Text

Description automatically generated

**School of Electrical and Computer Engineering**

**Embedded Systems Design - ECE 5721 - Fall 2022**

**Homework 3**

**Daniel Funke**

**November 8, 2022**

# Chapter 7

7. Specify how the control registers must be configured so that TPM2 generates interrupts at an approximate frequency of 2017.0101 Hz. What is the actual frequency of interrupts?

Assuming an input clock of 48MHz and a target interrupt frequency of 2017.0101Hz, the MOD register would need to be configured to contain the value of

Since the MOD register is a 16 bit binary value that represents an integer, we round this value to the nearest integer value, or 23, 797. This yields **an actual interrupt frequency of 2017.061Hz**. Setting this value looks like this:

1. TPM2->MOD = (23,797);

We also have to enable interrupts, which requires setting the Timer Overflow Interrupt Enable bit of the TMP2\_SC register to 1. We also need to set the CMOD bit field to 1 to tell the counter to increment on every clock cycle of the LPTPM. Finally, since we did not base our calculations on a prescaled input clock, we also have to ensure that the PS bit field of this register is set to 000, which is the default value, so no action needs to be taken. This looks like the following:

1. TPM2->SC = TMP\_SC\_CMOD(1) | TPM\_SC\_TOIE\_MASK;
2. NVIC\_SetPriority(TPM2\_IRQn, 3);
3. NVIC\_ClearPendingIRQ(TPM2\_IRQn);
4. NVIC\_EnableIRQ(TPM2\_IRQn);

8. Specify how the control registers must be configured so that TPM0 channel 3 generates pulses that are high for 150s and low for 27s, assuming an input clock of 24 MHz. No interrupts are to be generated, but the pulses are to be generated on MCU Port D bit 3. What are the actual high and low times?

First, Pin 3 of Port D needs to be configured to output the waveform generated by the TPM module. This is done by the following:

1. SIM->SCGC5 |= SIM\_SCGC5\_PORTD\_MASK;
2. PORTD->PCR[3] &= ~PORT\_PCR\_MUX\_MASK;
3. PORTD->PCR[3] |= PORT\_PCR\_MUX(4);

Next, the TMP module needs to be configured to generate the waveform described by the problem. To create a waveform with a high pulse lasting 150s means setting the CnV register to be

The MOD register would need to be set to the following to force the waveform to toggle low for 27s:

This is done with the following lines of code:

1. // Enable clock to TPM module
2. SIM->SCGC6 |= SIM\_SCGC6\_TPM0\_MASK;
3. //Set clock prescaler to 2 for 24MHz
4. TPM0->SC = TPM\_SC\_PS(1);
5. // Load the mod register with period value
6. TPM0->MOD = 4247;
7. // Set Duty Cycle
8. TPM0->CONTROLS[3].CnV = 3600;
9. // Set channel 3 to edge-aligned high-true PWM
10. TPM0->CONTROLS[3].CnSC = TPM\_CnSC\_MSB\_MASK | TPM\_CnSC\_ELSB\_MASK;
11. // Start TPM
12. TPM0-SC |= TPM\_SC\_CMOD(1);

This allows for actual high and low times of **150s** and **26.95s**, respectively.

9. Specify how the control registers must be configured so that one of the channels in TPM0 measures the delay until the pulse applied to Port A bit 0 changes from one to zero. TPM also must generate an interrupt at that time. Each count in CnV must represent one-third of a microsecond.

1. // Enable Clock to TPM0 and Port A
2. SIM->SCGC6 |= SIM\_SCGC6\_TPM0\_MASK;
3. SIM->SCGC5 |= SIM\_SCGC5\_PORTA\_MASK;
4. // Select pin mux to connect to timer module
5. PORTA->PCR[0] &= ~PORT\_PCR\_MUX\_MASK;
6. PORTA->PCR[0] |= PORT\_PCR\_MUX(3);
8. // Set clock source for TPM
9. SIM-SOPT2 |= SIM\_SOPT2\_TPMSRC(1) | SIM\_SOPT2\_PLLFLLSEL\_MASK;
10. // Load the mod with 16 (48MHz / 16) = 3MHz = 1/3
11. TPM0->MOD = 0x000F;
12. // Set channel to input capture with falling edge polarity and
13. // enable channel interrupt
14. TPM0->CONTROLS[0].CnSC = TPM\_CNSC\_ELSB\_MASK | TPM\_CnSC\_CHIE\_MASK;
15. // Enable counter and interrupts
16. TPM0->SC = TPM\_SC\_CMOD(1) |TPM\_SC\_TOIE\_MASK;
17. NVIC\_SetPriority(TPM0\_IRQn,3);
18. NVIC\_ClearPendingIRQ(TPM0\_IRQn);
19. NVIC\_EnableIRQ(TPM0\_IRQn);

10. What is the lowest interrupt frequency that a Timer/PWM module on KL25z can generate? Assume the CPU clock is 48 MHz.

The lowest interrupt frequency that the TPM module can produce on this chip occurs when the prescaler is set to the maximum value (128) and the MOD register is set to its highest value, which for a 16 bit integer is 216-1 =65,535. Additionally, if up-down counting is selected, the final frequency can be halved once more to produce the following:

# References

|  |  |
| --- | --- |
| [1] | A. Dean, Embedded Systems Fundamentals with Arm Cortex-M based Microcontrollers, Cambridge: Arm Education Media, 2017. |