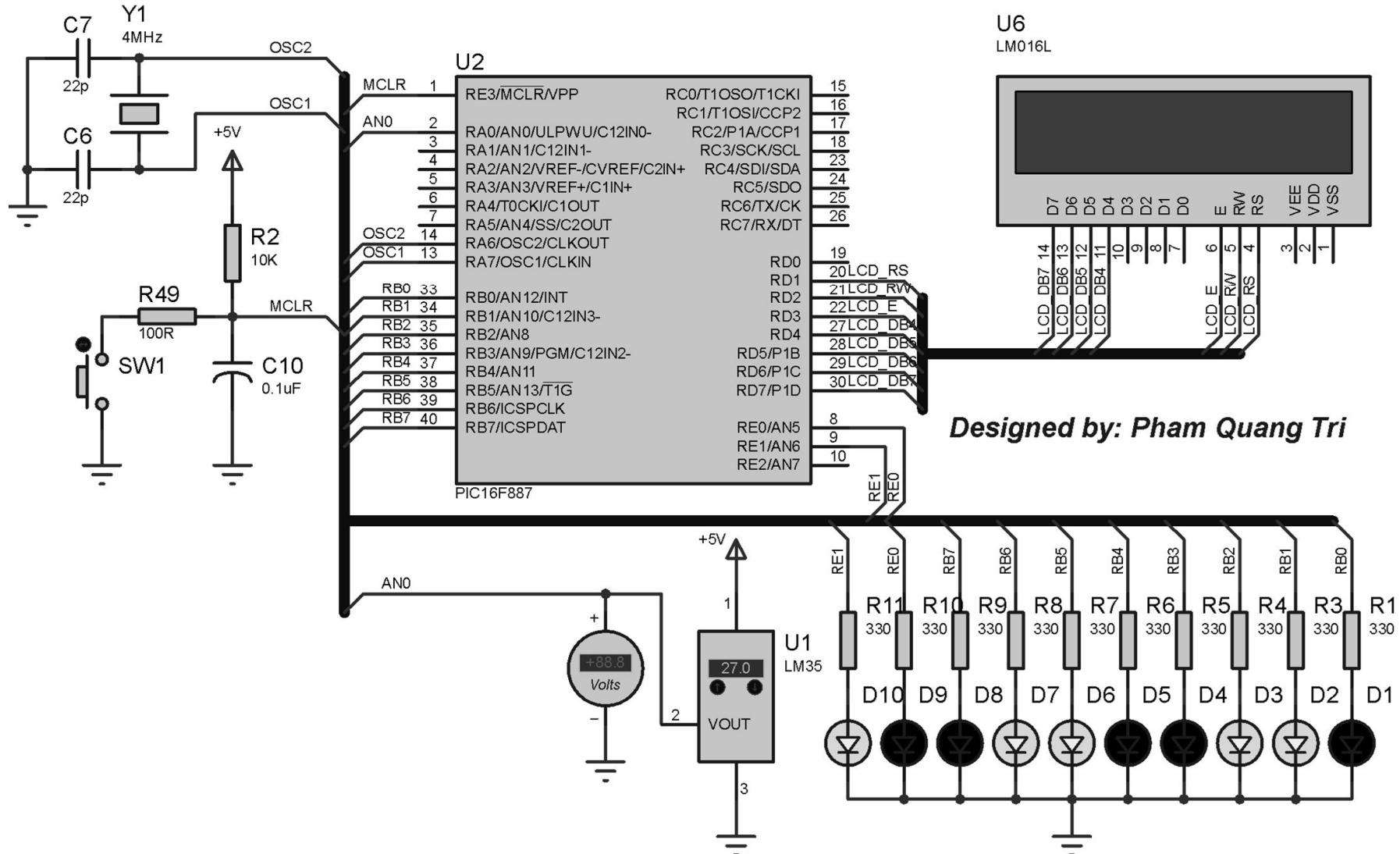


# BÀI TẬP ỨNG DỤNG

## ➤ Bài tập 2:



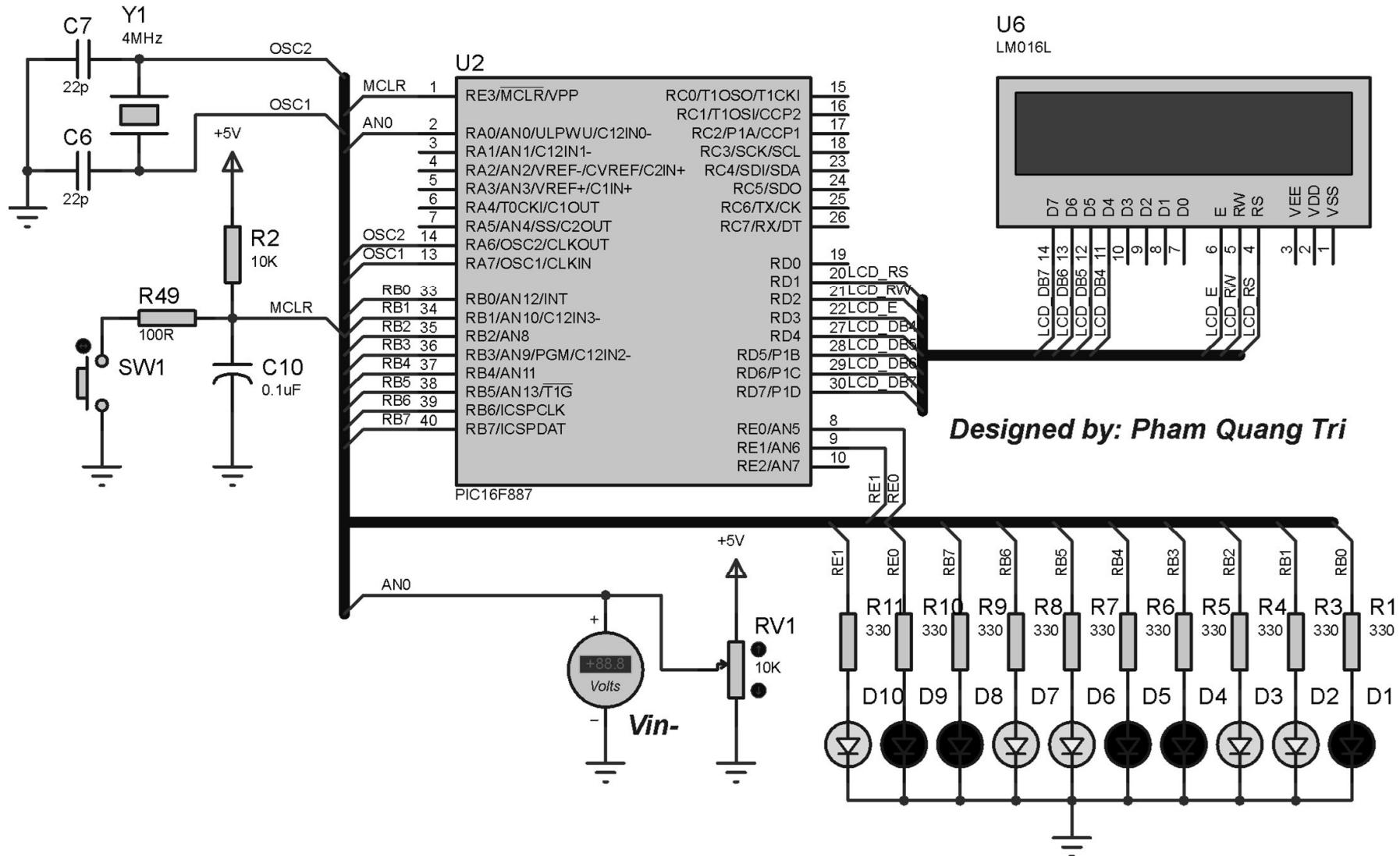
Designed by: Pham Quang Tri

## 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 (10mV / 1°C). Hiển thị giá trị nhiệt độ lên LCD theo đơn vị là °C và hiển thị giá trị số (Digital) sau khi chuyển đổi lên 10 LED. Sử dụng mô-đun ADC.**

# BÀI TẬP ỨNG DỤNG

## ➤ Bài tập 1:



## BÀI TẬP ỨNG DỤNG

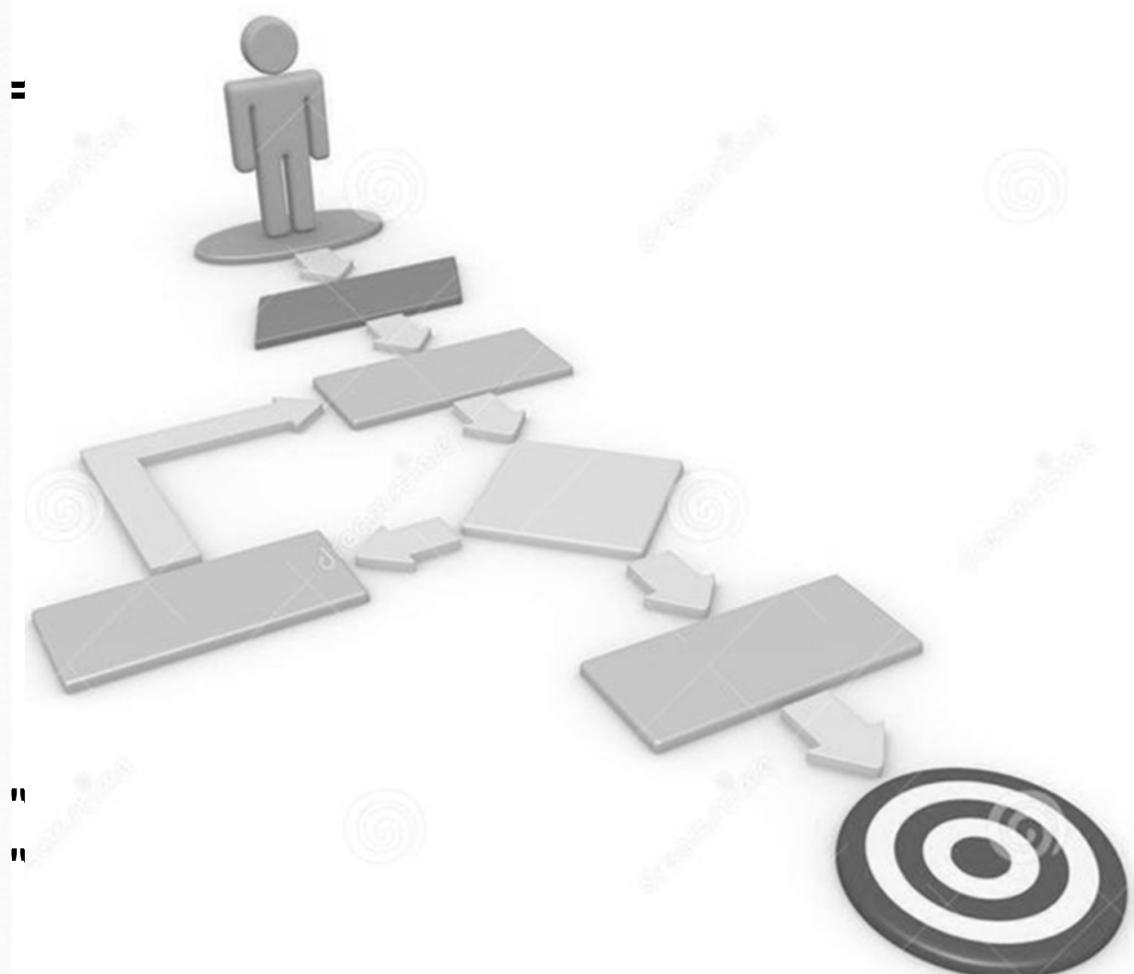
- **Bài tập 1:** Dựa vào sơ đồ, viết chương trình điều khiển đọc giá trị điện áp từ biến trở (0V – 5V). Hiển thị giá trị điện áp lên LCD theo đơn vị là mV và hiển thị giá trị số (Digital) sau khi chuyển đổi lên 10 LED. Sử dụng module ADC.

# VÍ DỤ MINH HỌA

- Chương trình (Hi-Tech C):

Source: C8\_VD2\_CodeC.c

SIM: C8\_VD2\_CodeC.pdsprj



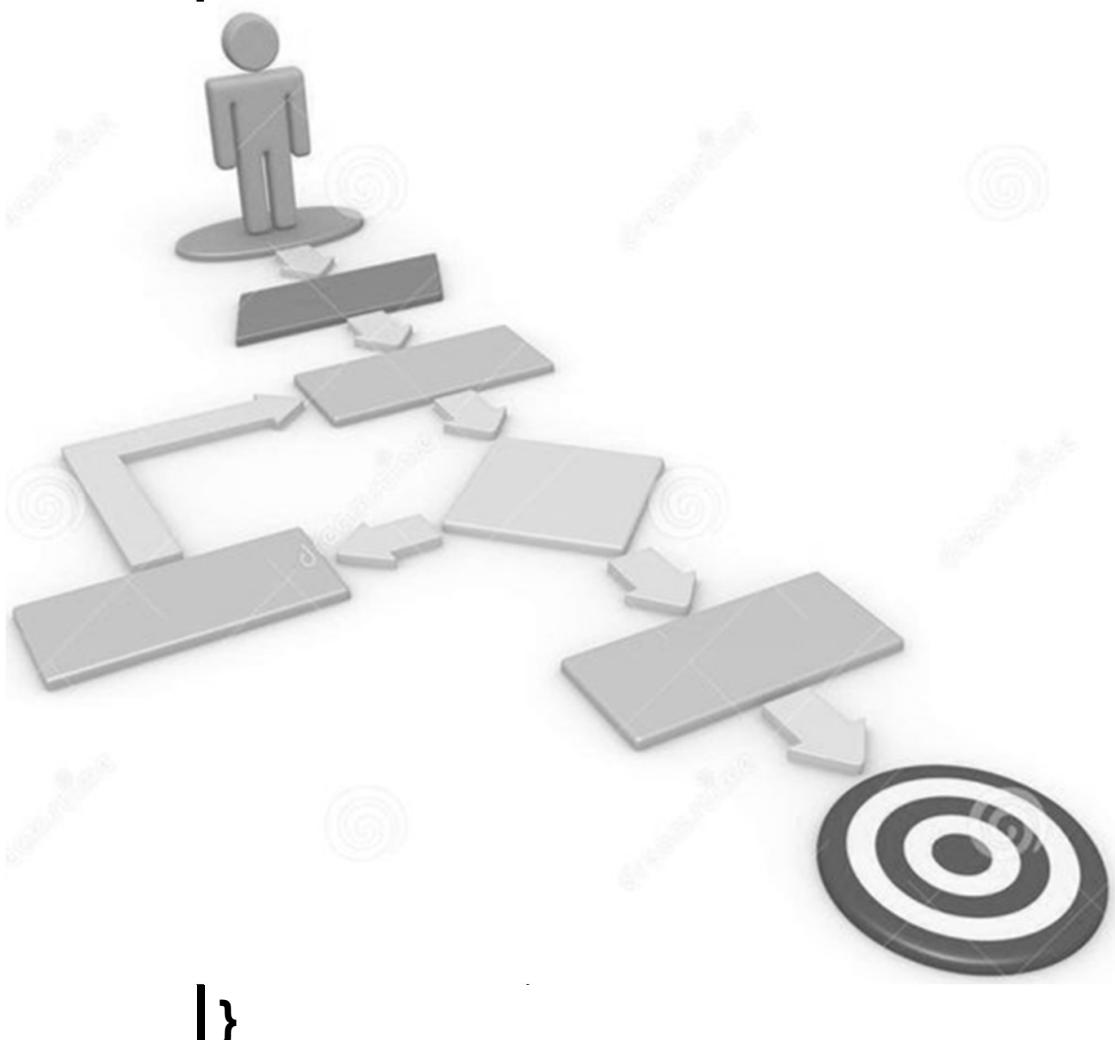
# VÍ DỤ MINH HỌA

Source: C8\_VD2\_CodeC.c

SIM: C8\_VD2\_CodeC.pdsprj

- Chương trình (Hi-Tech C):

```
| void CCP1_init(void)
```

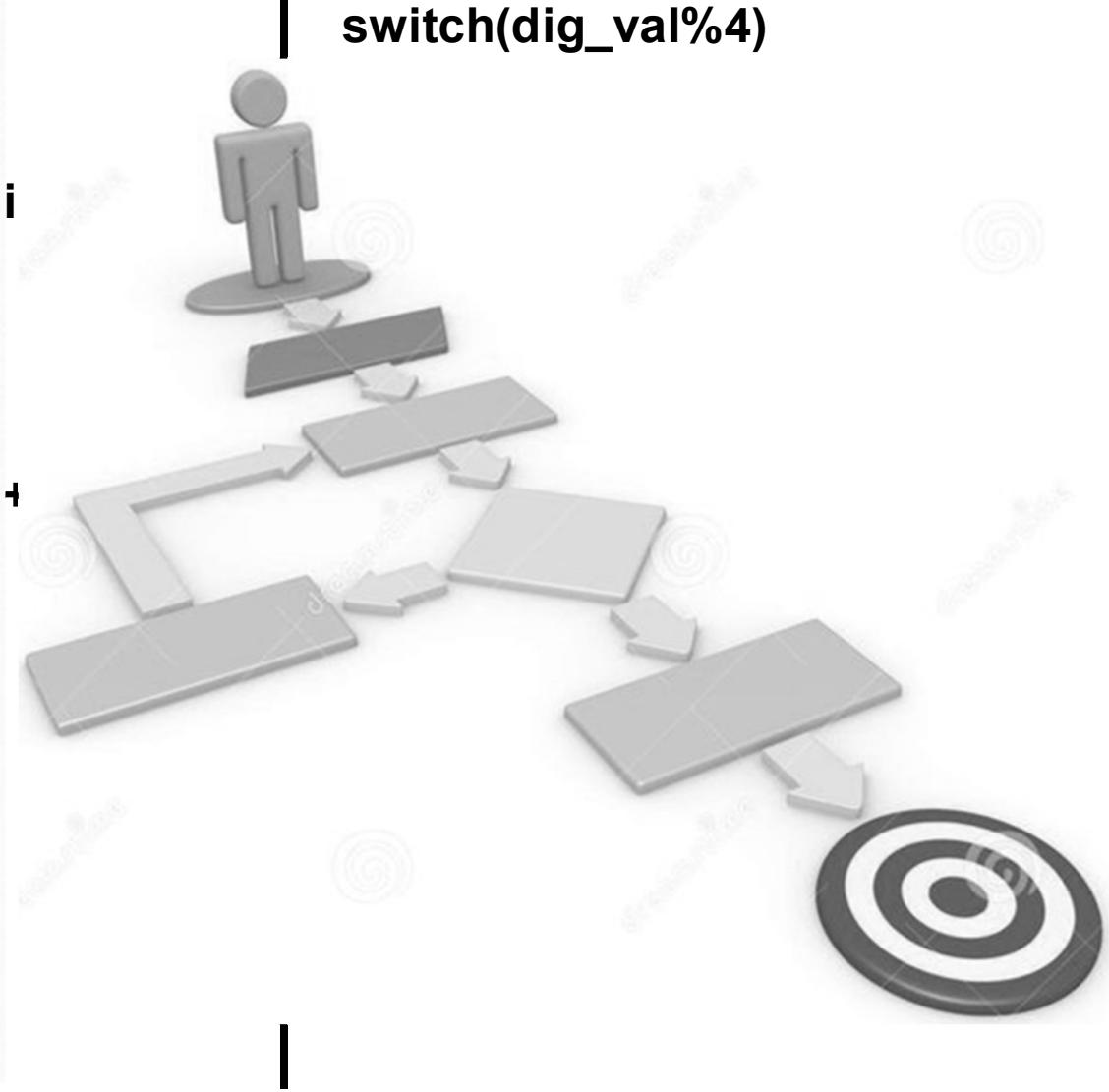


# VÍ DỤ MINH HỌA

Source: C8\_VD2\_CodeC.c

SIM: C8\_VD2\_CodeC.pdsprj

- Chương trình (Hi-Tech C):



# VÍ DỤ MINH HỌA

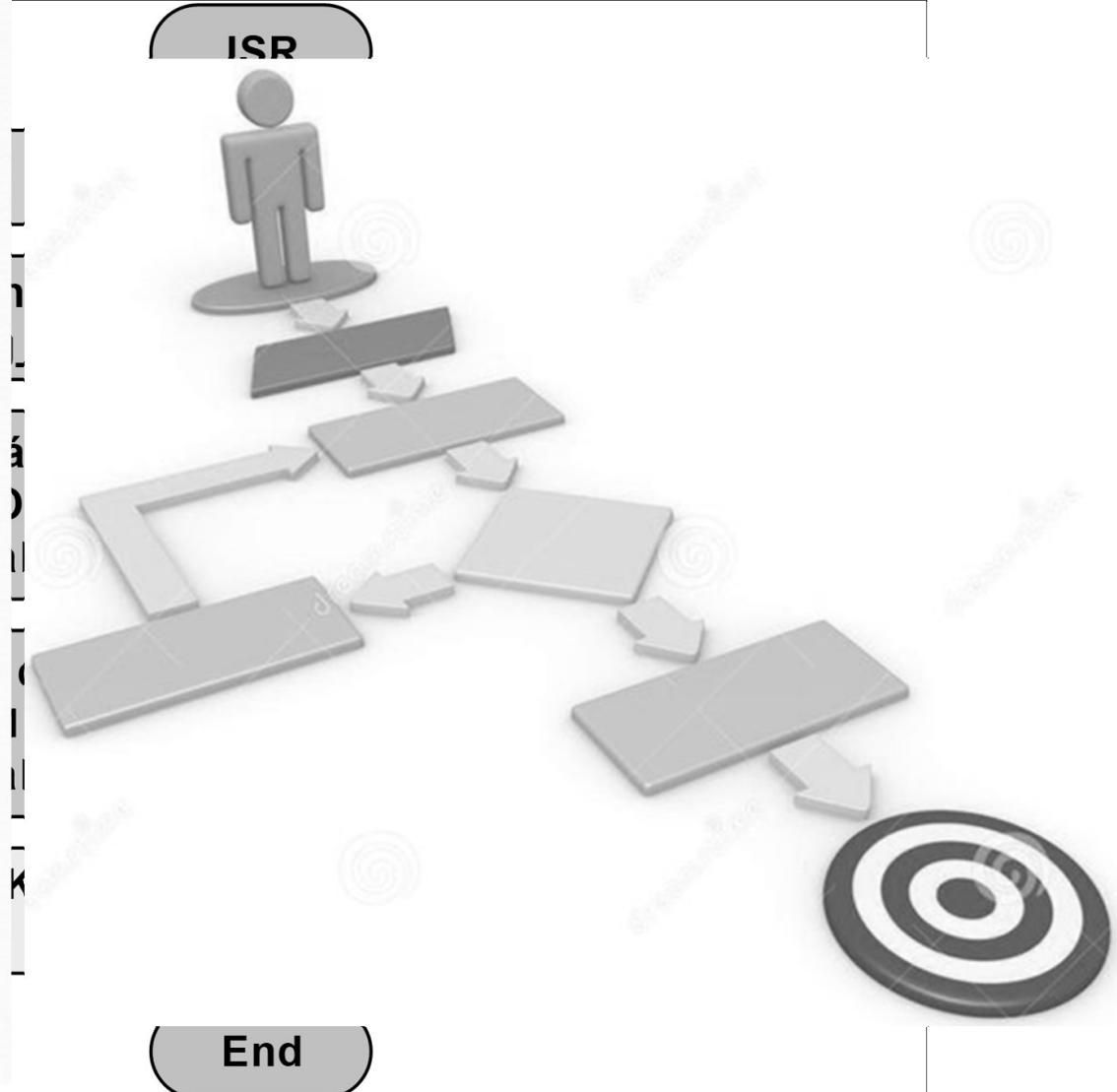
- Chương trình (Hi-Tech C):

Source: C8\_VD2\_CodeC.c

SIM: C8\_VD2\_CodeC.pdsprj

# VÍ DỤ MINH HỌA

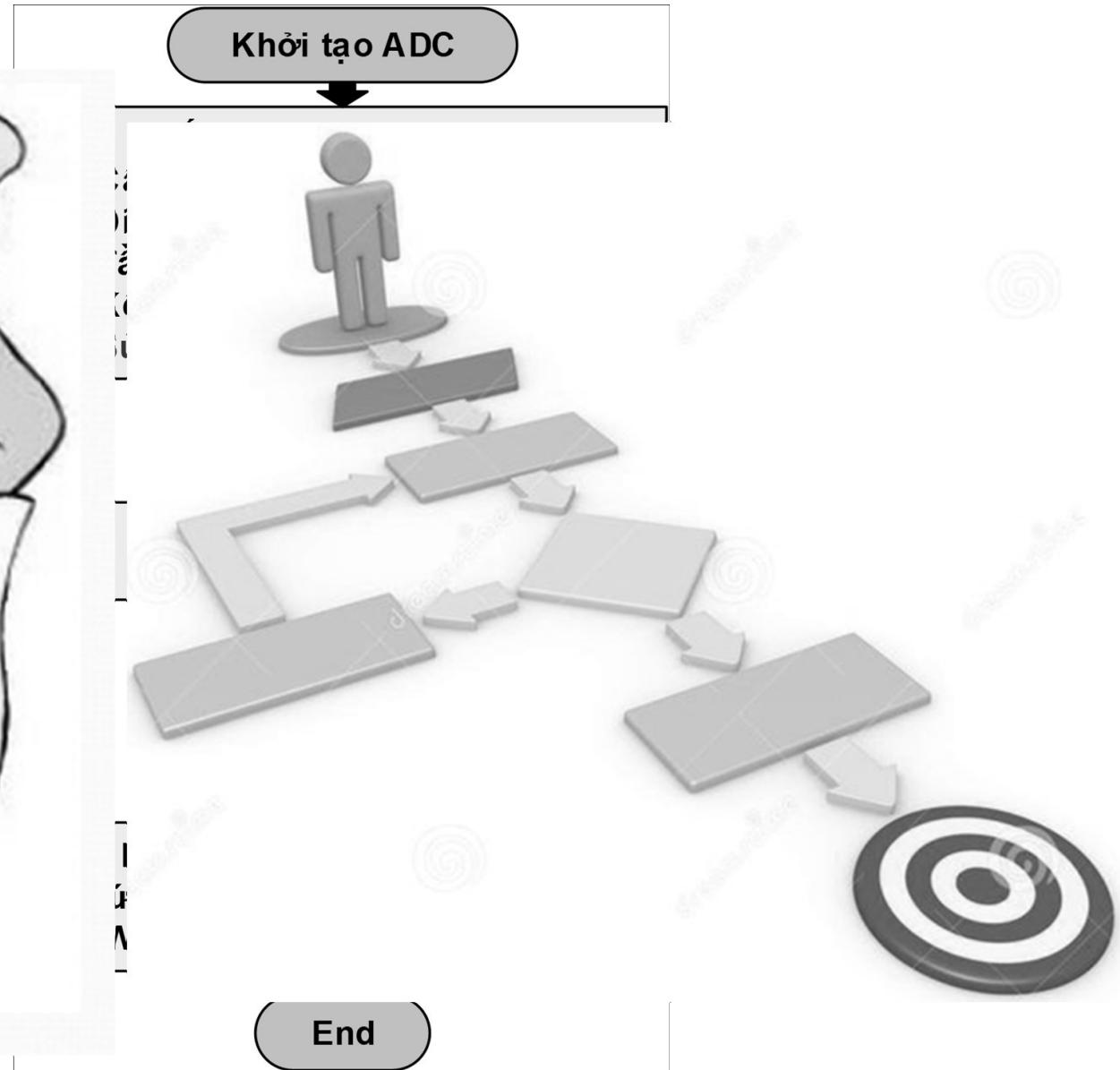
- Giải thuật:



( End )

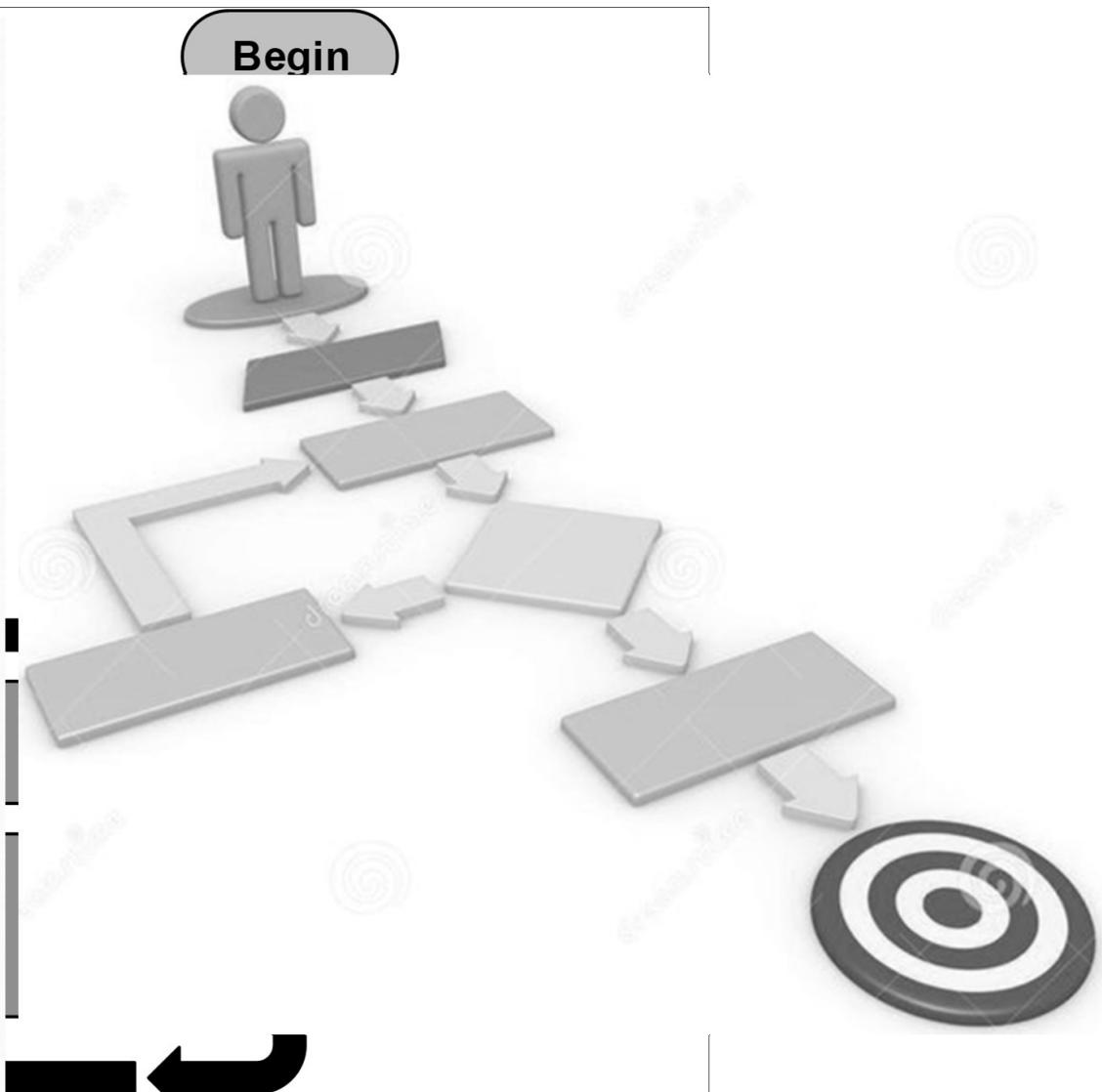
# VÍ DỤ MINH HỌA

- Giải thuật:



# VÍ DỤ MINH HỌA

- Giải thuật:



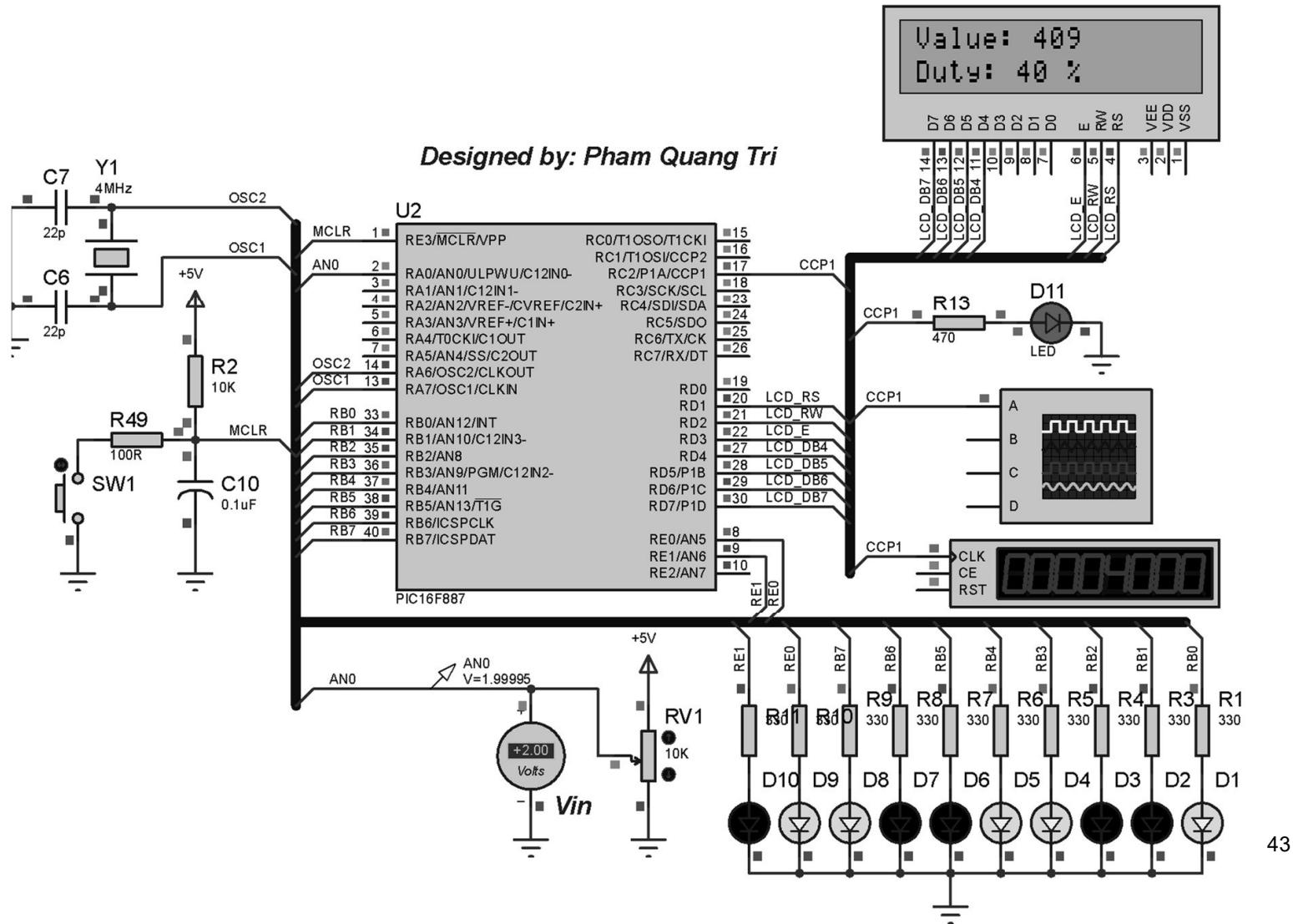
# VÍ DỤ MINH HỌA

- Giải thuật:

# VÍ DỤ MINH HỌA

- ## • Sơ đồ nguyên lý:

*Designed by: Pham Quang Tri*



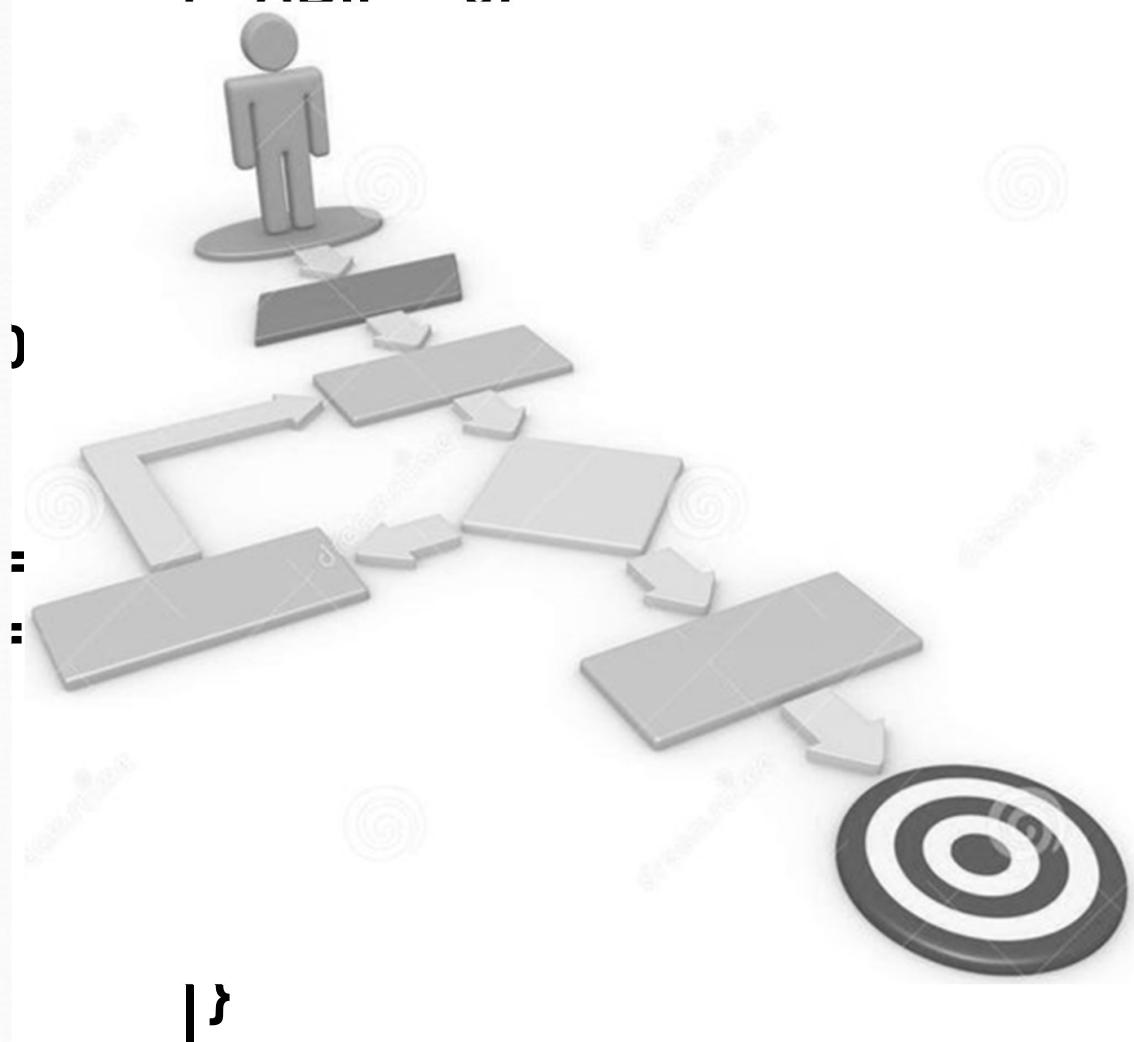
## VÍ DỤ MINH HỌA

- 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 giá trị số này để điều chỉnh độ rộng xung của xung PWM ( $f_{PWM} = 4KHz$ ). Hiển thị giá trị số (Digital) sau khi chuyển đổi lên LCD và 10 LED. Sử dụng mô-đun ADC và CCP (PWM)

# VÍ DỤ MINH HỌA

- Chương trình (Hi-Tech C):

| ADIF = 0;



Source: C8\_VD1\_CodeC.c

SIM: C8\_VD1\_CodeC.pdsprj

# VÍ DỤ MINH HỌA

- Chương trình (Hi-Tech C):



Source: C8\_VD1\_CodeC.c

SIM: C8\_VD1\_CodeC.pdsprj

# VÍ DỤ MINH HỌA

- Chương trình (Hi-Tech C):

Source: C8\_VD1\_CodeC.c

SIM: C8\_VD1\_CodeC.pdsprj

# VÍ DỤ MINH HỌA

- Giải thuật:

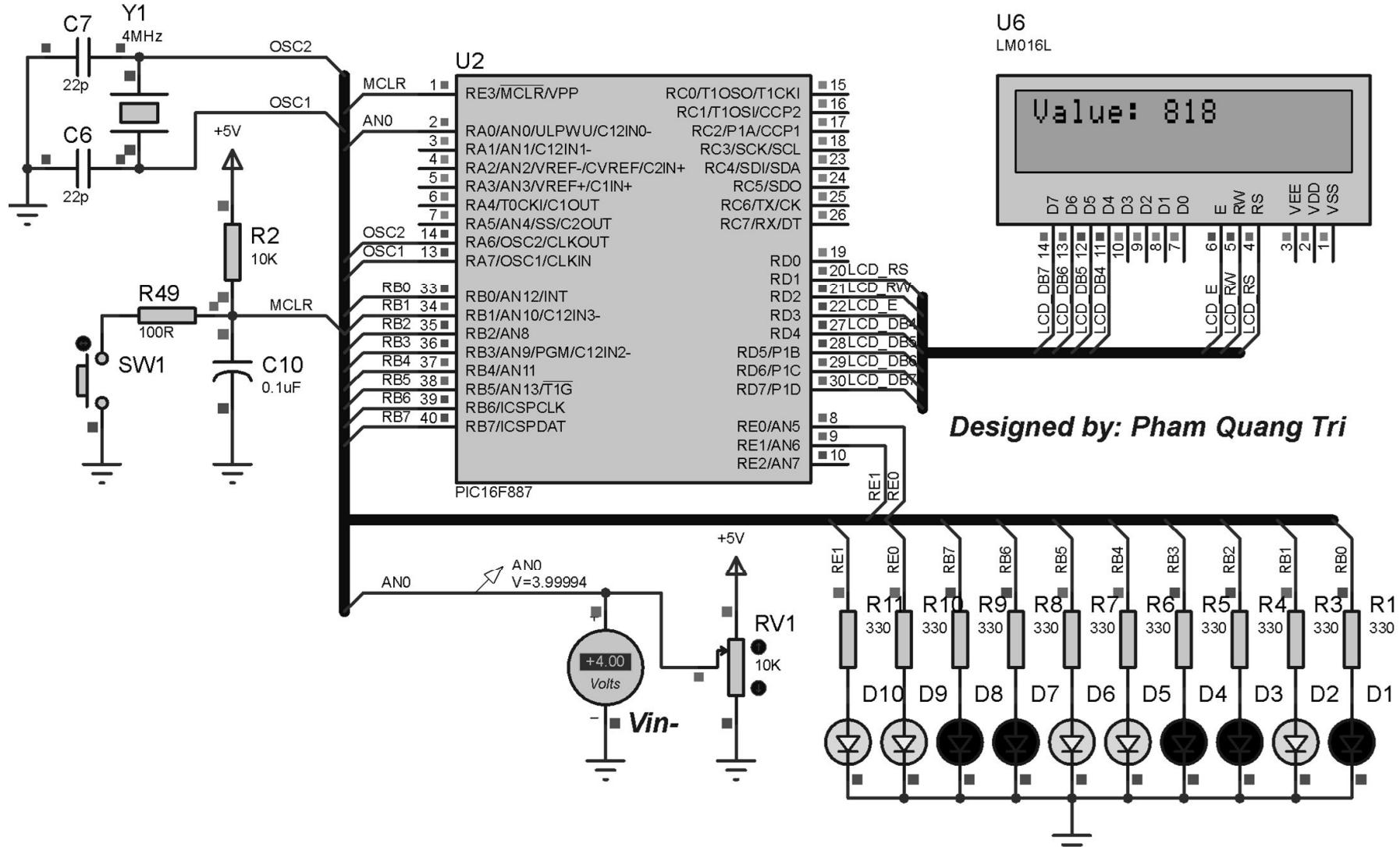


# VÍ DỤ MINH HỌA

- Giải thuật:

# VÍ DỤ MINH HỌA

- Sơ đồ nguyên lý:



## 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). Hiển thị giá trị số (Digital) sau khi chuyển đổi lên LCD và 10 LED. Sử dụng mô-đun ADC.**

# ADC: GIẢI THUẬT ĐIỀU KHIỂN

## ➤ Giải thuật điều khiển mô-đun ADC:

**Lưu ý rằng nếu chúng ta muốn thực hiện lại một lần nữa quá trình chuyển đổi tín hiệu thì chỉ cần quay trở về thực hiện từ Bước 5 đến Bước 7**

# ADC: GIẢI THUẬT ĐIỀU KHIỂN

## ➤ Giải thuật điều khiển mô-đun ADC:

- **BƯỚC 6: Xác định thời điểm quá trình chuyển đổi AD hoàn tất bằng một trong 2 cách sau:**
  - Polling: Kiểm tra bit GO/DONE  
→ Kiểm tra GO = 0 ?
  - Interrupt: Xử lý ISR của ADC  
→ Viết hàm ISR
- **BƯỚC 7: Đọc về và xử lý giá trị ADC sau khi chuyển đổi trong thanh ghi ADRESH:ADRESL**

$$Digital\ Value = (ADRESH \times 256) + ADRESL$$

# ADC: GIẢI THUẬT ĐIỀU KHIỂN

## ➤ Giải thuật điều khiển mô-đun ADC:

- **BƯỚC 3: Cấu hình ngắt ADC (tùy chọn)**

- Xóa cò ngắt  $\rightarrow ADIF = 0$
  - Vô hiệu hóa hoặc kích hoạt ngắt  
 $\rightarrow GIE = PEIE = ADIE = 0/1$

- **BƯỚC 4: Chờ một khoảng thời gian (xấp xỉ 20 $\mu$ s) để mô-đun ADC ổn định trạng thái**

- $\rightarrow$  Chờ đợi trong khoảng 20 $\mu$ s

- **BƯỚC 5: Bắt đầu quá trình chuyển đổi**

- $\rightarrow GO = 1$

# ADC: GIẢI THUẬT ĐIỀU KHIỂN

## ➤ Giải thuật điều khiển mô-đun ADC:

- **BƯỚC 2: Cấu hình mô-đun ADC**

- Chọn giá trị xung clock

→  $\text{ADCS1} = 00000000$

- Chọn điện áp tham chiếu

→  $\text{VCFG1} = 00000000$

- Chọn kênh vào ADC

→  $\text{CHS3} = 0/1, \text{CHS2} = 0/1, \text{CHS1} = 0/1, \text{CHS0} = 0/1$

- Chọn định dạng (conversion mode)

→  $\text{ADCON} = 00000000$

- Bật mô-đun ADC

→  $\text{ADEN} = 11111111$

CHS3	CHS2	CHS1	CHS0	Kênh ADC
0	0	0	0	AN0
0	0	0	1	AN1
0	0	1	0	AN2
0	0	1	1	AN3
0	1	0	0	AN4
0	1	0	1	AN5
0	1	1	0	AN6
0	1	1	1	AN7
1	0	0	0	AN8
1	0	0	1	AN9
1	0	1	0	AN10
1	0	1	1	AN11
1	1	0	0	AN12
1	1	0	1	AN13

# ADC: GIẢI THUẬT ĐIỀU KHIỂN

## ➤ Giải thuật điều khiển mô-đun ADC:

### • BƯỚC 2: Cấu hình mô-đun ADC

- Chọn giá trị xung clock chuyển đổi AD

→  $\text{ADCS1} = 0/1, \text{ADCS0} = 0/1$

- Chọn điện áp tham chiếu

→  $\text{VCFG1} = 0/1,$

- Chọn kênh vào ADC

→  $\text{CHS3} = 0/1, \text{CHS2} = 0/1,$

- Chọn định dạng (canh trái  
chuyển đổi)

→  $\text{ADFM} = 0/1$

- Bật mô-đun ADC

→  $\text{ADON} = 1$

$\text{ADCS1}$	$\text{ADCS0}$	Tần số chuyển đổi
0	0	$f_{\text{osc}} / 2$
0	1	$f_{\text{osc}} / 8$
1	0	$f_{\text{osc}} / 32$
1	1	$f_{\text{RC}}$ (Bộ dao động nội)

# ADC: GIẢI THUẬT ĐIỀU KHIỂN

## ➤ Giải thuật điều khiển mô-đun ADC:

### • BƯỚC 2: Cấu hình mô-đun ADC

- Chọn giá trị xung clock chuyển đổi AD  
→  $\text{ADCS1} = 0/1, \text{ADCS0} = 0/1$
- Chọn điện áp tham chiếu (bên trong hoặc bên ngoài)  
→  $\text{VCFG1} = 0/1, \text{VCFG0} = 0/1$
- Chọn kênh vào ADC  
→  $\text{CHS3} = 0/1, \text{CHS2} = 0/1, \text{CHS1} = 0/1, \text{CHS0} = 0/1$
- Chọn định dạng (canh trái hoặc canh phải) kết quả chuyển đổi  
→  $\text{ADFM} = 0/1$
- Bật mô-đun ADC  
→  $\text{ADON} = 1$

# ADC: GIẢI THUẬT ĐIỀU KHIỂN

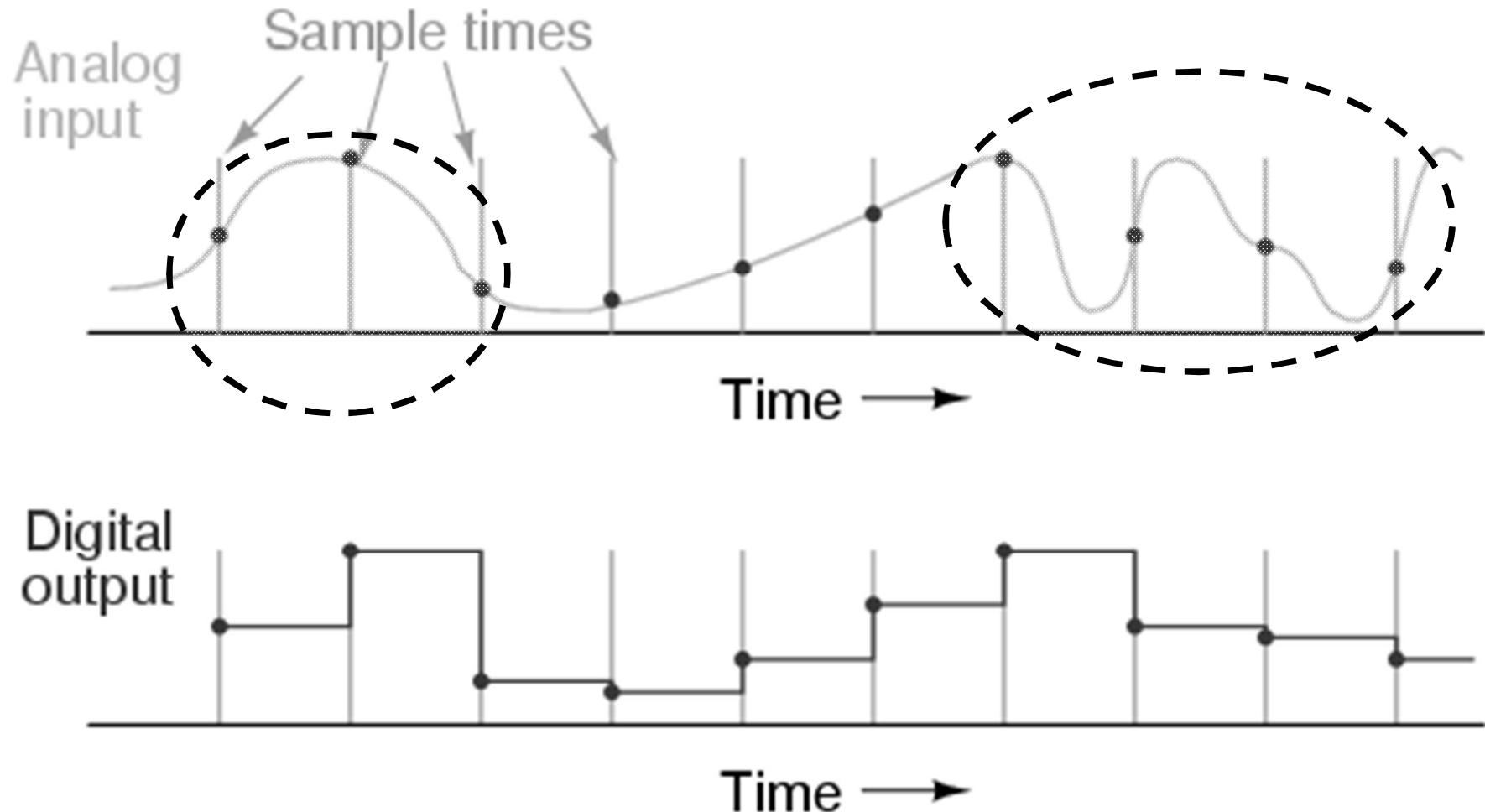
## ➤ Giải thuật điều khiển mô-đun ADC:

### • BƯỚC 1: Cấu hình Port sử dụng nhận tín hiệu

- Cấu hình chiều là ngõ vào → TRISx = 1
  - Port A (Kênh 0 – Kênh 4): x = A0, A1, A2, A3, A5
  - Port E (Kênh 5 – Kênh 7): x = E0, E1, E2
  - Port B (Kênh 8 – Kênh 13): x = B2, B3, B1, B4, B0, B5
- Cấu hình kiểu là tương tự → ANSx = 1
  - Port A (Kênh 0 – Kênh 4): x = 0, 1, 2, 3, 4
  - Port E (Kênh 5 – Kênh 7): x = 5, 6, 7
  - Port B (Kênh 8 – Kênh 13): x = 8, 9, 10, 11, 12, 13
- Hoặc có thể ngắn gọn hơn cho việc cấu hình kiểu là tương tự → ANSEL = ?, ANSELH = ?

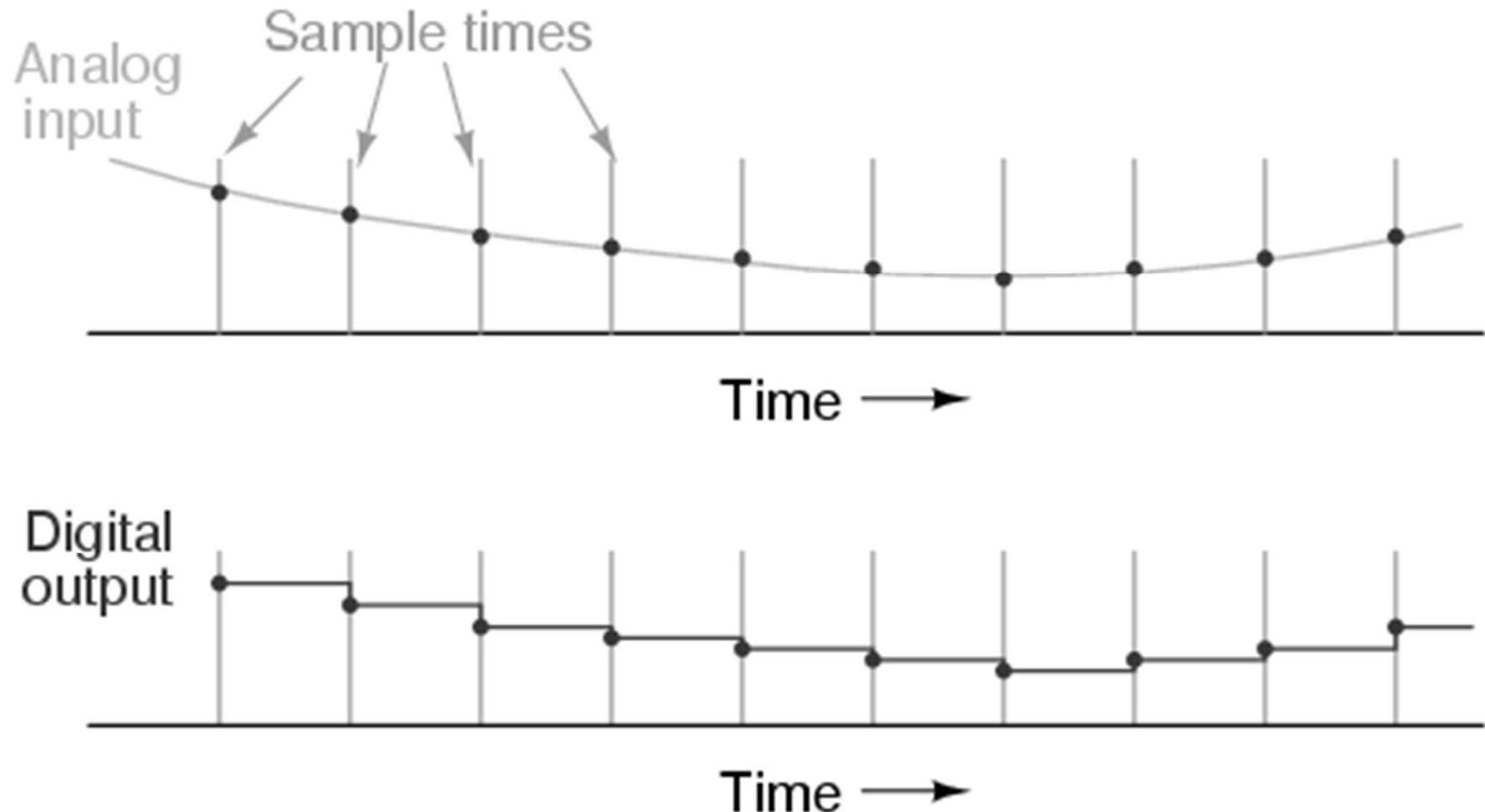
<b>ANSEL</b>	ANS7	ANS6	ANS5	ANS4	ANS3	ANS2	ANS1	ANS0
<b>ANSELH</b>	0	0	ANS13	ANS12	ANS11	ANS10	ANS9	ANS8

# VẤN ĐỀ THỜI GIAN TRONG ADC



Khi khoảng thời gian lấy mẫu quá dài (quá chậm), các chi tiết quan trọng của tín hiệu tương tự đưa vào sẽ bị bỏ qua<sup>4</sup>

# VẤN ĐỀ THỜI GIAN TRONG ADC



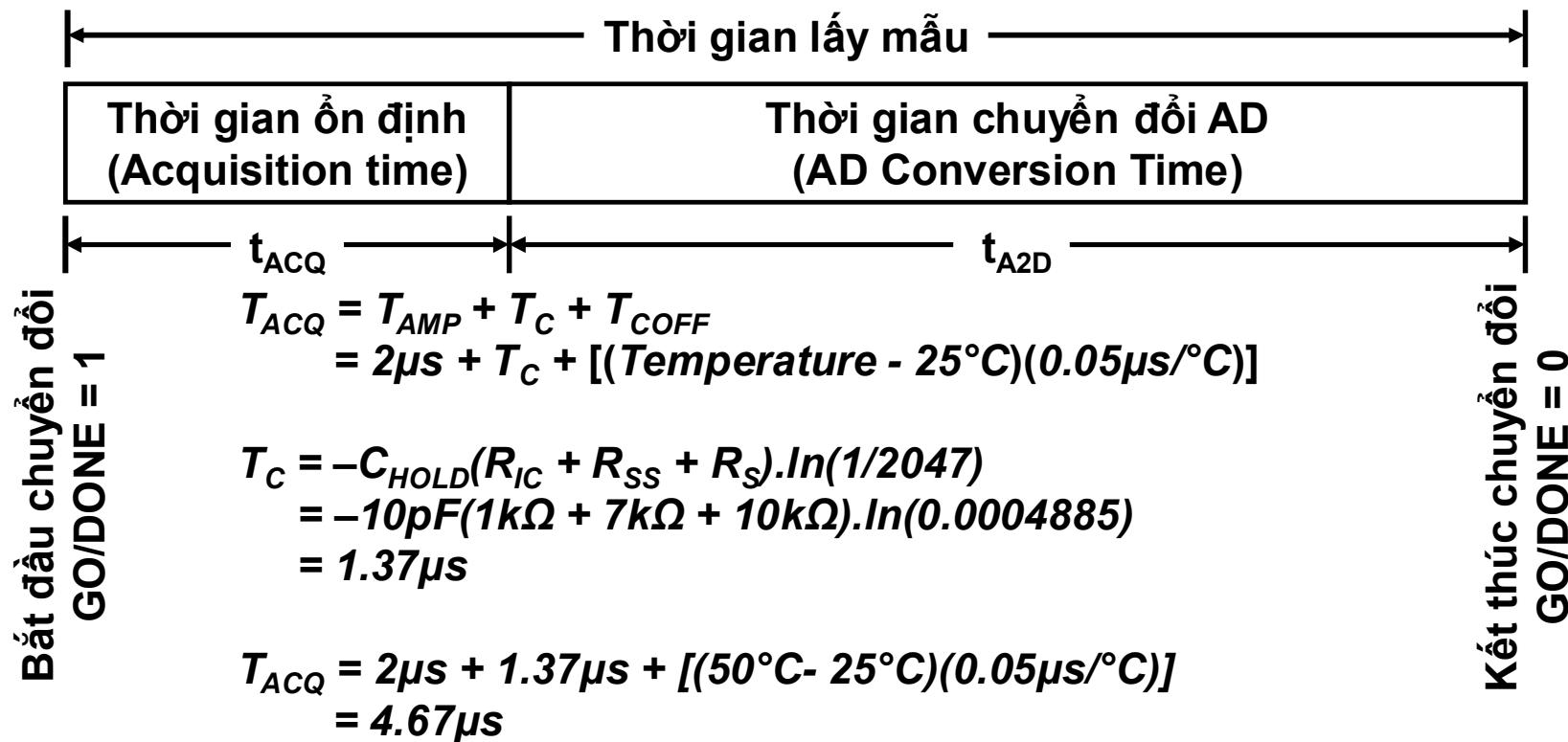
Tín hiệu tương tự đưa vào thay đổi chậm, thời gian lấy mẫu ở đây là quá đủ để nắm bắt xu hướng chung của nó <sup>4</sup>

# VẤN ĐỀ THỜI GIAN TRONG ADC

- Điều bắt buộc là thời gian lấy mẫu của ADC phải đủ nhanh để nắm bắt những thay đổi cần thiết trong dạng sóng tín hiệu tương tự.
- Phải chọn xung clock ADC phù hợp cho từng trường hợp dạng sóng tín hiệu tương tự đưa vào
- Về mặt lý thuyết tần số cao nhất của tín hiệu tương tự mà một ADC có thể nắm bắt là tần số Nyquist, bằng  $\frac{1}{2}$  tần số lấy mẫu của ADC.
- Ví dụ: ADC có tần số lấy mẫu là 5000 Hz
  - Tín hiệu tương tự đưa vào phải có tần số không được vượt quá 2500 Hz.

# VĂN ĐỀ THỜI GIAN TRONG ADC

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



(Chi tiết xem thêm trong datasheet của PIC16F887)

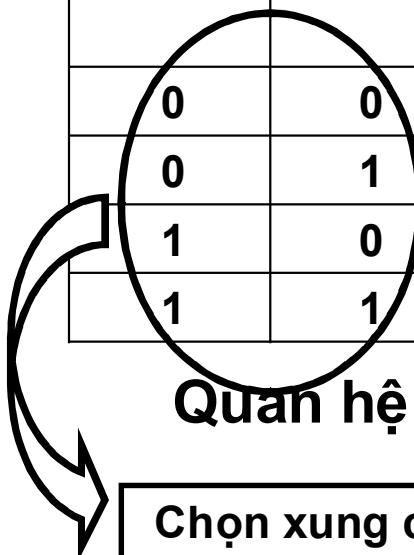
$$t_{ACQ} > 5\mu s \quad t_{A2D} = 11 \cdot T_{AD}$$

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

# VẤN ĐỀ THỜI GIAN TRONG ADC

- **Bất kỳ sự thay đổi nào của tần số xung clock hệ thống ( $f_{osc}$ ) đều ảnh hưởng đến chu kỳ xung clock ADC ( $T_{AD}$ ), từ đó có thể ảnh hưởng bất lợi đến kết quả ADC**

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



Quan hệ giữa  $T_{AD}$  với tần số hoạt động của PIC

Chọn xung clock ADC sao cho  $T_{AD} \geq 1.6\mu$ s

# VẤN ĐỀ THỜI GIAN TRONG ADC

- **Bất kỳ sự thay đổi nào của tần số xung clock hệ thống ( $f_{osc}$ ) đều ảnh hưởng đến chu kỳ xung clock ADC ( $T_{AD}$ ), từ đó có thể ảnh hưởng bất lợi đến kết quả ADC**

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

**Quan hệ giữa  $T_{AD}$  với tần số hoạt động của PIC**

# VĂN ĐỀ THỜI GIAN TRONG ADC

- Thời gian cần thiết để hoàn thành việc chuyển đổi một bit được gọi là  $T_{AD}$  (ADC Clock Period)
- Bắt buộc  $T_{AD}$  tối thiểu phải là  $1,6\mu s$
- Để hoàn tất quá trình chuyển đổi A/D 10 bit thì cần  $11 \cdot T_{AD}$
- Bất kỳ sự thay đổi nào của tần số xung clock hệ thống ( $f_{osc}$ ) đều ảnh hưởng đến chu kỳ xung clock ADC ( $T_{AD}$ ), từ đó có thể ảnh hưởng bất lợi đến kết quả ADC.

# 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

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

Trong đó:

DV: Giá trị số (Digital Value) sau khi chuyển đổi

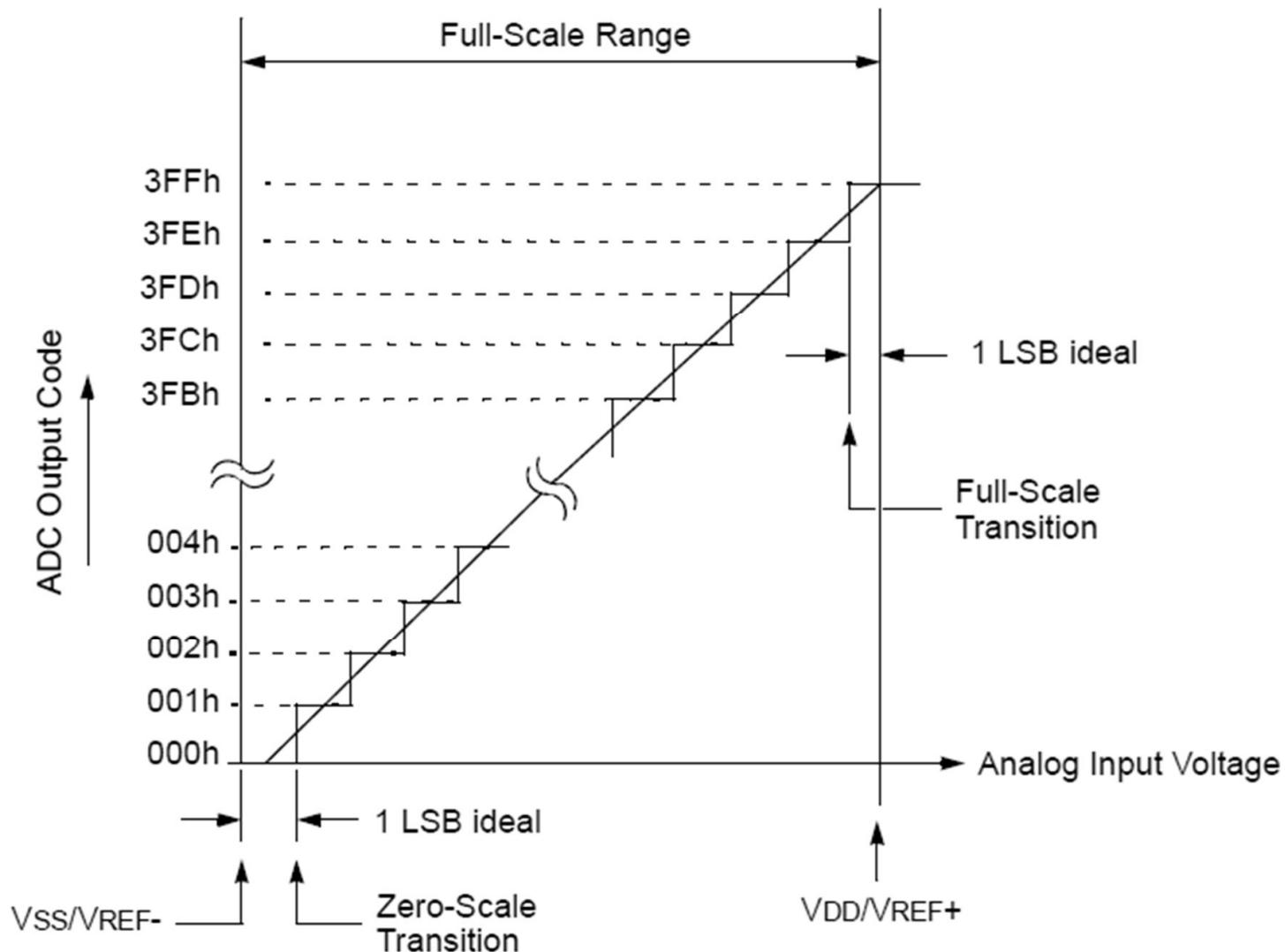
$V_{IN}$ : Điện áp tương tự ngõ vào (V)

$V_{REF(+)}, V_{REF(-)}$ : Điện áp tham chiếu (V)

n: Số bit của ADC.

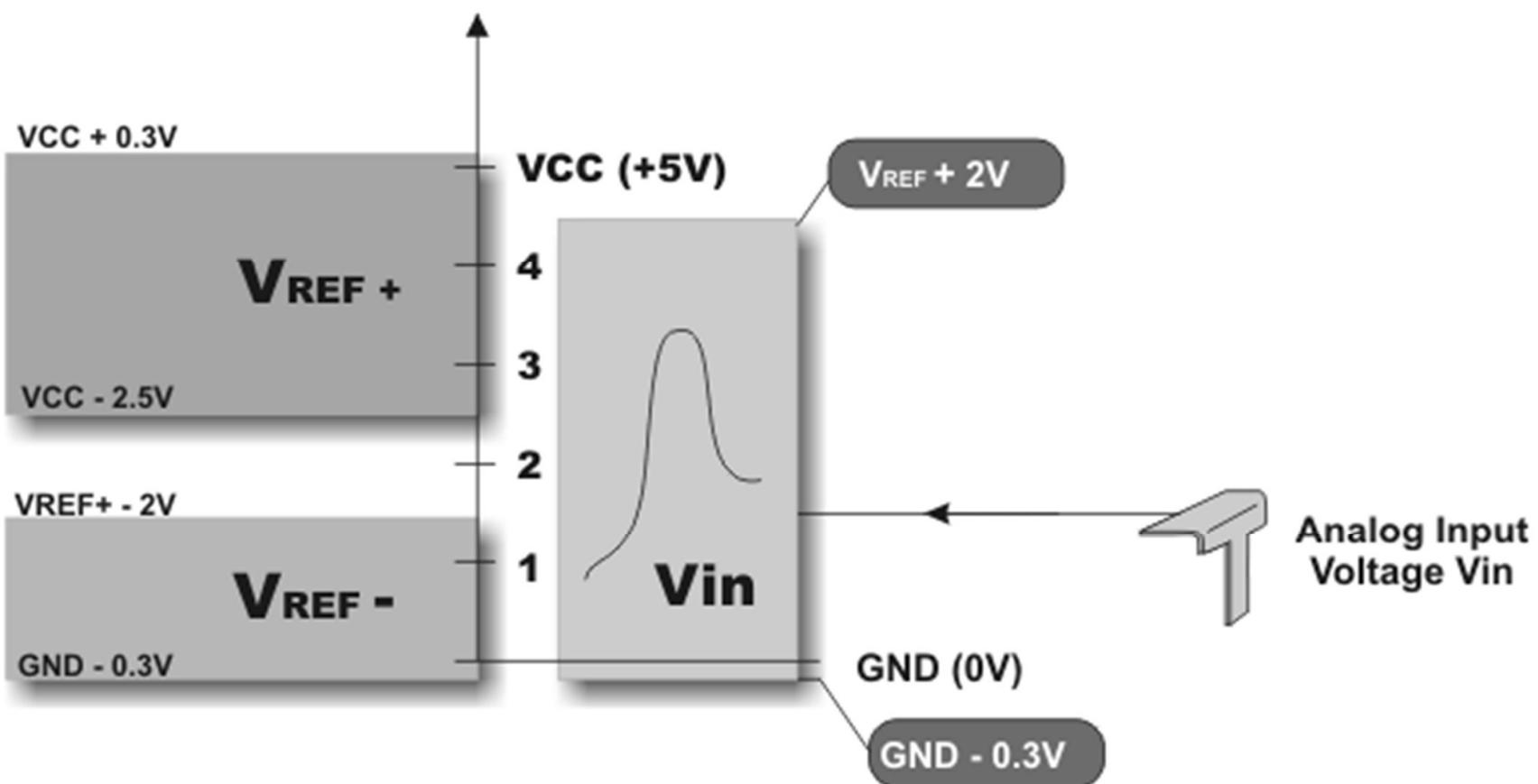
# MỘT SỐ ĐIỂM CẦN LƯU Ý

## ➤ **Hàm truyền ADC (ADC Transfer Function)**



# MỘT SỐ ĐIỂM CẦN LƯU Ý

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



# MỘT SỐ ĐIỂM CẦN LƯU Ý

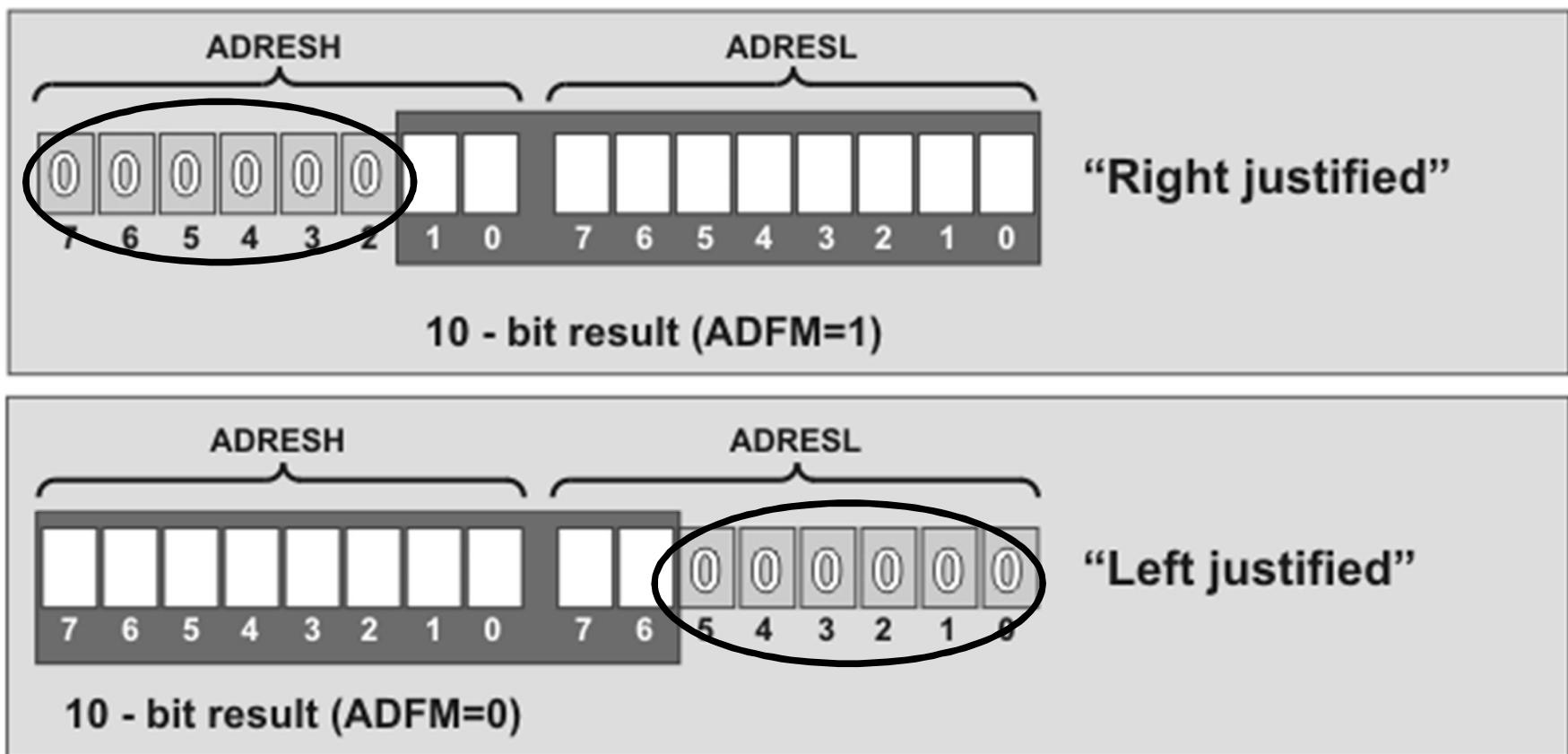
- **Bắt đầu quá trình chuyển đổi AD:**
  - GO/DONE = 1 bởi người sử dụng
- **Kết thúc quá trình chuyển đổi:**
  - Bit GO/DONE = 0
  - Cờ ADIF = 1
  - Giá trị ADRESH:ADRESL được cập nhật giá trị
- **Hủy quá trình chuyển đổi:**
  - GO/DONE = 0 bởi người sử dụng.
- **Để tránh sai sót khi đo lường hoặc hư hỏng chip, các chân nên được cấu hình là ngõ vào tương tự trước khi bắt đầu chuyển đổi.**

## MỘT SỐ ĐIỂM CẦN LƯU Ý

- Để việc chuyển đổi được chính xác cần phải tạo ra một khoảng thời gian trễ nhất định giữa việc lựa chọn một ngõ vào tương tự cụ thể và hoạt động chuyển đổi.
- Độ trễ thời gian như vậy được gọi là 'thời gian chuyển đổi' và chủ yếu phụ thuộc vào trở kháng nguồn.
- Có một phương trình được sử dụng để tính toán độ trễ thời gian này một cách chính xác, trong trường hợp xấu nhất là khoảng  $20\mu s$ . Vì vậy, nếu bạn muốn chuyển đổi A/D chính xác, đừng quên chi tiết nhỏ này.

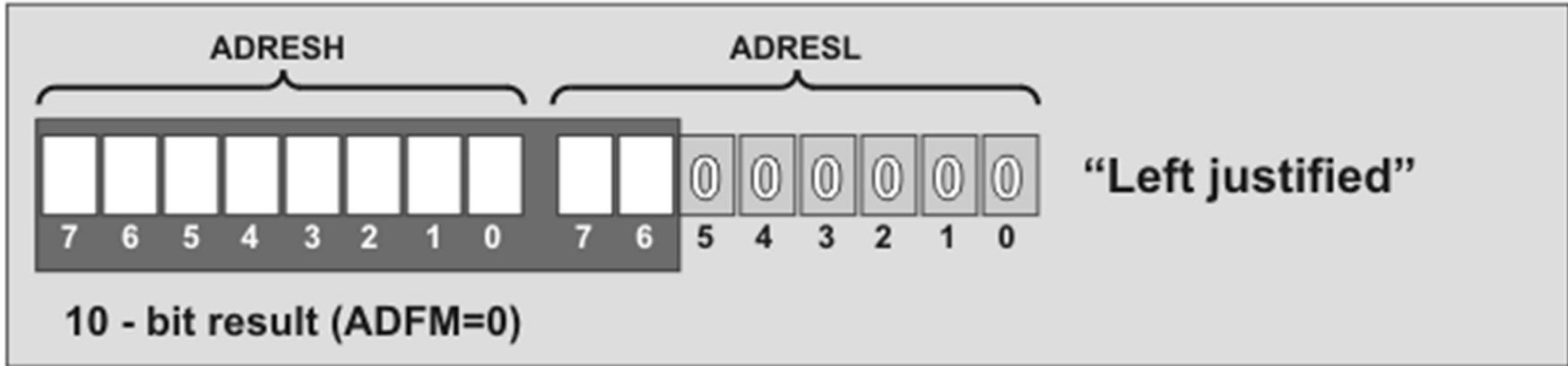
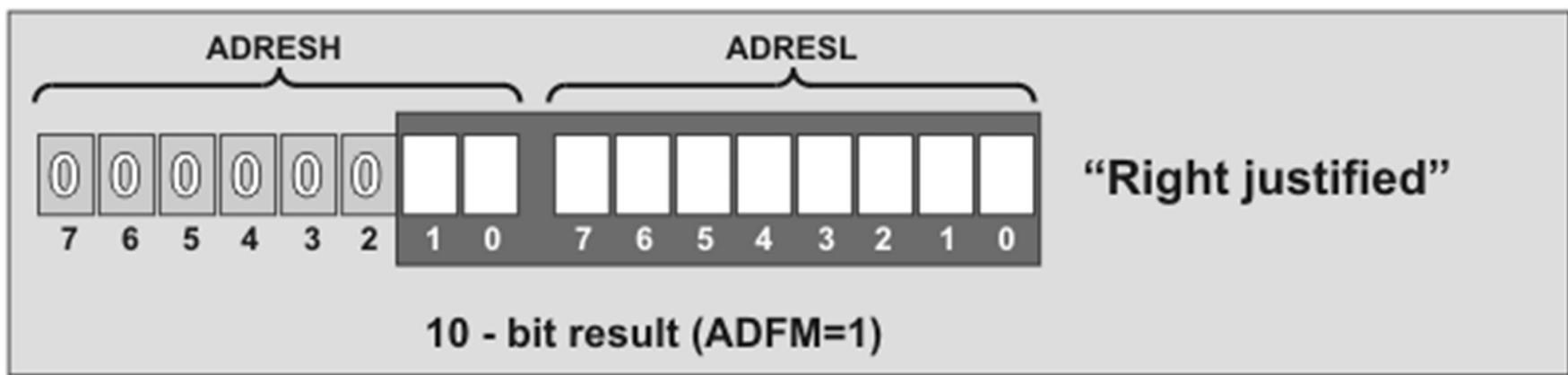
# CÁC THANH GHI MÔ-ĐUN ADC

- Kết quả ADC 10 bit có thể được canh trái hoặc canh phải

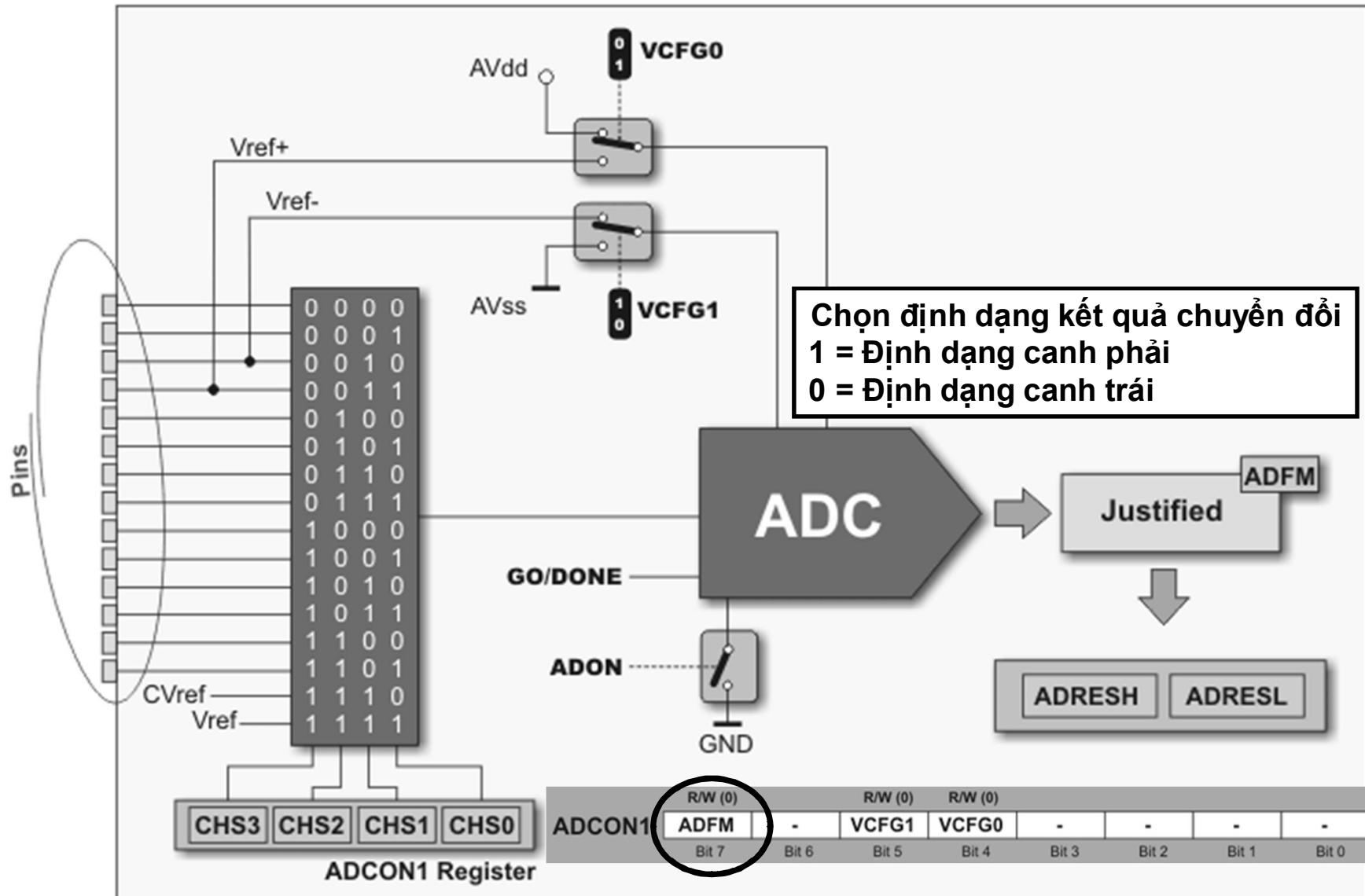


# CÁC THANH GHI MÔ-ĐUN ADC

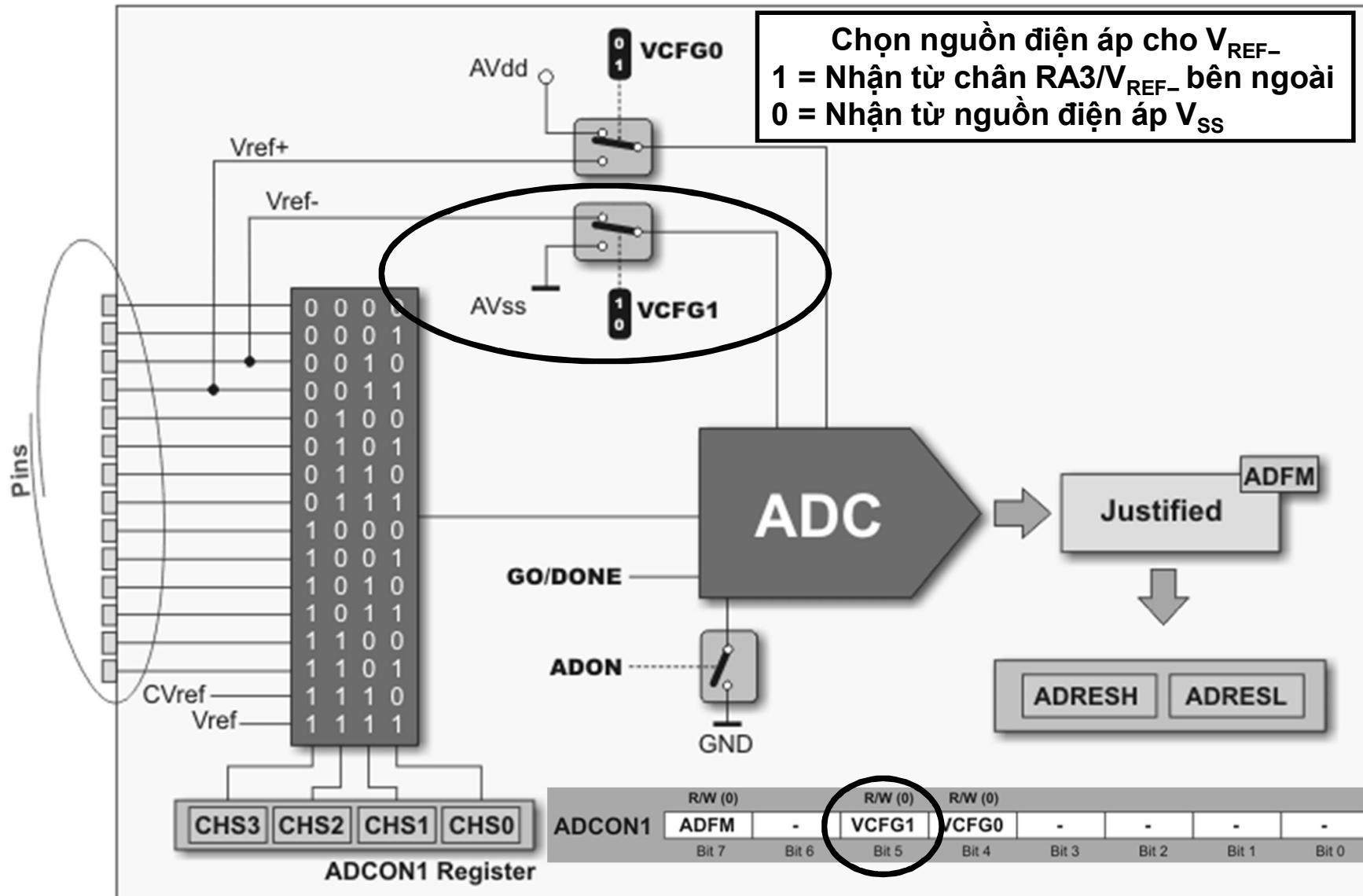
- Sau khi chuyển đổi, kết quả ADC được đặt vào trong hai thanh ghi kết quả: ADRESH và ADRESL



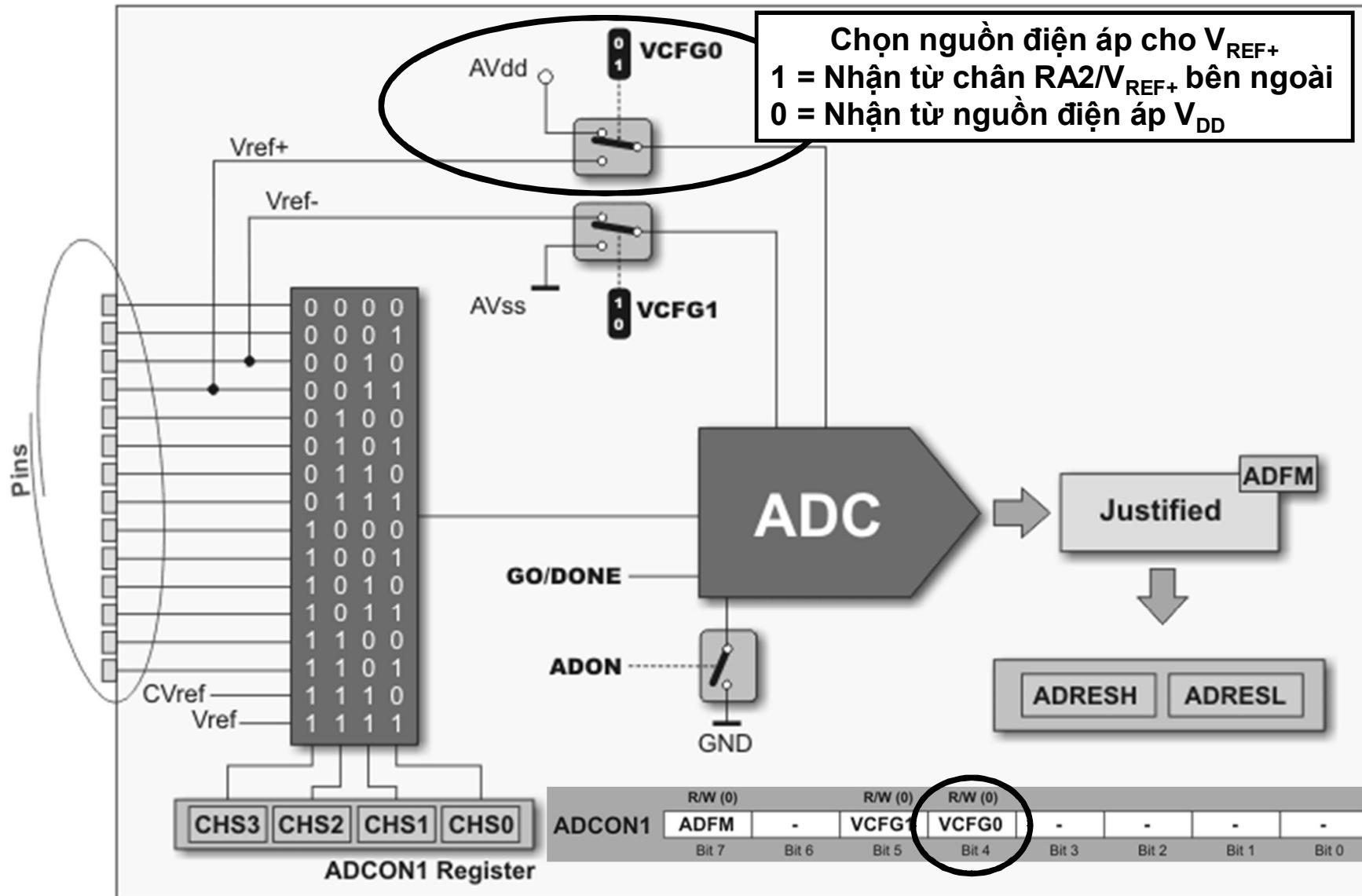
# CÁC THANH GHI MÔ-ĐUN ADC



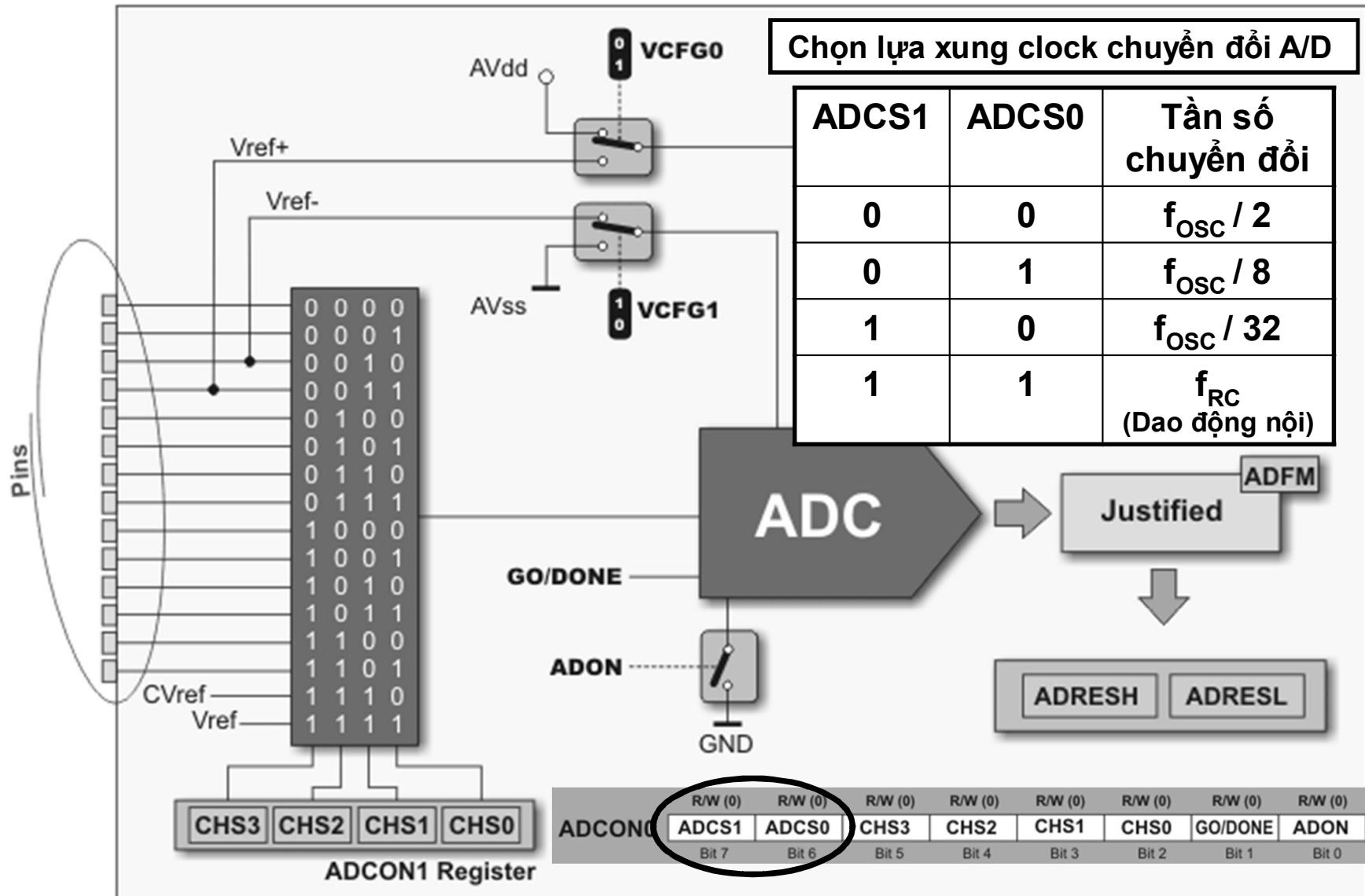
# CÁC THANH GHI MÔ-ĐUN ADC



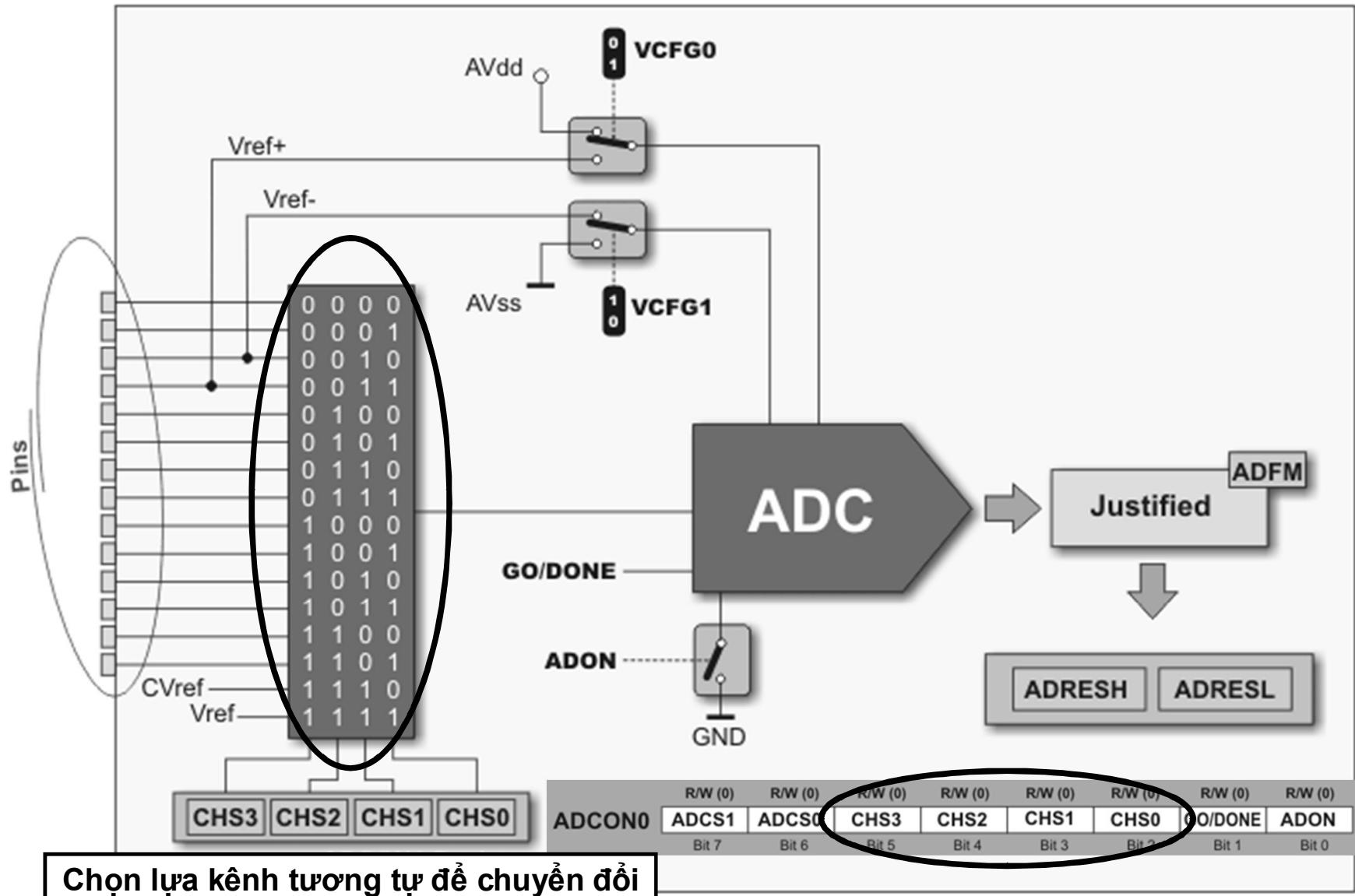
# CÁC THANH GHI MÔ-ĐUN ADC



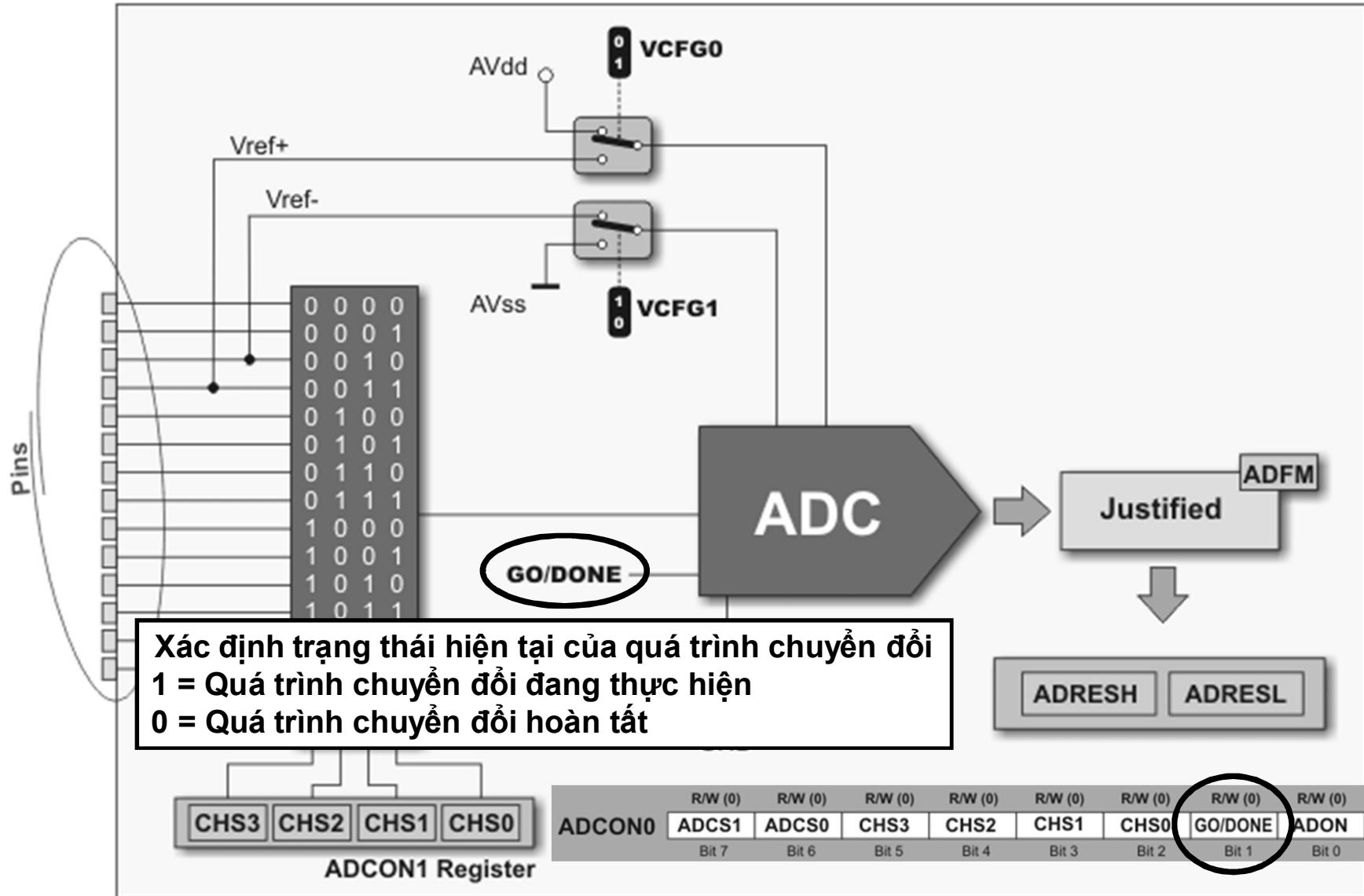
# CÁC THANH GHI MÔ-ĐUN ADC



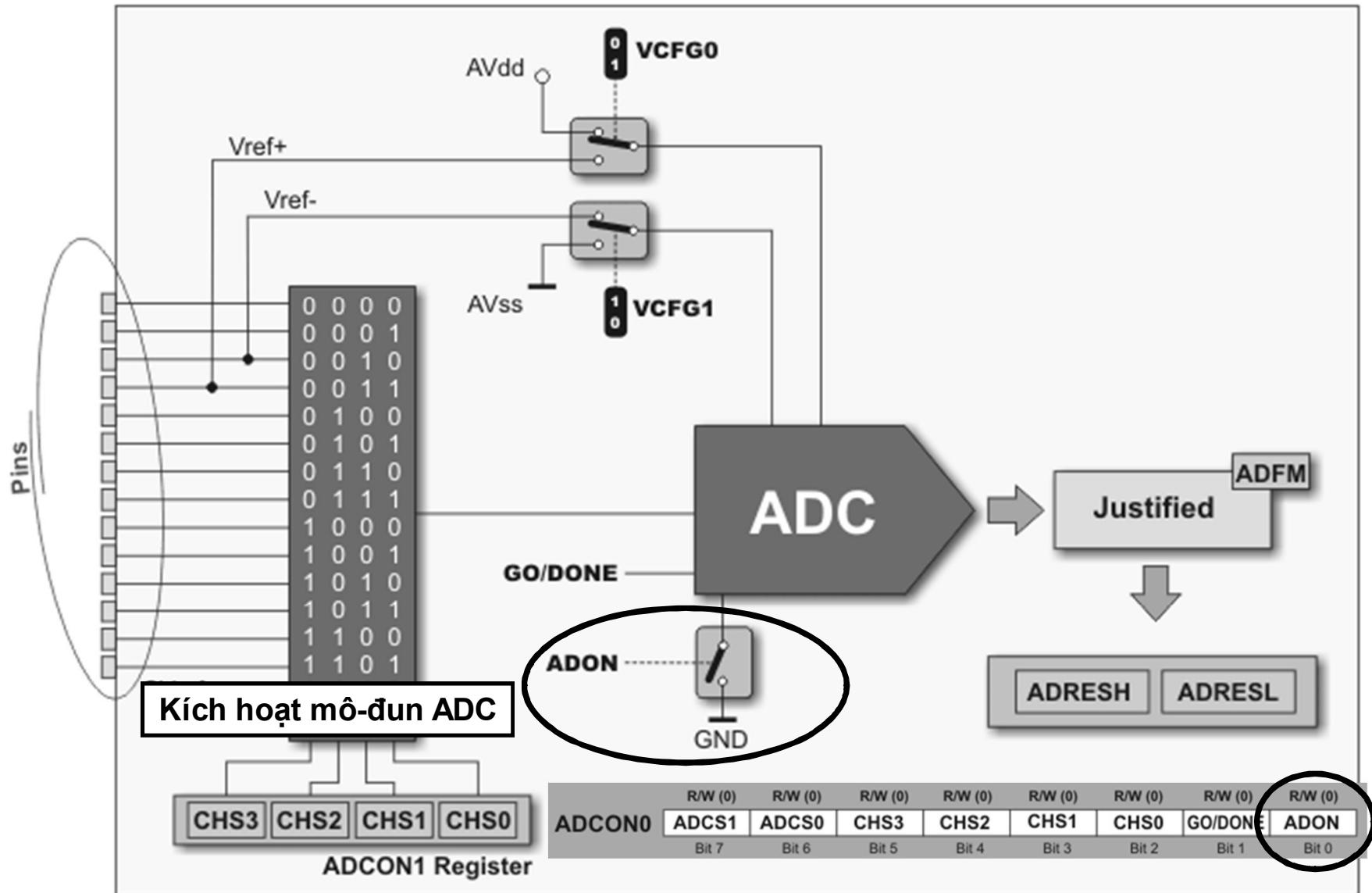
# CÁC THANH GHI MÔ-ĐUN ADC



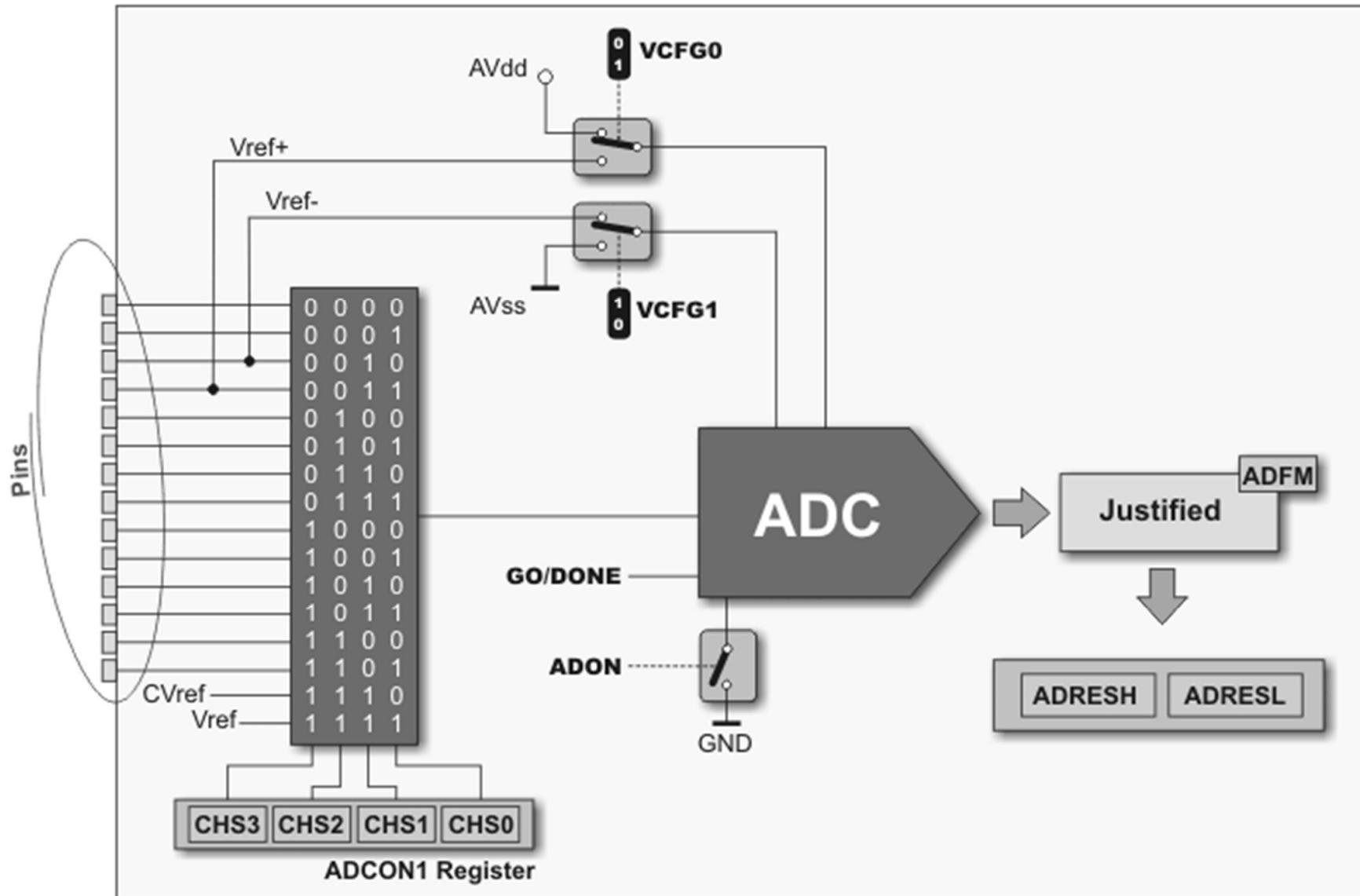
# CÁC THANH GHI MÔ-ĐUN ADC



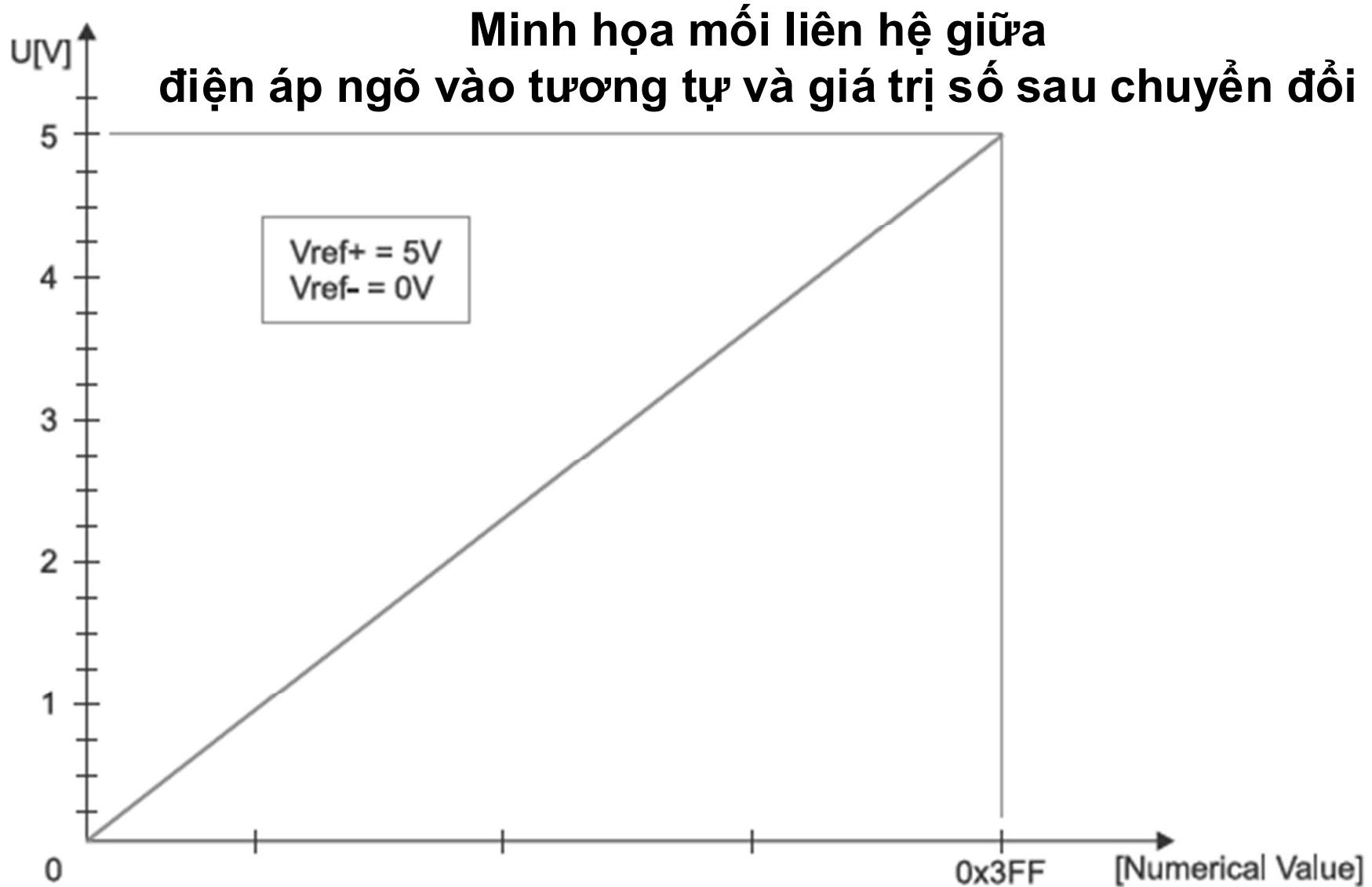
# CÁC THANH GHI MÔ-ĐUN ADC



# SƠ ĐỒ KHỐI KHÔNG ĐƠN GIẢN MÔ-ĐUN ADC

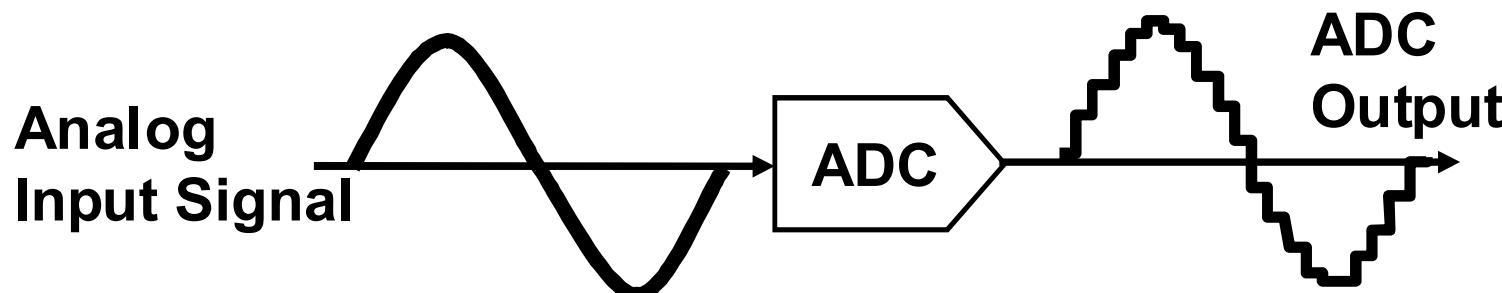


# TỔNG QUAN VỀ MÔ-ĐUN ADC



# TỔNG QUAN VỀ MÔ-ĐUN ADC

- Chuyển đổi tín hiệu tương tự (dạng điện áp) thành giá trị số nhị phân 10 bit
- Vi điều khiển PIC16F887 có 14 chân sử dụng làm ngõ vào tín hiệu tương tự
- Có khả năng chọn lựa điện áp tham chiếu bên trong hoặc bên ngoài vi điều khiển
- Độ phân giải (chất lượng chuyển đổi) có thể được điều chỉnh bằng cách chọn lựa nguồn điện áp tham chiếu.



## CHƯƠNG 8: MÔ-ĐUN CHUYỂN ĐỔI TƯƠNG TỰ SANG SỐ

**Nội dung bao gồm:**

- **Tổng quan về mô-đun ADC**
- **Các thanh ghi của mô-đun ADC**
- **Sơ đồ khối đơn giản của mô-đun ADC**
- **Các thông số thời gian ảnh hưởng đến việc chuyển đổi AD**
- **Một số vấn đề cần lưu ý**
- **Ví dụ minh họa và bài tập ứng dụng.**

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

**CHƯƠNG 8  
MÔ-ĐUN  
CHUYỂN ĐỔI TƯƠNG TỰ  
SANG SỐ  
(ANALOG TO DIGITAL  
CONVERTER)**