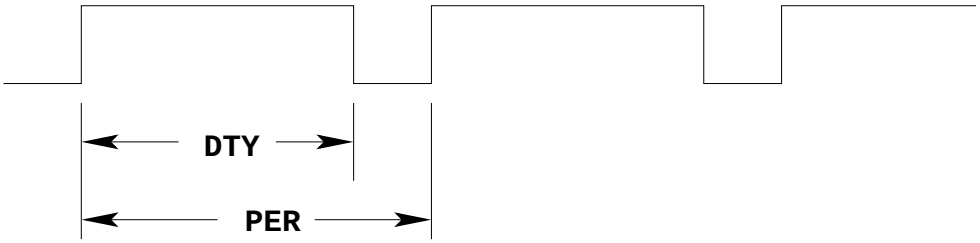
The HC12 Pulse Width Modulation System

Pulse Width Modulation

Control speed of motor by adjusting percent of time power is applied to the motor.

Need to choose period, and have a way to adjust duty cycle



- The HC12 has a flexible, and complicated, PWM system
- There are four 8-bit PWM channels
  - Two 8-bit channels can be combined into a single 16-bit channel
  - We will discuss only 8-bit mode
- You can select center-aligned or left-aligned PWM
  - We will discuss only left-aligned mode
- You can select high polarity or low polarity
  - We will discuss only high polarity mode

The HC12 Pulse Width Modulation System

- There are 14 registers you need to program to set up the four channels of the PWM
- To select 8-bit mode, write a 0 to Bits 7 and 6 of `PWCLK` register
- To select left-aligned mode, write a 0 to Bit 3 of `PWCTL` register
- To select high polarity mode, write a 1 to Bits 0, 1, 2, and 3 of `PWPOL` register
- To set the period for a PWM channel you need to program bits in 4 of the PWM registers
  - For Channel 0 the 4 registers are `PWCLK`, `PWPOL`, `PWSCAL0` and `PWPER0`
  - For Channel 1 the 4 registers are `PWCLK`, `PWPOL`, `PWSCAL0` and `PWPER1`
  - For Channel 2 the 4 registers are `PWCLK`, `PWPOL`, `PWSCAL1` and `PWPER2`
  - For Channel 3 the 4 registers are `PWCLK`, `PWPOL`, `PWSCAL1` and `PWPER3`
- To set the duty cycle for a PWM channel you need to write to the `PWDTY` register for that channel

CON23	CON01	PCKA2	PCKA1	PCKA0	PCKB2	PCKB1	PCKB0	0x0040	PWCLK

CON23 – Concatenate PWM 2 & PWM 3 into one 16-bit PWM
CON01 – Concatenate PWM 0 & PWM 1 into one 16-bit PWM

We will discuss 8-bit PWM only -- write 0 to CON23 and CON01

PWCLK = PWCLK & ~0xC0;

PCKA[2-0]: Prescaler for Clock A Divide 8 MHz clock by 2 PCKA[2-0]

PCKB[2-0]: Prescaler for Clock B Divide 8 MHz clock by 2 PCKB[2-0]

PCLK3	PCLK2	PCLK1	PCLK0	PPOL3	PPOL2	PPOL1	PPOL0	0x0041	PWPOL

PCLKn – Choose clock source for Channel n

CH3 can use either B (0) or S1 (1)

CH2 can use either B (0) or S1 (1)

CH1 can use either A (0) or S0 (1)

CH0 can use either A (0) or S0 (1)

$$S1 = \frac{B}{2 \times (PWSCAL1 + 1)}$$

$$S0 = \frac{A}{2 \times (PWSCAL0 + 1)}$$

PPOLn – Choose polarity

1 => high polarity
0 => low polarity

We will use high polarity only

PWPOL = PWPOL | 0x0F;

With high polarity, duty cycle will be the amount of time output is high.

Bit 7	6	5	4	3	2	1	Bit 0	0x0044	PSCAL0

PWSCAL0 adjusts frequency of Clock S0

Bit 7	6	5	4	3	2	1	Bit 0	0x0046	PSCAL1

PWSCAL1 adjusts frequency of Clock S1

PWPERn sets period of CHn

PWM Period = (PWPERn + 1) x Period of PWM Clock n

PWDTYn adjusts duty cycle of CHn

PWM Duty Cycle = (PWDTYn + 1)/(PWPERn + 1) x 100%

0	0	0	PSWAI	CENTR	RDPP	PUPP	PSBCK	0x0054	PWCTL

The only bit we will use is CENTR

Always set CENTR = 0 to choose left-aligned mode

```
PWCTL = 0x00;
```

0	0	0	0	PWEN3	PWEN2	PWEN1	PWEN0	0x0042	PWEN

Set PWENn = 1 to enable PWM on CHn

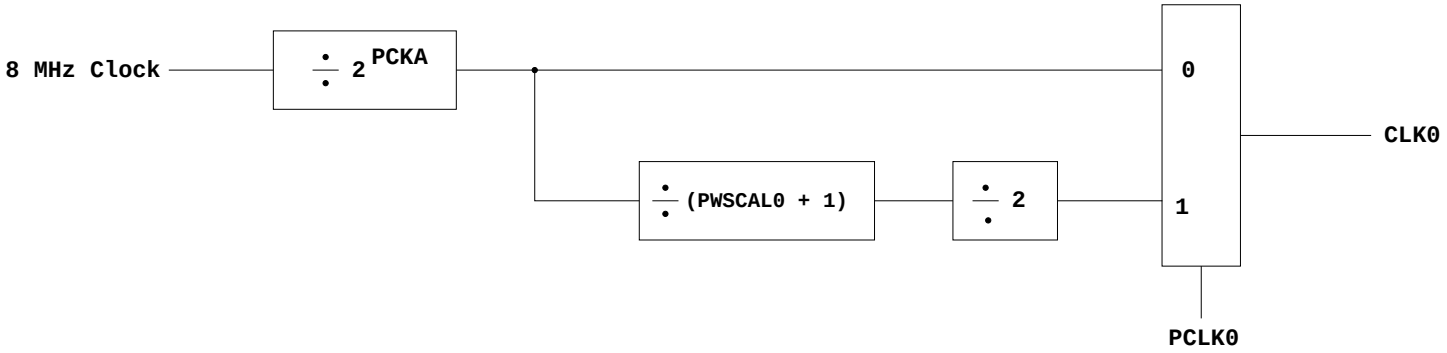
If PWENn = 0, the Port P Bit n can be used as general purpose I/O

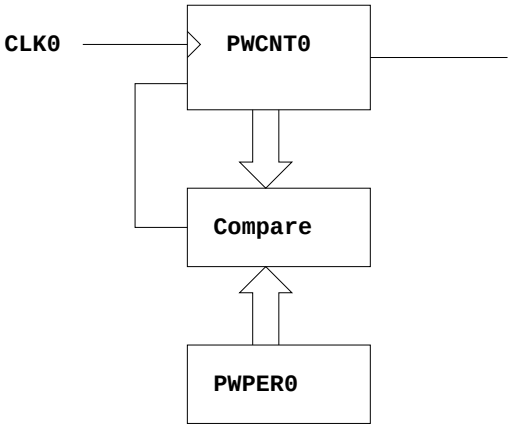
Bits 6–4 of Port P are always used for general purpose I/O

```
To enable PWM Channel 2:  PWEN = PWEN | 0x04:
```

Clock Select for PWM Channel 0

You need to set PCKA, PWSCAL0, PCLK0, and PWPER0





PWCNT0 counts from 0 to PWPER0, then back to zero

It takes (PWPER0 + 1) periods of CLK0 to make one Ch0 period

Ch0 Period = (PWPER0 + 1) x CLK0 Period

$$= \begin{cases} (PWPER0 + 1) \times (2^{PCKA}) & PCLK0 = 0 \\ (PWWER0 + 1) \times (2^{PCKA + 1}) \times (PWSCAL0 + 1) & PCLK0 = 1 \end{cases}$$

How to set the Period for PWM Channel 0

- To set the period for PWM Channel 0:
  - Set the PWM Period register for Channel 0, `PWPER0`
  - CLK0, the clock for Channel 0, drives a counter (`PWCNT0`)
  - `PWCNT0` counts from 0 to `PWPER0`
  - The period for PWM Channel 0 is (`PWPER0` + 1) $\times$ Period of CLK0
- There are two modes for the clock for PWM Channel 0
  - You select the mode by the `PCLK0` bit
  - If `PCLK0` == 0, CLK0 is generated by dividing the 8 MHz clock by 2^{PCKA}, where PCKA is between 0 and 7
  - If `PCLK0` == 1, CLK0 is generated by dividing the 8 MHz clock by $2^{\mathrm{PCKA}+1}\times$ (`PWSCAL0` + 1), where PCKA is between 0 and 7 and `PWSCAL0` is between 0 and 255
- The Period for PWM Channel 0 (in number of 0.125 μs cycles) is calculated by

$$\mathrm{Period} = \begin{cases} (\mathtt{PWPER}+1)\times 2^{\mathtt{PCKA}} & \text{if } \mathtt{PCLK0} == 0 \\ (\mathtt{PWPER}+1)\times 2^{\mathtt{PCKA}+1}\times(\mathtt{PWSCAL0}+1) & \text{if } \mathtt{PCLK0} == 1 \end{cases}$$

- With `PCLK0` == 0, the maximum possible PWM period is 4.1 ms
- With `PCLK0` == 1, the maximum possible PWM period is 2.1 s

- To get a 1 ms PWM period, you need 8,000 cycles.

$$8,000 = \begin{cases} (\texttt{PWPER}+1) \times 2^{\texttt{PCKA}} & \text{if } \texttt{PCLK0} == 0 \\ (\texttt{PWPER}+1) \times 2^{\texttt{PCKA}+1} \times (\texttt{PWSCAL0}+1) & \text{if } \texttt{PCLK0} == 1 \end{cases}$$

- You can do this in many ways
  - With PCLK0 = 0, can have

PCKA	PWPER0	
5	249	Exact
6	124	Exact
7	62	Close

  - With PCLK0 = 1, can have

PCKA	PWSCAL0	PWPER0	
0	15	249	Exact
0	19	199	Exact
0	31	124	Exact
0	79	49	Exact
0	159	24	Exact
1	7	249	Exact
1	15	124	Exact
2	3	249	Exact
2	7	124	Exact
3	1	249	Exact

and many other combinations

- You want PWPER0 to be large (say, 99 or larger)
  - If PWPER0 is small, you don't have much control over the duty cycle
  - For example, if PWPER0 = 4, you can only have 0%, 25%, 50%, 75% or 100% duty cycle
- Once you choose a way to set the PWM period, you can program the PWM registers
- For example, to get a 1 ms period, let's use PCLK0 = 1, PCKA = 0, PWSCAL0 = 19, and PWPER0 = 199
- We need to do the following:
  - Write 0 to bits 7 and 6 of PWCLK (to set up 8-bit mode)
  - Write 000 to Bits 5,4,3 of PWCLK (to set PCKA to 0)
  - Write 1 to Bit 4 of PWPOL (to set PCLK0 = 1)
  - Write 1 to Bit 0 of PWPOL (to select high polarity mode)
  - Write 19 to PWSCAL0
  - Write 0 to Bit 3 of PWCTL (to select left aligned mode)
  - Write 199 to PWPER0
  - Write 1 to Bit 0 of PWEN (to enable PWM on Channel 0)
  - Write the appropriate value to PWDTY0 to get the desired duty cycle (e.g., PWDTY0 = 119 will give 60% duty cycle)

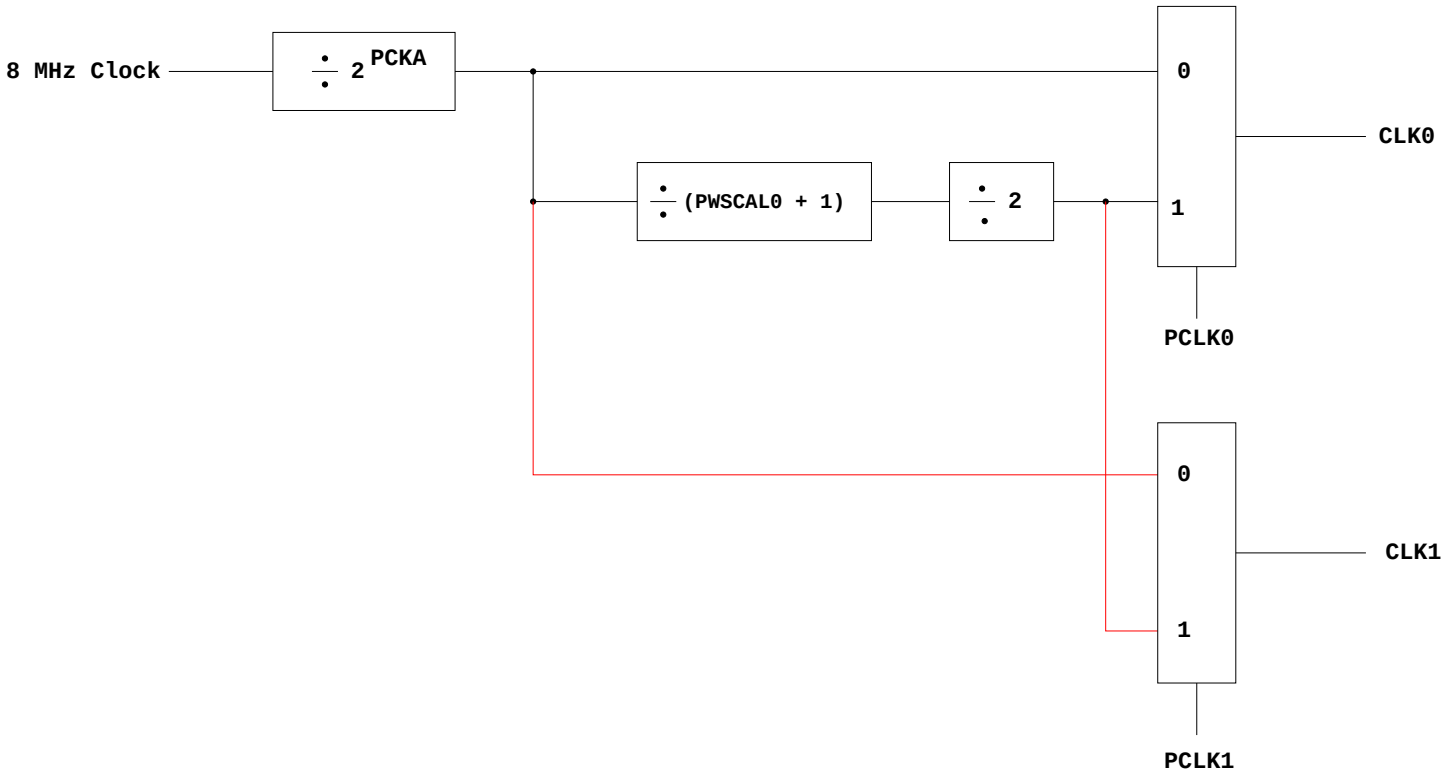
C code to set up PWM Channel 0 for 1 ms period (1 kHz frequency) PWM with 60% duty cycle

```
PWCLK = PWCLK & ~0xC0;  /* 8-bit Mode */
PWCLK = PWCLK & ~0x38;  /* PCKA = 0 */
PWPOL = PWPOL | 0x10;   /* PCLK0 = 1 */
PWPOL = PWPOL | 0x01;   /* High polarity mode */
PWSCAL0 = 19;
PWCTL = PWCTL & ~0x08;  /* Left-Aligned */
PWPER0 = 199;
PWEN = PWEN | 0x01;     /* Enable PWM Channel 0 */
PWDTY0 = 119;           /* 60% duty cycle on Channel 0 */
```

Interdependence of clocks for Channels 0 and 1

- The clocks for Channels 0 and 1 are interdependent
- They both use PCKA and PWSCAL0
- To set the clock for Channel 0, you need to set PCKA, PCLK0, PWSCAL0 (if PCLK0 == 1) and PWPER0
- To set the clock for Channel 1, you need to set PCKA, PCLK1, PWSCAL0 (if PCLK1 == 1) and PWPER1

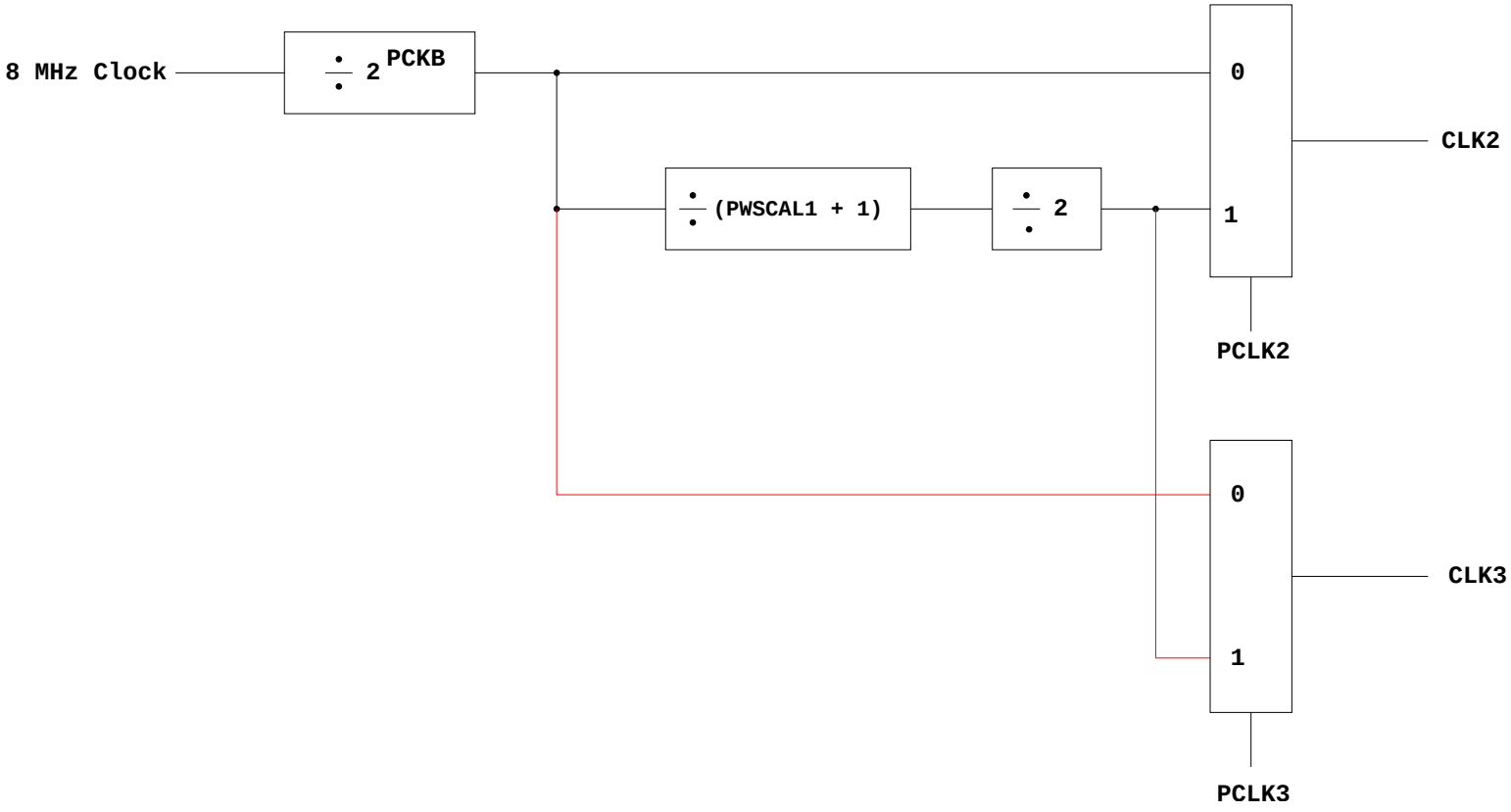
Clock Select for PWM Channels 0 and 1



PWM Channels 2 and 3

- PWM channels 2 and 3 are similar to PWM channels 0 and 1
- To set the clock for Channel 2, you need to set PCKB, PCLK2, PWSCAL1 (if PCLK2 == 1) and PWPER2
- To set the clock for Channel 3, you need to set PCKB, PCLK3, PWSCAL1 (if PCLK3 == 1) and PWPER3

Clock Select for PWM Channels 2 and 3



Using the HC12 PWM

1. Choose 8-bit mode (`PWCLK 7:6` = 00)
2. Choose left-aligned (`PWCTL 3` = 0)
3. Choose polarity = 1 (`PWPOL 3:0` = 1111)
4. Select clock mode in `PWPOL`: `PCLKx` = 0 for 2^N,
   `PCLKx` = 1 for $2^{(N+1)} \times (M+1)$,
5. Select N in `PWCLK` register: `PCKA 2:0` for channels 1, 0; `PCKB 2:0` for channels 3, 2.
6. If `PCLKx` = 1, select M: `PWSCAL0 7:0` = M for channels 1, 0; `PWSCAL1 7:0` = M for channels 3, 2.
7. Select `PWPERx`, normally between 99 and 255.
8. Enable desired PWM channels: `PWEN 3:0`.
9. Select `PWDTYx`, normally between 0 and `PWPERx`. Then

$$\text{Duty Cycle x} = \frac{\texttt{PWDTYx} + 1}{\texttt{PWPERx} + 1} \times 100\%$$

   Change duty cycle to control speed of motor or intensity of light, etc.
10. For 0% duty cycle, choose `PWDTYx` = 0xFF.

Program to use the HC12 PWM System

```
/*
 * Program to generate 5.2 kHz pulse width modulation
 * on Port P Bits 0 and 1
 *
 * To get 5.2 kHz:   8,000,000/5,200 = 1538.5
 *
 * Cannot get exactly 1538.5

 * Use 1536, which is 2^9 x 3
 *
 * Lots of ways to set up PWM to achieve this.  One way is 2^3 x 192
 * Set PCKA to 3, do not use PSCAL0, set PWPER to 191
 *
 */
#include "hc12b32.h"

main()
{
    /* Choose 8-bit mode */
    PWCLK = PWCLK & ~0xC0;
    /* Choose left-aligned */
    PWCTL = PWCTL & ~0x08;
    /* Choose high polarity on all channels */
    PWPOL = PWPOL | 0x0f;
    /* Select clock mode 0 for Channels 1 and 0 (no PWSCAL0) */
    PWPOL = PWPOL & ~0x30;
    /* Select PCKA = 3 for Channels 1 and 0 */
    PWCLK = (PWCLK & ~0x20) | 0x18;    /* PCKA = 3 */
    /* Select period of 192 for Channels 1 and 0 */
    PWPER1 = 191;             /* Period 1 = PWPER1 + 1 */
    PWPER0 = 191;             /* Period 0 = PWPER0 + 1 */
    /* Enable PWM on Channels 1 and 0 */
    PWEN = PWEN | 0x03;

    PWDTY1 = 95;  /* 50% duty cycle on Channel 1 */
    PWDTY0 = 47;   /* 25% duty cycle on Channel 0 */

while (1)
{ /* Code to adjust duty cycle to meet requirements */ }
}
```