PWM PULSE WIDTH MODULATION

1. CONTROL OF DC MOTORS

Peatman P 112 + KLAFTER

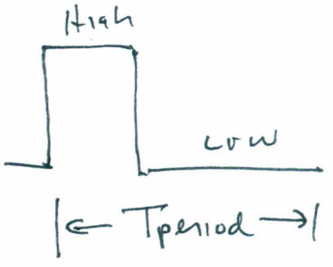
↓ POSITIVE PULSES

↓ SQUARE WAVE −V TO +V

2. POWER CONTROL (POSITIVE PULSES)

$P \propto V^2$

$$V_{avg}^2 = \frac{1}{T_{period}} \int_0^{T_{High}} V^2 dt$$



ie say DUTY cycle 50%

$$V_{avg}^2 = \frac{1}{T} \int_0^{T/2} V^2 dt = \frac{1}{T} V^2 t \Big|_0^{T/2}$$

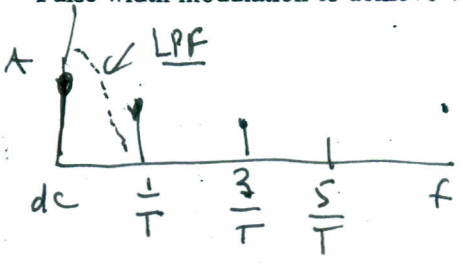
$$= \underline{V^2/2} \quad (50\%)$$

3. USE WITH FILTERS TO CREATE OTHER WAVES

FIGURE 2.21
Features of the Intel 8096 and Motorola 68HC11 A/D converters.

Waveform	Duty cycle	Average dc output
+5 V	100%	5.00 V
+5 V / 0 V	75%	3.75 V
+5 V / 0 V	50%	2.50 V
+5 V / 0 V	25%	1.25 V
0 V	0%	0.00 V

FIGURE 2.22
Pulse-width modulation to achieve variable average dc output.



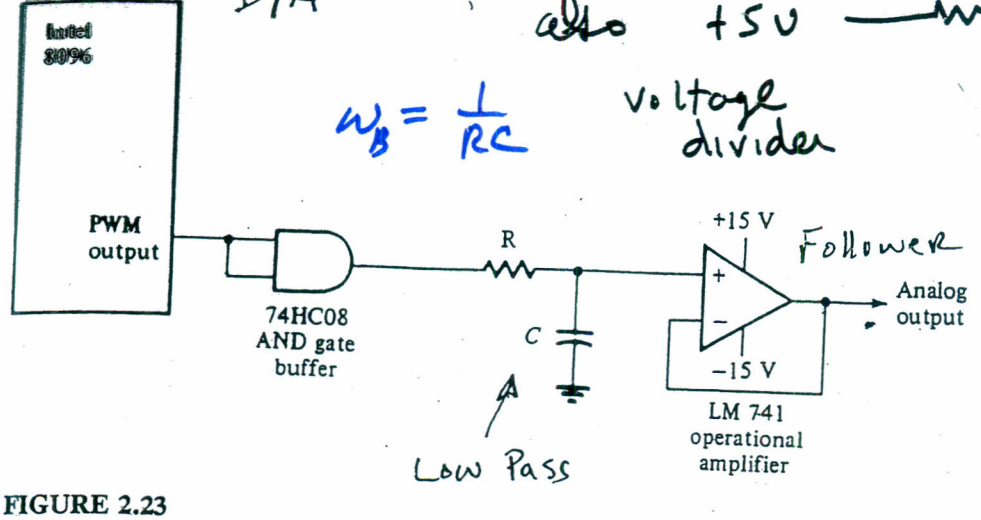


FIGURE 2.23
D/A conversion via the dc component of a pulse-width modulated waveform.

and the same comments concerning the advantages and disadvantages of both are pertinent (see Figures 4.11.1 and 4.11.2). However, unlike the linear case, the output voltage of the T or H circuit will be almost equal to the *full value* of either the positive or negative dc supply voltage (see Figure 4.11.3).

How can these types of signals provide the required variation in armature voltage and hence rotor speed? The answer to this question is found by recognizing that the servomotor is a low-pass filter [e.g., see the transfer function in Eq. (4.3.5)]. With T_S defined as the period of the switching signal waveform, then if the radian switching frequency $\omega_S = 2\pi/T_S >> \omega_E$, the electrical pole of the motor (i.e., $\omega_S > 100\,\omega_E$), the filtering action of the motor will cause the effective armature voltage to be the "average value" of the waveforms in Figure 4.11.3.* Mathematically, this means that

$$(V_{\text{arm}})_{\text{ave}} = \frac{1}{T_S}\int_0^{T_S} V_{\text{arm}}(t)\,dt \tag{4.11.1}$$

Thus applying Eq. (4.11.1) to the waveforms in Figure 4.11.3, it is seen by inspection that the motor will not move for the square wave in part (a) because $(V_{\text{arm}})_{\text{ave}} = 0$, whereas the nonzero average value of this quantity for the waveforms in (b) and (c) will produce rotor motion. It is important to understand that Eq. (4.11.1) will not be strictly correct if the switching frequency is too low. For example, if it is only about 10 times higher than the electrical pole of the motor, the effective armature voltage will be somewhat less than the average value and the armature current may exhibit significant ripple (see Problem 4.33).

In actual use, a PWM servomotor drive can be made to produce practically any type of acceleration, velocity, or position profile that might be required in a given application. For example, if it is desired to cause a servomotor to turn with a trapezoidal velocity profile (see Figure 4.6.7), this can be achieved by making the pulse width, T_p in Figure 4.11.3, vary trapezoidally with time (see Problem 4.34). In a robotic application the joint processor converts the velocity error samples into equivalent values of T_p. This is accomplished by causing the associated control logic to command the appropriate power transistor(s) in the PWM amplifier to turn on for T_p milliseconds. In view of the discussion of the preceding paragraph, faithful reproduction of the desired profile will occur only provided that the switching frequency is "high enough." This statement, in effect, implies that the frequency must be chosen so that the *sampling theorem* is satisfied.

Unlike the linear servo amplifier, there is another cause of power dissipation in a PWM device, and this places a practical upper limit on the switching frequency.

---

*Recall that a periodic waveform such as a square wave can be represented by a Fourier series:

$$V_{\text{arm}}(t) = V_{\text{dc}} + \sum_{n=1}^{\infty} [A_n \cos(n\omega_S t) + B_n \sin(n\omega_S t)]$$

If this signal is passed through a *low-pass* filter network with a cutoff frequency below ω_S (and hence $n\omega_S$), only the dc term will be transmitted, and the output will be V_{dc}.

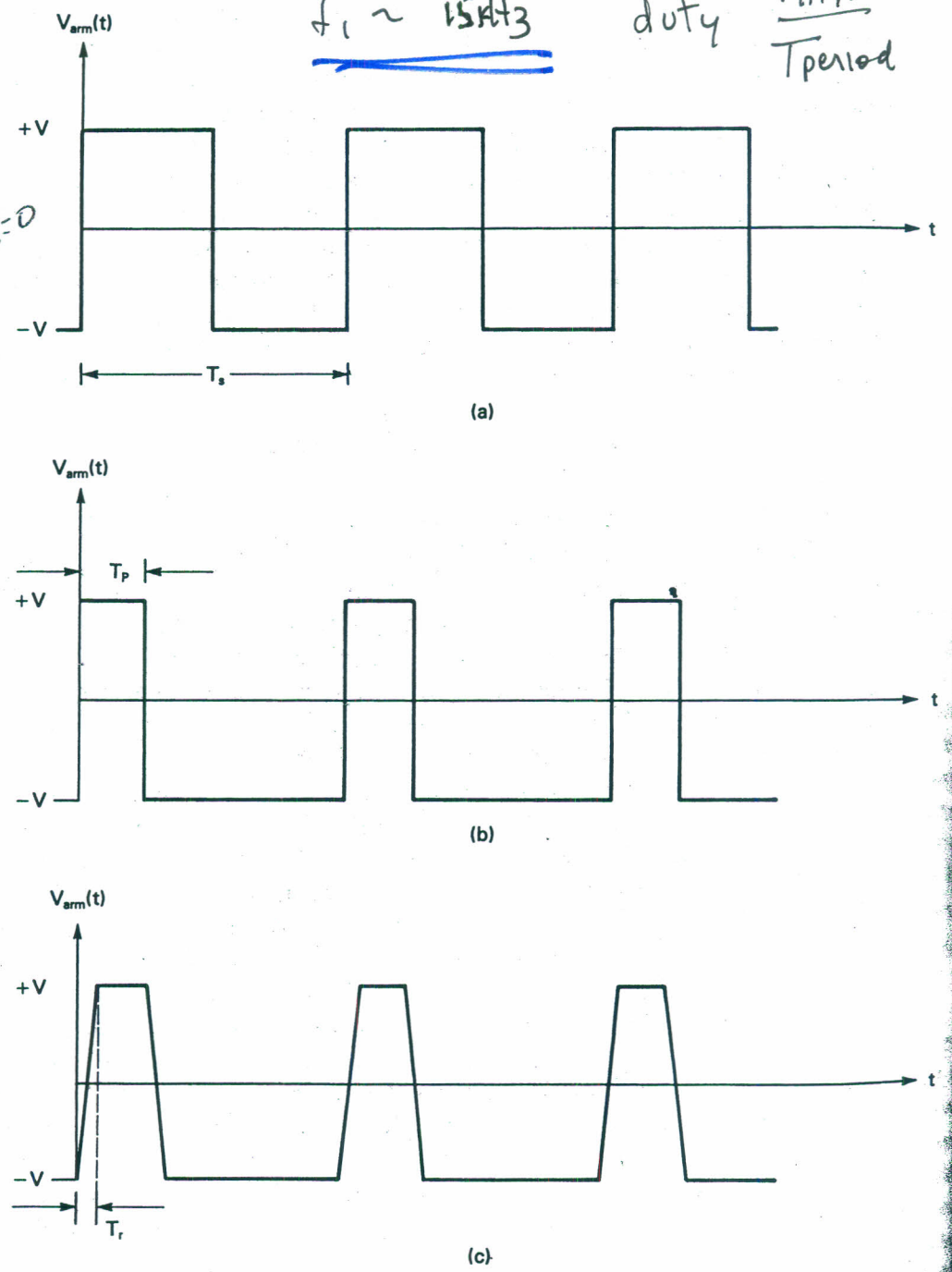


Figure 4.11.3. Typical PWM waveforms: (a) no load PWM output, ideal switch, $(V_{arm})_{ave} = 0$, motor does not turn; (b) loaded PWM output, ideal switch, $(V_{arm})_{ave} = -V/2$, motor turns CCW; (c) same as part (b), except nonideal switch and power transistors in active region during T_r.

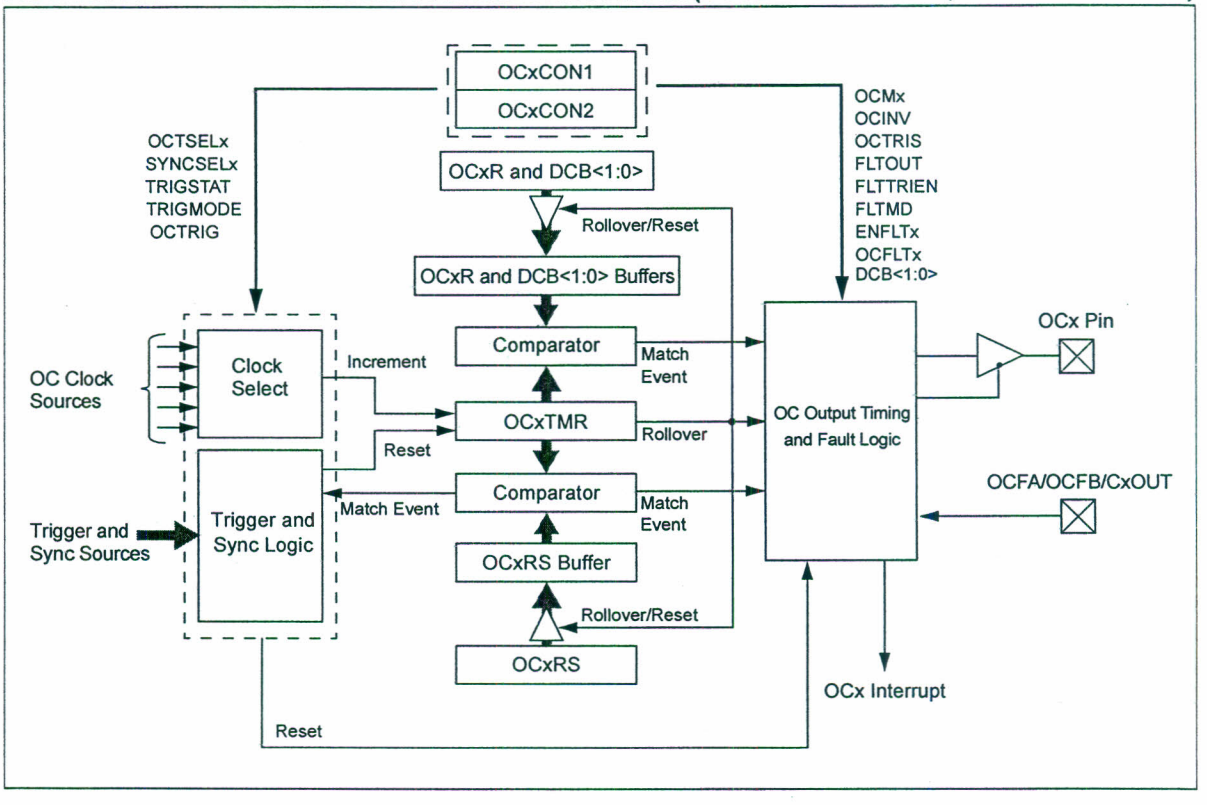
15.3 Pulse-Width Modulation (PWM) Mode

In PWM mode, the output compare module can be configured for edge-aligned or center-aligned pulse waveform generation. All PWM operations are double-buffered (buffer registers are internal to the module and are not mapped into SFR space).

To configure the output compare module for edge-aligned PWM operation:

1. Calculate the desired on-time and load it into the OCxR register.
2. Calculate the desired period and load it into the OCxRS register.
3. Select the current OCx as the synchronization source by writing 0x1F to SYNCSEL<4:0> (OCxCON2<4:0>) and '0' to OCTRIG (OCxCON2<7>).
4. Select a clock source by writing the OCTSEL2<2:0> (OCxCON<12:10>) bits.
5. Enable interrupts, if required, for the timer and output compare modules. The output compare interrupt is required for PWM Fault pin utilization.
6. Select the desired PWM mode in the OCM<2:0> (OCxCON1<2:0>) bits.
7. If a timer is selected as a clock source, set the TMRy prescale value and enable the time base by setting the TON (TxCON<15>) bit.

FIGURE 15-2: OUTPUT COMPARE BLOCK DIAGRAM (DOUBLE-BUFFERED, 16-BIT PWM MODE)



15.3.1 PWM PERIOD

In Edge-Aligned PWM mode, the period is specified by the value of the OCxRS register. In Center-Aligned PWM mode, the period of the synchronization source, such as the Timers' PRy, specifies the period. The period in both cases can be calculated using Equation 15-1.

EQUATION 15-1: CALCULATING THE PWM PERIOD(1)

PWM Period = [Value + 1] x TCY x (Prescaler Value)

Where:

Value = OCxRS in Edge-Aligned PWM mode and can be PRy in Center-Aligned PWM mode (if TMRy is the sync source).

Note 1: Based on TCY = TOSC * 2; Doze mode and PLL are disabled.

15.3.2 PWM DUTY CYCLE

The PWM duty cycle is specified by writing to the OCxRS and OCxR registers. The OCxRS and OCxR registers can be written to at any time, but the duty cycle value is not latched until a period is complete. This provides a double buffer for the PWM duty cycle and is essential for glitchless PWM operation.

Some important boundary parameters of the PWM duty cycle include:

- Edge-Aligned PWM:
  - If OCxR and OCxRS are loaded with 0000h, the OCx pin will remain low (0% duty cycle).
  - If OCxRS is greater than OCxR, the pin will remain high (100% duty cycle).
- Center-Aligned PWM (with TMRy as the sync source):
  - If OCxR, OCxRS and PRy are all loaded with 0000h, the OCx pin will remain low (0% duty cycle).
  - If OCxRS is greater than PRy, the pin will go high (100% duty cycle).

See Example 15-3 for PWM mode timing details. Table 15-1 and Table 15-2 show example PWM frequencies and resolutions for a device operating at 4 MIPS and 10 MIPS, respectively.

$f_{PWM} = \frac{1}{T}$

EQUATION 15-2: CALCULATION FOR MAXIMUM PWM RESOLUTION(1)

$$\text{Maximum PWM Resolution (bits)} = \frac{\log_{10}\left(\frac{F_{CY}}{F_{PWM} \bullet (\text{Prescale Value})}\right)}{\log_{10}(2)} \text{ bits}$$

Note 1: Based on FCY = FOSC/2, Doze mode and PLL are disabled.

EQUATION 15-3: PWM PERIOD AND DUTY CYCLE CALCULATIONS(1)

1. Find the OCxRS register value for a desired PWM frequency of 52.08 kHz, where FOSC = 8 MHz with PLL (32 MHz device clock rate) and a prescaler setting of 1:1 using Edge-Aligned PWM mode:

   TCY = 2 * TOSC = 62.5 ns
   PWM Period = 1/PWM Frequency = 1/52.08 kHz = 19.2 µs
   PWM Period = (OCxRS + 1) • TCY • (OCx Prescale Value)
   19.2 µs = (OCxRS + 1) • 62.5 ns • 1
   OCxRS = 306

2. Find the maximum resolution of the duty cycle that can be used with a 52.08 kHz frequency and a 32 MHz device clock rate:

   PWM Resolution = $\log_{10}(F_{CY}/F_{PWM})/\log_{10}2$) bits
   = ($\log_{10}$(16 MHz/52.08 kHz)/$\log_{10}$2) bits
   = 8.3 bits

Note 1: Based on TCY = 2 * TOSC; Doze mode and PLL are disabled.

15.4 Subcycle Resolution

The DCB bits (OCxCON2<10:9>) provide for resolution better than one instruction cycle. When used, they delay the falling edge generated from a match event by a portion of an instruction cycle.

For example, setting DCB<1:0> = 10 causes the falling edge to occur halfway through the instruction cycle in which the match event occurs, instead of at the beginning. These bits cannot be used when OCM<2:0> = 001. When operating the module in PWM mode (OCM<2:0> = 110 or 111), the DCB bits will be double-buffered.

The DCB bits are intended for use with a clock source identical to the system clock. When an OCx module with enabled prescaler is used, the falling edge delay caused by the DCB bits will be referenced to the system clock period rather than the OCx module's period.

TABLE 15-1: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 4 MIPS (FCY = 4 MHz)(1)

PWM Frequency	7.6 Hz	61 Hz	122 Hz	977 Hz	3.9 kHz	31.3 kHz	125 kHz
Prescaler Ratio	8	1	1	1	1	1	1
Period Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.

TABLE 15-2: EXAMPLE PWM FREQUENCIES AND RESOLUTIONS AT 16 MIPS (FCY = 16 MHz)(1)

PWM Frequency	30.5 Hz	244 Hz	488 Hz	3.9 kHz	15.6 kHz	125 kHz	500 kHz
Prescaler Ratio	8	1	1	1	1	1	1
Period Value	FFFFh	FFFFh	7FFFh	0FFFh	03FFh	007Fh	001Fh
Resolution (bits)	16	16	15	12	10	7	5

Note 1: Based on FCY = FOSC/2; Doze mode and PLL are disabled.