุกบิต (ในที่นี้มี 8 บิต) หรือ Va = Vax เอาท์พุท $\overline{EOC}$ จะเท่ากับ 0 วงจรจะหยุดทำงาน เป็นการเสร็จสิ้นการทำงาน

วิธีการแปลงแบบนี้ให้ความเร็วมากกว่าแบบ การนับ อีกทั้งราคาก็ถูกจึงเป็นวิธีที่นิยมใช้กันมาก

14.3.3 Parallel Comparator (“Flash") ADC

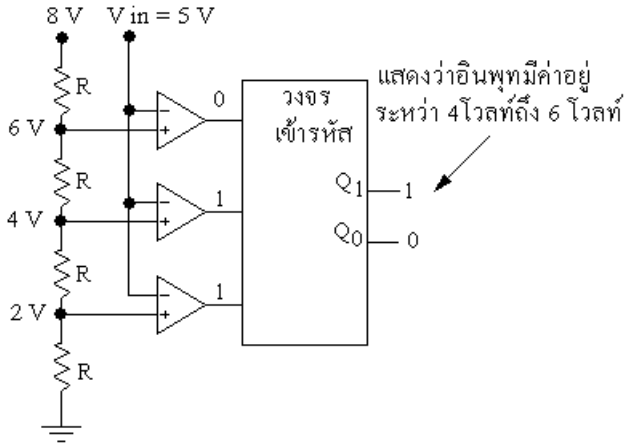
เป็นวิธีที่เร็วที่สุด ใช้ในสโคปแบบดิจิตอล สัญญาณวิดีโอ และการใช้งานที่ต้องการความเร็วสูง แต่ก็มีข้อเสียในด้านราคา เพราะว่าใช้ฮาร์ดแวร์ในการสร้างมาก ดังนั้นจึงเหมาะกับการใช้งานที่ต้องการจำนวนบิตน้อยๆ

ลักษณะของวงจรประกอบด้วยตัวเปรียบเทียบและวงจรเข้ารหัส จำนวนของตัวเปรียบเทียบเช่นถ้า



รูปที่ 14.19 ลักษณะของ ADC แบบ Flash

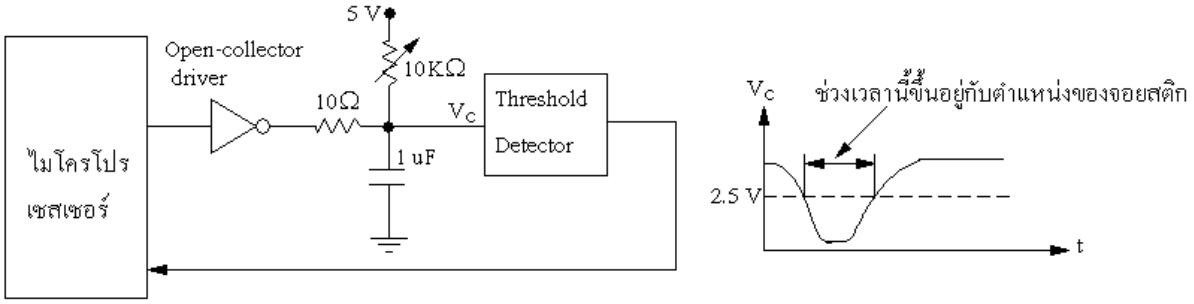
ต้องการ ADC ขนาด 2 บิตมีจำนวน 4 สเต็ป ซึ่งมีแรงดันอ้างอิง 8 โวลท์ ดังนั้นแต่ละสเต็ป 0 โวลท์ 2 โวลท์ 4 โวลท์ และ 6 โวลท์ ดังนั้นต้องใช้ตัวเปรียบเทียบ 3 ตัว ตามรูปที่ 14.20 ดังนั้นเมื่อป้อนแรงดันอินพุท เช่น 5 V เข้ามา ตัวเปรียบเทียบทั้ง 3 ตัวจะทำการเปรียบเทียบกับแรงดันอ้างอิงของแต่ละระดับได้เอาท์พุทเป็น 0 1 1 วงจรเข้ารหัสจึงให้เอาท์พุทเป็นรหัสไบนารี 10 ตามรูปที่ 14.20



รูปที่ 14.20 ตัวอย่าง ADC แบบ Flash ขาด 2 บิต

14.3.4 ADC แบบอื่นๆ

นอกจากแบบที่ได้กล่าวมาแล้ว ยังมีADC แบบอื่นๆอีกเช่น ADC ที่ใช้ใน จอยสติก (Joystick) ของคคอมพิวเตอร์ Apple II โดยมีรูปแบบตามรูปที่ 14.21 สำหรับ Threshold Detector อาจใช้ Op-amp ที่ใช้แรงดันอ้างอิงที่ขา inverting เป็น 2.5 โวลท์ หรือใช้ 74LS14 หรือ 74HCT14 ก็ได้ ส่วนตำแหน่งของจอยสติกกำหนดด้วย Potentiometer ซึ่งเป็นความต้านทานที่ปรับค่าได้ หลักการทำงานจะใช้การวัดค่า RC time constant ที่เกิดจากการประจุให้กับ C 1 μF ทั้งนี้การชาร์ตและดิสชาร์ต ประจุนี้ สั่งงานด้วยโปรแกรมของไมโครโปรเซสเซอร์



รูปที่ 14.21 ลักษณะของ ADC แบบ Apple II A-to-D for joystick

14.3.5 ข้อกำหนดของ ADC

Resolution

เหมือนกับใน DAC จะเป็นตัวกำหนดค่าความผิดพลาดของการ quantizing ซึ่งเป็นการผิดพลาดจากค่าที่แท้จริง

Accuracy

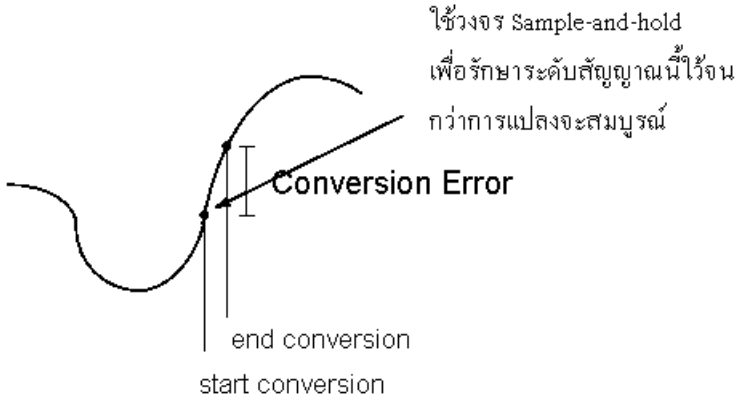
เหมือนกับใน DAC เป็นการแสดงค่าที่ถูกต้องเมื่อแทียบกันระหว่างค่าที่แท้จริงกับค่าที่กำหนดได้

Conversion Time

ระยะเวลาที่ใช้ในการแปลงสัญญาณ

Sample and Hold

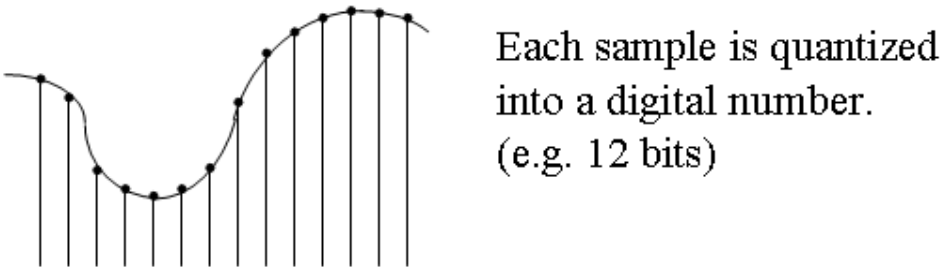
เมื่อสัญญาณมีความถี่สูงถ้าไม่ต้องการให้เกิดความผิดพลาดเนื่องจากระยะเวลาที่ใช้แปลง ควรใช้ Sample and Hold



รูปที่ 14.22 การใช้ Sample and Hold

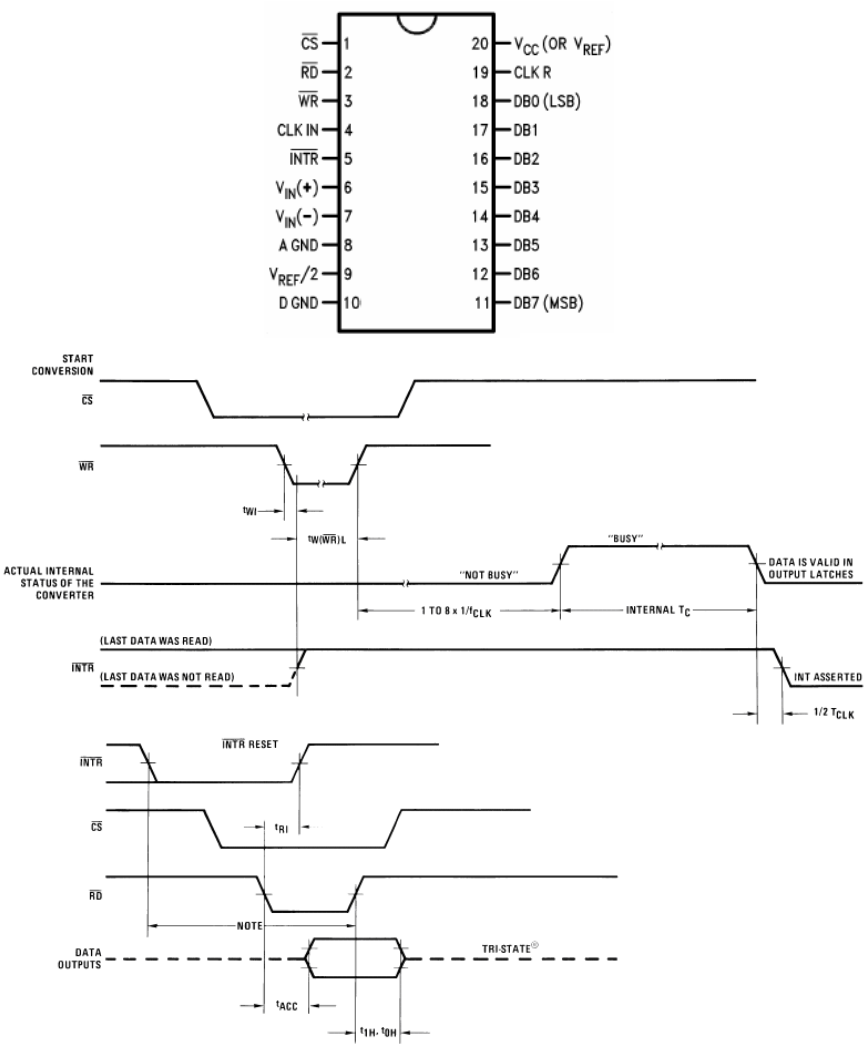
Sampling Rate

ความถี่ของการแปลงสัญญาณ เช่นถ้าเป็น 44 kHz ก็หมายถึงว่าการแปลงสัญญาณจะเกิดขึ้นทุกๆ 22.7 μS



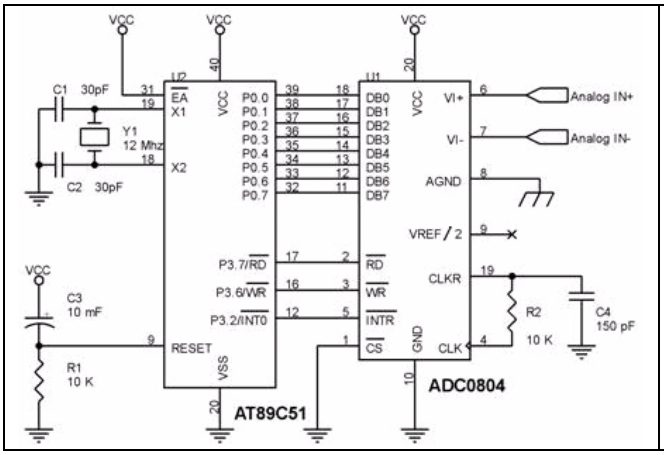
รูปที่ 14.23 ลักษณะของการสุ่มสัญญาณ

14.3.6 ADC0804 8-Bit µP Compatible A/D Converters



รูปที่ 14.24 ลักษณะของ ADC 0804 และไดอะแกรมเวลาของการแปลงสัญญาณ

ตัวอย่างการใช้งาน ADC0804 กับ AT89C51

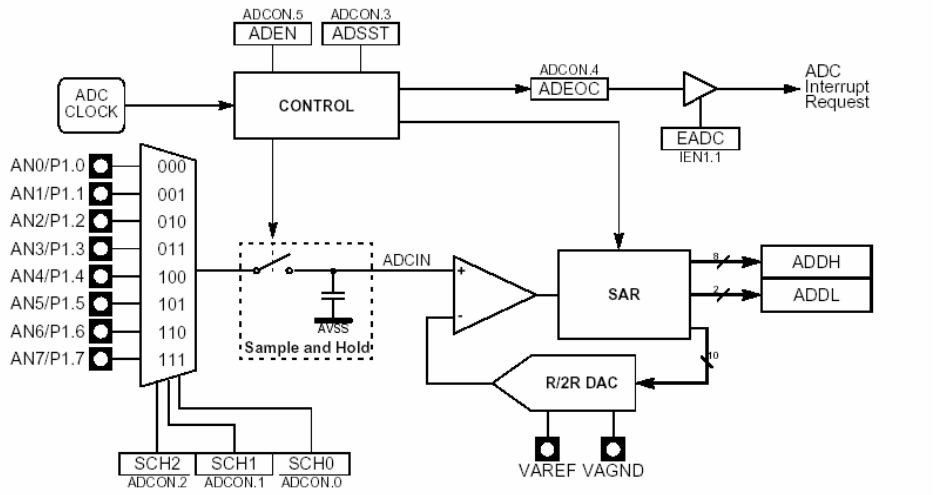



```
ReadADC0804:
        MOVX    @R0,A
        JB      INT0,$
        MOVX    A,@R0
        RET
```

รูปที่ 14.25 การใช้งาน ADC0804 กับ AT89C51

14.3.7 ADC ขนาด 10 บิตของ T89C51AC2

T89C51AC2 มี ADC ขนาด 10 บิตอยู่ภายในมีคุณสมบัติเด่นๆดังนี้



รูปที่ 14.26 บล็อกไดอะแกรม ADC ของ T89C51AC2

- รองรับอินพุทได้ 8 ช่องแบบมัลติเพลกซ์
- เป็น ADC ขนาด 10 บิต
- เวลาในการแปลง 20 μS
- ใช้แรงดันอ้างอิง +2.4 ถึง +3 โวลท์
- รับสัญญาณอินพุทได้ตั้งแต่ 0 ถึง 3 โวลท์
- สามารถอินเตอร์รัพท์ซีพียูได้
- กำหนดความถี่ของสัญญาณนาฬิกาได้
- Integral non-linearity ทั่วๆไป 1 LSB และสูงสุดไม่เกิน 2 LSB
- Differential non-linearity ทั่วๆไป 0.5 LSB และสูงสุดไม่เกิน 1 LSB

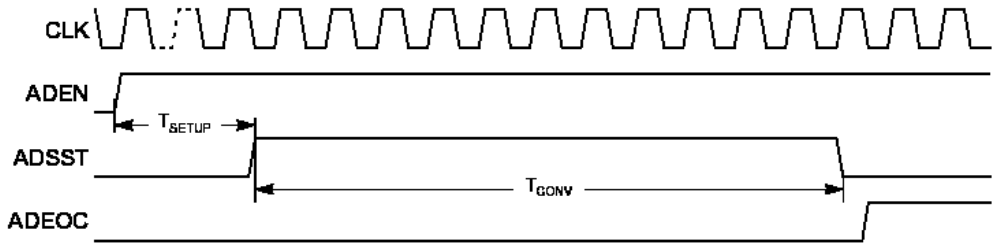
ข้อกำหนดในการใช้งานมีดังนี้

ADC Port1 I/O Functions

ADC นี้ใช้ขาพอร์ท 1 ร่วมกับ I/O โดยกำหนดที่รีจิสเตอร์ ADCF ถ้าให้เป็น 1 แสดงว่าให้ทำหน้าที่เป็น สัญญาณอินพุทของ ADC ถ้าให้เป็น 0 แสดงว่าให้ทำหน้าที่เป็นพอร์ท I/O

ระยะเวลาในการแปลง

ใช้เวลาในการแปลงสัญญาณ 11 สัญญาณนาฬิกา และเวลา setup ไม่น้อยกว่า 4 μS มีไดอะแกรมเวลาการแปลงตามรูปที่ 14.27

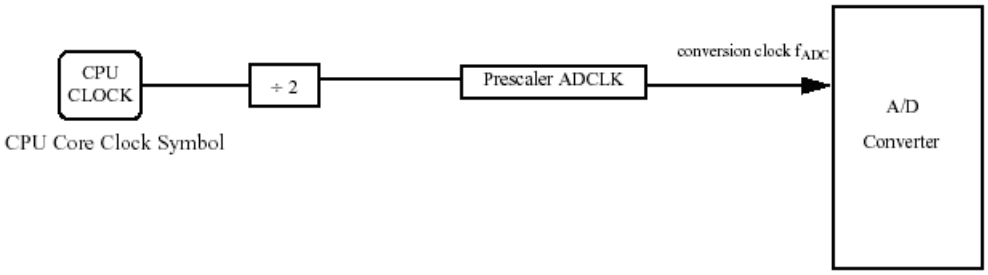


T_{SETUP} น้อยสุด = 4 uS

T_{CONV} = 11 สัญญาณนาฬิกาของ ADC = 1 sample and hold + 10 bit conversion

รูปที่ 14.27 ไดอะแกรมการทำงานของ ADC ของ T89C51AC2

A/D Converter clock



รูปที่ 14.28 ADC Converter clock

ADC Standby Mode

ถ้าไม่ได้ใช้ ADC สามารถให้ ADC อยู่ใน standby mode ได้โดยให้บิต ADEN ในรีจิสเตอร์ ADCON เป็น 0 ในโหมดนี้ กำลังงานสูญเสียประมาณ 1 ไมโครวัตต์

รีจิสเตอร์ที่เกี่ยวข้อง

ADCF (ADC Configuration Register) ใช้กำหนดหน้าที่ของพอร์ท P1

ADCF (S:F6h)
ADC Configuration

7	6	5	4	3	2	1	0
CH 7	CH 6	CH 5	CH 4	CH 3	CH 2	CH 1	CH 0

Bit Number	Bit Mnemonic	Description
7-0	CH 0:7	**Channel Configuration** Set to use P1.x as ADC input. Clear tu use P1.x as standart I/O port.

Reset Value=0000 0000b

รูปที่ 14.29 หน้าที่ของบิตแต่ละบิตของรีจิสเตอร์ ADCF

ADCON (ADC Control Register) ใช้กำหนดการทำงานและเป็นสถานะการทำงาน

ADCON (S:F3h)
ADC Control Register

7	6	5	4	3	2	1	0
-	PSIDLE	ADEN	ADEOC	ADSST	SCH2	SCH1	SCH0

Bit Number	Bit Mnemonic	Description
7	-	
6	PSIDLE	**Pseudo Idle mode (best precision)** Set to put in idle mode during conversion Clear to converte without idle mode.
5	ADEN	**Enable/Standby Mode** Set to enable ADC Clear for Standby mode (power dissipation 1 uW).
4	ADEOC	**End Of Conversion** Set by hardware when ADC result is ready to be read. This flag can generate an interrupt. Must be cleared by software.
3	ADSST	**Start and Status** Set to start an A/D conversion. Cleared by hardware after completion of the conversion
2-0	SCH2:0	**Selection of channel to convert** see Table 21

Reset Value=X000 0000b

รูปที่ 14.30 หน้าที่ของบิตแต่ละบิตของรีจิสเตอร์ ADCON

ADCLK (ADC Clock prescaler) ใช้กำหนดค่าสัญญาณนาฬิกา

ADCLK (S:F2h)
ADC Clock Prescaler

7	6	5	4	3	2	1	0
-	-	-	PRS 4	PRS 3	PRS 2	PRS 1	PRS 0

Bit Number	Bit Mnemonic	Description
7-5	-	**Reserved** The value read from these bits are indeterminate. Do not set these bits.
4-0	PRS4:0	**Clock Prescaler** f_{ADC} = fosc / (4 (or 2 in X2 mode)* PRS)

Reset Value: XXX0 0000b

รูปที่ 14.31 หน้าที่ของบิตแต่ละบิตของรีจิสเตอร์ ADCLK ADDH และ ADDL

ADDH และ ADDL (ADC Data high byte and low byte register) เป็นค่าที่แปลงได้

ADDH (S:F5h Read Only)
ADC Data High byte register

7	6	5	4	3	2	1	0
ADAT 9	ADAT 8	ADAT 7	ADAT 6	ADAT 5	ADAT 4	ADAT 3	ADAT 2

Bit Number	Bit Mnemonic	Description
7-0	ADAT9:2	**ADC result** bits 9-2

Reset Value: 00h

ADDL (S:F4h Read Only)
ADC Data Low byte register

7	6	5	4	3	2	1	0
-	-	-	-	-	-	ADAT 1	ADAT 0

Bit Number	Bit Mnemonic	Description
7-2	-	**Reserved** The value read from these bits are indeterminate. Do not set these bits.
1-0	ADAT1:0	**ADC result** bits 1-0

Reset Value: 00h

รูปที่ 14.32 หน้าที่ของบิตแต่ละบิตของรีจิสเตอร์ ADCLK ADDH และ ADDL

ตัวอย่างการควบคุม ADC

[1] Configure P1.2 and P1.3 in ADC channels

```
ADCF = 0Ch                  // configure channel P1.2 and P1.3 for ADC
ADCON = 20h                 // Enable the ADC
```

[2] Start a standard conversion

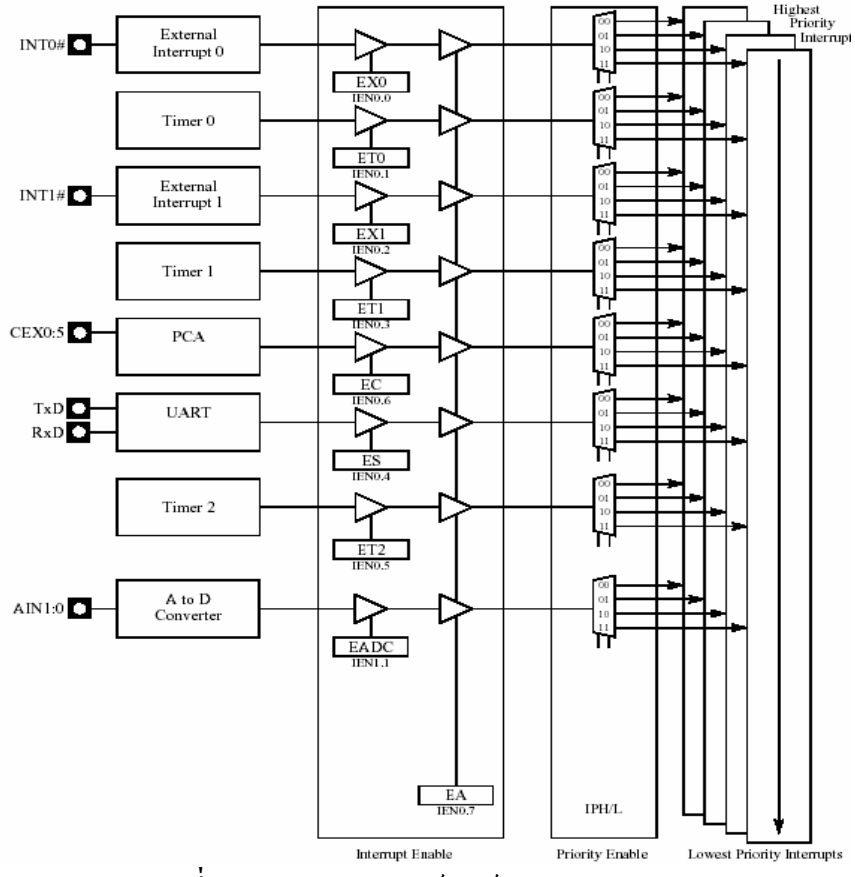
```
// The variable "channel" contains the channel to convert
// The variable "value_converted" is an unsigned int
ADCON and = F8h                     // Clear the field SCH[2:0]
ADCON | = channel                   // Select channel
ADCON | = 08h               // Start conversion in standard mode
while((ADCON and 01h)! = 01h)       // Wait flag End of conversion
ADCON and = EFh                     // Clear the End of conversion flag
value_converted = (ADDH << 2)+(ADDL)        // read the value
```

[3] Start a precision conversion (need interrupt ADC)

```
// The variable "channel" contains the channel to convert
EADC = 1                            // Enable ADC
ADCON and = F8h                     // clear the field SCH[2:0]
ADCON | = channel                   // Select the channel
ADCON | = 48h               // Start conversion in precision mode
```

ถ้าต้องการให้ ADC สามารถอินเตอร์รัพท์ได้ ต้องให้ EA = 1 ด้วย

ระบบอินเตอร์รัพท์ของ T89C51AC2



รูปที่ 14.33 ระบบอินเตอร์รัพท์ของ T89C51AC2

ระดับความสำคัญของสัญญาณอินเตอร์รัพท์

การกำหนดค่าความสำคัญของการอินเตอร์รัพท์

IPH.x	IPL.x	Interrupt Level Priority
0	0	0 (Lowest)
0	1	1
1	0	2
1	1	3 (Highest)

ลำดับความสำคัญของอินเตอร์รัพท์ต่างๆของ T89C51AC2

Interrupt Name	Interrupt Address Vector	Priority Number
external interrupt (INT0)	0003h	1
Timer0 (TF0)	000Bh	2
external interrupt (INT1)	0013h	3
Timer1 (TF1)	001Bh	4
PCA (CF or CCFn)	0033h	5
UART (RI or TI)	0023h	6
Timer2 (TF2)	002Bh	7
-	-	-
ADC (ADCI)	0043h	8