

Phân Lý thuyết

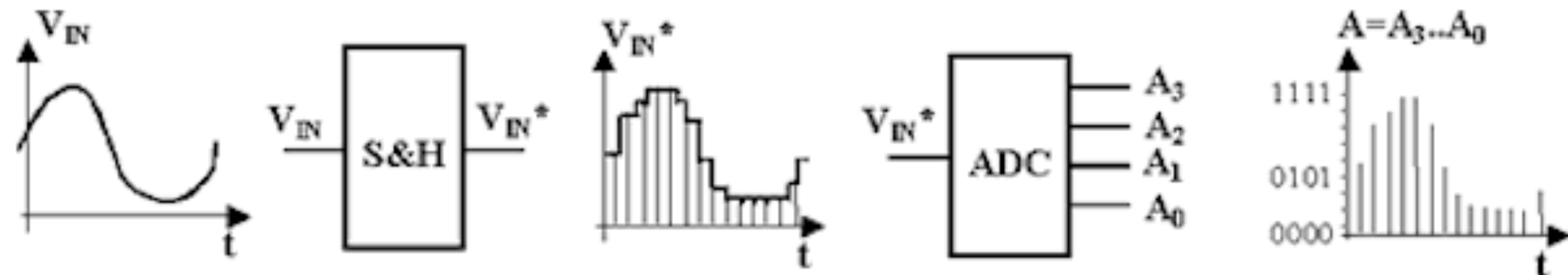
1. Giới thiệu chung - Mô hình hệ VXL - Nguyên tắc hoạt động
2. Cấu trúc và hoạt động của vi xử lý 8085
3. Quá trình thực hiện 1 lệnh trong VXL 8085
4. Giới thiệu về vi điều khiển PIC
5. Bộ công cụ nạp chương trình, công cụ mô phỏng vi điều khiển
6. Bộ định thời Timer
7. Ghép nối với bộ hiển thị
8. ADC
9. Giao tiếp truyền dữ liệu
10. Ngắt
11. PWM

Nội dung buổi học

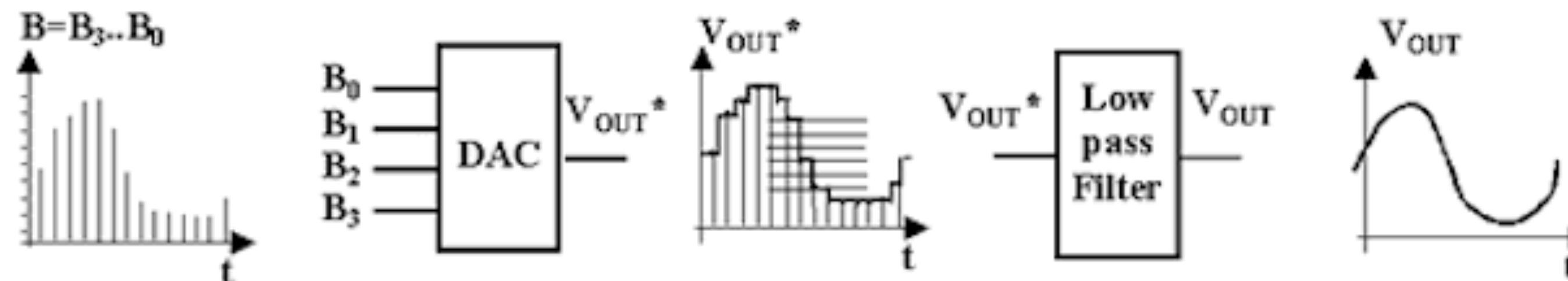
1. ADC
2. Bộ so sánh

Chuyển đổi giữa các tín hiệu

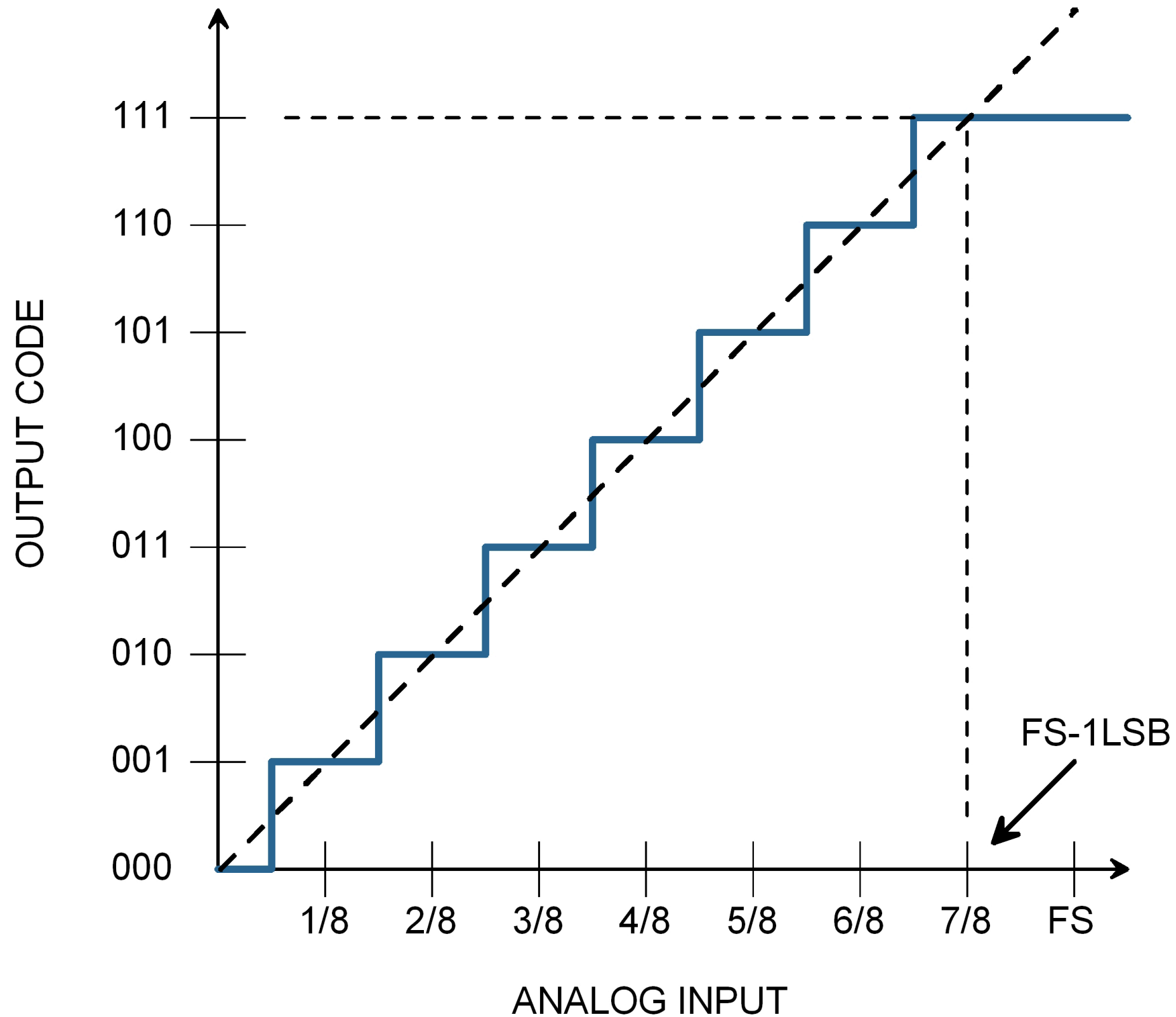
Analog to Digital Converter converts an analog input to a digital output



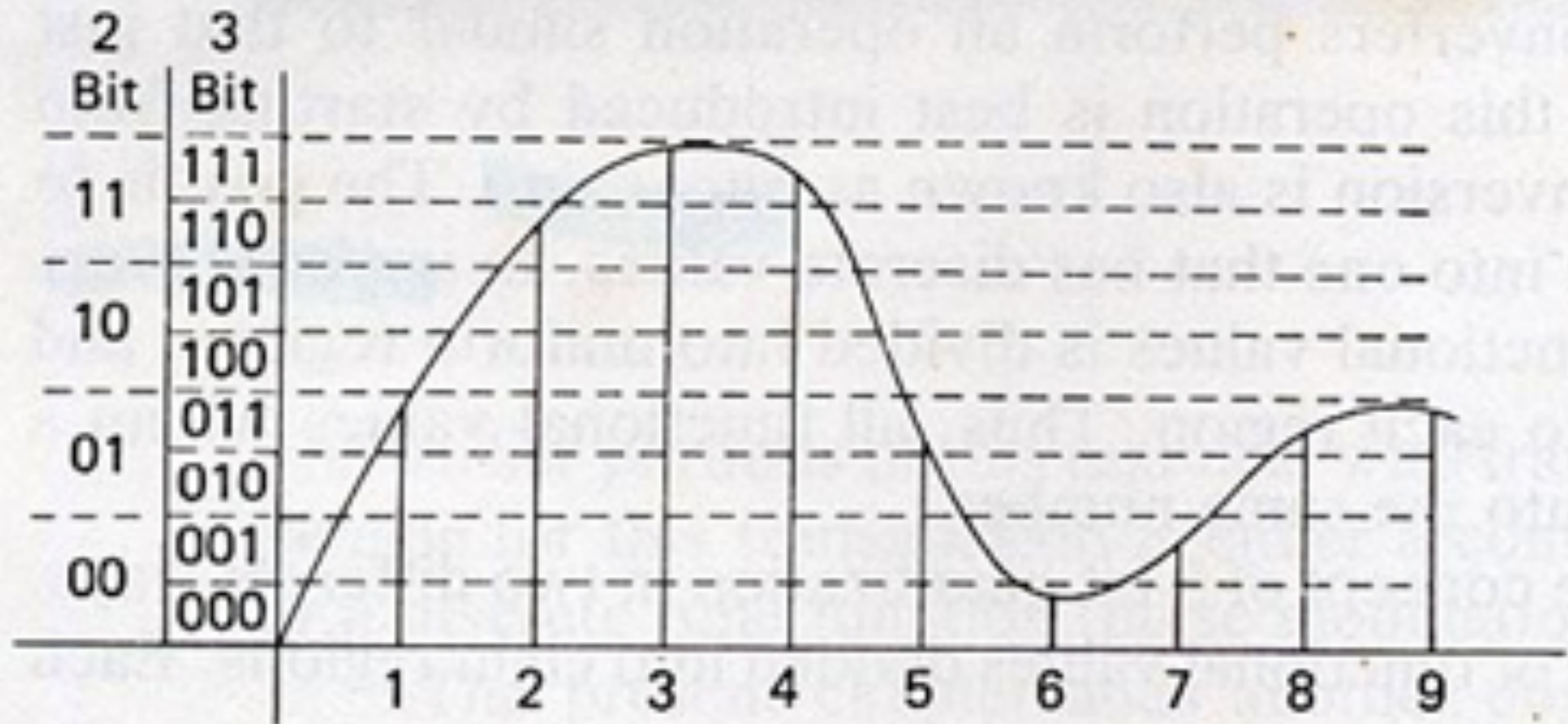
Digital to Analog Converter converts a digital signal to an analog output



Bộ chuyển đổi tương tự-số ADC



Bộ chuyển đổi tương tự-số ADC



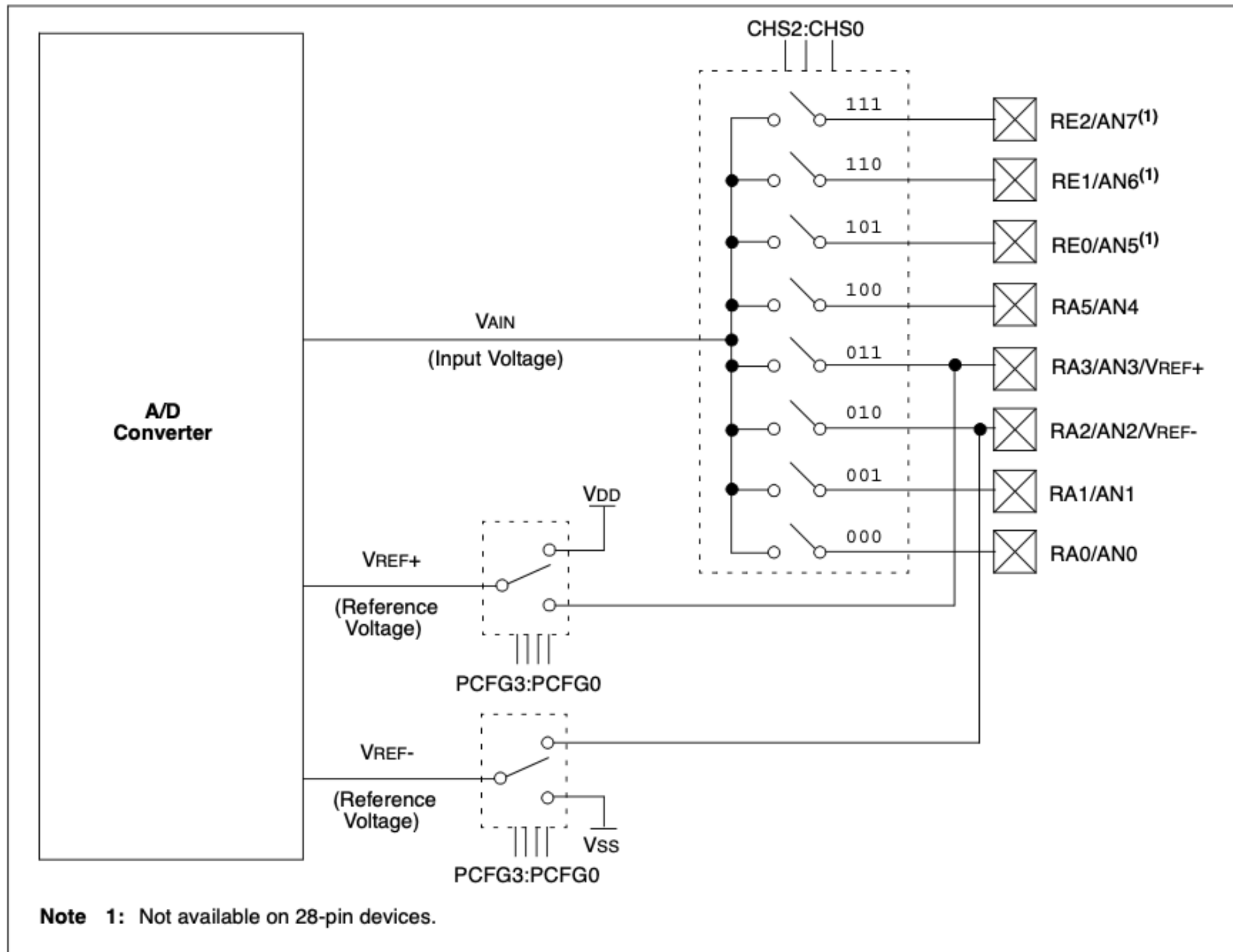
2-bit	01	11	11	11	01	00	00	01	01
3-bit	011	110	111	111	011	000	001	011	011

ADC trong 16F877A

- 8 đầu vào ADC 10 bit (5 chân PortA và 3 chân PortE)
- Dữ liệu sau khi chuyển ADC xong được lưu tạm thời tại 2 thanh ghi ADRESH và ADRESL
- Thanh ghi ADCON0 điều khiển chức năng hoạt động của khối ADC
- Thanh ghi ADCON1 thiết lập chức năng cho các chân của port

Adc Channel	Pic16f877a Pin	Pin Function
0	RA0	AN0
1	RA1	AN1
2	RA2	AN2/VREF-
3	RA3	AN3/VREF+
4	RA5	AN4
2	RE0	AN5
3	RE1	AN6
4	RE2	AN7

ADC trong 16F877A



ADC trong 16F877A

- Thanh ghi ADCON0

ADCON0 REGISTER (ADDRESS 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7							bit 0

ADCS1 và **ADCS0** là các bit lựa chọn tần số xung chuyển đổi ADC

ADCON1 <ADCS2>	ADCON0 <ADCS1:ADCS0>	Clock Conversion
0	00	FOSC/2
0	01	FOSC/8
0	10	FOSC/32
0	11	FRC (clock derived from the internal A/D RC oscillator)
1	00	FOSC/4
1	01	FOSC/16
1	10	FOSC/64
1	11	FRC (clock derived from the internal A/D RC oscillator)

ADC trong 16F877A

CHS2:CHS0 là các bit lựa chọn kênh vào ADC

CHS2:CHS0: Analog Channel Select bits

000 = Channel 0 (AN0)

001 = Channel 1 (AN1)

010 = Channel 2 (AN2)

011 = Channel 3 (AN3)

100 = Channel 4 (AN4)

101 = Channel 5 (AN5)

110 = Channel 6 (AN6)

111 = Channel 7 (AN7)

GO/DONE: báo trạng thái của ADC

ADON: Bit mở nguồn cho ADC hoạt động

ADONE =0 : Khối ADC bị cắt nguồn để giảm công suất tiêu thụ

ADONE =1 : Khối ADC được cấp nguồn. Khi đó:

GO/DONE =1 : quá trình chuyển đổi ADC đang diễn ra

GO/DONE =0 : quá trình chuyển đổi ADC **không** diễn ra

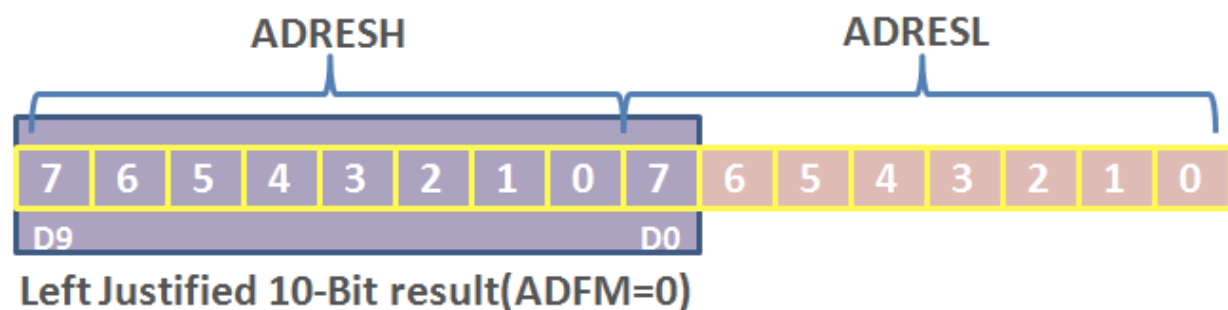
ADC trong 16F877A

- Thanh ghi ADCON1

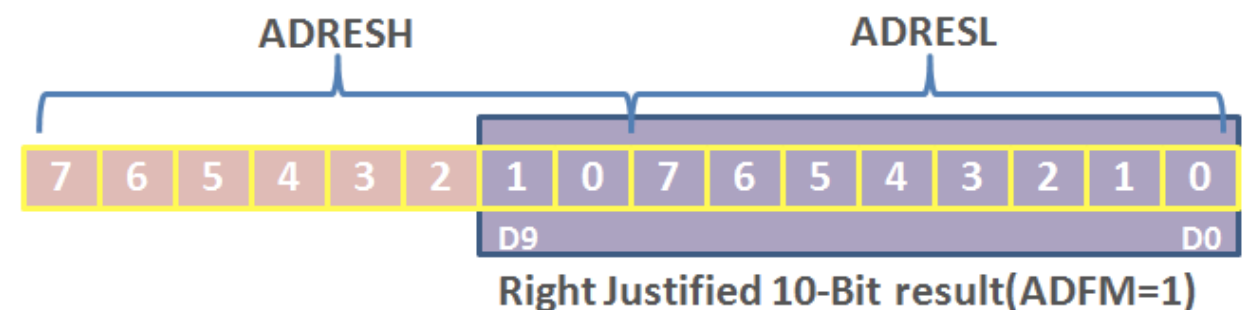
ADCON1 REGISTER (ADDRESS 9Fh)

R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	ADCS2	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

ADFM Bit định dạng kết quả đầu ra của ADC. Nên có ADFM=1



Left Justified 10-Bit result(ADFM=0)



Right Justified 10-Bit result(ADFM=1)

ADCS2 Lựa chọn tần số xung chuyển đổi. Kết hợp với ADCS0 và ADCS1 ở thanh ghi ADCON0

ADC trong 16F877A

PCFG3:PCFG0 Bit điều khiển ADC

PCFG3:PCFG0: A/D Port Configuration Control bits

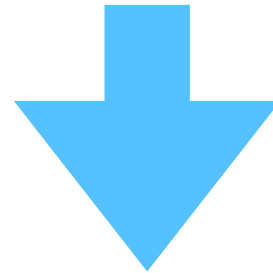
PCFG <3:0>	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0	VREF+	VREF-	C/R
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	AN3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	AN3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	AN3	VSS	2/1
011x	D	D	D	D	D	D	D	D	—	—	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	AN3	AN2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	AN3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	AN3	AN2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	AN3	AN2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	AN3	AN2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	AN3	AN2	1/2

A = Analog input D = Digital I/O

C/R = # of analog input channels/# of A/D voltage references

ADC trong 16F877A

Quá trình chuyển đổi ADC hoàn thành



Kết quả 10 bit được nạp vào
ADRESH:ADRESL



Bit GO/DONE bị xoá về 0

Bit ADIF được set lên 1

Các bước thực hiện ADC

1. Cấu hình module ADC

Cấu hình các chân analog/digital và Vref - ADCON1

Chọn các chân đầu vào cho A/D - ADCON0

Chọn xung nhịp chuyển đổi ADC - ADCON0

Cấp nguồn cho ADC - ADCON0

2. Cấu hình ngắt ADC nếu có

Xoá bit ADIF

Set bit ADIE

Set bit PEIE

Set bit GIE

3. Chờ 1 khoảng thời gian yêu cầu khoảng $19,72\mu s \rightarrow$ thường chọn $20\mu s$

Các bước thực hiện ADC

4. Bắt đầu chuyển đổi: set bit $\overline{GO/DONE}$

5. Chờ cho ADC hoàn thành

Kiểm tra liên tục bit $\overline{GO/DONE}$ xem về 0 hay chưa

Đợi ngắt ADC

6. Đọc kết quả từ ADRESH:ADRESL. Xoá ADIF nếu được yêu cầu

7. Quay về bước 1 hoặc 2 nếu cần. Thời gian dẫn cách chuyển đổi là T_{AD}

AD Clock Source (T_{AD})		Maximum Device Frequency
Operation	ADCS2:ADCS1:ADCS0	
2 TOSC	000	1.25 MHz
4 TOSC	100	2.5 MHz
8 TOSC	001	5 MHz
16 TOSC	101	10 MHz
32 TOSC	010	20 MHz
64 TOSC	110	20 MHz
RC ^(1, 2, 3)	x11	(Note 1)

Các hàm khai báo, sử dụng trên CCS

1. setup_adc(mode)

Hàm không trả về giá trị

Các tham số cho từng VDK có thể mở file.h tương ứng, xem tại phần ADC

Cơ bản có những mode sau

adc_off	tắt hoạt động của ADC
adc_clock_internal	thời gian lấy mẫu bằng xung clock IC, 200ns với thạch anh 20MHz
adc_clock_div_2	Thời gian lấy mẫu bằng xung clock/2 (400ns - 0.4μs)
adc_clock_div_8	Thời gian lấy mẫu bằng xung clock/8 (1.6μs)
adc_clock_div_32	Thời gian lấy mẫu bằng xung clock/32 (6.4μs)

Các hàm khai báo, sử dụng trên CCS

2. `setup_adc_ports(value)`

Xác định chân vào analog và các điện thế chuẩn để sử dụng

Tham số chi tiết cần xem trong thư viện. Khai báo các chân như ví dụ dưới đây

	Chân đầu vào analog	Điện áp so sánh
<code>all_analogs</code>	A0 A1 A2 A3 A5 E0 E1 E2	Vref=Vdd
<code>no_analog</code>	Không dùng ADC, các chân chỉ là cổng I/O	
<code>an0_an1_an3</code>	A0 A1 A3	Vref=Vdd
<code>an0_an1_an4_an5_vss_vref</code>	A0 A1 A5 E0	Vref=A3
<code>an0_an1_an4_vref_vref</code>	A0 A1 A5	Vrefh=A3, Vrefl=A2

Các hàm khai báo, sử dụng trên CCS

3. `set_adc_channel(channel)`

Hàm không trả về giá trị

Dùng để chọn chân vào analog trước khi đọc giá trị analog

Giá trị từ 0 đến 7

Nên delay khoảng $10\mu s$ trước khi dùng lệnh `read_adc()`

4. `read_adc(mode)`

trả về kết quả số 8 bit (hoặc 10 bit) từ chân được chọn trong lệnh `set_adc_channel()` ở trước đó

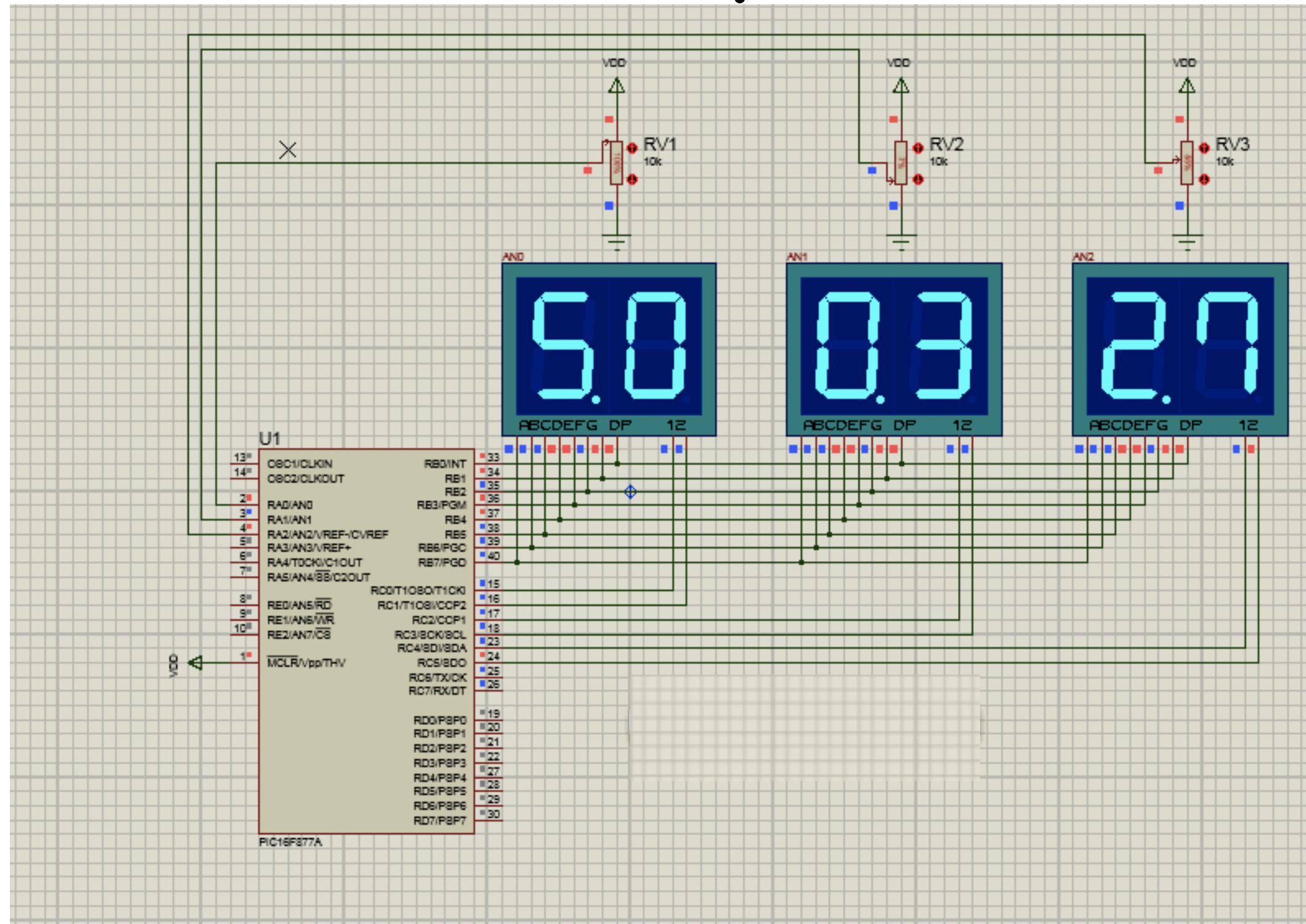
Chú ý cần khai báo để chọn độ phân giải adc `mode` có thể có/có thể không

`#device adc=8`

hoặc `#device adc=10`

<code>adc_start_and_read</code>	Giá trị mặc định
<code>adc_start_only</code>	bắt đầu chuyển đổi và trả về
<code>adc_read_only</code>	đọc KQ chuyển lần cuối

Ví dụ



Đọc tín hiệu analog từ 3 biến trở, hiển thị kết quả điện áp tương ứng trên led 7 thanh

Ví dụ

```
#include <16F877A.h>
```

```
#device ADC=10
```

```
#fuses HS, NOWDT, PROTECT
```

```
#use delay(clock=20M)
```

```
#use fast_io(b)
```

```
#use fast_io(c)
```

```
static unsigned char So[]={0x03, 0x9f, 0x25, 0x0d, 0x99, 0x49, 0x41, 0x1b, 0x01, 0x09};
```

```
unsigned int8 DV, TP;          Biến lưu tạm thời số đơn vị, thập phân
```

```
unsigned int8 Value1, Value2, Value3;    Biến lưu giá trị từ ADC
```

```
float Temp;
```

```
}
```

Ví dụ

```
void Hienthi()  
{  
    TP=Value1%10;  
    DV=Value1/10;  
    output_c(0x01);  
    output_b(So[DV]-0x01);  
    delay_ms(1);  
    output_c(0x02);  
    output_b(So[TP]);  
    delay_ms(1);  
  
    TP=Value2%10;  
    DV=Value2/10;  
    output_c(0x04);  
    output_b(So[DV]-0x01);  
    delay_ms(1);  
    output_c(0x08);  
    output_b(So[TP]);  
    delay_ms(1);
```

```
    TP=Value3%10;  
    DV=Value3/10;  
    output_c(0x10);  
    output_b(So[DV]-0x01);  
    delay_ms(1);  
    output_c(0x20);  
    output_b(So[TP]);  
    delay_ms(1);  
  
    output_c(0x00);  
    output_b(0xff);  
    delay_ms(1);  
}
```

Hàm chuyển đổi mã 7 thanh, chia tách thập phân - đơn vị rồi hiển thị trên led 7 thanh

Ví dụ

```
void Doc_ADC()
{
    set_adc_channel(0); //Chọn kênh AN0
    delay_us(20);        //Trễ tối thiểu 20us
    temp=read_adc();
    Value1=(int8)((temp*50)/1023);

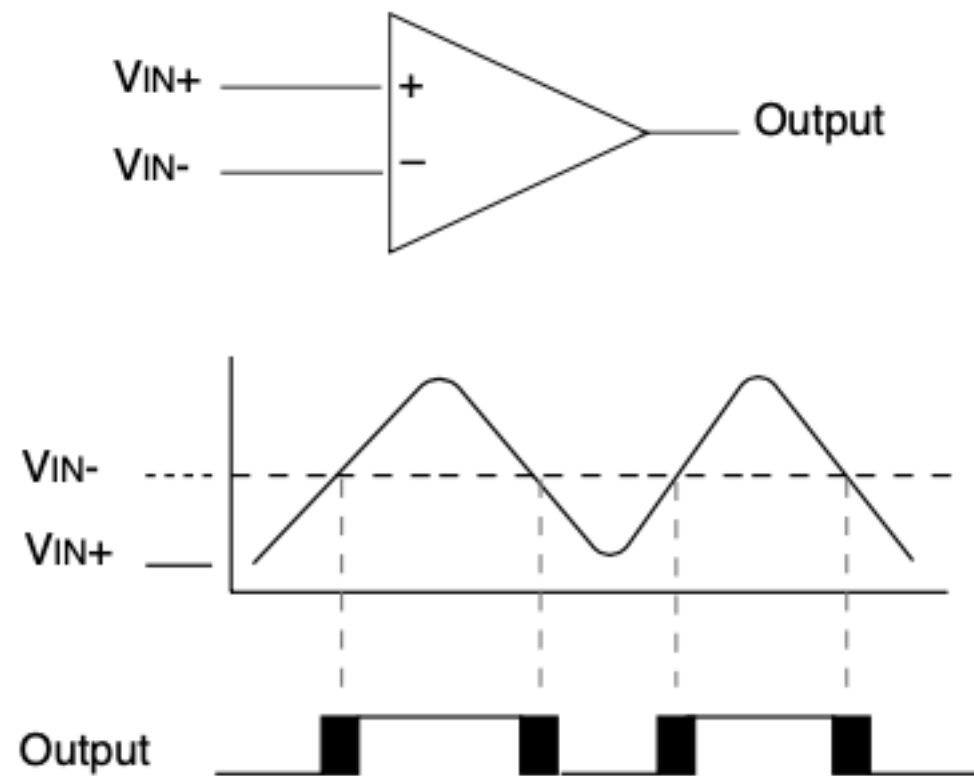
    set_adc_channel(1); // Chọn Kênh AN1
    delay_us(20);        //Trễ tối thiểu 20us
    temp=read_adc();
    Value2=(int8)((temp*50)/1023);

    set_adc_channel(2); //Chọn Kênh AN2
    delay_us(20);        //Trễ tối thiểu 20us
    temp=read_adc();
    Value3=(int8)((temp*50)/1023);
}
```

```
void main()
{
    set_tris_b(0x00);
    set_tris_c(0x00);
    setup_adc(ADC_CLOCK_INTERNAL);
    setup_adc_ports(AN0_AN1_AN2);
    while(true)
    {
        Doc_ADC();
        Hienthi();
    }
}
```

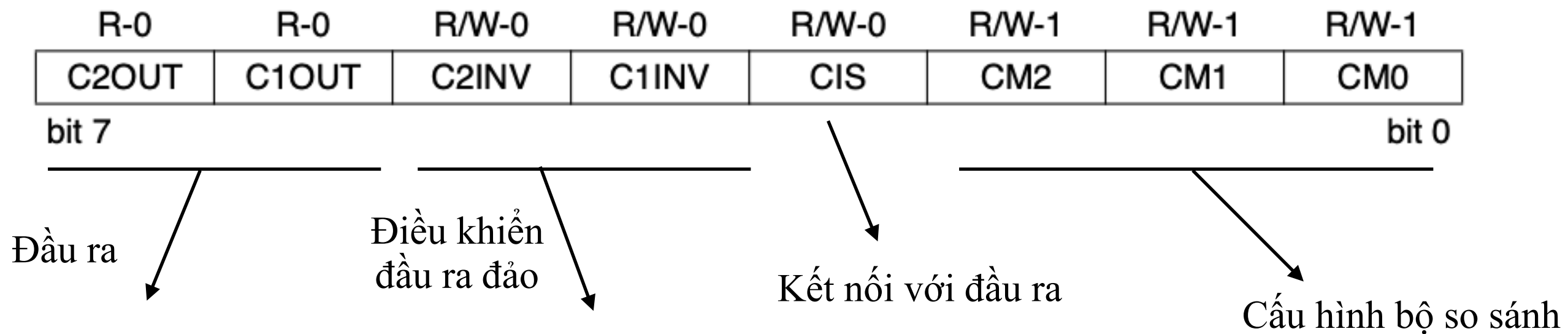
Bộ so sánh Comparator

- Khối so sánh trong PIC16F877A có 2 bộ so sánh
- Các đầu vào được tích hợp trong chân RA0-RA3
- Tín hiệu ra được tích hợp trong chân RA4, RA5
- Điện áp tham chiếu on-chip cũng có thể được coi là 1 đầu input



Thanh ghi điều khiển bộ so sánh

CMCON REGISTER



C2OUT: Comparator 2 Output bit

When C2INV = 0:

1 = C2 $V_{IN+} > C2 V_{IN-}$
0 = C2 $V_{IN+} < C2 V_{IN-}$

When C2INV = 1:

1 = C2 $V_{IN+} < C2 V_{IN-}$
0 = C2 $V_{IN+} > C2 V_{IN-}$

C1OUT: Comparator 1 Output bit

When C1INV = 0:

1 = C1 $V_{IN+} > C1 V_{IN-}$
0 = C1 $V_{IN+} < C1 V_{IN-}$

When C1INV = 1:

1 = C1 $V_{IN+} < C1 V_{IN-}$
0 = C1 $V_{IN+} > C1 V_{IN-}$

C2INV: Comparator 2 Output Inversion bit

1 = C2 output inverted
0 = C2 output not inverted

C1INV: Comparator 1 Output Inversion bit

1 = C1 output inverted
0 = C1 output not inverted

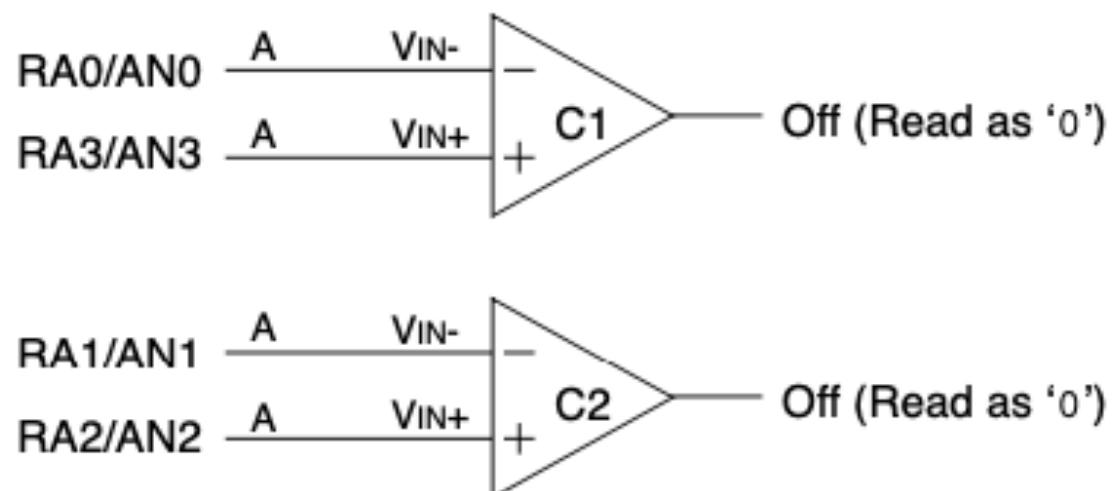
CIS: Comparator Input Switch bit

When CM2:CM0 = 110:

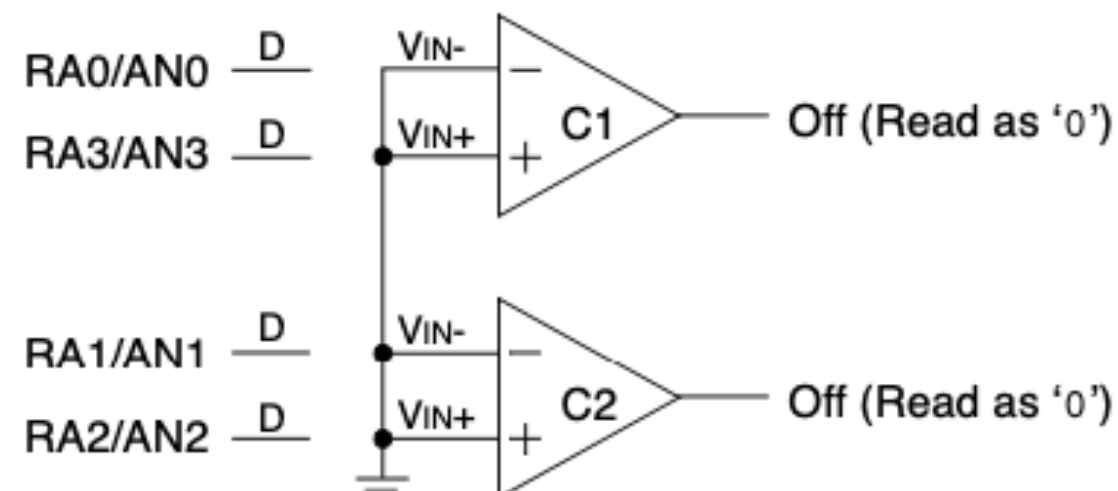
1 = C1 V_{IN-} connects to RA3/AN3
C2 V_{IN-} connects to RA2/AN2
0 = C1 V_{IN-} connects to RA0/AN0
C2 V_{IN-} connects to RA1/AN1

8 cấu hình bộ so sánh

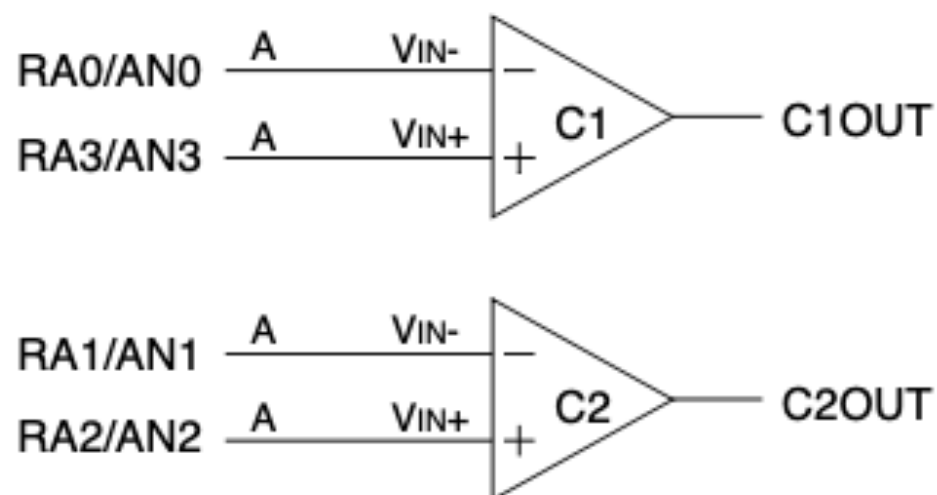
Comparators Reset CM2:CM0 = 000



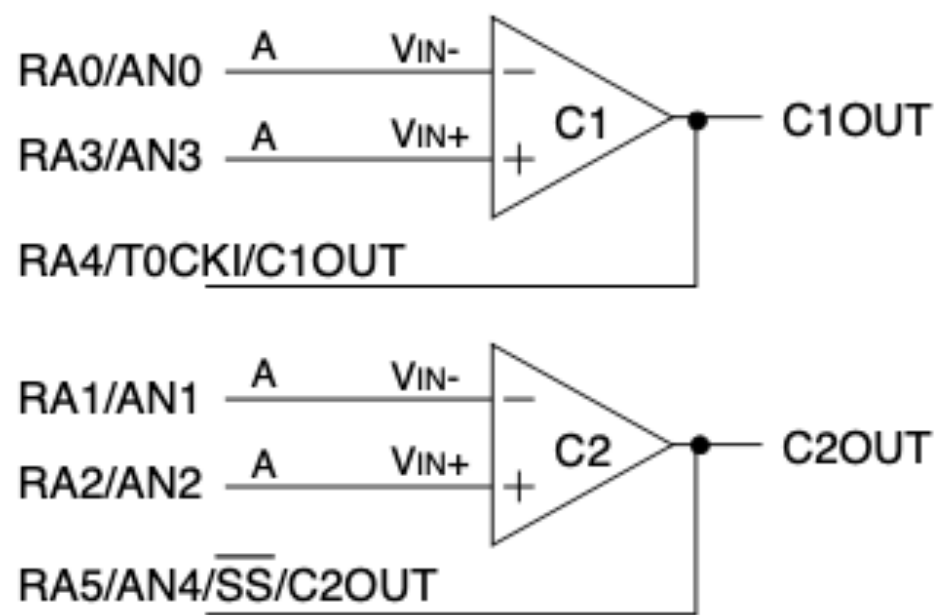
Comparators Off (POR Default Value) CM2:CM0 = 111



Two Independent Comparators CM2:CM0 = 010



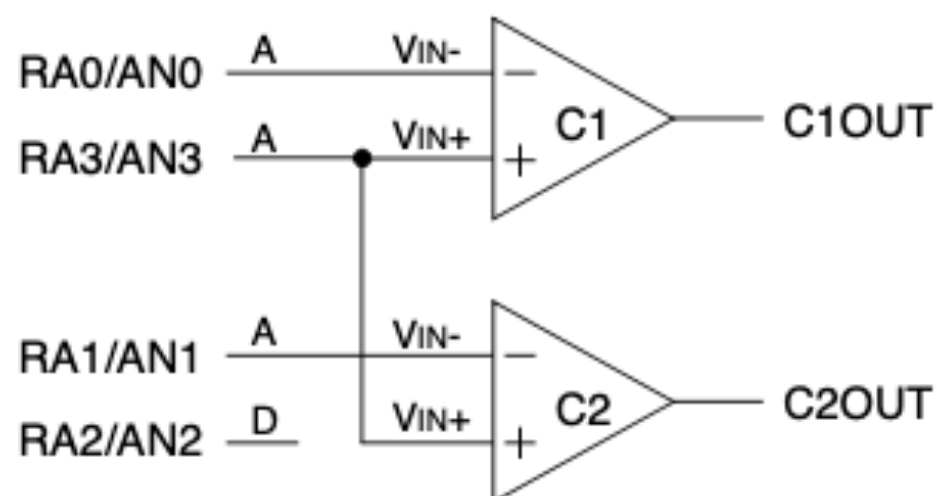
Two Independent Comparators with Outputs CM2:CM0 = 011



8 cấu hình bộ so sánh

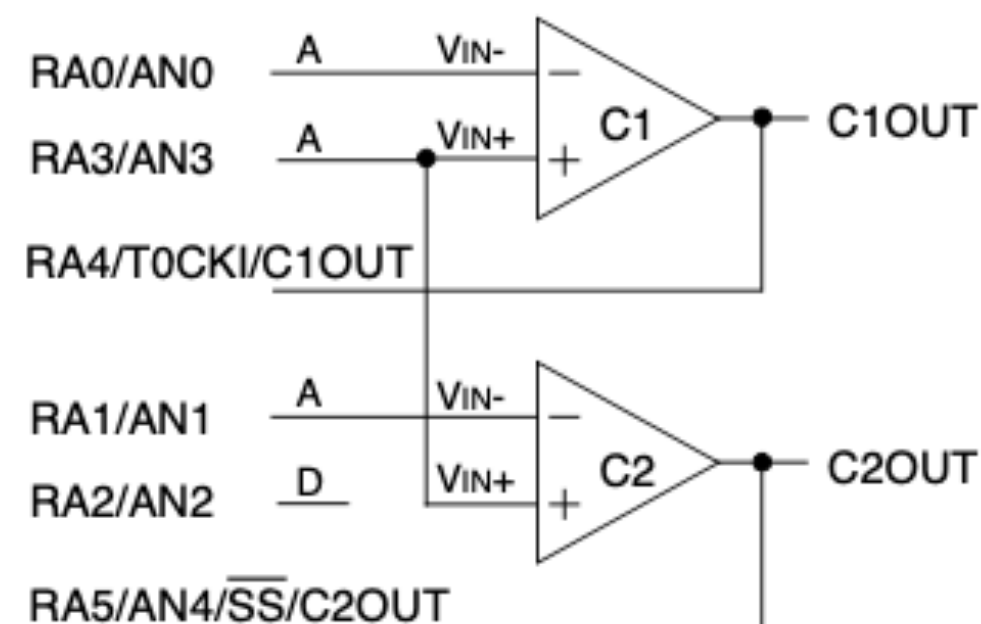
Two Common Reference Comparators

CM2:CM0 = 100



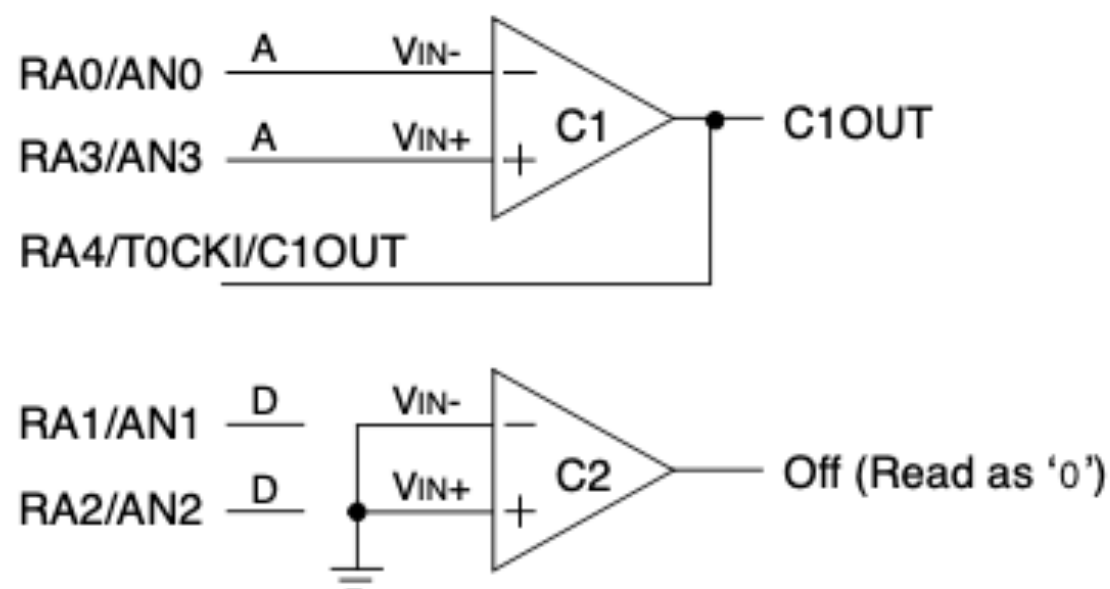
Two Common Reference Comparators with Outputs

CM2:CM0 = 101



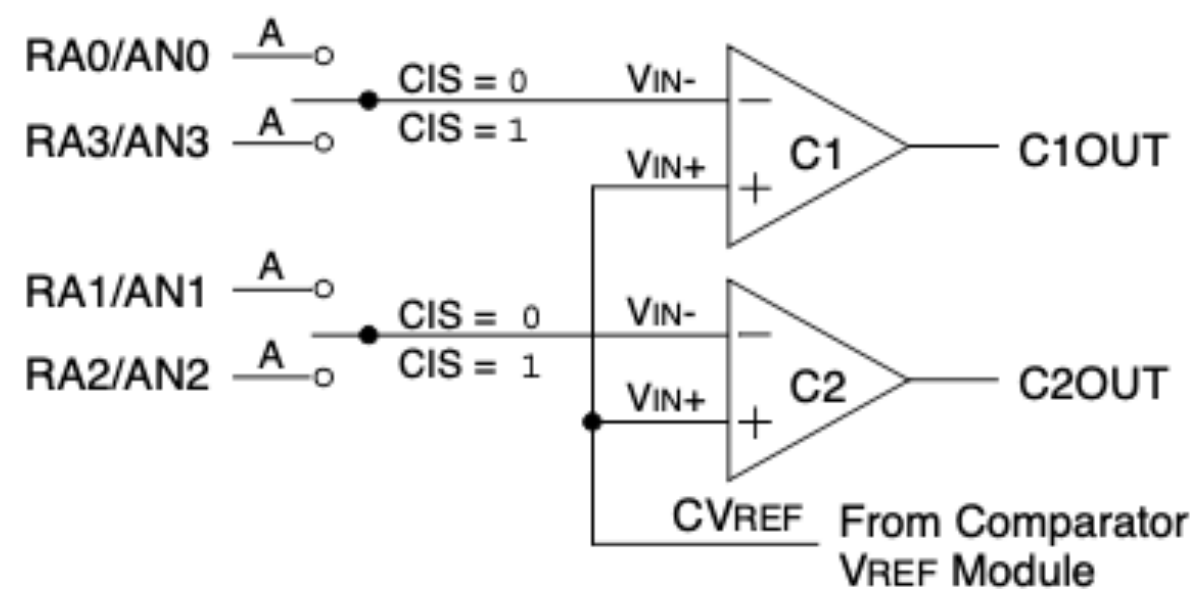
One Independent Comparator with Output

CM2:CM0 = 001

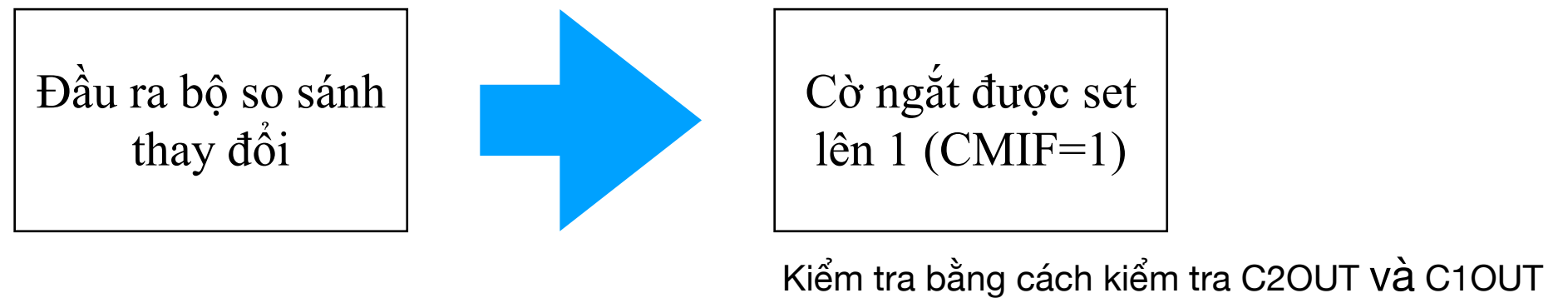


Four Inputs Multiplexed to Two Comparators

CM2:CM0 = 110



Ngắt của bộ so sánh



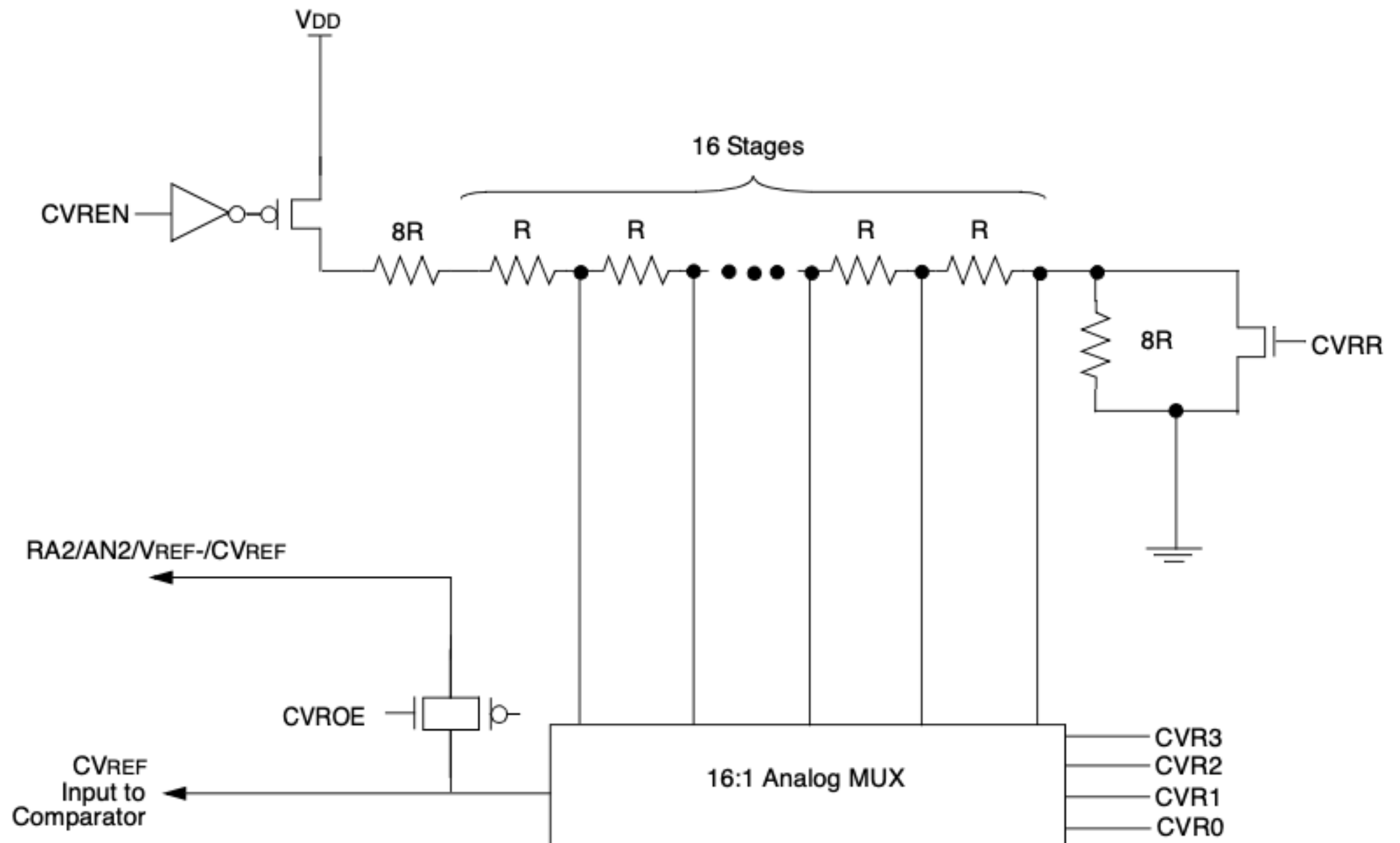
- CMIF (trong PIR) là cờ báo ngắt bộ so sánh
- CMIF được reset bằng cách xoá nó
- Để cho phép thực hiện ngắt so sánh, cần set các bit CMIE (trong PIE) và PEIE (trong INTCON) lên 1. Ngoài ra bit GIE cũng phải được set lên 1
- Trong chương trình ngắt, người sử dụng xoá ngắt theo cách sau

Thực hiện lệnh bất kỳ đọc/ghi CMCON để kết thúc điều kiện không thích ứng

Xoá CMIF

Bộ tạo điện áp chuẩn so sánh

- Là mạng điện trở bậc thang 16 cấp
- Tạo điện áp chuẩn cố định khi bộ so sánh làm việc kiểu 101



Điện áp ra
Đưa vào bộ so sánh

Bộ tạo điện áp chuẩn so sánh

CVRCON CONTROL REGISTER (ADDRESS 9Dh)

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
CVREN	CVROE	CVRR	—	CVR3	CVR2	CVR1	CVR0
bit 7							bit 0

CVREN: Cho phép sử dụng bộ điện áp chuẩn so sánh

- 1 = CVREF circuit powered on
- 0 = CVREF circuit powered down

CVRR: Lựa chọn dãy điện áp của bộ so sánh

- 1 = 0 to 0.75 CVRSRC, with CVRSRC/24 step size
- 0 = 0.25 CVRSRC to 0.75 CVRSRC, with CVRSRC/32 step size

CVROE: Cho phép đưa điện áp ra ngoài chân CV_{REF}

- 1 = CVREF voltage level is output on RA2/AN2/VREF-/CVREF pin
- 0 = CVREF voltage level is disconnected from RA2/AN2/VREF-/CVREF pin

CVR3:CVR0: Lựa chọn giá trị V_{REF} của bộ so sánh từ 0 đến 15

When CVRR = 1:

$$CV_{REF} = (VR_{<3:0>} / 24) \cdot (CVRSRC)$$

When CVRR = 0:

$$CV_{REF} = 1/4 \cdot (CVRSRC) + (VR_3:VR_0 / 32) \cdot (CVRSRC)$$

CV_{RSRC} là điện áp nguồn cho bộ so sánh - thường đến trực tiếp từ V_{DD}