dsPIC33EP512MC810

PLL PWM PPT

SETUP FOR USING PLL WITH THE PRIMARY OSCILLATOR (POSC)

The following process is used to set up the PLL to operate the device at 60 MIPS with a **10 MHz** external crystal:

1. To execute instructions at 60 MHz, ensure that the required system clock frequency is:

   FOSC = 2 x FCY = 120 MHz

2. To set up the PLL and meet the requirements of the PLL, follow these steps:

a) Select the PLL postscaler to meet the VCO output frequency requirement (120 MHz < **FVCO** < 340 MHz).

Select a PLL postscaler ratio of N2 = 2

Ensure that FVCO = (FPLLO x N2) = **240** MHz

b) Select the PLL prescaler to meet the PFD input frequency requirement (0.8 MHz < FPLLI < 8.0 MHz).

- Select a PLL prescaler ratio of N1 = 2

  Ensure that FPLLI = (FIN ÷ N1) = **5** MHz

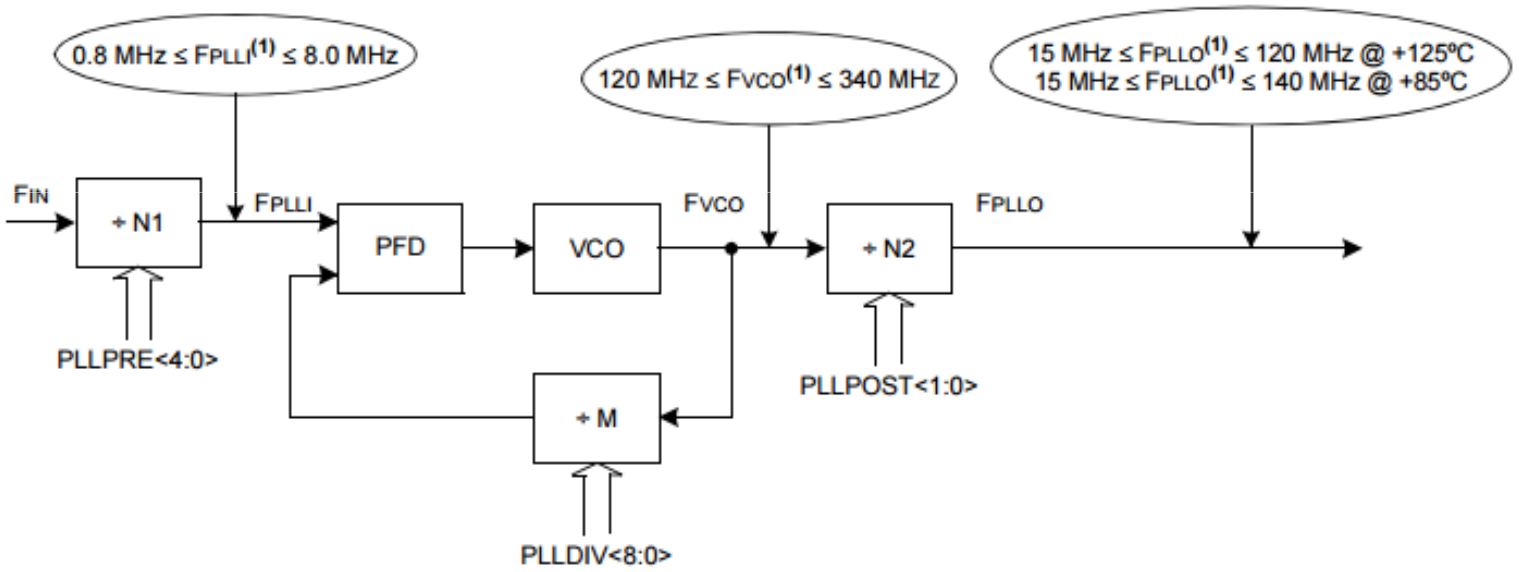
c) Select the PLL feedback divisor to generate the required VCO output frequency based on the PFD input frequency.

- FVCO = FPLLI x M
- M = FVCO ÷ FPLLI = 48

d) Configure the FNOSC bits (FOSCSEL) to select a clock source without the PLL (for example, Internal FRC) at Power-on Reset.

e) In the main program, change the PLL prescaler, PLL postscaler and PLL feedback divisor values to those just decided in the previous steps, and then perform a clock switch to the PLL mode.

Figure 7-1: dsPIC33/PIC24 Family PLL Block Diagram



Note 1: This frequency range must be met at all times.

Equation 7-1: FVCO Calculation

$$F_{VCO} = F_{IN} \times \left(\frac{M}{N1}\right) = F_{IN} \times \left(\frac{(PLLDIV + 2)}{(PLLPRE + 2)}\right)$$

Equation 7-2 provides the relation between Input Frequency (FIN) and Output Frequency (FPLLO).

Equation 7-2: FPLLO Calculation

$$F_{PLLO} = F_{IN} \times \left(\frac{M}{N1 \times N2}\right) = F_{IN} \times \left(\frac{(PLLDIV + 2)}{(PLLPRE + 2) \times 2(PLLPOST + 1)}\right)$$

Where,

$N1 = PLLPRE + 2$

$N2 = 2 \times (PLLPOST + 1)$

$M = PLLDIV + 2$

7.2.2 SETUP FOR USING PLL WITH 7.37 MHZ INTERNAL FRC

The following process is used to set up the PLL to operate the device at 60 MIPS with a 7.37 MHz Internal FRC.

1. To execute instructions at 60 MHz, ensure that the system clock frequency is: FOSC = 2 x FCY = 120 MHz 2. To set up the PLL and meet the requirements of the PLL, follow these steps:

a) Select the PLL postscaler to meet the VCO output frequency requirement (120 MHz < FVCO < 340 MHz). • Select a PLL postscaler ratio of N2 = 2 • Ensure that FVCO = (FPLLO x N2) = 240 MHz

b) Select the PLL prescaler to meet the PFD input frequency requirement (0.8 MHz < FPLLI < 8.0 MHz). • Select a PLL prescaler ratio of N1 = 2 • Ensure that FPLLI = (FIN ÷ N1) = 3.68 MHz

c) Select the PLL feedback divisor to generate the required VCO output frequency based on the PFD input frequency. • FVCO = FPLLI x M • M = FVCO ÷ FPLLI = 65

d) Configure the FNOSC bits (FOSCSEL) to select a clock source without PLL (for example, Internal FRC) at Power-on Reset.

e) In the main program, change the PLL prescaler, PLL postscaler and PLL feedback divisor to meet the user and PLL requirements, and then perform a clock switch to the PLL mode.

dsPIC33EP512MC810 PWM Code External 8MHz Crystal

```
// Configure PLL prescaler, PLL postscaler,
//PLL divisor
// ((8MHz / N1) x 60) / N2 = 120Mhz
//FOSC = 120Mhz
//Fcy = 120Mhz/2 = 60Mhz = 60MIPS

   PLLFBD = 58;              // M=60
   CLKDIVbits.PLLPOST=0;   // N2=2
   CLKDIVbits.PLLPRE=0;    // N1=2
```

dsPIC33EP512MC810 PWM Code External 8MHz Crystal

```
// Initiate Clock Switch to Primary Oscillator with PLL
// (NOSC=0b011)
   __builtin_write_OSCCONH(0x03);
   __builtin_write_OSCCONL(OSCCON | 0x01);

// Wait for Clock switch to occur
   while (OSCCONbits.COSC!= 0b011);

// Wait for PLL to lock
   while (OSCCONbits.LOCK!= 1);
}
```

dsPIC33EP512MC810 PWM Code External 7.37MHz Crystal

```
// Select Internal FRC at POR
 _FOSCSEL(FNOSC_FRC & IESO_OFF);
// Enable Clock Switching and Configure Primary
// Oscillator in XT mode
_FOSC(FCKSM_CSECMD & OSCIOFNC_OFF & POSCMD_NONE);
// ((7.37 MHz / N1) x 65) / N2 = 120Mhz
//FOSC = 120Mhz
//Fcy = 120Mhz/2 = 60Mhz = 60MIPS
int main()
{
// Configure PLL prescaler, PLL postscaler, PLL divisor
PLLFBD=63;                          // M=65
CLKDIVbits.PLLPOST=0;               // N2=2
 CLKDIVbits.PLLPRE=1;               // N1=3
```

dsPIC33EP512MC810 PWM Code External 7.37MHz Crystal

```
// Initiate Clock Switch to FRC oscillator with PLL
//(NOSC=0b001) __builtin_write_OSCCONH(0x01);
   __builtin_write_OSCCONL(OSCCON | 0x01);
 // Wait for Clock switch to occur
while (OSCCONbits.COSC!= 0b001);
// Wait for PLL to lock
while (OSCCONbits.LOCK!= 1);
```

dsPIC33EP512MU810 ..Page 71

Equation 14-1: PERIOD, PHASEx and SPHASEx Register Value Calculation for Edge-Aligned Mode

$$PTPER,\ PHASEx,\ SPHASEx = \frac{FOSC}{FPWM * PWM\ Input\ Clock\ Prescaler}$$

Where:
FPWM = Desired PWM frequency
FOSC = Oscillator output (120 MHz for 60 MIPS)
PWM Input Clock Prescaler = Value defined in the PCLKDIV<2:0> bits (PTCON2<2:0)

Based on Equation 14-1, while operating in the master time base (PTPER register) or the independent time base (PHASEx and SPHASEx registers), the register value to be loaded is shown in Example 14-38.

Example 14-38: PWM Time Period Calculation for Edge-Aligned Mode

$$PTPER = \frac{120MHz}{20kHz \times 1} = 6000$$

Where:
Desired PWM Switching Frequency = 20 kHz
PWM Input Clock Prescaler = 1:1
System Oscillator Frequency (FOSC) = 120 MHz

dsPIC33EP512MU810 .. Page 72

Equation 14-2: PHASEx or SPHASEx Register Value Calculation in Center-Aligned Mode

$$PHASEx, SPHASEx = \frac{FOSC}{FPWM \cdot PWM\ Input\ Clock\ Prescaler \cdot 2}$$

Based on Equation 14-2, when operating in independent time bases (PHASEx and SPHASEx registers), the register value to be loaded is shown in Example 14-39.

Example 14-39: PWM Time Period Calculation Example in Center-Aligned Mode

$$PHASEx, SPHASEx = \frac{120\ MHz}{20\ kHz \cdot 1 \cdot 2} = 3000$$

Where:
PWM Frequency (FPWM) = 20 kHz
PWM Input Clock Prescaler = 1:1
System Oscillator Frequency (Fosc) = 120 MHz

dsPIC33EP512MU810 .. Page 72

Example 14-40: Clock Prescaler Selection

```
/* Select PWM time base input clock prescaler */
/* Choose divide ratio of 1:2 */

PTCON2bits.PCLKDIV = 1;
```

Example 14-41: PWM Time Period Selection

```
/* Select Time Base Period Control */
/* Choose one of these options */

PWMCON1bits.ITB = 0;   /* PTPER provides the PWM time period value */
PWMCON1bits.ITB = 1;   /* PHASEx/SPHASEx provides the PWM time period value */
```

Example 14-42: PWM Time Period Initialization

```
/* Choose PWM time period based on FRC input clock */
/* PWM frequency is 100 kHz */
/* Choose one of the following options */

PTPER = 4808;

PHASEx = 4808;

SPHASEx = 4808;
```

Equation 14-3: MDC, PDCx and SDCx Calculation

$$MDC,\ PDCx,\ and\ SDCx = \frac{Fosc}{FPWM \cdot PWM\ Input\ Clock\ Prescaler} \cdot Desired\ Duty\ Cycle$$

Where:

$FPWM$ = PWM Frequency

$Fosc$ = System Oscillator Output

PWM Input Clock Prescaler = Value defined in the PCLKDIV<2:0> bits (PTCON<2:0>)

Desired Duty Cycle = Value between 0 and 1 for desired duty cycle

dsPIC33EP512MU810 .. Page 75

Equation 14-4: Bit Resolution Calculation for Edge-Aligned Mode

$$Bit\ Resolution = \log 2\left[\frac{FOSC}{FPWM \cdot PWM\ Input\ Clock\ Prescaler}\right]$$

Equation 14-5: Bit Resolution Calculation for Center-Aligned Mode

$$Bit\ Resolution = \log 2\left[\frac{FOSC}{FPWM \cdot PWM\ Input\ Clock\ Prescaler \cdot 2}\right]$$

Example 14-43: PWM Duty Cycle Selection

```
PWMCON1bits.MDCS = 0;          /* PDCx/SDCx provides duty cycle value */
PWMCON1bits.MDCS = 1;          /* MDC provides duty cycle value */
```

Example 14-44: PWM Duty Cycle Initialization

```
/* Initialize PWM Duty Cycle Value */

PDC1 = 2404;    /* Independent Primary Duty Cycle is 50% of the period */

SDC1 = 2404;    /* Independent Secondary Duty Cycle is 50% of the period */

MDC = 2404;     /* Master Duty Cycle is 50% of the period */
```

dsPIC33EP512MU810 .. Page 92

Equation 14-6: Dead Time Calculation

$$ALTDTRx, DTRx = FOSC * \frac{Desired\ Dead\ Time}{PWM\ Input\ Clock\ Prescaler}$$

Desired Dead Time Calculation

//ALDTRx, DTRx = Fosc x

(Desired Dead Time /

PreScaler)

Fosc = 120MHz

Desired Dead Timer = 500ns

PreScaler = 1

// Set Dead Time Values ??

DTR1 = DTR2 = DTR3 = 60;

ALTDTR1 = ALTDTR2 = ALTDTR3 = 60;