Embedded Systems
LA2 (C3)

Name: Ahmed ElShaarany

Date: 6/9/2015

Lab 4 Worksheet

Reset and Operating Modes & Watchdog Timers

1. Name all 3 types of resets:

   a. Brownout Reset (BOR)
   b. Power-On Reset (POR)
   c. Power-Up Clear (PUC)

2. If the Watchdog (WDT) times out, which reset does it invoke?

   Power-Up Clear (PUC)

3. Write the DriverLib function that stops (halts) the watchdog timer:

   WDT_A_hold (WDT_A_BASE);

Power Management

4. (‘F5529 Launchpad users only) Write the DriverLib function that sets the core voltage needed to run MCLK at 8MHz.

```
PMM_setVCore(PMM_CORE_LEVEL_0);
```

Clocking

5. Why does MSP430 provide 3 different types of internal clocks?

   MSP430 provide 3 different types of internal clocks because it promotes power efficiency. This is achieved by turning off different clocks in low power modes. Therefore, having 3 clocks provides flexibility in tuning system’s power vs performance.

   These 3 clocks are:
   a. Auxiliary clock (ACLK)
   b. Sub-system master clock (SMCLK)
   c. Master clock (MCLK)

6. What is the speed of the crystal oscillators on your board?

```
#define LF_CRYSTAL_FREQUENCY_IN_HZ 32768

#define HF_CRYSTAL_FREQUENCY_IN_HZ 4000000
```

7. What function specifies these crystal frequencies to the DriverLib?

```
UCS_setExternalClockSource(LF_CRYSTAL_FREQUENCY_IN_HZ,
HF_CRYSTAL_FREQUENCY_IN_HZ);
```

8. At what frequencies are the clocks running? There's an API for that…Write the code that returns your current clock frequencies:

```
uint32_t myACLK = 0;

uint32_t mySMCLK = 0;

uint32_t myMCLK = 0;

myACLK  = UCS_getACLK();

mySMCLK = UCS_getSMCLK();

myMCLK  = UCS_getMCLK();
```

9. We didn't set up the clocks (or power level) in our previous labs, how come our code worked?

   Because the clocks are set to their default in that case.

   Don't spend too much time pondering this, but what speed do you think each clock is running at before we configure them? (You can compare this to your results when running the code.)

   ACLK: 32768 Hz SMCLK: 1048576 Hz MCLK: 1048576 Hz

10. Set up ACLK:

- Use REFO for the F5529 device

```
UCS_clockSignalInit(
        UCS_ACLK,                                   // Clock you're configuring
        UCS_REFOCLK_SELECT,                        // Clock source
        UCS_CLOCK_DIVIDER_1                         // Divide down clock source by this much
);
```

11. (F5529 User's) Write the code to setup MCLK. It should be running at 8MHz using the DCO+FLL as its oscillator source.

```
#define MCLK_DESIRED_FREQUENCY_IN_KHZ  8000
#define MCLK_FLLREF_RATIO MCLK_DESIRED_FREQUENCY_IN_KHZ / ( UCS_REFOCLK_FREQUENCY/1024 )
// Set the FLL's clock reference clock to REFO
   UCS_clockSignalInit(
            UCS_FLLREF,                                  // Clock you're configuring
            UCS_REFOCLK_SELECT,                         // Clock source
            UCS_CLOCK_DIVIDER_1);

// Configure the FLL's frequency and set MCLK & SMCLK to use the FLL as their source (8MHz)
      UCS_initFLLSettle(
        MCLK_DESIRED_FREQUENCY_IN_KHZ,
        MCLK_FLLREF_RATIO );
```

Answer the following questions:

- What is the FLLREF ratio?

  244

- What is the actual MCLK & SMCLK frequency after the correction?

  7995392 Hz

Step 19. Run the code to the first breakpoint and write down the Express values

myACLK/1000: 32.768

mySMCLK/1000: 1048.576

myMCLK/1000: 1048.576

Are these the values that you expected? Yes

Step 20. Run to the next breakpoint – at the end of the initClocks() function.

Check on the values again:

myACLK/1000: 32.768

mySMCLK/1000: 7995.392

myMCLK/1000: 7995.392

Are these the values we were asked to implement? As close to them as possible yes.

Step 25. Did the value for MCLK change? Yes

Extra credit: (20 points)

In lab 4a, when using REFO oscillator as MCLK source, what should be the delay count to get LED to blink at 10s intervals? Test and verify (measure time between blinks).

The delay count should be 327680.

It was tested and verified and the code is included.

(Optional) Lab 4b – Exploring the Watchdog Timer

Step 4. How fast is the LED blinking now?

The LED is not blinking at all in this case.

A couple of Questions about Watchdogs

Step 6. Complete the code needed to enable the Watchdog Timer using ACLK:

```
WDT_A_watchdogTimerInit( //Initialize the WDT as a watchdog
WDT_A_BASE,
WDT_A_CLOCKSOURCE_ACLK, //Which clock should WDT use?
//WDT_A_CLOCKDIVIDER_64 ); //Divide the WDT clock input?
WDT_A_CLOCKDIVIDER_512 ); //Here are 3 (of 8) different div choices
//WDT_A_CLOCKDIVIDER_32K );

WDT_A_start ( WDT_A_BASE ); //Start the watchdog
```

Step 7. Write the code to reset the Watchdog Timer. What driverlib function can you used to reset the timer?

void WDT_A_resetTimer (uint16_t baseAddress)

Step 16. Build the code to test that it's error-free (syntax wise). Where are these values defined?

They are defined in wdt_a.h

Step 16. Launch the debugger and run the program. Write down the results. How many times does printf() run before the count restarts? Terminate, change divisor, and retest.

Number of times printf() runs before watchdog reset:

WDT_A_CLOCKDIVIDER_64: 1

WDT_A_CLOCKDIVIDER_512: 9

WDT_A_CLOCKDIVIDER_32K: 541

Step 21. Build and run the program to observe the watchdog resetting the MSP430. How many times will it run now?

It will keep running forever because we are resetting the watchdog timer in every loop iteration.