

ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH  
KHOA CÔNG NGHỆ ĐIỆN TỬ  
BỘ MÔN ĐIỆN TỬ MÁY TÍNH

# CHƯƠNG 8

## MODUL

## CHUYỂN ĐỔI TƯƠNG

## TỰ SANG SỐ

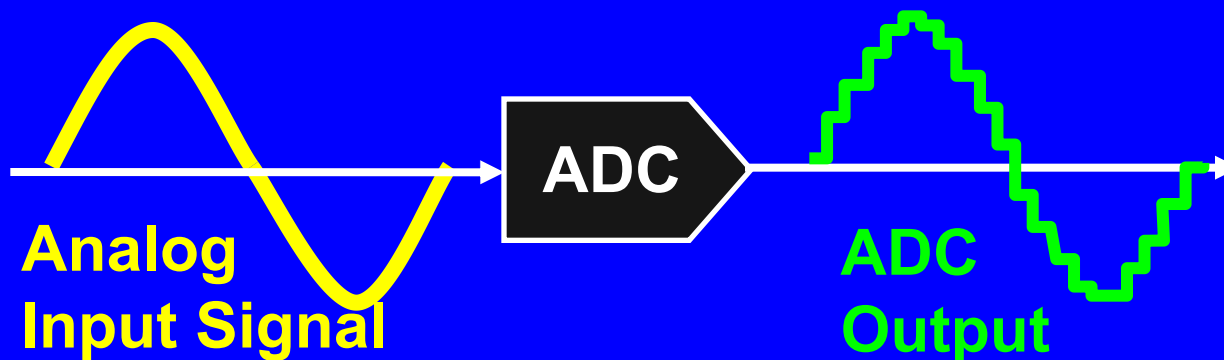
(ADC: ANALOG TO  
DIGITAL CONVERTER)

### Nội dung bao gồm:

- Tổng quan về modul ADC
- Các thanh ghi của modul ADC
- Sơ đồ khối đơn giản của modul ADC
- Vấn đề thời gian trong ADC
- Một số vấn đề cần lưu ý
- Ví dụ minh họa và bài tập ứng dụng.

# TỔNG QUAN VỀ MODUL ADC

- PIC16F887 chuyển đổi tín hiệu vào tương tự thành giá trị nhị phân 10 bit với 14 ngõ vào
- Có khả năng chọn lựa điện áp tham chiếu bên trong hoặc bên ngoài
- Sự kiện ngắt có thể được tạo ra sau khi quá trình chuyển đổi hoàn tất
- Sự kiện ngắt có thể được dùng để đánh thức PIC khỏi chế độ "ngủ".



# CÁC THANH GHI MODUL ADC

- Modul ADC của PIC16F887 có hai thanh ghi điều khiển là: **ADCON0** và **ADCON1**

## ADCON0

ADCS1	ADCS0	CHS3	CHS2	CHS1	CHS0	GO/DONE	ADON
-------	-------	------	------	------	------	---------	------

BIT	Chức năng
ADCS<1:0>	Các bit chọn xung clock chuyển đổi A/D 00 = $F_{osc}/2$ , 01 = $F_{osc}/8$ , 10 = $F_{osc}/32$ , 11 = $F_{RC}$ (Bộ dao động bên trong)
CHS<3:0>	Các bit chọn kênh tương tự
GO/DONE	1 = Bộ chuyển đổi A/D đang xử lý 0 = Bộ chuyển đổi A/D đã hoàn tất quá trình chuyển đổi
ADON	Cho phép modul ADC hoạt động.

# CÁC THANH GHI MODUL ADC

- Modul ADC của PIC16F887 có hai thanh ghi điều khiển là: **ADCON0** và **ADCON1**

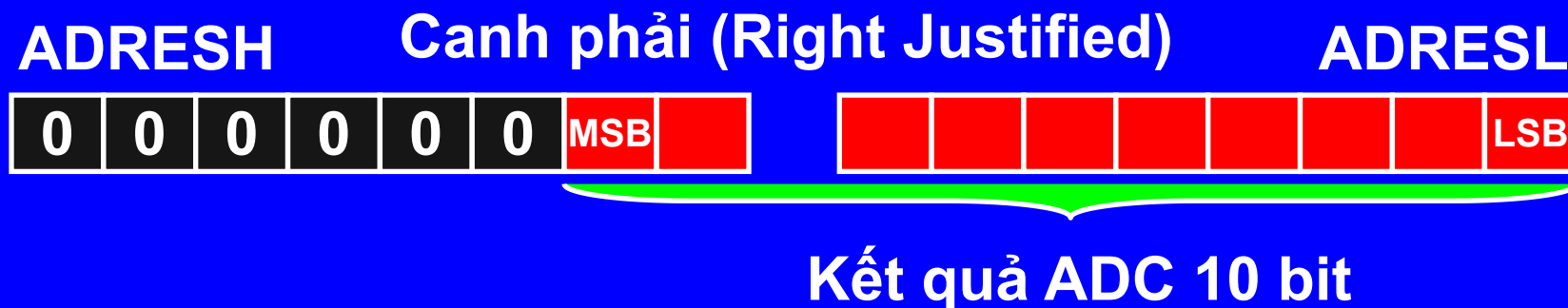
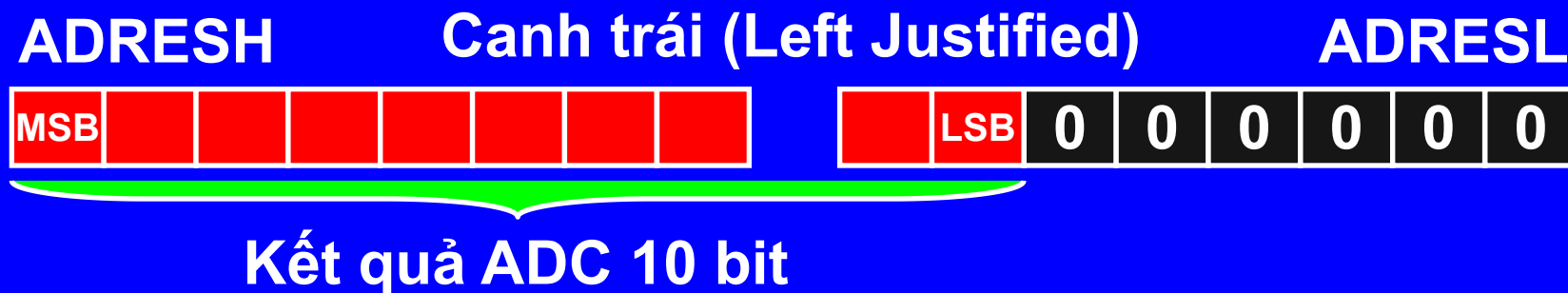
## ADCON1

<b>ADFM</b>	---	<b>VCFG1</b>	<b>VCFG0</b>	---	---	---	---
-------------	-----	--------------	--------------	-----	-----	-----	-----

BIT	Chức năng
<b>ADFM</b>	Điều chỉnh giá trị các thanh ghi kết quả 1 = Canh phải, 0 = Canh trái
<b>VCFG1</b>	Điện áp tham chiếu âm ( $V_{REF-}$ ) 1 = Nguồn ngoài tại chân Vref-, 0 = Vss
<b>VCFG0</b>	Điện áp tham chiếu dương ( $V_{REF+}$ ) 1 = Nguồn ngoài tại chân Vref+, 0 = Vdd

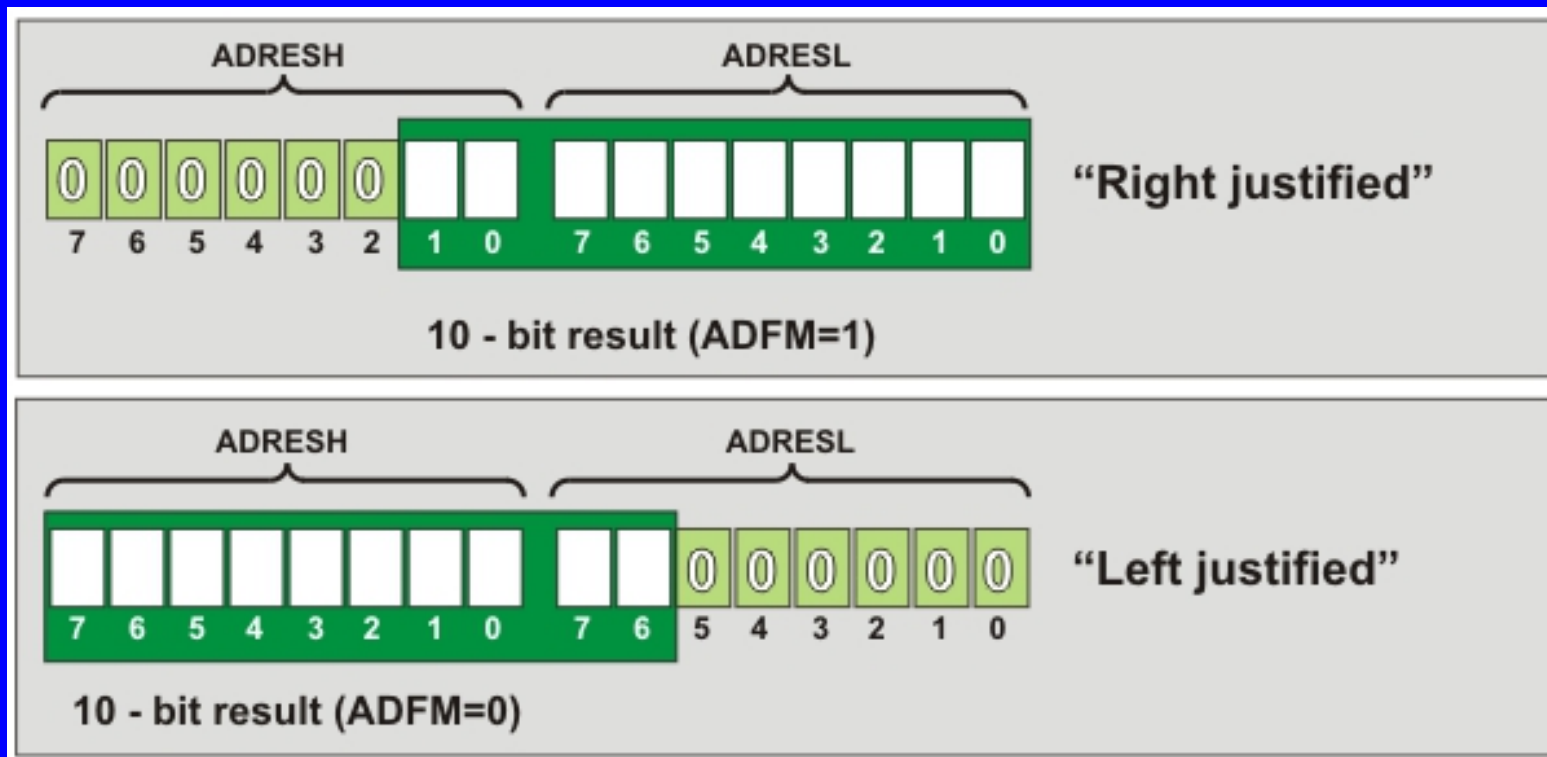
# CÁC THANH GHI MODUL ADC

- Sau khi chuyển đổi, kết quả ADC được đặt vào trong hai thanh ghi kết quả: **ADRESH** và **ADRESL**
- Kết quả ADC 10 bit có thể được **canh trái** hoặc **canh phải**

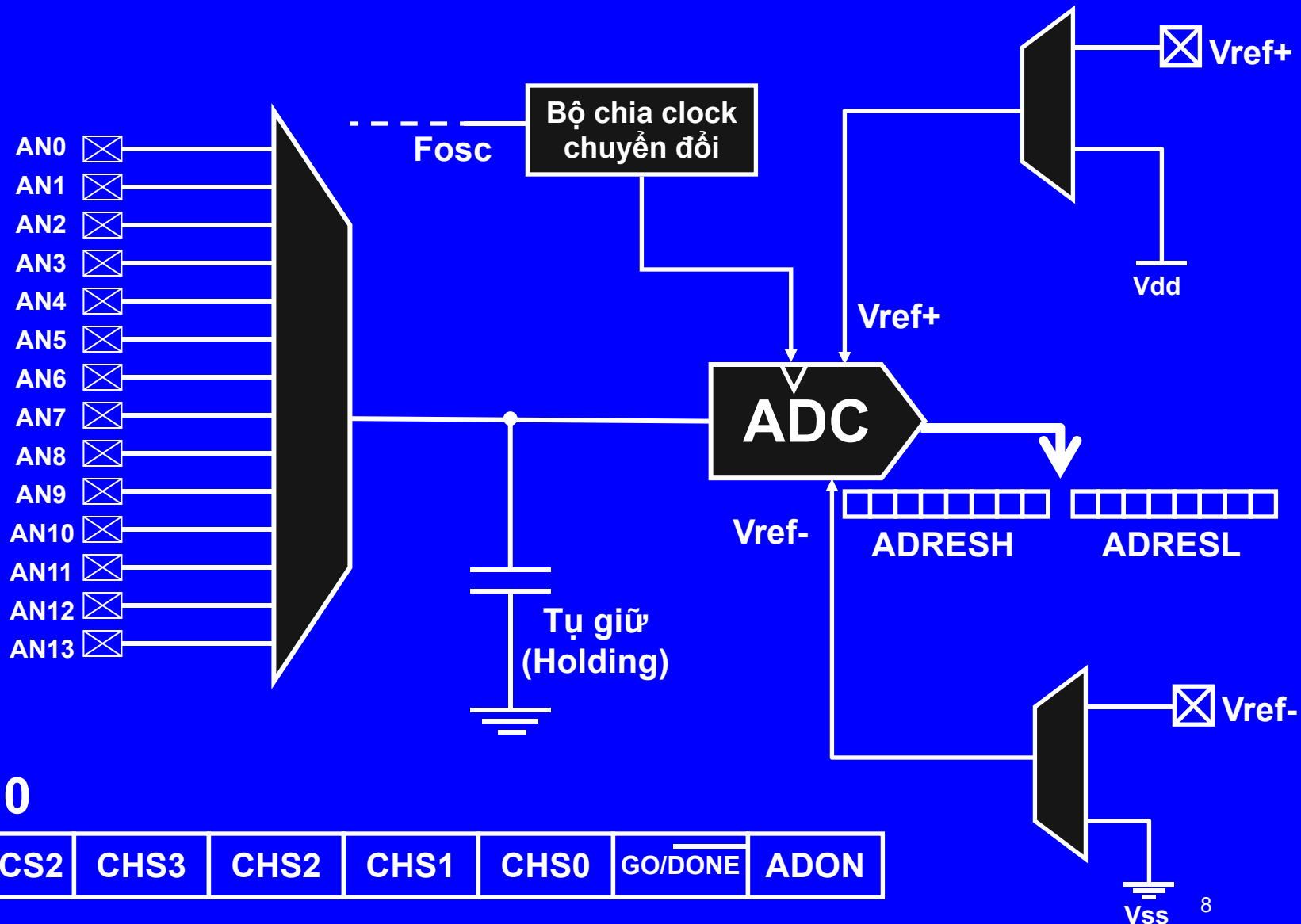


# CÁC THANH GHI MODUL ADC

- Sau khi chuyển đổi, kết quả ADC được đặt vào trong hai thanh ghi kết quả: **ADRESH** và **ADRESL**
- Kết quả ADC 10 bit có thể được **canh trái** hoặc **canh phải**

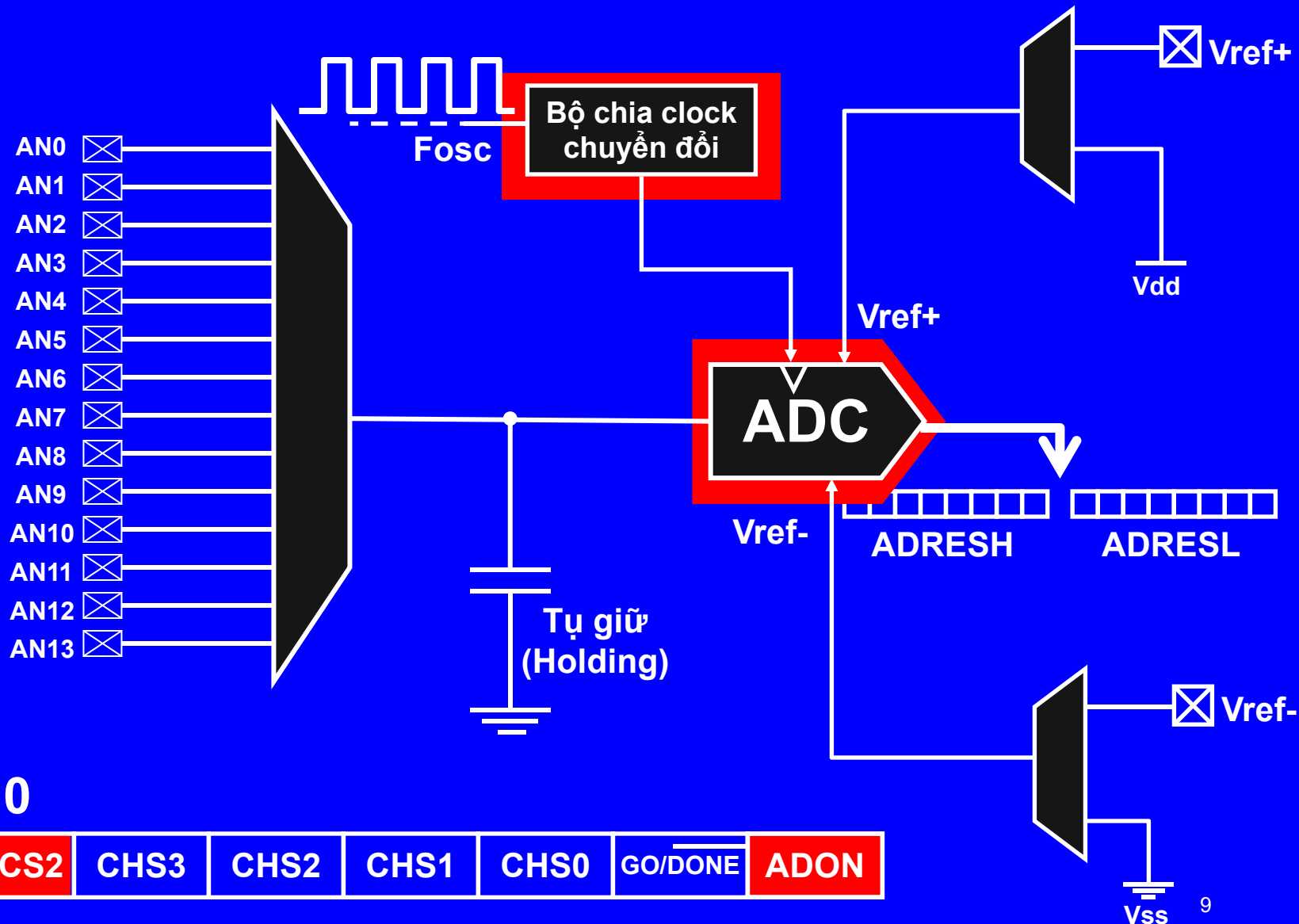


# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC

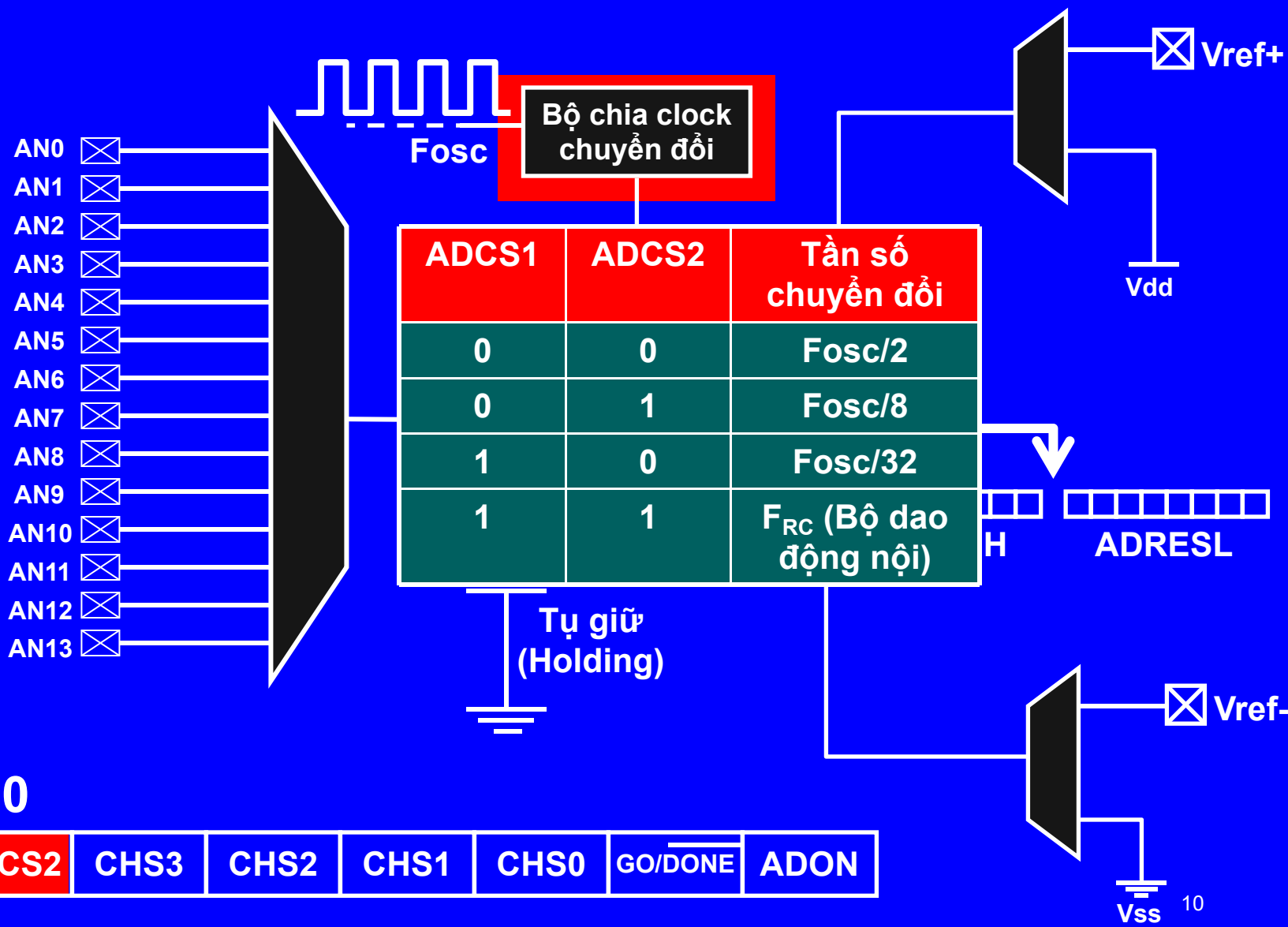




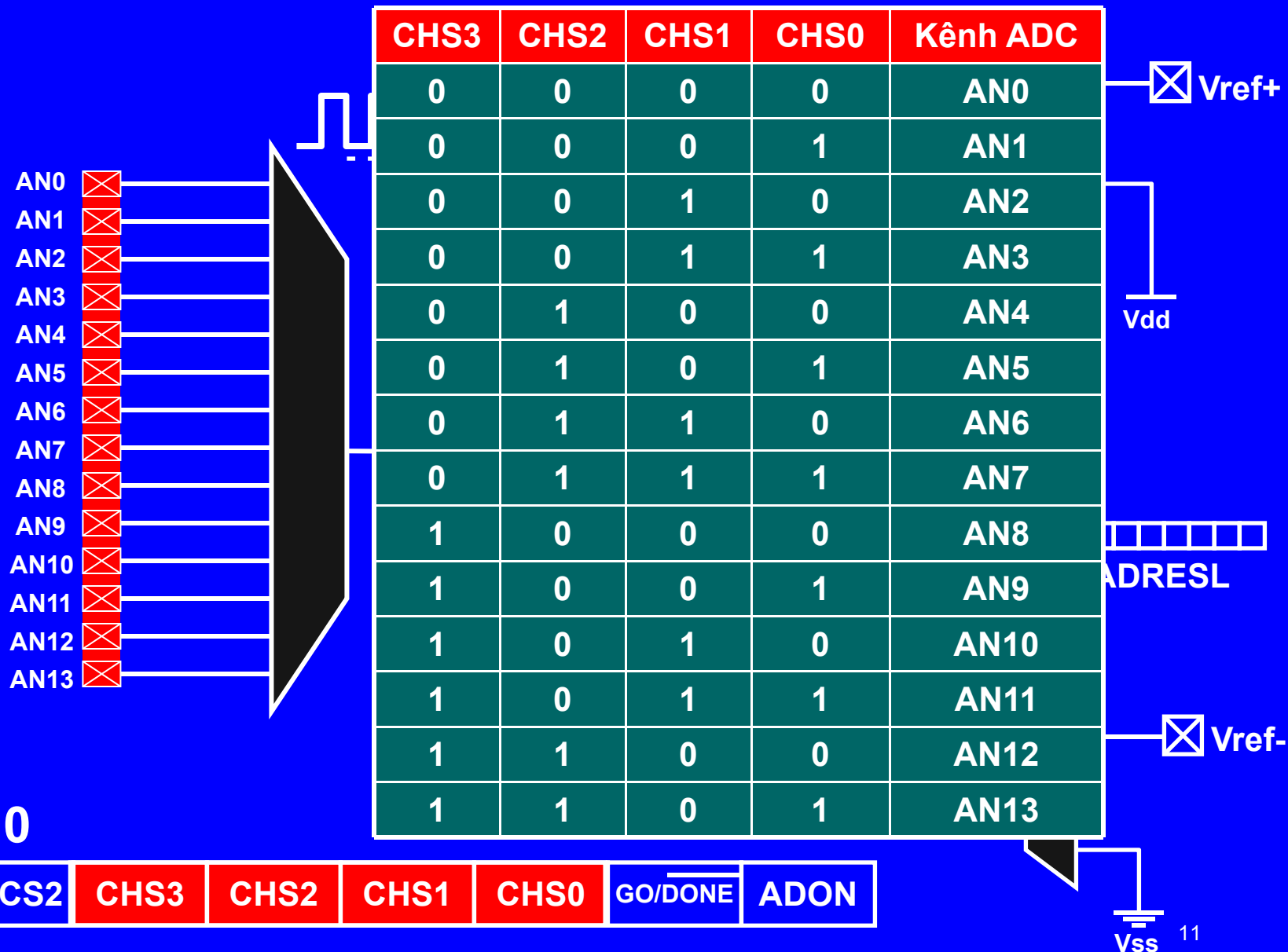
# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



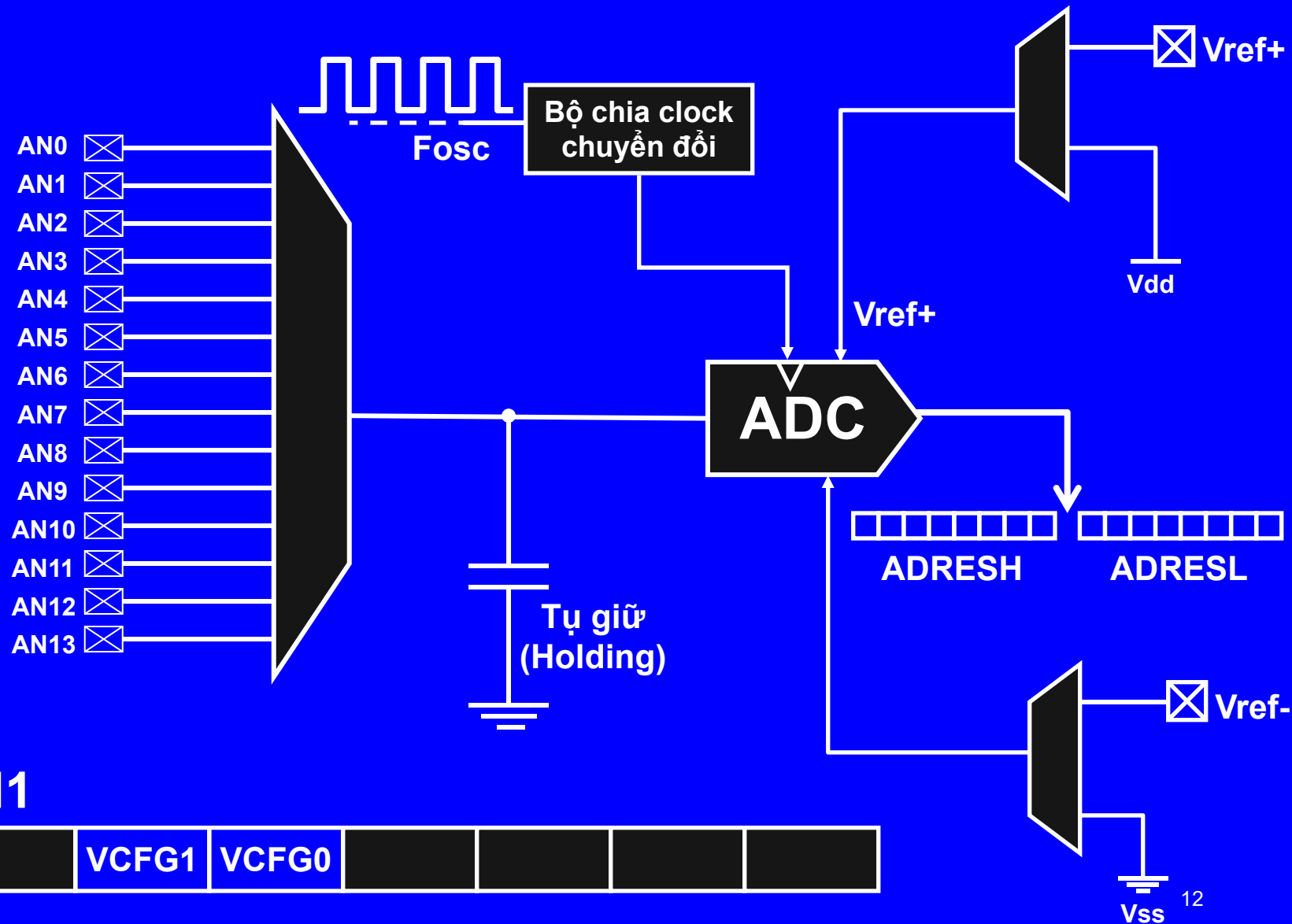
# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



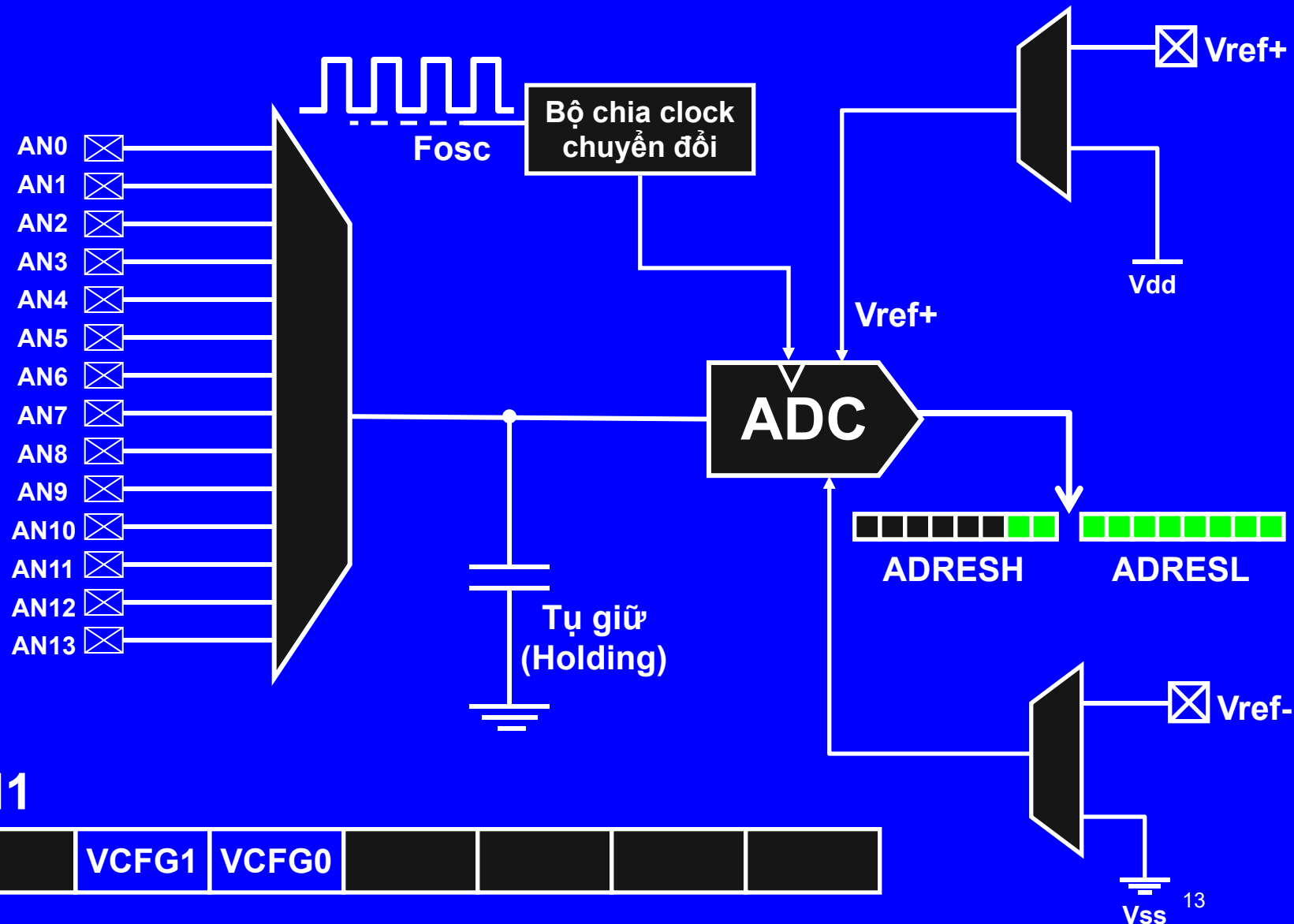
# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



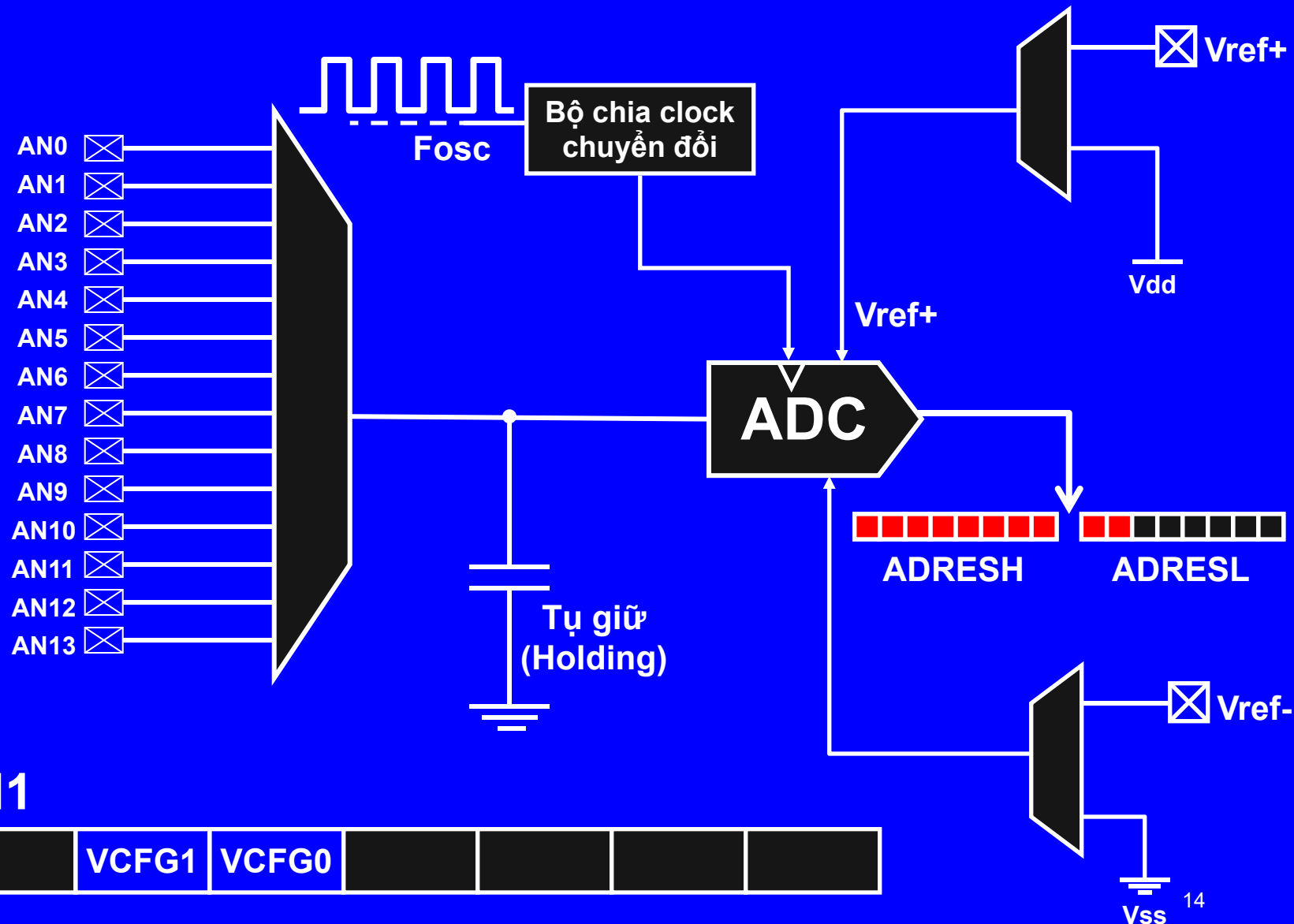
# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



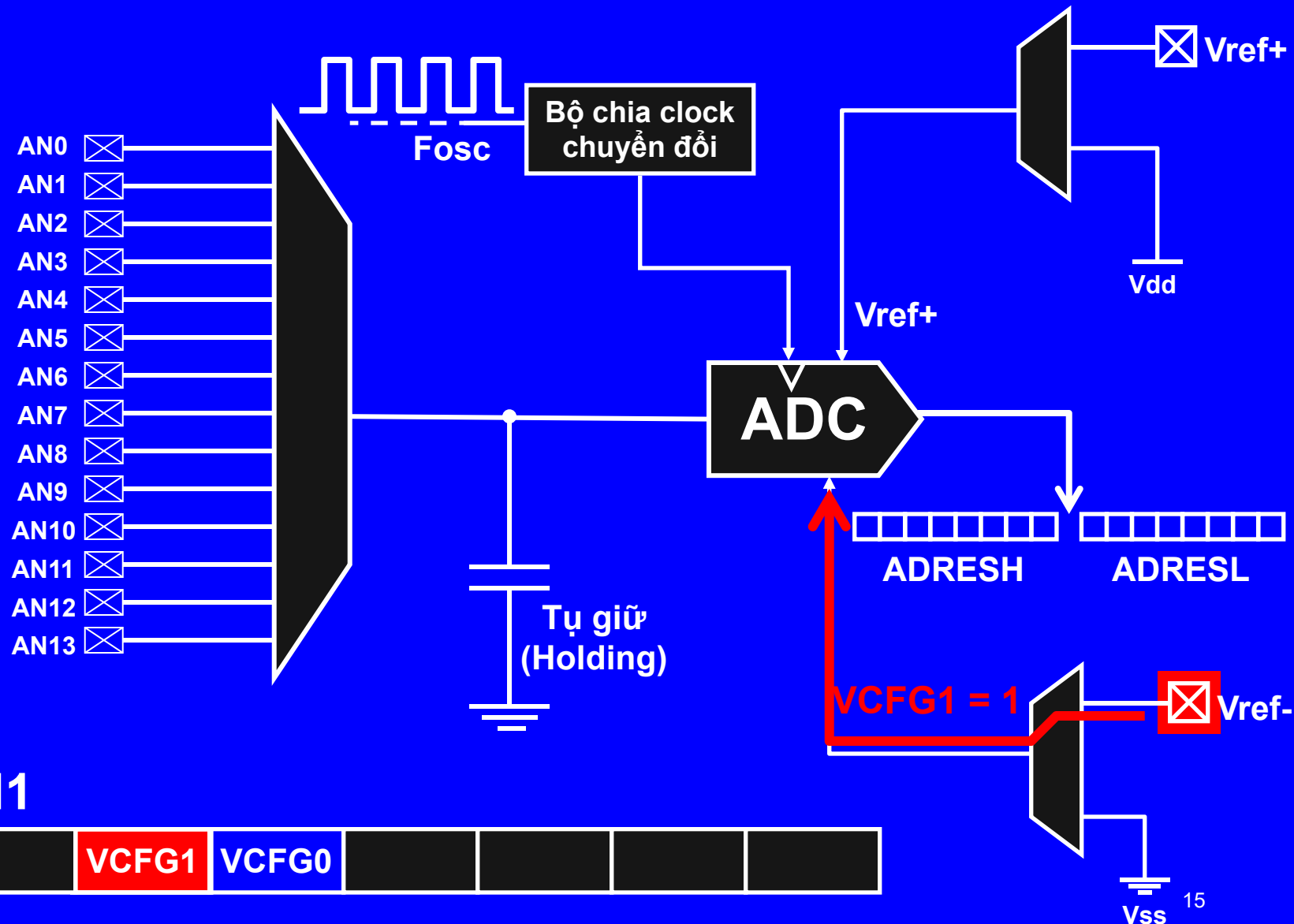
# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



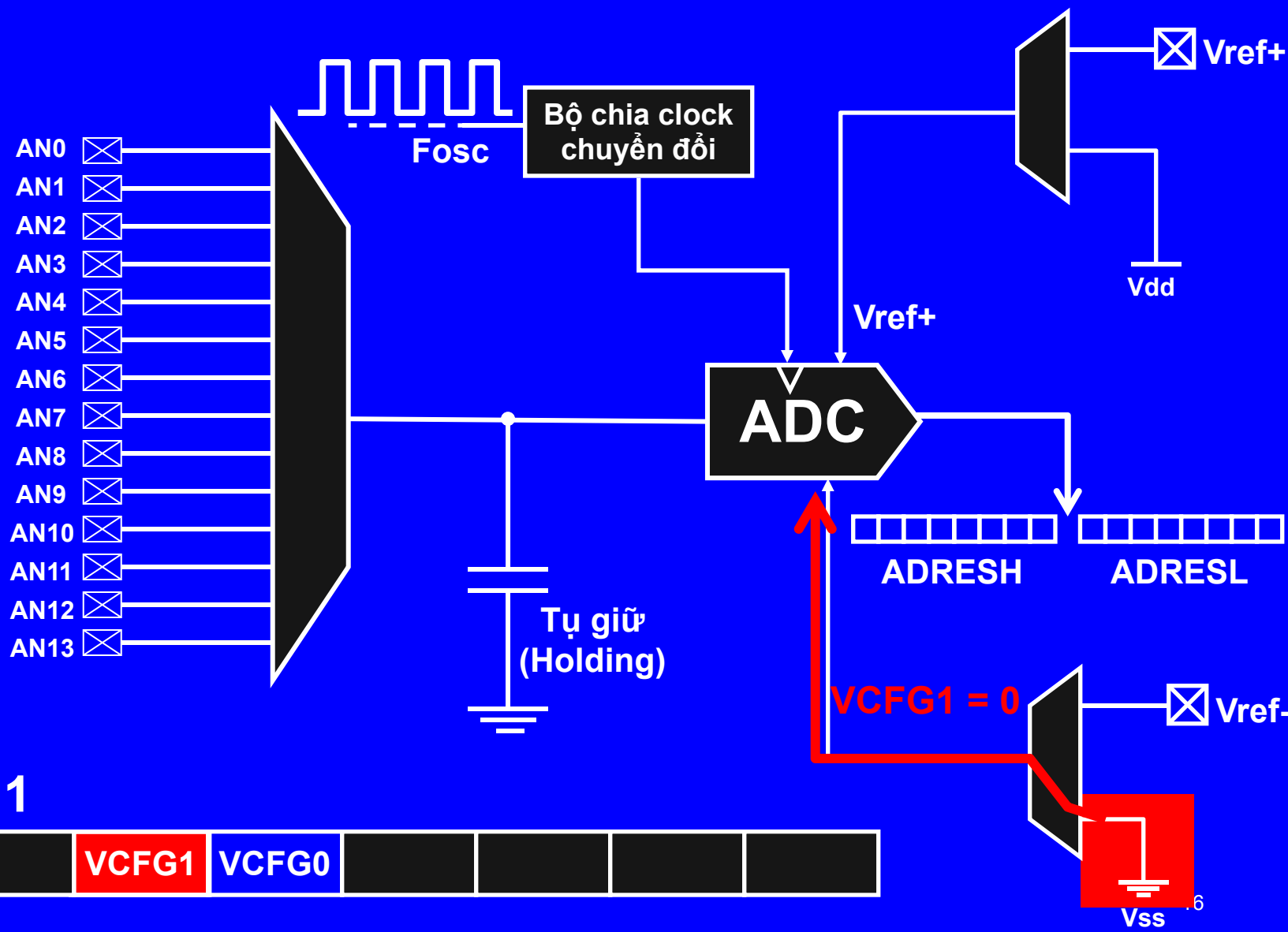
# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC

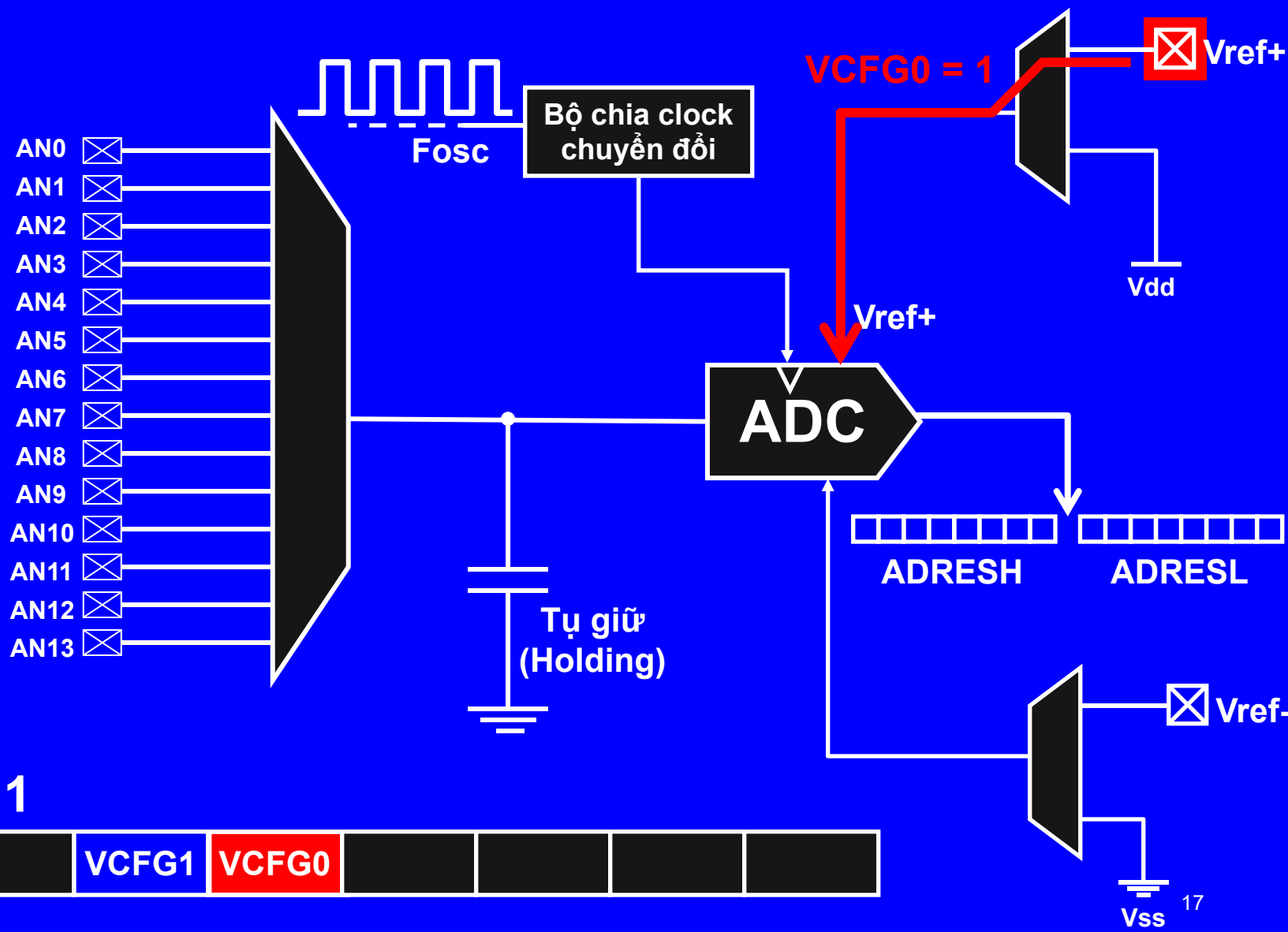


**ADCON1**

ADFM		VCFG1	VCFG0				
------	--	-------	-------	--	--	--	--



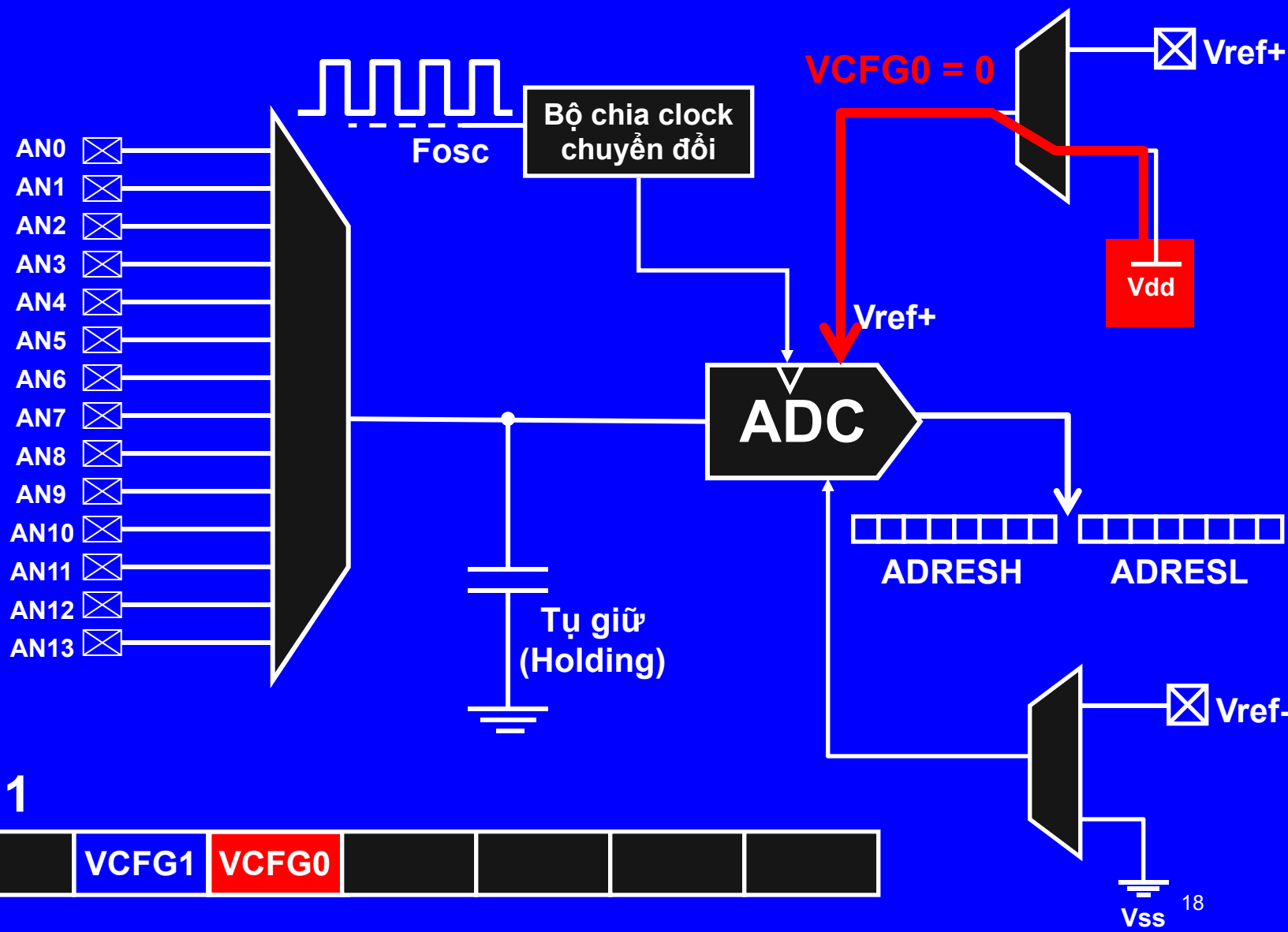
# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



**ADCON1**

ADFM		VCFG1	VCFG0				
------	--	-------	-------	--	--	--	--

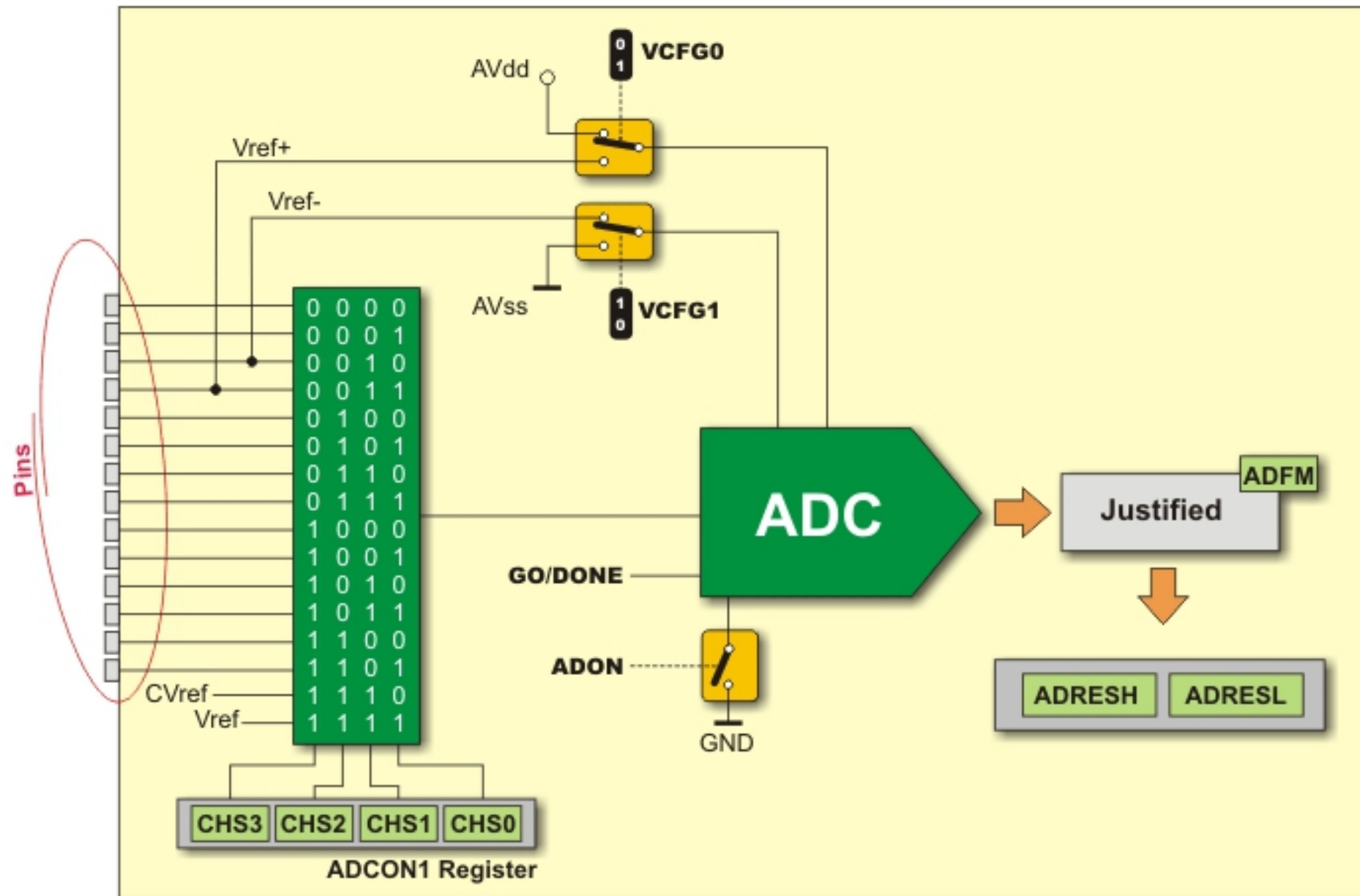
# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



ADCON1

ADFM		VCFG1	VCFG0				
------	--	-------	-------	--	--	--	--

# SƠ ĐỒ KHỐI ĐƠN GIẢN MODUL ADC



# VẤN ĐỀ THỜI GIAN TRONG ADC

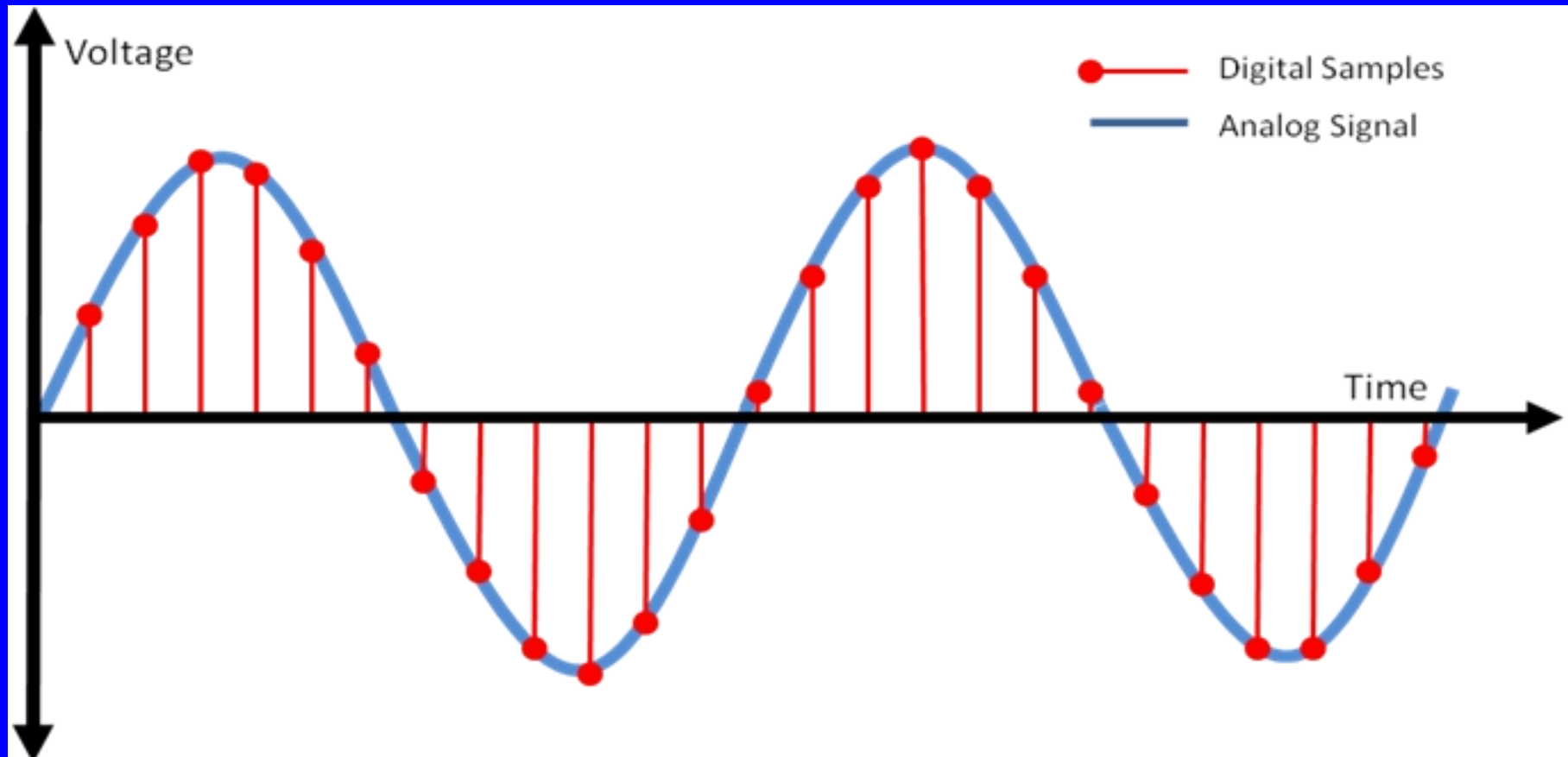
- Khi một kênh AD được chọn, cần phải mất một khoảng thời gian để tụ giữ (Holding Capacitor) nạp điện
- Để hoàn tất quá trình chuyển đổi AD 10 bit thì cần  $11.T_{AD}$  (ADC Clock Period)

NGUỒN XUNG CLOCK ADC	ADCS1	ADCS0	TẦN SỐ HOẠT ĐỘNG CỦA PIC ( $F_{osc}$ )			
			20 MHz	8 MHz	4 MHz	1 MHz
$F_{osc}/2$	0	0	100 ns	250 ns	500 ns	2 $\mu s$
$F_{osc}/8$	0	1	400 ns	1 $\mu s$	2 $\mu s$	8 $\mu s$
$F_{osc}/32$	1	0	1.6 $\mu s$	4 $\mu s$	8 $\mu s$	32 $\mu s$
$F_{RC}$	1	1	2 - 6 $\mu s$	2 - 6 $\mu s$	2 - 6 $\mu s$	2 - 6 $\mu s$

Quan hệ giữa chu kỳ clock ADC ( $T_{AD}$ ) với tần số hoạt động của PIC.

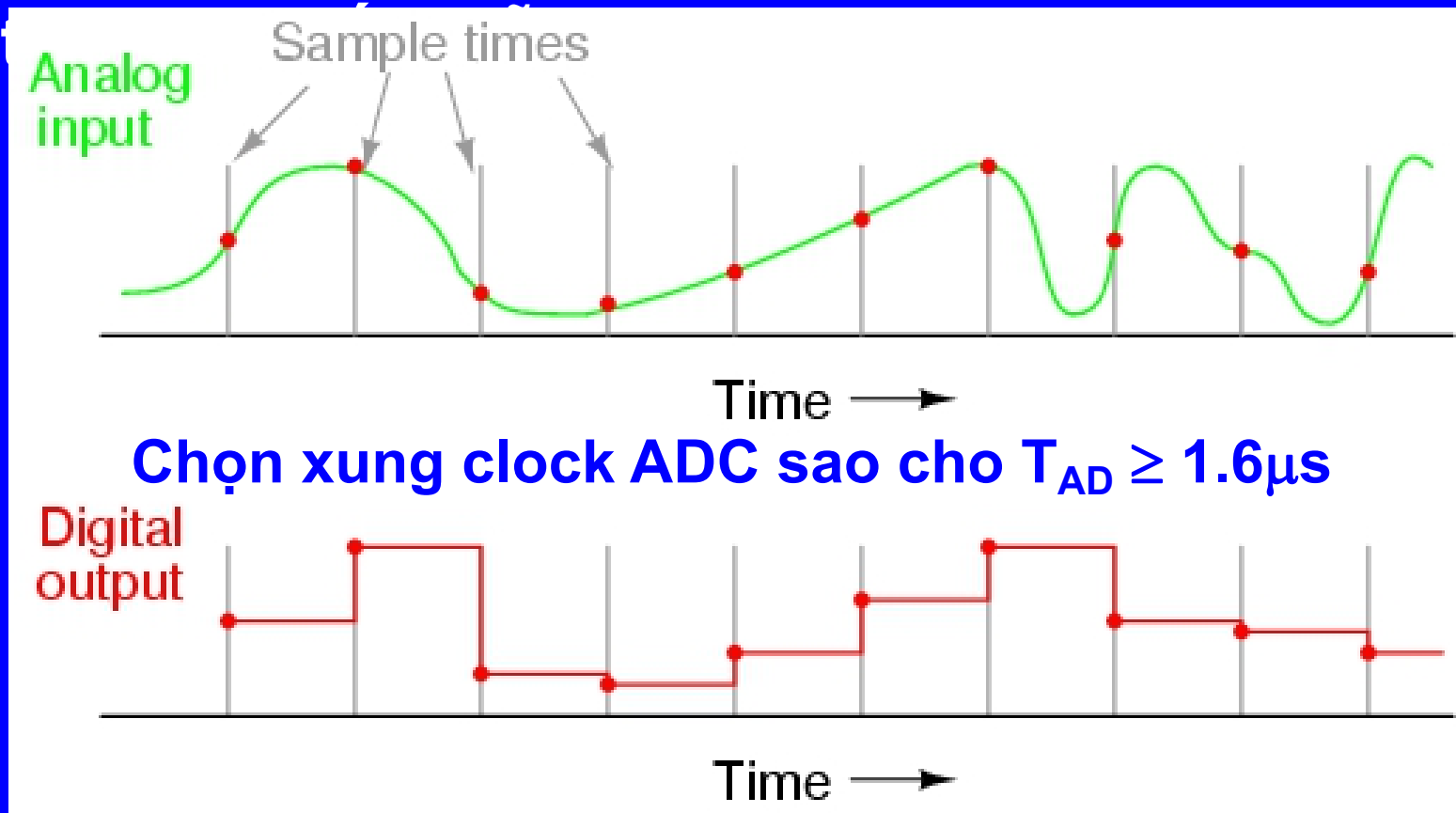
# VẤN ĐỀ THỜI GIAN TRONG ADC

- Cần phải chọn xung clock ADC phù hợp cho từng trường hợp tín hiệu tương tự đưa vào
  - Xem xét giữa chu kỳ của tín hiệu tương tự và



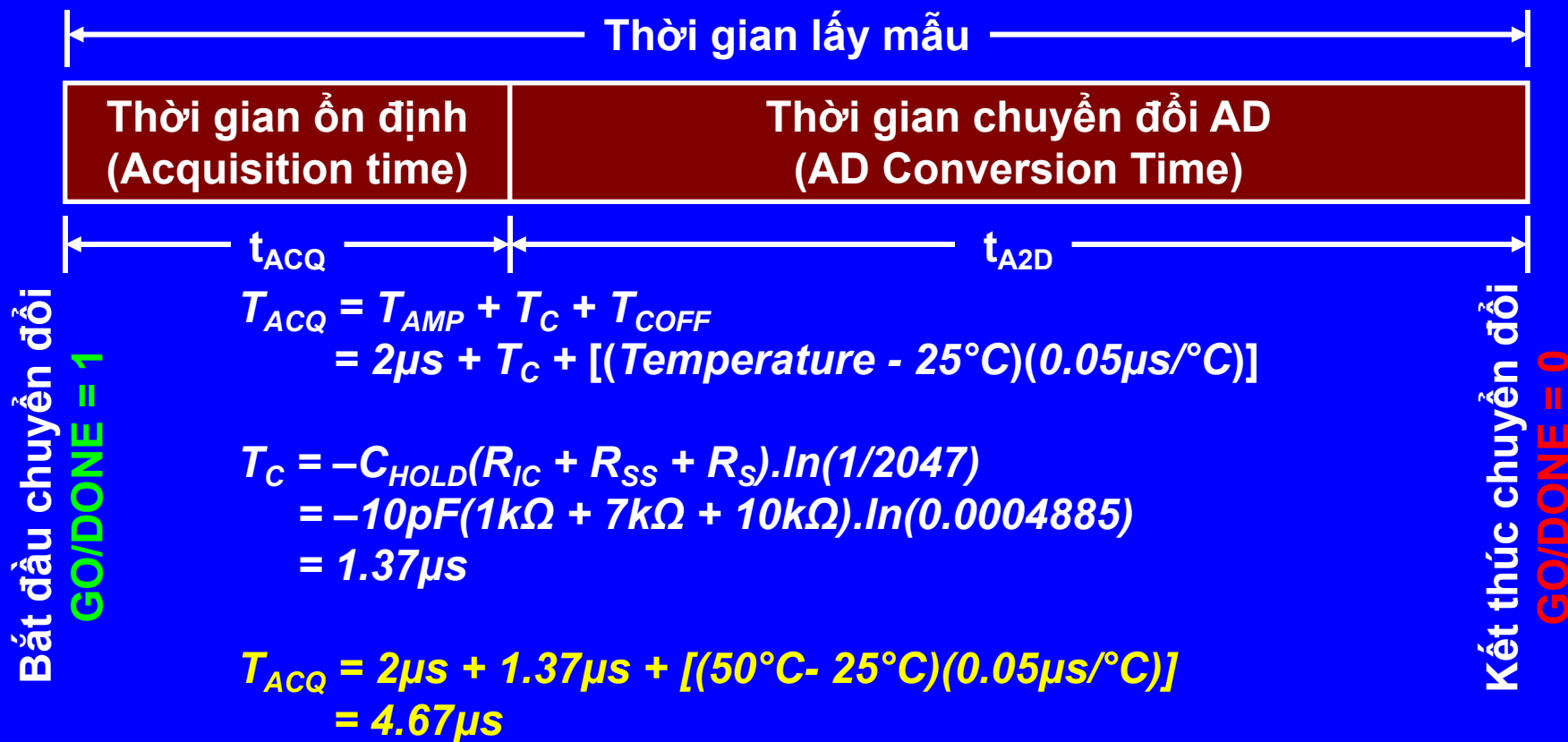
# VẤN ĐỀ THỜI GIAN TRONG ADC

- Cần phải chọn xung clock ADC phù hợp cho từng trường hợp tín hiệu tương tự đưa vào
  - Xem xét giữa chu kỳ của tín hiệu tương tự và



# VẤN ĐỀ THỜI GIAN TRONG ADC

## ➤ Thời gian lấy mẫu AD (AD Sample Time)



(Chi tiết xem thêm trong datasheet của PIC16F887)

$$t_{ACQ} > 5\mu s \quad t_{A2D} = 11.T_{AD}$$

(Với  $t = 50^{\circ}C$ ,  $R_S = 10K\Omega$ ,  $V_{DD} = 5.0V$ )

# MỘT SỐ ĐIỂM CẦN LƯU Ý

- Khi cấu hình sử dụng modul ADC cần phải:
  - Cấu hình port
  - Chọn kênh ADC
  - Chọn điện áp tham chiếu ADC
  - Chọn nguồn xung clock chuyển đổi ADC
  - Điều khiển ngắt
  - Định dạng kết quả.

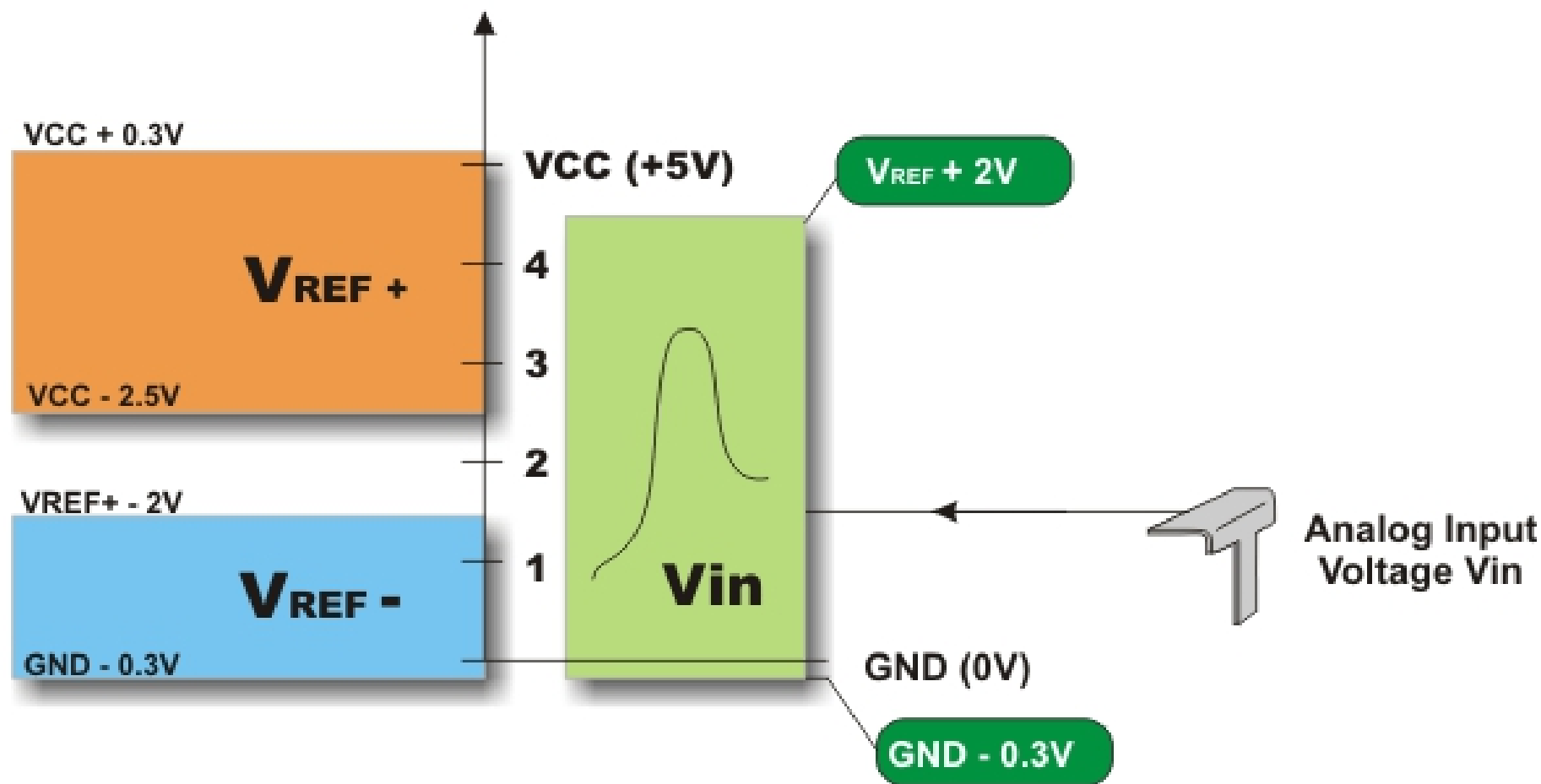


# MỘT SỐ ĐIỂM CẦN LƯU Ý

- Bắt đầu quá trình chuyển đổi AD:
  - Đặt bit GO/DONE bởi người sử dụng (**cần lưu ý rằng bit này không nên được cùng thời điểm với việc kích hoạt modul ADC**)
- Kết thúc quá trình chuyển đổi:
  - Bit GO/DONE được xóa bởi modul ADC
  - Cờ ADIF được đặt bởi modul ADC
  - Giá trị ADRESH:ADRESL được cập nhật bởi modul ADC
- Hủy quá trình chuyển đổi:
  - Xóa bit GO/DONE bởi người sử dụng.

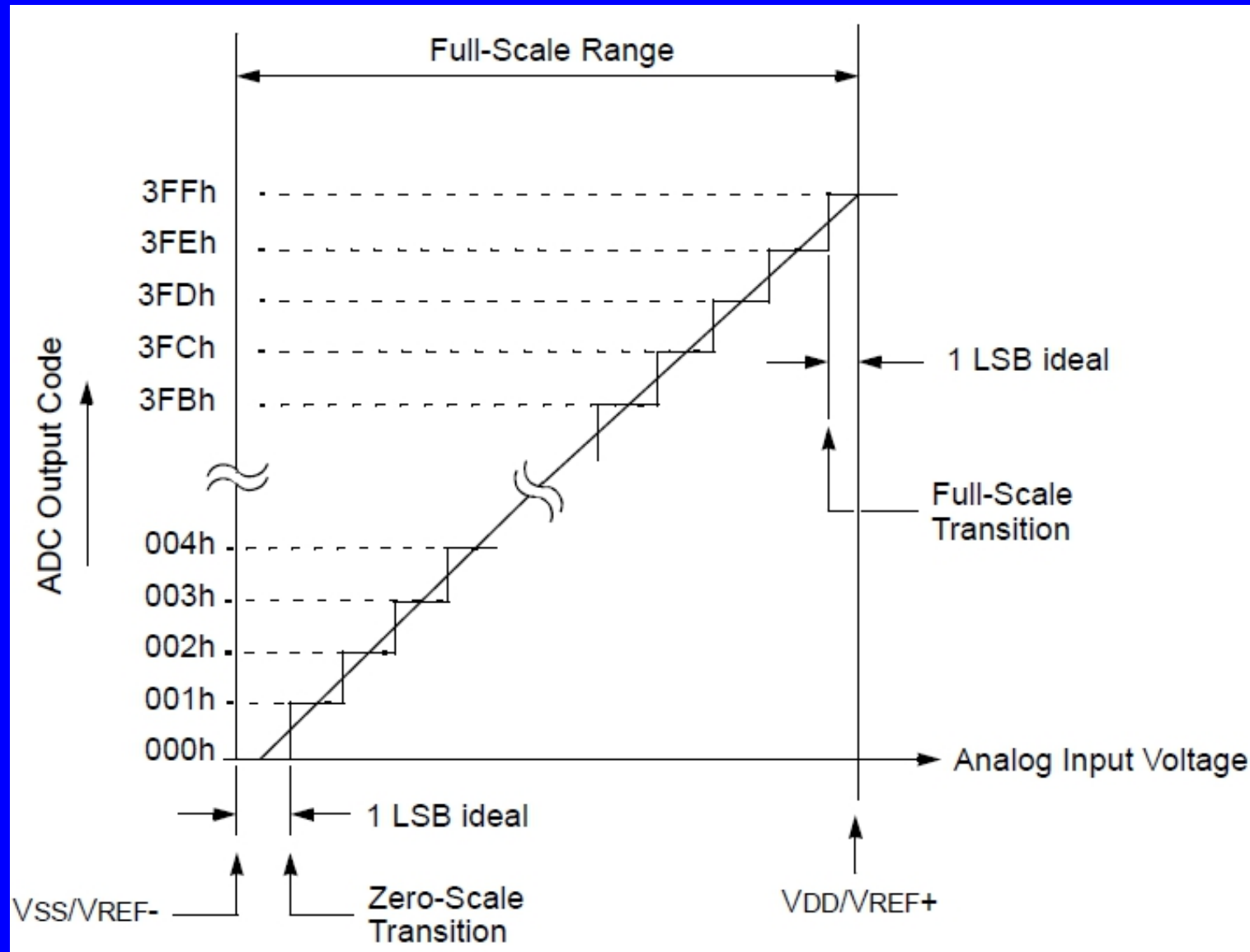
# MỘT SỐ ĐIỂM CẦN LƯU Ý

- Giới hạn tốt nhất cho  $V_{REF+}$  và  $V_{REF-}$  khi chọn từ nguồn bên ngoài



# MỘT SỐ ĐIỂM CẦN LƯU Ý

## ➤ Hàm truyền ADC (ADC Transfer Function)



# MỘT SỐ ĐIỂM CẦN LƯU Ý

- Công thức quan hệ giữa điện áp tương tự ngõ vào và giá trị số sau khi chuyển đổi

$$DV = \frac{(V_{IN} - V_{REF(-)}) \times (2^n - 1)}{V_{REF(+)} - V_{REF(-)}}$$

Trong đó:

**DV:** Giá trị số sau khi chuyển đổi

**VIN:** Điện áp tương tự ngõ vào

**VREF(+), VREF(-):** Điện áp tham chiếu

**n:** Số bit của ADC.

# MỘT SỐ ĐIỂM CẦN LƯU Ý

## ➤ Thủ tục (Procedure) chuyển đổi AD:

- **BƯỚC 1:** Cấu hình Port
  - Cấu hình chân PIC là ngõ vào (**TRISx**)
  - Cấu hình chân PIC là tương tự (**ANSELH, ANSEL**)
- **BƯỚC 2:** Cấu hình modul ADC
  - Chọn xung clock chuyển đổi ADC (**ADCSx**)
  - Chọn điện áp tham chiếu (**VCFGx**)
  - Chọn kênh vào ADC (**CHSx**)
  - Chọn định dạng kết quả (**ADFM**)
  - Bật modul ADC (**ADON**).

# MỘT SỐ ĐIỂM CẦN LƯU Ý

- Thủ tục (Procedure) chuyển đổi AD:
  - **BƯỚC 3:** Cấu hình ngắt ADC (tùy chọn)
    - Xóa cờ ngắt ADC (**ADIF**)
    - Cho phép ngắt ADC (**ADIE**)
    - Cho phép ngắt ngoại vi (**PEIE**)
    - Cho phép ngắt toàn cục (**GIE**)
  - **BƯỚC 4:** Chờ một khoảng thời gian (khoảng  $20\mu s$ ) để modul ADC ổn định ( $t_{ACQ}$ )
  - **BƯỚC 5:** Đặt bit GO/DONE để bắt đầu quá trình chuyển đổi.

# MỘT SỐ ĐIỂM CẦN LƯU Ý

- **Thủ tục (Procedure) chuyển đổi AD:**
  - **BƯỚC 6:** Chờ quá trình chuyển đổi AD hoàn tất bằng một trong các cách sau:
    - Kiểm tra bit GO/DONE ( $GO/DONE = 0$ ) → **Polling**
    - Chờ ngắt ADC (nếu ngắt được cho phép) → **Interrupt**
  - **BƯỚC 7:** Đọc kết quả ADC
  - **BƯỚC 8:** Xóa cờ ngắt ADC (bỏ qua bước này nếu ngắt không được cho phép).

# MINH HỌA KHỞI ĐỘNG ADC

```

banksel TRISA      ; RA0 là ngõ vào
bsf      TRISA,RA0  ; tương tự.
banksel ANSEL
bsf      ANSEL,AN0
banksel ADCON1      ; Dữ liệu canh phải,
movlw    b'10000000' ; Vref+ = Vdd,
movwf    ADCON1      ; Vref- = Vss.
banksel ADCON0      ; Xung clock chuyển
movlw    b'11000001' ; đổi ADC: FRC, kênh
movwf    ADCON0      ; AN0 và bật ADC.
call     ACQTime      ; TACQ
bsf      ADCON0,GO    ; Bắt đầu chuyển đổi
btfsc    ADCON0,GO    ; Hoàn tất ?
goto     $-1
banksel ADRESH
movf     ADRESH,W     ; Đọc 2 bit cao và lưu
movwf    RESULTHI     ; trữ vào GPR
banksel ADRESL
movf     ADRESL,W     ; Đọc 8 bit thấp và
movwf    RESULTLO     ; lưu trữ vào GPR
    
```

TRISA

							1
--	--	--	--	--	--	--	---

ANSEL

							1
--	--	--	--	--	--	--	---

ADCON1

1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

ADFM VCFG<1:0> Chọn điện áp  
Định dạng dữ liệu tham chiếu

ADCON0

1	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

ADCS,1:2> Chọn xung  
clock chuyển đổi

CHS<3:0> Chọn kênh  
ngõ vào

ADON Bật ADC



# MINH HỌA KHỞI ĐỘNG ADC

```

banksel TRISA      ; RA0 là ngõ vào
bsf     TRISA,RA0   ; tương tự.
banksel ANSEL
bsf     ANSEL,AN0
banksel ADCON1      ; Dữ liệu canh phải,
movlw   b'10000000' ; Vref+ = Vdd,
movwf   ADCON1      ; Vref- = Vss.
banksel ADCON0      ; Xung clock chuyển
movlw   b'11000001' ; đổi ADC: FRC, kênh
movwf   ADCON0      ; AN0 và bật ADC.
call    SampleTime  ; TAcq
bsf     ADCON0,GO  ; Bắt đầu chuyển đổi
btfsc   ADCON0,GO ; Hoàn tất ?
goto    $-1
banksel ADRESH
movf    ADRESH,W    ; Đọc 2 bit cao và lưu
movwf   RESULTHI    ; trữ vào GPR
banksel ADRESL
movf    ADRESL,W    ; Đọc 8 bit thấp và
movwf   RESULTLO    ; lưu trữ vào GPR
    
```

TRISA

							1
--	--	--	--	--	--	--	---

ANSEL

							1
--	--	--	--	--	--	--	---

ADCON1

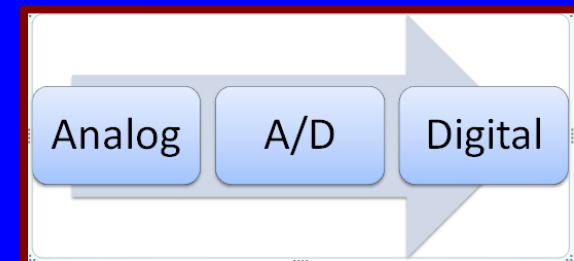
1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

ADFM VCFG<1:0> **Chọn điện áp tham chiếu**  
**Định dạng dữ liệu**

ADCON0

1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---

ADCS,1:2> **Chọn xung clock chuyển đổi**  
 CHS<3:0> **Chọn kênh ngõ vào**  
 ADON **Bật ADC**  
**Bắt đầu**



# MINH HỌA KHỞI ĐỘNG ADC

```

banksel TRISA      ; RA0 là ngõ vào
bsf     TRISA,RA0   ; tương tự.
banksel ANSEL
bsf     ANSEL,AN0
banksel ADCON1      ; Dữ liệu canh phải,
movlw   b'10000000' ; Vref+ = Vdd,
movwf   ADCON1      ; Vref- = Vss.
banksel ADCON0      ; Xung clock chuyển
movlw   b'11000001' ; đổi ADC: FRC, kênh
movwf   ADCON0      ; AN0 và bật ADC.
call    SampleTime  ; TAcq
bsf     ADCON0,GO    ; Bắt đầu chuyển đổi
btfscl ADCON0,GO   ; Hoàn tất ?
goto   $-1
banksel ADRESH
movf   ADRESH,W    ; Đọc 2 bit cao và lưu
movwf  RESULTHI    ; trữ vào GPR
banksel ADRESL
movf   ADRESL,W    ; Đọc 8 bit thấp và
movwf  RESULTLO    ; lưu trữ vào GPR
    
```

TRISA

							1
--	--	--	--	--	--	--	---

ANSEL

							1
--	--	--	--	--	--	--	---

ADCON1

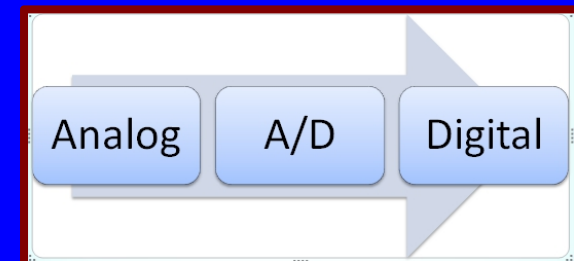
1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

ADFM VCFG<1:0> **Chọn điện áp tham chiếu**  
**Định dạng dữ liệu**

ADCON0

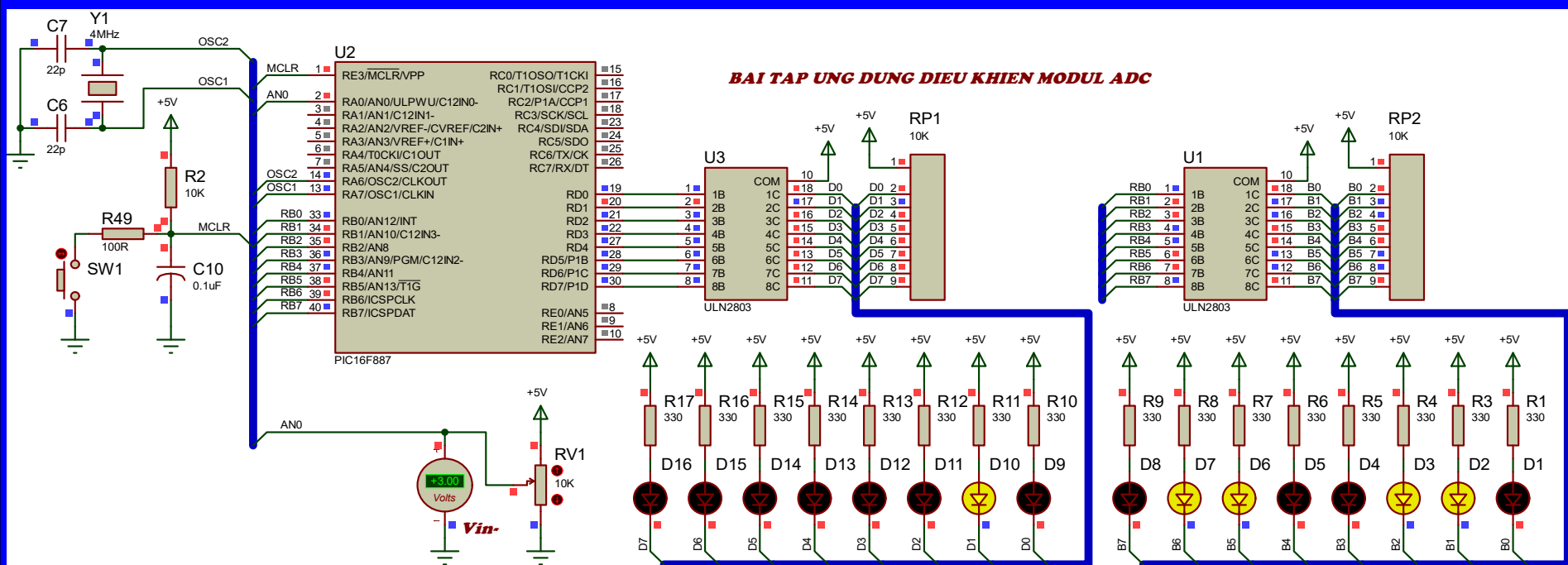
1	1	0	0	0	0	0	1
---	---	---	---	---	---	---	---

ADCS,1:2> **Chọn xung clock chuyển đổi**  
 CHS<3:0> **Chọn kênh ngõ vào**  
 ADON **Bật ADC**  
**Kết thúc**



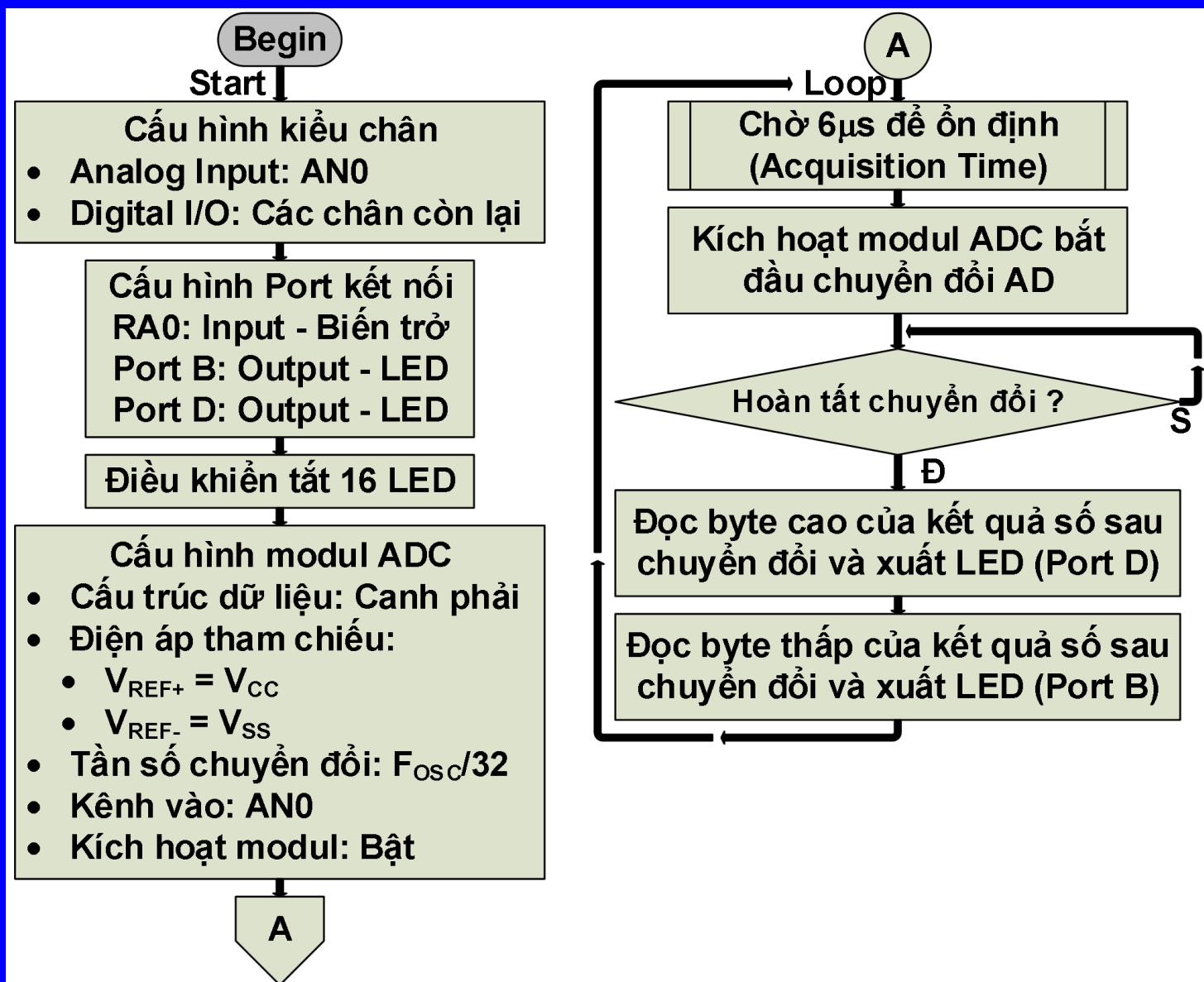
## VÍ DỤ MINH HỌA

- **Ví dụ 1:** Dựa vào sơ đồ, viết chương trình điều khiển đọc giá trị điện áp (Analog) từ biến trở, chuyển đổi thành giá trị số nhị phân 10 bit (Digital) và hiển thị lên LED. Sử dụng modul ADC.
  - **Sơ đồ nguyên lý:**



# VÍ DỤ MINH HỌA

## • Giải thuật:



# VÍ DỤ MINH HỌA

- **Cấu hình (Hi-Tech C):**

```
__CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

- Chương trình (Hi-Tech C):

```
void main(void)
```

```
{  
  ANSEL = 0x01;  
  ANSELH = 0;
```

```
  
  TRISA0 = 1;  
  TRISB = 0;  
  TRISD = 0;
```

```
  
  PORTD = 0;  
  PORTB = 0;
```

```
  
  ADCON1 = 0b10000000;
```

```
  
  ADCON0 = 0b10000001;
```

```
while(1)
```

```
{  
  __delay_us(6);
```

```
  
  GO = 1;  
  while(GO);
```

```
  
  PORTD = ADRESH;  
  PORTB = ADRESL;  
}
```

```
}
```

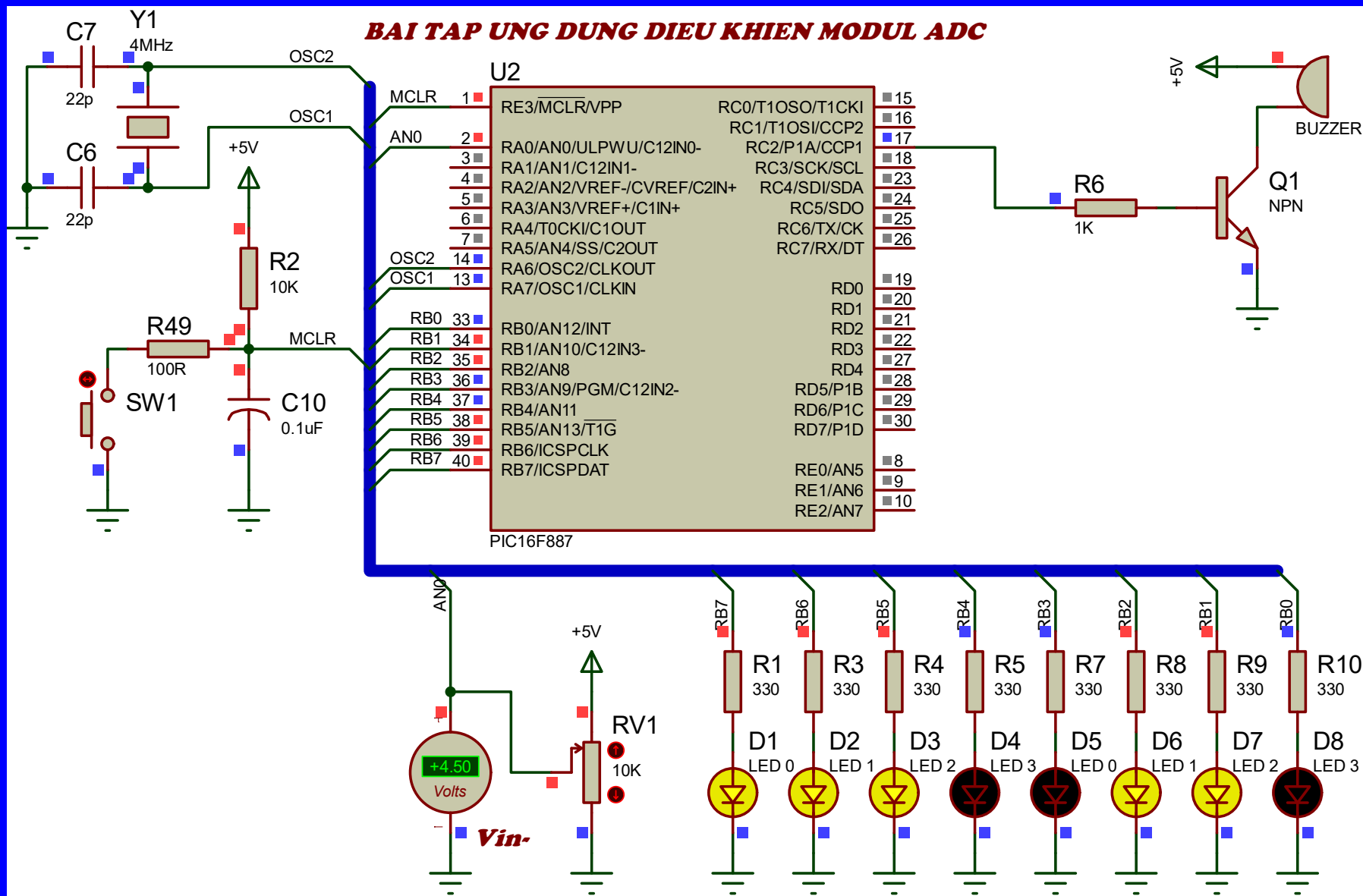
# VÍ DỤ MINH HỌA

- **Ví dụ 2:** Dựa vào sơ đồ, viết chương trình điều khiển đọc giá trị điện áp (Analog) từ biến trở, chuyển đổi thành giá trị số nhị phân 10 bit (Digital). Sử dụng 8 bit cao của giá trị này để điều chỉnh độ rộng xung của xung PWM. Xung PWM sau đó được dùng để tạo âm thanh tại chuông và hiển thị giá trị độ rộng xung lên 8 LED. Sử dụng modul ADC và CCP (PWM)

# VÍ DỤ MINH HỌA

## Sơ đồ nguyên lý:

### BAI TAP UNG DUNG DIEU KHIEN MODUL ADC





# VÍ DỤ MINH HỌA

## • Giải thuật:

Begin

Main

Cấu hình kiểu chân

- Analog Input: AN0
- Digital I/O: Các chân còn lại

Cấu hình Port kết nối  
RA0: Input - Biến trở  
Port B: Output - LED

Cắm xuất PWM  
CCP1 (RC2): Input

Cấu hình modul ADC

- Cấu trúc dữ liệu: Canh trái
- Điện áp tham chiếu:
  - $V_{REF+} = V_{CC}$
  - $V_{REF-} = V_{SS}$
- Tần số chuyển đổi:  $F_{osc}/32$
- Kênh vào: AN0
- Kích hoạt modul: Bật

A

A

Cấu hình modul CCP1

- Mode: PWM
- Chu kỳ:  $250\mu S$  ( $f = 4KHz$ )
- Độ rộng: 50% ( $125\mu S$ )
- CCP1  $\rightarrow$  ON

Cho phép xuất PWM  
CCP1 (RC2): Output

Cấu hình modul Timer 2

- Mode: Timer
- Prescaler: 1:1
- Postscaler: 1:1
- Timer 2  $\rightarrow$  ON

B

Loop

Chờ  $6\mu S$  để ổn định  
(Acquisition Time)

Kích hoạt modul ADC bắt  
đầu chuyển đổi AD

Hoàn tất chuyển đổi ?

Đ

Đọc byte cao giá trị ADC và gán  
vào thanh ghi qui định độ rộng  
xung PWM

Xuất giá trị độ rộng xung PWM  
lên LED hiển thị

S

# VÍ DỤ MINH HỌA

- **Cấu hình (Hi-Tech C):**

```
__CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

- Chương trình (Hi-Tech C):

```
void main(void)
```

```
{
```

```
  ANSEL = 0x01;
```

```
  ANSELH = 0;
```

```
  RA0 = 1;
```

```
  RC2 = 0;
```

```
  TRISB = 0;
```

```
  PR2 = 0xFF;
```

```
  CCPR1L = 0x80;
```

```
  CCP1CON = 0x0C;
```

```
  TMR2ON = 1;
```

```
  ADCON0 = 0;
```

```
  ADCS1 = 1;
```

```
  ADON = 1;
```

```
  ADCON1 = 0x0E;
```

```
  while(1)
```

```
  {
```

```
    __delay_us(6);
```

```
    GO = 1;
```

```
    while(GO);
```

```
    CCPR1L = ADRESH;
```

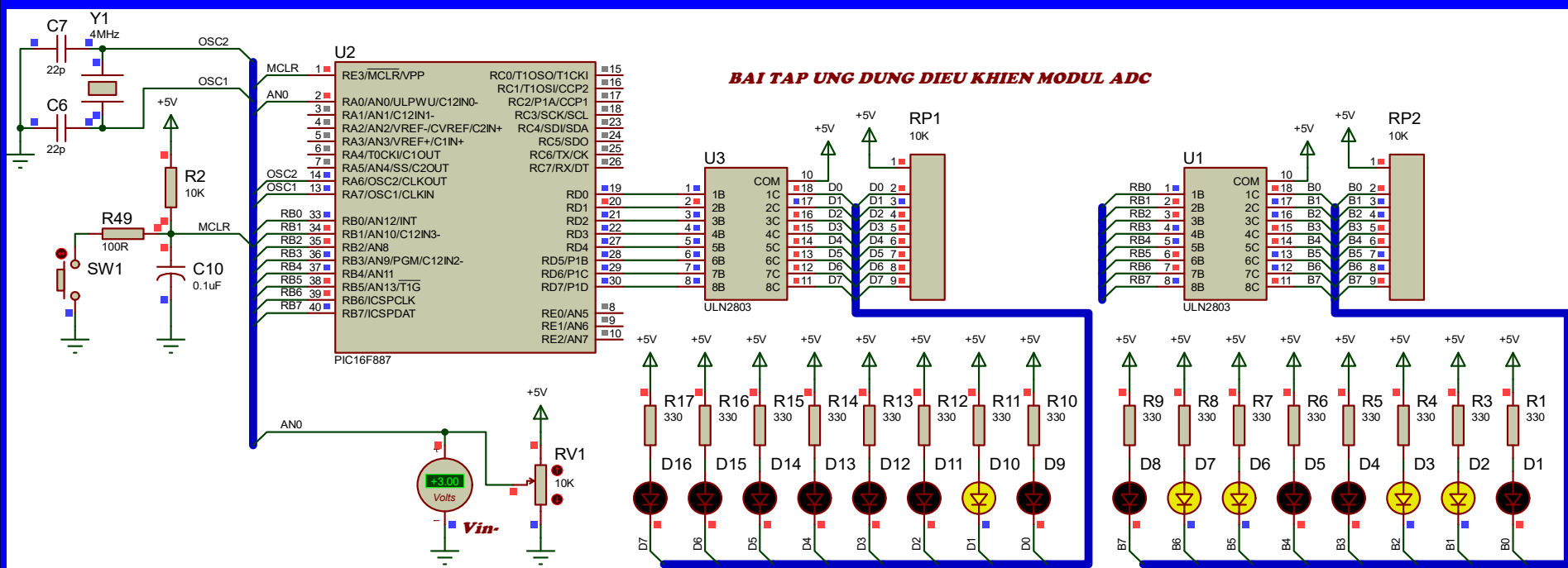
```
    PORTB = CCPR1L;
```

```
  }
```

```
}
```

# BÀI TẬP ỨNG DỤNG

➤ **Bài tập 1:** Dựa vào sơ đồ, viết chương trình điều khiển đọc giá trị điện áp từ biến trở (0V – 5V) và hiển thị lên 16 LED giá trị này theo đơn vị là mV (0 – 5000). Sử dụng modul ADC.



# BÀI TẬP ỨNG DỤNG

- **Bài tập 2:** Dựa vào sơ đồ, viết chương trình điều khiển đọc giá trị nhiệt độ từ cảm biến LM35 ( $10\text{mV} / 1^{\circ}\text{C}$ ) và hiển thị lên 8 LED giá trị này theo đơn vị là  $^{\circ}\text{C}$  (0 – 255). Sử dụng modul ADC.

# BÀI TẬP ỨNG DỤNG

## ➤ Bài tập 2: Dựa vào sơ đồ, viết chương trình điều khiển đọc giá trị nhiệt độ từ cảm biến

