Solutions - Homework 3

(Due date: November 6th @ 5:30 pm)
Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (25 PTS)

- Using the HCS12D Timer, write a C program (*provide a printout*) that measures the period (in cycles) of a square waveform on `PT2`. Use a pre-scale factor of 4 with E-clock = 24 MHz. Note that the period of the signal can be longer than 2^{16} cycles.
  - ✓ What is the period (in units of time) of the Timer Clock?
  - ✓ What is the smallest period (in units of time) that we can measure?

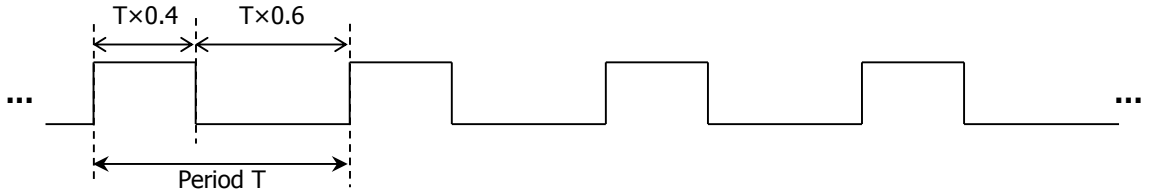
- $Timer\ Clock = \frac{24}{4} = 6MHz \rightarrow Timer\ Clock\ Period = \frac{1}{6}us$
- $Smallest\ Period\ we\ can\ measure = Timer\ Clock\ Period = \frac{1}{6}us$
- Process:
  1. Enable Input Capture on Channel 2: `TIOS(2)=0` → `TIOS = 0x00`
  2. Select rising edge on Input Capture Channel 2: `TCTL4(5..4) = 01` → `TCTL4 = 0x10`
  3. Set pre-scaler to 4. → `TSCR2=0x02`
  4. Enable timer counter (starts from 0) → `TSCR1 = 0x80`
  5. Clear *C2F* flag: `TFLG1(2) = 1` → `TFLG1 = 0x04`
  6. Wait until `TFLG1(2) = 1`
  7. Here we enable the TOF Interrupt:
     - Clear `TOF`: `TFLG2 = 0x80`, write '1' on `TOF` to clear it.
     - Local Enable for TOF: `TOI=1` → `TSCR2 = TSCR2|0x80.`
     - Global Enable: `asm("cli");`
  8. edge1 = `TC2` (16-bit Input Capture Channel 2 Register). On the rising edge, `TC2` gets the counter value.
  9. Clear *C2F* flag: `TFLG1(2) = 1` → `TFLG1 = 0x04`
  10. Wait until `TLFG1(2) = 1`
  11. edge2 = `TC2`
  12. diff = edge2-edge1.
  13. If edge2 < edge1 then overflow = overflow -1
  14. $Period = Overflow \times 2^{16} + diff$

- *Interrupt Service Routine*: The TOF interrupt was enabled on Step 7. The global variable overflow is initialized with 0, and every time the TOF interrupt arrives, the ISR clears the TOF flag and increments the count on overflow.

C Code: `hw3p1.c`

PROBLEM 2 (25 PTS)

- Using the HCS12D Timer, write a C program (*provide a printout*) that generates an active high 2-kHz digital waveform with a 40% duty cycle on `PT5`. Use a pre-scale factor of 2 with E-clock = 24 Mhz. Try it by playing it on the Buzzer.



  - ✓ What is the period (in units of time) of the Timer Clock?
  - ✓ How many Timer cycles are in one period of the 2-KHz digital waveform?
  - ✓ How many Timer cycles are required for the high level portion of the period and for the low level portion of the period?

- $Timer\ Clock = \frac{24}{2} = 12\ MHz \rightarrow Timer\ Clock\ Period = \frac{1}{12}us$
- One period of the 2 KHz digital waveform amounts to 500 us. So, the number of Timer Cycles in one period of the 2 KHz waveform is given by: $Timer\ Cycles = \frac{500}{1/12} = 6000\ cycles$.

- For the 40% duty cycle, we need $HCYCLES = 0.4 \times 6000 = 2400$, and $LCYCLES = 6000 - 2400 = 3600$
- Process:
  1. Enable Output Compare on Channel 5: `TIOS(5)=1` → `TIOS = 0x20`
  2. Select OC5 action to pull to high: `TCTL1(3..2)=11` → `TCTL1 = 0x0C`
  3. Set Pre-scaler factor to 2. → `TSCR2=0x01`
  4. Enable timer counter (starts from 0). Enable fast clear for TOF and *C5F*. → `TSCR1 = 0x90`
  5. Clear all *CnF* flags (just in case) → `TFLG1 = 0xFF`
  6. Start an OC5 operation with a delay of 10 cycles: `TC5 = TCNT+10`. This is so that we start with 0 for just 10 cycles.
  7. Wait until `TLFG1(5) = 1`.
  8. Set OC5 pin to toggle. `TCTL1(3..2) = 01` → `TCTL1 = 0x04`
  9. Set new Output Compare operation with a delay of HCYCLES cycles: `TC5 = TC5+HCYCLES`
  10. HiLoflag = 0 (Global variable)
  11. Enable OC5 Interrupt:
      - Local Enable for OC5: `TIE(5)=1` → `TIE=0x20`
      - Global Enable: `asm("cli")`
- *Interrupt Service Routine*: The OC5 interrupt was enabled on Step 11. If HiLoflag = 0, then we add `LCYCLES` cycles to `TC5` and make HiLoflag=1. If HiLoflag=1, then we add `HCYCLES` cycles to `TC5` and make HiLoflag=0. Note that the addition will wraparound if it is larger than $2^{16} - 1$. Also, recall that TFLG1 is cleared every time we write on `TC5`.

C Code: `hw3p2.c`

PROBLEM 3 (20 PTS)

- If we want to measure the period of a signal using just one iteration of the count, i.e., within a 2^{16} cycles time window, the period has to be lower or equal than 2^{15} cycles. Assuming an E-clock of 24 MHz, what is the minimum frequency that we can measure for each of the following pre-scale factors: 1, 2, 4, 8, 16, 32, 64, and 128?

Pre-scale factor	Timer Clock Period	Max. period we can measure	Min. frequency we can measure
1	$\frac{1}{24} us$	$2^{15} \times \frac{1}{24} us = 1.3653\ ms$	$\frac{1}{1.3653} = 0.7324\ KHz$
2	$\frac{2}{24} = \frac{1}{12} us$	$2^{15} \times \frac{1}{12} us = 2.7306\ ms$	$\frac{1}{2.7306} = 0.36621\ KHz$
4	$\frac{4}{24} = \frac{1}{6} us$	$2^{15} \times \frac{1}{6} us = 5.4613\ ms$	$\frac{1}{5.4613} = 0.183105\ KHz$
8	$\frac{8}{24} = \frac{1}{3} us$	$2^{15} \times \frac{1}{3} us = 10.9226\ ms$	$\frac{1}{10.9226} = 91.55\ Hz$
16	$\frac{16}{24} = \frac{2}{3} us$	$2^{15} \times \frac{2}{3} us = 21.845\ ms$	$\frac{1}{21.845} = 45.77\ Hz$
32	$\frac{32}{24} = \frac{4}{3} us$	$2^{15} \times \frac{4}{3} us = 43.69\ ms$	$\frac{1}{43.69} = 22.88\ Hz$
64	$\frac{64}{24} = \frac{8}{3} us$	$2^{15} \times \frac{8}{3} us = 87.381\ ms$	$\frac{1}{87.381} = 11.44\ Hz$
128	$\frac{128}{24} = \frac{16}{3} us$	$2^{15} \times \frac{16}{3} us = 174.762\ ms$	$\frac{1}{174.762} = 5{,}72\ Hz$

- To create a delay using the Output Compare Channel 6, we add a number of cycles (DCYCLES) to `TC6` and then wait until `TCNT` is equal to `TC6`. This happens when `TLFG1(6)=1`. Assuming an E-clock of 24 MHz:
  - ✓ Complete the following table in order to generate the given delays:

DCYCLES	Pre-scale Factor	Delay
24	1	1 us
75	32	100 us
375	32	500 us
375	128	2 ms
15000	128	80 ms
56250	128	300 ms

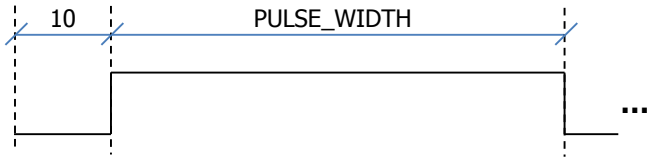
There can be more than only solution for each case. In particular, the higher the pre-scale factor, the lower the Timer Clock frequency, and the lower the power consumption. So, it is a good idea to aim for the largest pre-scale factor.

$$Delay = DCYCLES \times \frac{1}{\frac{24\ MHz}{Pre-scale\ factor}} \rightarrow (Pre-scale\ factor) \times DYCLES = 24 \times 10^6 \times Delay$$

- ✓ What is the largest delay (in units of time) that we can generate with Output Compare Channel 6? Provide DCYCLES and the Pre-scale Factor as well.

$$Largest\ Delay = DCYCLES \times \frac{1}{\frac{24\ MHz}{Pre\text{-}scale\ factor}} = 65535 \times \frac{1}{\frac{24\ MHz}{128}} = 349.52ms$$

- To create a single pulse on `PT6`, we set the OC6 pin action to toggle when a comparison is successful. Then:
  1. We start with `PT6=0`, and wait some cycles (say 10) before we toggle to `PT6=1`: `TC6=TC6+10` and then check whether `TFLG1(6)=1`.
  2. With `PT6=1`, we wait a number of cycles before we toggle to `PT6=0`: `TC6=TC6 + PULSE_WIDTH` and then check whether `TFLG1(6)=1`.



- ✓ Provide a PULSE_WIDTH value and a pre-scale factor in order to generate a pulse of 5 us. Assume an E-clock of 24 MHz.

$$5\ us = PULSE\_WIDTH \times \frac{1}{\frac{24\ MHz}{Pre\text{-}scale\ factor}} \rightarrow PULSE\_WIDTH\ \times (Pre - scale\ factor) = 120$$

There is more than one solution. For example: $PULSE_{WIDTH} = 60,\ Pre - scale\ factor = 2$