Following describes the correct way of updating PWM registers in a 4-phase variable frequency application where 4 PWM modules are operating at a constant 90 degrees phase shift.

1. The user should use shadow load mode for CMP and Period registers for all 4 PWM modules.

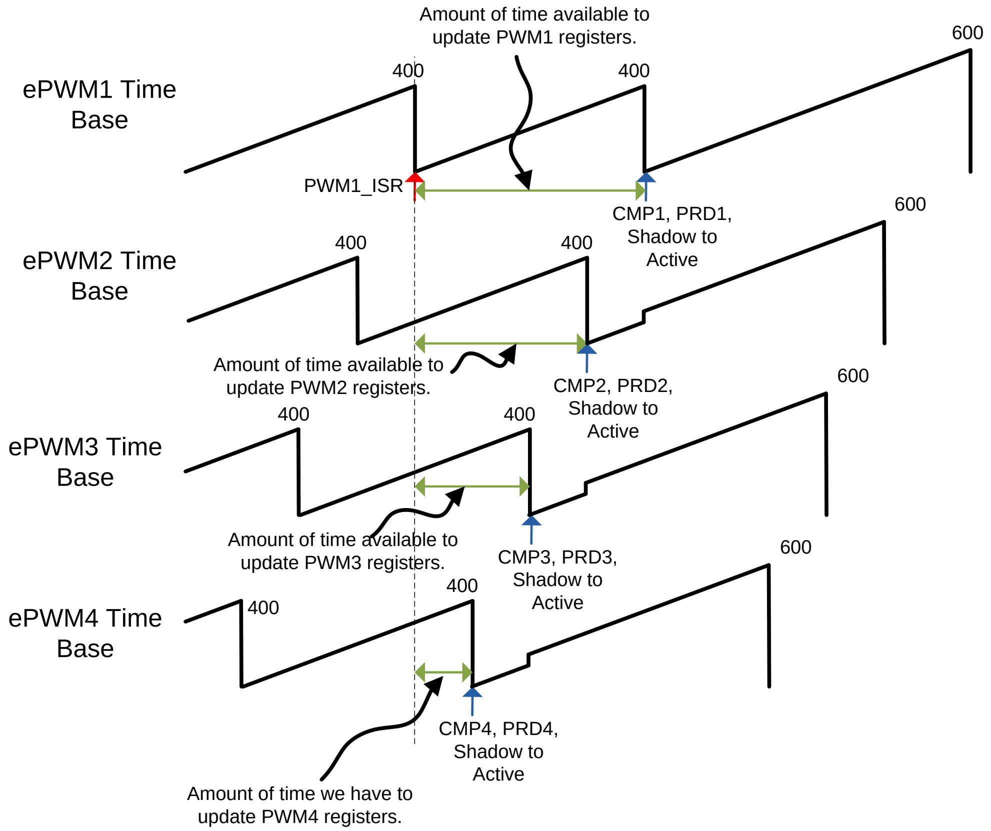
2. The user should update the Period, CMP and Phase registers **inside a PWM ISR** which should be configured to trigger at a known good point in the switching cycle. Here, an up-count mode operation of the PWM is assumed. But the following discussion is equally applicable to up-down and down count modes of operation as well.

3. If the control loop is run inside a separate slower/fixed frequency ISR/loop that ISR/loop should write the new PWM register values to dummy variables first. Once new register values are calculated and written to dummy variables, the PWM ISR should be enabled. The PWM ISR will then trigger at a known good point in the cycle. PWM ISR can be disabled from within the ISR at the end of its code. This enabling and disabling of the PWM ISR can save valuable processing bandwidth.

4. PWM registers should be updated from the dummy variables inside the PWM ISR.

5. Here, assuming PWM1 as master, PWM1_ISR is generated at CTR = ZERO of PWM1 time base counter as shown below. PWM registers for all 4 PWM modules are updated inside this ISR.

6. It is very important how and when the registers are updated inside this ISR. The amount of time/cycle counts available to update PWM registers for all PWM modules should be calculated for the fastest possible frequency. These update times must be met for each of the PWM modules. For this example (see figure below): Old period = 400 counts, Old phase = 100, New period = 600 counts and New phase = 150. If the highest possible frequency corresponds to PWM period of 400, the user will have 100 cycles from the point where PWM1_ISR is triggered to the point where PWM4 registers are updated. Inside the ISR the user should write to PWM4 registers first, followed by PWM3, then PWM2 and finally PWM1 registers.


Amount of time available to update PWM1 registers.
600
400
400
ePWM1 Time Base
PWM1_ISR
CMP1, PRD1, Shadow to Active
600
400
400
ePWM2 Time Base
Amount of time available to update PWM2 registers.
CMP2, PRD2, Shadow to Active
600
400
400
ePWM3 Time Base
Amount of time available to update PWM3 registers.
CMP3, PRD3, Shadow to Active
600
400
400
ePWM4 Time Base
CMP4, PRD4, Shadow to Active
Amount of time we have to update PWM4 registers.

7. The PWM1_ISR can be disabled at the end of its ISR code such that it is only triggered when new values have been calculated and written to the dummy variables.

This particular configuration is one way of implementing this solution. There are other valid variations of this solution that are possible.