LAB 4 TURN-IN QUESTIONS

Explain what the BRR in the UART setup is for and how to determine a value for it.

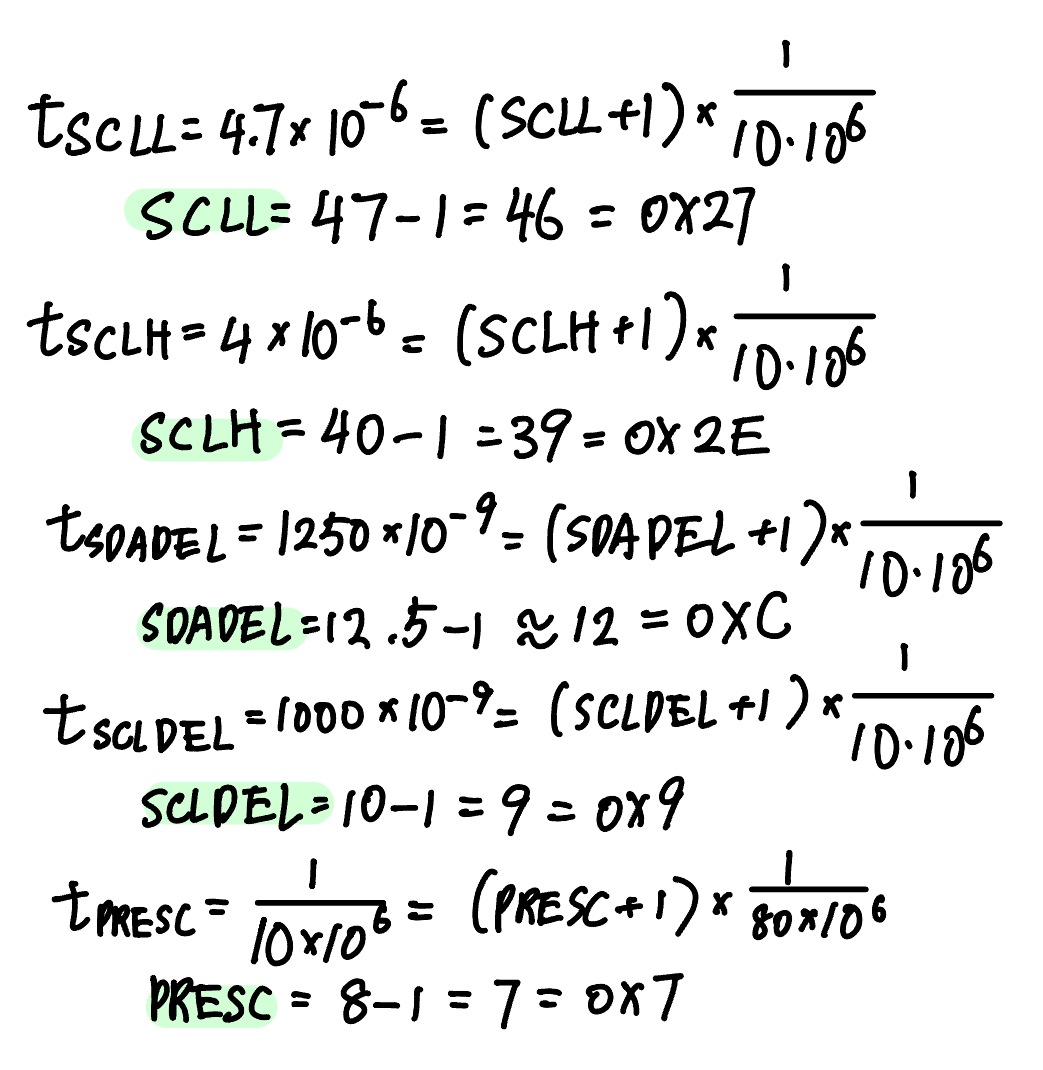
* The BRR in the UART setup is the Baud Rate Register, which we can use to select a desired baud rate.
* To obtain 9600 baud with fCK = 8 MHz:
  + Oversample by 16 (what we do)
    - USARTDIV = 8 000 000/9600
    - BRR = USARTDIV = 833d = 0341h
  + Oversample by 8:
    - USARTDIV = 2 \* 8 000 000/9600
    - USARTDIV = 1666,66 (1667d = 683h)
    - BRR[3:0] = 3h << 1 = 1h
    - BRR = 0x681
* Essentially, we determine a value for BRR by taking the clock frequency and dividing it by the desired baud rate.

Explain your choice of prescaler as well as your timings for your I2C connection to the temperature sensor. Show your computations that lead to your choice.

We use a prescaler of 7 in order to achieve a frequency of 10MHz.

80MHz/(7+1) = 10MHz

Timings explanation: Using the minimum timing needed, we used the following process for finding each of the corresponding values.



What messages do you send to the accelerometer to receive data from it? What does the data you send represent? What does the data you receive represent?

* To receive data from the accelerometer, we send the 6-bit address of the register that we want to read data from. When actually calling the SPI read function, we add two bits to the front of this 6-bit address to make it 8 bits: the first bit tells us whether to read/not write, which we set to 1 to read/receive data, and a bit that tells us whether we are using multi-byte or singular byte communication (we use singular byte communication for this lab, so we set this to 0). We also shift this data over by 8 (empty) bits (since we’re not writing anything over) and send our newly formed 8-bit number as a 16-bit unsigned integer. We send a 16-bit integer: 1 R/W bit (1), 1 MB bit (0), 6 register address bits, and 8 data bits.
* When writing to the accelerometer, we send the 6-bit address of the register that we want to write to, the first two read/not write and MB bits as described above, shift it, and add the 8 bits of data that we want to store in the register. Once again, we send a 16-bit integer: 1 R/W bit (0), 1 MB bit (0), 6 register address bits, and 8 data bits.
* The data we receive represents the data that is stored in the register. Depending on what register’s address we send, we can receive different data that represents information about the accelerometer’s settings, such as bandwidth, range, or frequency.