Lab 4 – Oscillator Calibration Test

ECE531 Digital TPE

In this lab, you will calibrate the internal oscillator of the pic16f883. The pic has an internal 8 MHz oscillator used for clocking, which can be downscaled at 7 levels from 125 kHz to 8 MHz. Information about the oscillator module can be found in chapter 4 of the data sheet. From the data sheet, we can find that the oscillator can be configured in one of 8 modes using the FOSC value in the CONFIG1 register. One of these modes uses the internal oscillator while outputting the instruction clock signal on the OSC2 pin (which is pin RA6). The instructions of the pic operate at $F_{osc}/4$, so at the maximum frequency of 8MHz, this would be a 2 MHz instruction clock. From the programming specification document, we can see that there is another register in the configuration address space which allows us to set a calibration value for the internal oscillator.

In this lab, you will set the device to operate using the internal oscillator with clock output on OSC2, and modify the calibration value to achieve a clock oscillator as near 8 MHz as possible. The clock signal output on pin OSC2 can be used for the measurement and testing of the current oscillator frequency. If F_{osc} is set to 4 MHz (the default) using internal special function registers, then your test program should be designed to identify the calibration value which produces an output on OSC2 which is nearest to 1 MHz.

Measurement of the clock signal on OSC2 can be accomplished using the Time Measurement Unit (TMU) of the ATE. On the digital DIB, pin RA6 is connected to TMU0. This will allow us to make precise measurements of time between the rising and falling edges of the signal on RA6. For greater accuracy, we will measure the time over a large number of periods of the clock signal on RA6 (say 10,000), and find the average clock period. We can specify what threshold voltage should be used to indicate a high → low or low → high transition for the TMU. For this, we can use the value 2.5 V. The following are a set of functions that will be useful in working with the TMU. Since RA6 is connected to TMU0 on the DIB, we can use Direct for the pin names and types (use QTMU_DIRECTA for the type). More information can be found in the ETS documentation. The mode, start, and stop setting functions can be performed just once, but the arming and measurement functions will need to be executed for each measurement. NOTE: QTMU should be turned off in cleanup code, just as other resources, using the mode setting function.

Quad Time Measurement Unit (QTMU) Utilities

qtmumode(tmu, mode, ArmOut=QTMU_ARMOUTOFF);

tmu: The TMU to program.
Valid range is QTMU_ALL (-1) or QTMU0 (0) through QTMU7 (7).

mode: The measurement mode for the TMU.
Valid values are:
QTMU_OFF (0),
QTMU_TIMER (1),
QTMU_FREQANALYSIS (2),
QTMU_EVENTCOUNTER (3).

ArmOut: Value of QTMU_ARMOUTON (1) will enable the TMU ArmOut signal on the DUT board for each TMU that will go high when the TMU is armed.
Value of QTMU_ARMOUTOFF (0) disables this signal

qtmustart(tmu, pinName, pinType, slope, Holdoff=0, RetriggerHoldoff= QTMU_NOFILTER, AnalogLevel=0.0);

tmu: The TMU to program.
Valid range is:
QTMU_ALL (-1) or QTMU0 (0) through QTMU7 (7).

pinName: The input pin to connect to the start channel of the TMU. The pin name can be a name defined in the Vector Editor.

Other keywords that are accepted are:

Any pin name	Driver/Receiver pins defined in vector file
"DP0" through "DP127"	Direct driver(I/O) pins
"RP0" through "RP127"	Direct receiver pins
"Direct"	Direct signals from the DUT board
"Off"	Disconnects the start channel

pinType: The actual input from the specified pinName.
Valid values are:
QTMU_RCVL (0),
QTMU_RCVH (1),
QTMU_DRV (2)*, (* Not valid with DPU-16 boards in the tester.)
QTMU_DIRECTA (3),
QTMU_DIRECTB (4).

slope: The slope of the source signal.
Valid values are: QTMU_POS (0) or QTMU_NEG (1).

Holdoff: The number of events to be ignored.
Range is 0 to 65535.

RetriggerHoldoff: The lowpass filter value in nsec.
Valid values are as follows:
For QTMU Modules revision 90.2 or greater (see NOTE 1 in the Description section below for information on determining your modules' revision number) – QTMU_NOFILTER=0|

1 to 1000 – any number will be accepted, but it will be rounded to 10 or 30 – 1000 in steps of 5 ns.

1001 to 3200 – any number will be accepted, but it will be rounded in steps of 12.5 ns.

3201 to 405000 – any number will be accepted, but it will be rounded in steps of 1.6 µs.

For QTMU Modules older than revision 90.2 (see NOTE 1 in the Description section below for information on determining your modules' revision number) – QTMU_NOFILTER=0, 10, 30 to 1000 in steps of 5.

AnalogLevel: The trigger level.
Only valid for direct sources. Range is ±50 V

qtmustop(tmu, pinName, pinType, slope, Holdoff=0, RetriggerHoldoff= QTMU_NOFILTER, AnalogLevel=0.0);

tmu: The TMU to program.

Valid range is:

QTMU_ALL (-1) or QTMU0 (0) through QTMU7 (7).

pinName: The input pin to connect to the stop channel of the TMU. The pin name can be a name defined in the Vector Editor.

Other keywords that are accepted are:

Any pin name	Driver/Receiver pins defined in vector file
"DP0" through "DP127"	Direct driver(I/O) pins
"RP0" through "RP127"	Direct receiver pins
"Direct"	Direct signals from the DUT board
"Off"	Disconnects the stop channel

pinType: The actual input from the specified pinName.

Valid values are:
QTMU_RCVL (0),
QTMU_RCVH (1),
QTMU_DRV (2)*, (* Not valid with DPU-16 boards in the tester.)
QTMU_DIRECTA (3),
QTMU_DIRECTB (4).

slope: The slope of the source signal.

Valid values are: QTMU_POS (0) or QTMU_NEG (1).

Holdoff: The number of events to be ignored.

Range is 0 – 65535.

RetriggerHoldoff: The low pass filter value in nsec.

Valid values are as follows:

For QTMU Modules revision 90.2 or greater (see NOTE 1 in the Description section below for information on determining your modules' revision number) –

QTMU_NOFILTER=0|

1 to 1000 – any number will be accepted, but it will be rounded to 10 or 30 – 1000 in steps of 5 ns.

1001 to 3200 – any number will be accepted, but it will be rounded in steps of 12.5 ns.

3201 to 405000 – any number will be accepted, but it will be rounded in steps of 1.6 µs.

For QTMU Modules older than revision 90.2 (see NOTE 1 in the Description section below for information on determining your modules' revision number) –

QTMU_NOFILTER=0, 10, 30 to 1000 in steps of 5.

AnalogLevel: The trigger level.

Only valid for direct sources. Range is ±50 V.

qtmuarm(tmu, source, count=1, trigger=QTMU_EDGE, slope=QTMU_NEG);

tmu: The TMU to program.
Valid range is QTMU_ALL (-1) or QTMU0 (0) through QTMU7 (7).

source: The input signal to connect to the arm channel of the TMU.
This can be a PSQ group name defined in the Vector Editor.
Other keywords that are accepted are:
"AUTO", "AUTOMCU", "DIRECT", "DIRECTMCU", "PSQ0" through "PSQ15", "OFF"

count: The max number of samples that will be taken after the TMU arms.
Valid range is 1 – 4096.

trigger: The trigger mode of the arming signal.
Valid values are QTMU_EDGE (0) or QTMU_LEVEL (1).

slope: The slope of the Direct arming signal.
This parameter is only valid if the 'source' is "DIRECT" or "DIRECTMCU".
Valid values are QTMU_POS (0) or QTMU_NEG (1).

qtmumt(tmu, mode, timeout, start=-1, stop=-1, slice=1);

tmu: The TMU to program.
Valid range is QTMU_ALL (-1) or QTMU0 (0) through QTMU7 (7).

mode: The form of the returned value.
QTMU_READTIME (0) Averaged time in nsec.
QTMU_READFREQ (1) Averaged time converted to a frequency in kHz.
QTMU_READCOUNTER (2) Count value for a QTMU_EVENTCOUNTER mode.
QTMU_READ_ELAPSEDTIME (3) Total time from start sample to stop sample in nsec.

timeout: The time in µsec to wait for the TMU to make the correct number of measurements.
Valid range is 0 µsec – 2147 Seconds.

start: The start sample point within the specified slice from the TMU. Starts at 1.

stop: The stop sample point within the specified slice from the TMU. Starts at 1.

slice: The time slice at which to read the data. Starts at 1

The CALIBRATION register is in configuration memory space. It stores calibration values for several components, including the FCAL, a signed 7-bit value which represents the calibration for the internal oscillator. To perform this test, you will need to enter Program/Verify mode to write to the calibration register, then restart the device to run the test application and measure the frequency of the clock signal during execution. This process will be repeated in a loop until the desired calibration value is found. NOTE: you should read and save the calibration values in the config registers prior to modifying them (you can do this during the WriteConfig test and store the value in a global variable). You do not want to modify the other calibration values in configuration, only OSC, so be sure to augment your data that you write to configuration to include the original calibration values for other modules.

Write-up your results
Describe the algorithm you used to scan across the range of calibration values to find the optimal value. Could your algorithm be improved to reduce the number of steps needed to find the optimal calibration value?

Datalog the calibration value, FCAL, and the measured frequency of the selected calibration value. Record your test time. Turn in your final code and datalogs. Discuss your test results.

Due: Wednesday, April 27, 2016