

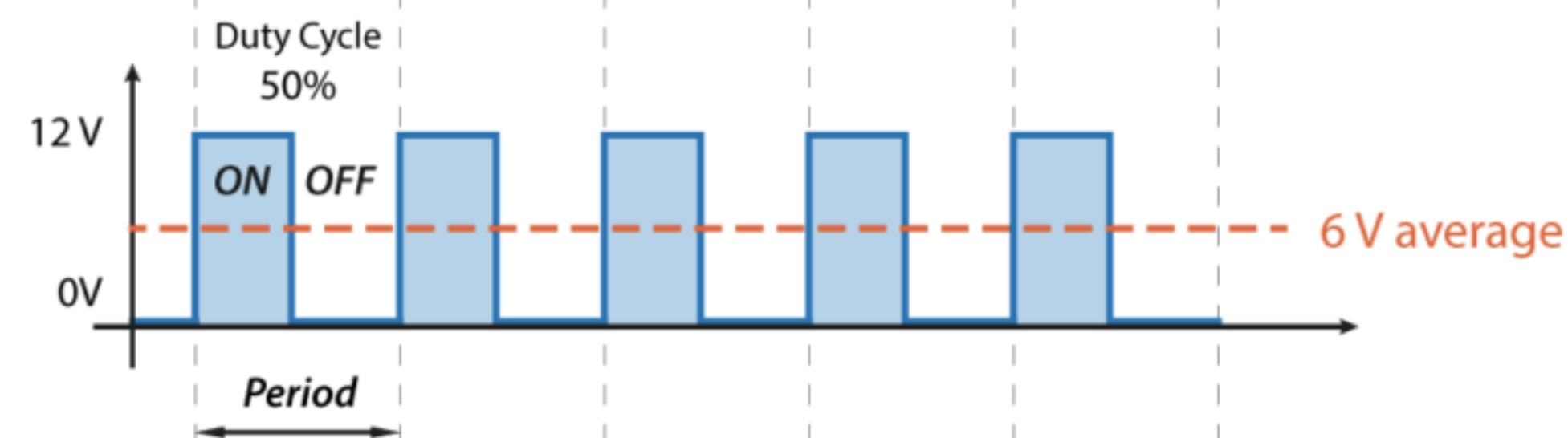
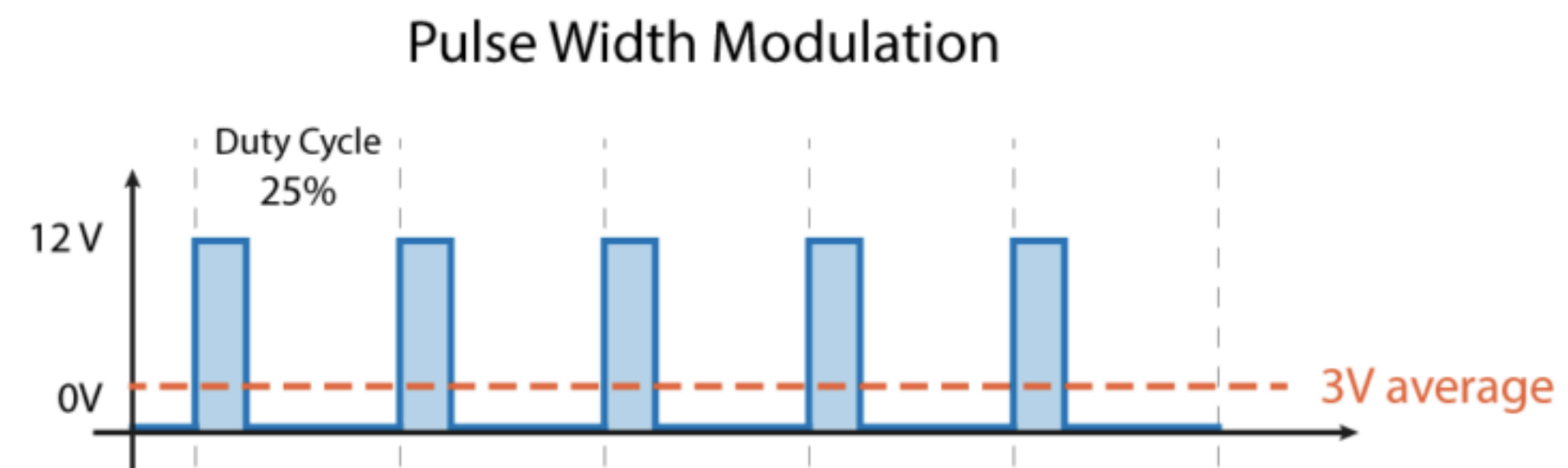
Phần Lý thuyết

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

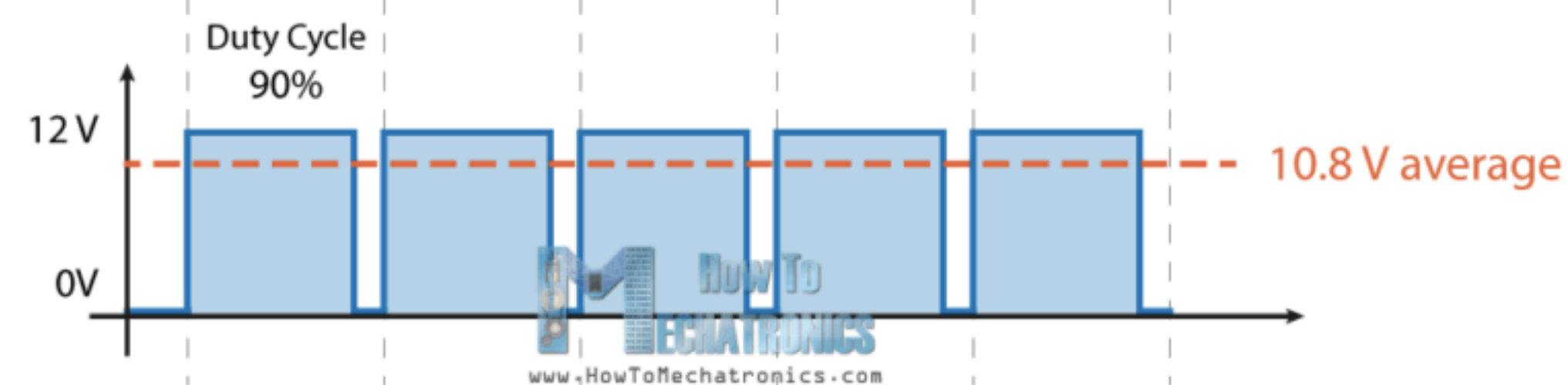
PWM là gì ?

- Điều chế độ rộng xung
- Mục đích là tạo ra xung vuông, trong đó phần năng lượng có ích tương đương một tín hiệu điện một chiều DC với điện áp điều chỉnh được

Điện áp thấp, thời gian
ON của xung ngắn

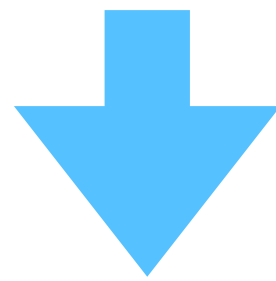


Điện áp cao, thời gian
ON của xung dài

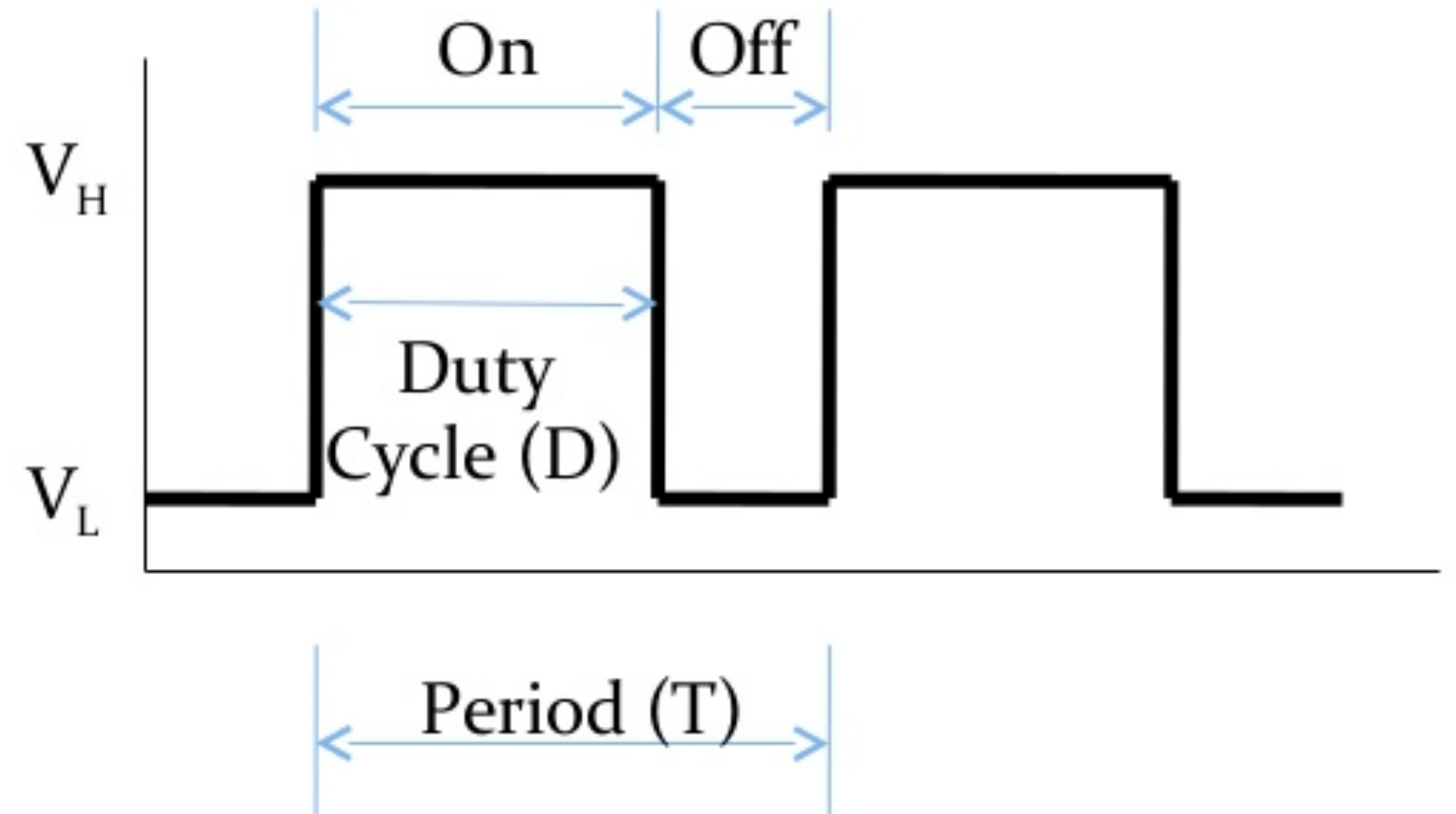


Nguyên lý chung của PWM

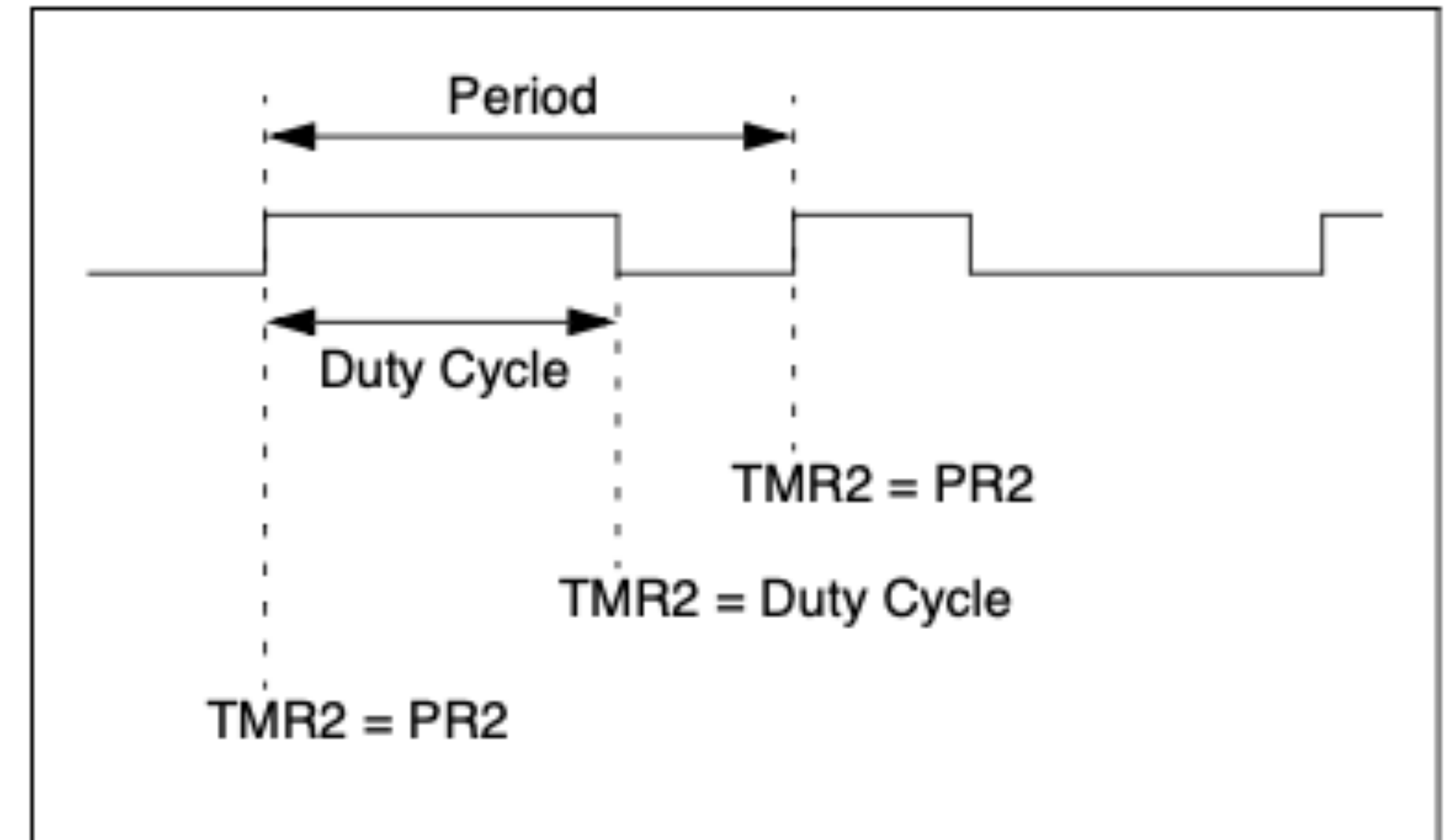
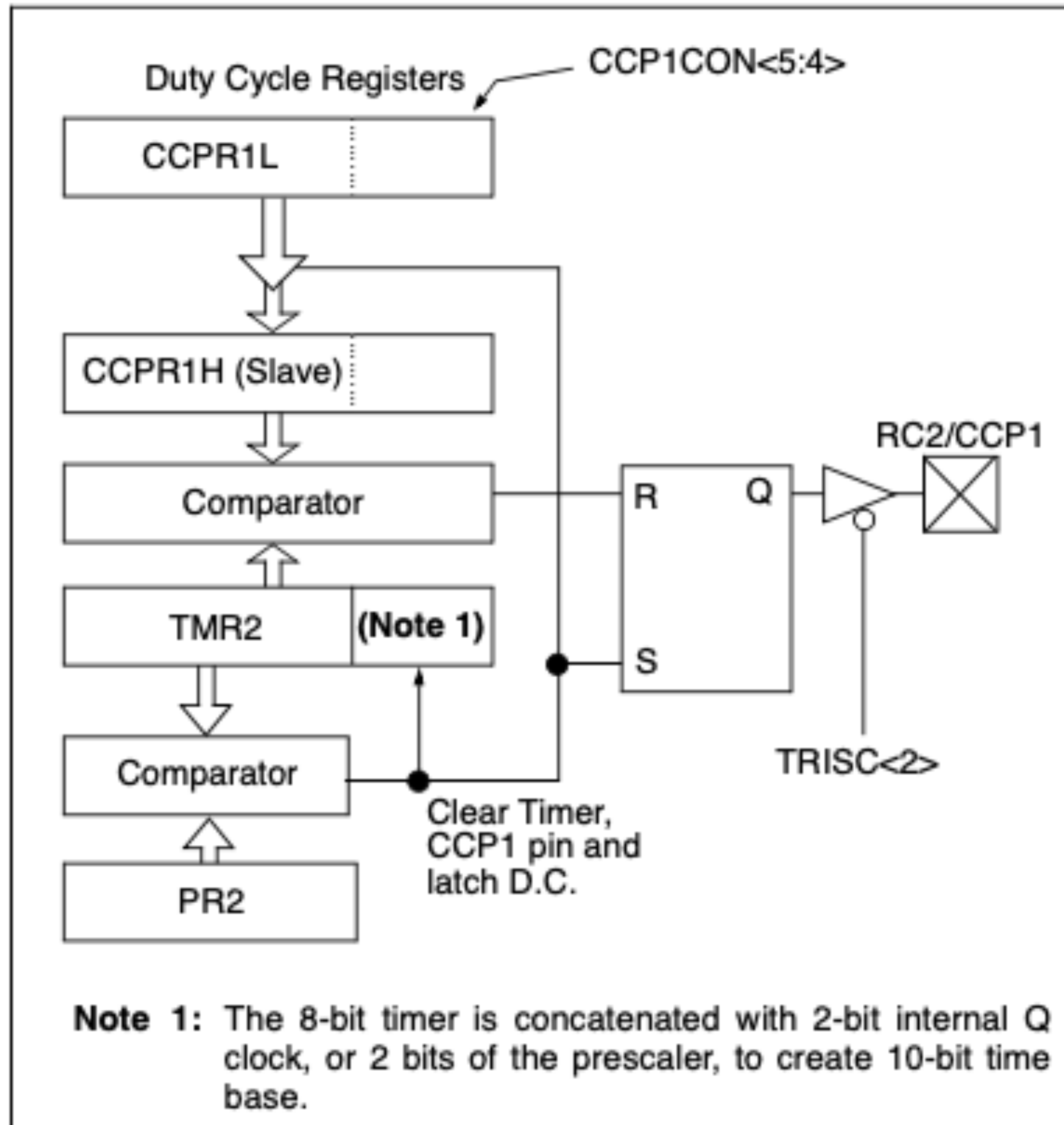
$$Duty\ Cycle = \frac{On\ Time}{Period} \times 100\%$$



Cố định T (Period) và thay đổi D (Duty Cycle)



PWM trong PIC16F877A



Set Period bằng cách nạp giá trị vào thanh ghi PR2

Set Duty Cycle bằng cách nạp giá trị vào CCPR1L và CCP1CON

Các bước để setup PWM

- Xác định Periode bằng cách nạp vào thanh ghi PR2 giá trị thích hợp

$$\text{PWM Period} = [(\text{PR2}) + 1] * 4 * T_{\text{Osc}} * (\text{TMR2 Prescale})$$

TMR2 Prescale được định nghĩa trong thanh ghi T2CON

T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

Các bước để setup PWM

- Xác định Duty Cycle bằng cách nạp vào thanh ghi CCPR1L và CCP1CON giá trị thích hợp

$$\text{PWM Duty Cycle} = \underbrace{(\text{CCPR1L}:\text{CCP1CON}_{<5:4>})}_{\text{Giá trị này gồm 10 bit bao gồm 8 bit trong CCPR1L và 2 bit thứ 5:4 trong thanh ghi CCP1CON}} * T_{\text{osc}} * (\text{TMR2}_{\text{Prescale}})$$

Giá trị này gồm 10 bit bao gồm 8 bit trong CCPR1L và 2 bit thứ 5:4 trong thanh ghi CCP1CON

- Mở cổng CCP1 (chân RC2) là cổng output

Các bước để setup PWM

- Set TMR2 Prescale và enable Timer2 bằng cách nạp giá trị cho T2CON

T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

TMR2ON: Timer2 On bit

1 = Timer2 is on

0 = Timer2 is off

T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits

00 = Prescaler is 1

01 = Prescaler is 4

1x = Prescaler is 16

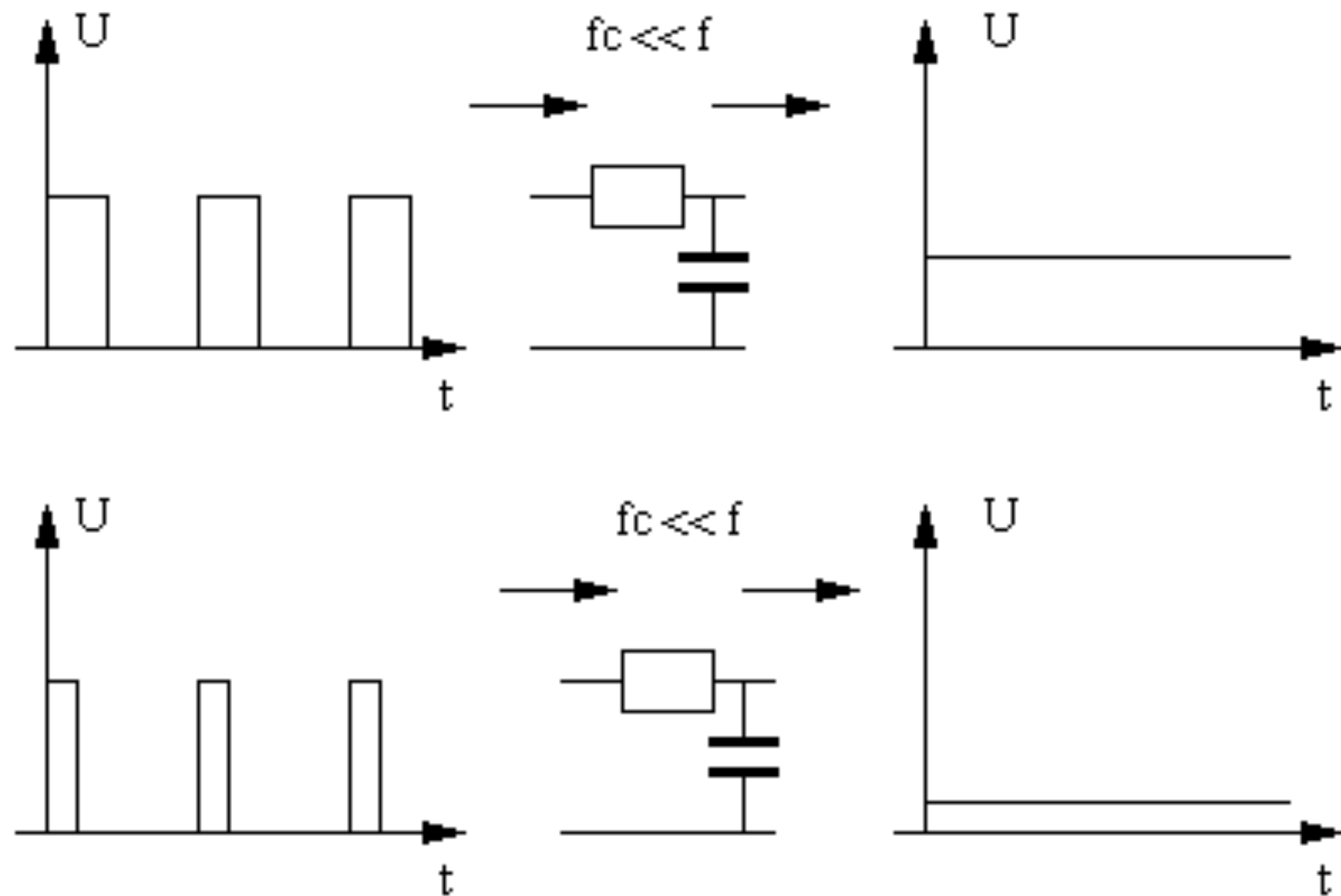
- Cấu hình CCP1 để bộ PWM hoạt động

CCP1CON REGISTER/CCP2CON REGISTER (ADDRESS 17h/1Dh)

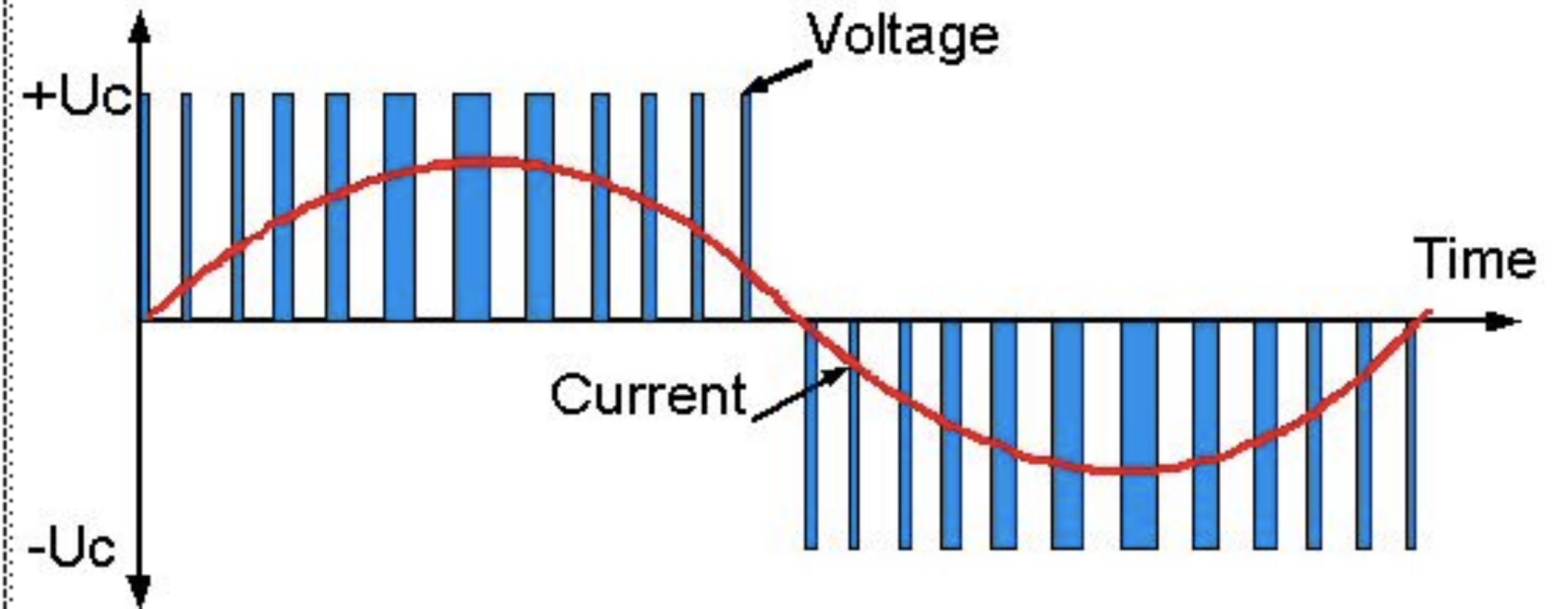
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
bit 7		bit 0					

11xx = PWM mode

Ứng dụng



Tạo ra điện áp DC



Tạo ra điện áp AC

Tín hiệu đưa ra từ PIC có công suất nhỏ nên cần KĐ, tín hiệu PWM từ PIC thường được dùng để đóng/mở các transistor trường khuếch đại công suất (không sử dụng trực tiếp)

PWM trong CCS

`#use pwm(options)`

Options có thể là một hay nhiều trường sau đây, phân biệt với nhau bởi dấu phẩy

pwm_x	Chọn pwm thứ x của VĐK	
output=pin_{xx}	chọn chân đầu ra cho tín hiệu PWM	là một trong các chân CCP
timer=x	chọn timer để thực hiện PWM	mặc định là Timer2
frequency=x	chọn thông số Period cho PWM	Khi dịch chương trình, chọn 1 trong 2, lỗi cái này chọn cái kia
period=x	Tương tự frequency=x	
duty=x	Chọn tỉ lệ duty cycle tính theo %	mặc định là 50%
pwm_on	Khởi tạo bộ PWM chế độ ON	
pwm_off	Khởi tạo bộ PWM chế độ OFF	

Chú ý cần có `#use delay` trước khi khai báo pwm

PWM trong CCS

`set_pwm_duty(x,value)` Cập nhật giá trị duty cycle cho bộ PWM thứ x một giá trị value mới

`setup_ccp1(mode)` Chọn chế độ capture/compare hoặc PWM cho bộ CCP1

`set_pwm1_duty(value)` Cập nhật giá trị duty cycle cho bộ PWM thứ 1 một giá trị value mới

`setup_pwm1(settings)` Cài đặt giá trị cho bộ PWM tương ứng (Ví dụ là bộ PWM1)

pwm_enabled	Khởi tạo bộ PWM tương ứng
pwm_output	Khởi tạo chân PWM mặc định
pwm_active_low	ON state ở mức thấp

`setup_pwm1(pwm_enabled|pwm_output)`