Review Questions: PWM

Timer (Part 2)

Review Questions

1. Can a Timer configured in PWM mode count UP?
   - No. It works in the count DOWN mode.

2. Describe how the prescaler is used in Timer PWM mode.
   - In PWM mode, the prescaler works as a timer extension. It is appended as the MSB bits of the Timer. Therefore, in 16-bit mode, the PWM timer works as a (16 + 8) = 24 bit count-down timer running at the system clock frequency.

3. Describe how a Timer works in **PWM mode**.
   - In PWM mode, the Timer counts down from the values in **T*n*ILR** & **T*n*PR** registers when the **T*n*EN** bit is set.
   - When the count value reach 0x0, the **T*n*ILR** & **T*n*PR** values are reloaded at the next clock cycle and the Timer continues to count down.
   - When the values in **T*n*ILR** & **T*n*PR** equals that in **T*n*MATCHR** & **T*n*PMR** respectively, the timer output toggles.
   - Polarity of timer output is controlled by the **T*n*PWML** bit in the **CTL** register.

Timer PWM Operation

4. A system has a system clock of **80 MHz**. We want to generate a **PWM output** waveform at **Timer 0 (GPIO PB6)** with a frequency of **50 KHz** and with duty cycle of **33%**. What are values we should load to GPTMTAILR, GPTMTAPR, GPTMTAMATCHR and GPTMTAPMR registers? Place your calculated values into an initialization routine and capture the output signal at Timer0 output (PB6).

- At 80 MHz bus clock, period = 12.5ns.
- With a 24-bit count down Timer,
  - maximum count = 2^{24}
  - maximum interval = 2^{24} x 12.5ns = 209.715 ms.

$$Period\ at\ 50\ KHz = \frac{1}{50{,}000} = 20\ us$$

$$no\ of\ counts = \frac{20us}{12.5ns} = 1600 = 0x0640$$

$$duty\ cycle = \frac{(1600 - n)}{1600} = 0.33$$

$$n = 1072 = 0x0430$$

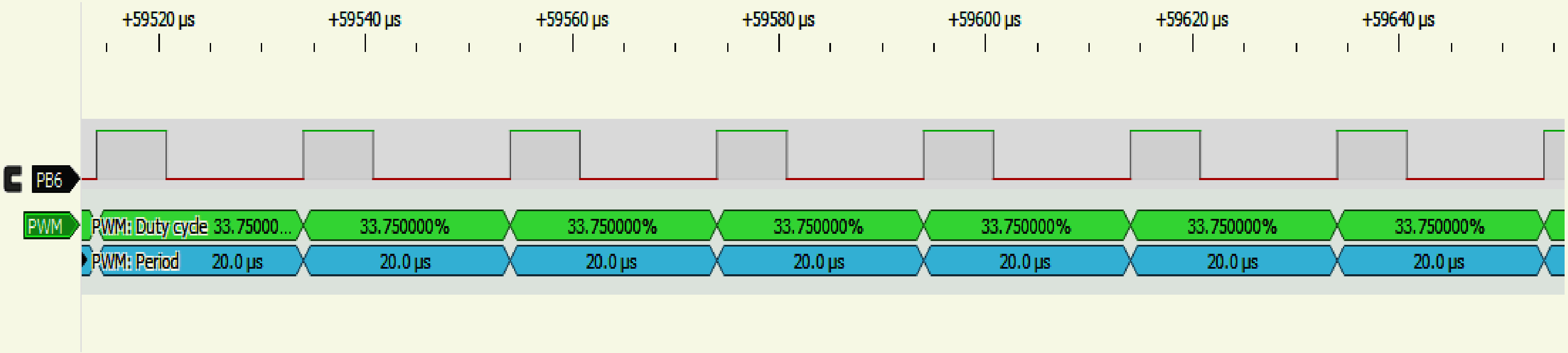
Reload (**TAILR)** = 0x063F; Pre-scaler (**TAPR**) = 0x00
Match (**TAMATCHR)** = 0x042F; Pre-scaler Match (**TAPMR**) = 0x00

Timer PWM Operation

```
#define PB_T0CCP0   6U        /* PB6  */
void Port_Init( void ){
    SYSCTL->RCGCTIMER |= SYSCTL_RCGCTIMER_R0;  /* Timer 0 */
    SYSCTL->RCGCGPIO |= SYSCTL_RCGCGPIO_R1     /* GPIO Port B */

    /* Wait for peripherals to be ready */
    while( 0 == (SYSCTL->PRTIMER & SYSCTL_PRTIMER_R1) );
    while( 0 == (SYSCTL->PRGPIO & SYSCTL_PRGPIO_R1) );

    /* setup Timer 0 GPIO pin at PB6 */
    GPIOB->AFSEL |= BIT (PB_T0CCP0);    // enable alternate function
    GPIOB->PCTL |= GPIO_PCTL_PB6_T0CCP0; // enable timer 0 in PB6
    GPIOB->DEN |= BIT (PB_T0CCP0);      // enable PB6
    GPIOB->DIR |= BIT (PB_T0CCP0);      // set as output

    /* setup timer 0 t0 PWM mode        */
    TIMER0->CTL &= ~TIMER_CTL_TAEN; // disable timer 0 during setup
    TIMER0->CFG |= TIMER_CFG_16_BIT; // set to 16-bit mode
    /* TAAMS-enable PWM; TAMR-Periodic mode       */
    TIMER0->TAMR |= TIMER_TAMR_TAAMS | TIMER_TAMR_TAMR_PERIOD;
    TIMER0->TAMR |= TIMER_TAMR_TACMR; /* TACMR-Edge-Time mode  */

    TIMER0->TAPR = 0x0; /* prescaler (8 bits)  */
    TIMER0->TAILR = 0x063F; /* TAILR-pulse period */
    TIMER0->TAMATCHR = 0x042A;
    TIMER0->TAPMR =  0x0;;
    TIMER0->CTL |= TIMER_CTL_TAEN; /* enable timer 0 */
}
```

Timer PWM Operation

Output signal at Timer 0 T0CCP0 pin (PB6):



Timer PWM Operation

5. A system has a system clock of **50 MHz**. We want to generate a **PWM output** waveform at **Timer 2 (GPIO PB0)** with a frequency of **500 Hz** and with duty cycle of **50%**. What are values we should load to GPTMTAILR, GPTMTAPR, GPTMTAMATCHR and GPTMTAPMR registers?

- At 50 MHz bus clock, period = 20ns.
- With a 24-bit count down Timer,
  - maximum count = 2^{24}
  - maximum interval = 2^{24} x 20ns = 335.44 ms.

$$Period\ at\ 500\ Hz = \frac{1}{500} = 2\ ms$$

$$\frac{2ms}{20ns} = 100{,}000 = 0x01.86A0$$

$$\frac{(100{,}000 - n)}{100{,}000} = 0.5$$

$$n = 50{,}000 = 0x0.C350$$

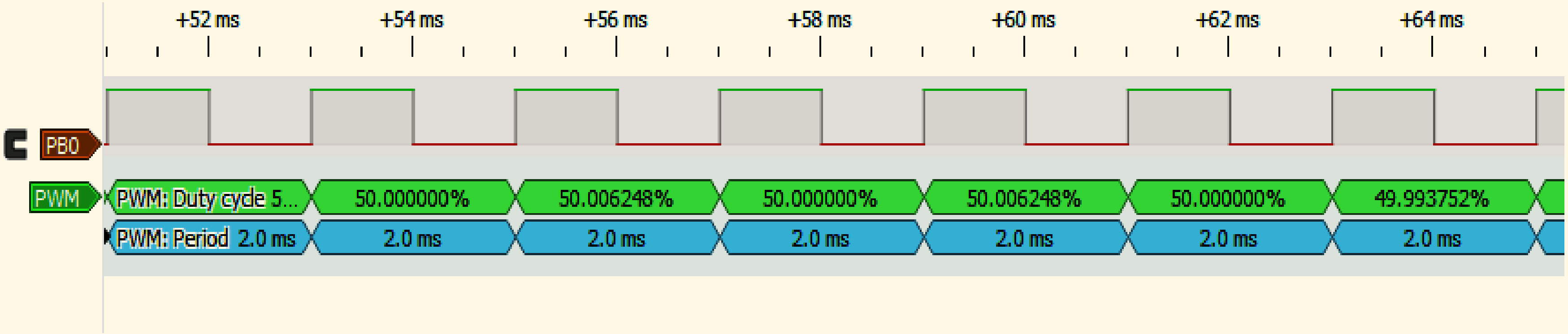
Reload (**TAILR)** = 0x869F; Pre-scaler (**TAPR**) = 0x01
Match (**TAMATCHR)** = 0xC34F; Pre-scaler Match (**TAPMR**) = 0x00

Timer PWM Operation

```
#define PB_T2CCP0   0U        /* PB0  */
void Port_Init( void ){
    SYSCTL->RCGCTIMER |= SYSCTL_RCGCTIMER_R2;  /* Timer 2 */
    SYSCTL->RCGCGPIO |= SYSCTL_RCGCGPIO_R1     /* GPIO Port B */

    /* Wait for peripherals to be ready */
    while( 0 == (SYSCTL->PRTIMER & SYSCTL_PRTIMER_R2) );
    while( 0 == (SYSCTL->PRGPIO & SYSCTL_PRGPIO_R1) );

    /* setup Timer 2 GPIO pin at PB0 */
    GPIOB->AFSEL |= BIT (PB_T2CCP0);    // enable alternate function
    GPIOB->PCTL |= GPIO_PCTL_PB0_T2CCP0; // enable timer 2 in PB0
    GPIOB->DEN |= BIT (PB_T2CCP0);      // enable PB0
    GPIOB->DIR |= BIT (PB_T2CCP0);      // set as output

    /* setup timer 2 t0 PWM mode        */
    TIMER2->CTL &= ~TIMER_CTL_TAEN; // disable timer 2 during setup
    TIMER2->CFG |= TIMER_CFG_16_BIT; // set to 16-bit mode
    /* TAAMS-enable PWM; TAMR-Periodic mode       */
    TIMER2->TAMR |= TIMER_TAMR_TAAMS | TIMER_TAMR_TAMR_PERIOD;
    TIMER2->TAMR |= TIMER_TAMR_TACMR; /* TACMR-Edge-Time mode  */

    TIMER2->TAPR = 0x01; /* prescaler (8 bits)  */
    TIMER2->TAILR = 0x0869F; /* TAILR-pulse period */
    TIMER2->TAMATCHR = 0x0C34F;
    TIMER2->TAPMR =  0x0;;
    TIMER2->CTL |= TIMER_CTL_TAEN; /* enable timer 2 */
}
```

Timer PWM Operation

Output signal at Timer 2 T2CCP0 pin (PB0):



+52 ms
+54 ms
+56 ms
+58 ms
+60 ms
+62 ms
+64 ms
PB0
PWM
PWM: Duty cycle 5...
50.000000%
50.006248%
50.000000%
50.006248%
50.000000%
49.993752%
PWM: Period 2.0 ms
2.0 ms
2.0 ms
2.0 ms
2.0 ms
2.0 ms
2.0 ms