1 Introduction

Power consumption is of great importance to today's embedded system designers. Embedded devices are increasingly relying on batteries for all their power needs, and as we all know, when that battery dies so does a little piece of you. It gets worse, not only is battery life a major concern, but heat generation has also become an issue as CMOS technology becomes faster and leakier. Failing to control the heat has obvious impacts to handheld devices (imagine burning your face on your phone) as well as plugged in devices such as TV set top boxes. It should be understood that software affects power consumption, and in this lab you will explore the effect that software has on the power consumption of the AT91SAM7L battery-powered microcontroller.

We are able to measure the power consumption on the microcontroller by measuring the voltage and current that flows out of the battery or power supply. Since P = V*I, we are now armed with enough information to calculate the instantaneous power. Furthermore, we can measure the current of just the core (the AT91SAM7L128) without considering the external components on the PCB. Both currents are measureable by inserting a current measuring device in place of the respective shorting block (jumper) located on the PCB of the AT91SAM7L-STK (Starter Kit). The battery's current flows through J9 while core's current flows through J8.

To obtain the current measurement we can either place an ammeter in place of the current shorting block (jumper) or we can place a resistor there instead and measure the voltage drop across it. The latter method allows us to use an oscilloscope to better visualize the current verses time relationship.

Section 5.9 of Computers as Components describes how to isolate and repeat the execution of a small segment of code using a while loop.

The datasheet for the AT91SAM7L128 states that the core should draw no more than 30mA in Active Mode (see page 13) at 36MHz, whereas the voltage regulator is capable of supplying up to 60mA. The voltage regulator is programmable to provide between 1.55V and 1.80V to the core, and the default setting is 1.80V. The power for the SLCD is supplied through a voltage pump/regulator that resides within the core. This power can be optionally supplied externally.

2 Lab Procedure

2.1 Measure the idle power

1. Download and unzip the lab07.zip file from the lab webpage.
2. Download the code into the microcontroller. This code is empty and simply loops in an empty while loop. Run the program either in the debugger or by pressing 'Reset' and then 'Wake Up' on the board.
3. Measure the voltage of the battery supply, Vbat. You may want to measure this voltage before each current measurement because the battery voltage could change during use. To avoid this hassle, use the provided voltage generator to produce a steady and reliable voltage.
4. Use an ammeter to measure both the core current, Icore, and the battery current, Ibat,
5. Calculate the power for each the core and the entire system by using the equation, P = V*I.

TA Initial Lab07 Box 1

2.2 Measure basic arithmetic operations

1. Measure the power consumption for the arithmetic operations: addition, subtraction, multiplication, and division.
2. Compare the results to each other and to the previous results. Comment on the findings.

2.3 Measure SRAM reads and writes

3. Write a small program that repeatedly reads data from the SRAM. An array access should do fine. Measure the power consumption during SRAM reads.
4. Modify the program to write into the SRAM. Measure the power consumption during SRAM writes.
5. Compare the results to each other and to the previous results. Comment on the findings.

TA Initial Lab07 Box 2

2.4 Measure each peripheral

1. Refer to or modify the source code from previous labs to measure the power consumed by the SLCD, ADC, and PWM, individually.
2. Compare the results to each other and to the previous results. Comment on the findings.

TA Initial Lab07 Box 3

2.5 Measure function calls

1. Measure the power consumption of an empty function call.
2. Try implementing the basic arithmetic operations from 2.2 above inside of the function call to observe the power effect of the loop overhead.
3. Implement a recursive function call such as the Euclidean algorithm for GCD (see http://en.wikipedia.org/wiki/Euclidean_algorithm).
4. Measure the power consumption for this recursive call.
5. Implement a non-recursive function call that performs the same calculation and measure its power consumption. (see the Wikipedia page for Euclidean Algorithm above or refer to http://en.wikipedia.org/wiki/Binary_GCD_algorithm for a better algorithm).
6. Compare the results to each other and to the previous results. Comment on the findings. How and why would recursion make a difference to the power consumption?

TA Initial Lab07 Box 4

2.6 Use a resistor to measure power

1. Place a resistor of measured resistance in place of the ammeter.
2. Measure the voltage drop across the resistor and use Ohm's Law (V = I*R) to calculate the current flowing through the resistor.
3. Use the equation P = V*I = V^2 / R to calculate the power.

4. Compare the results to the measurements obtained when using an ammeter. Explain any differences.
5. Use an oscilloscope to see the current waveform as the change in voltage drops across the resistor.
6. Describe the shape of the waveform and explain how your software can change it.

TA Initial Lab07 Box 5

2.7 BONUS: Measure internal flash reads and writes

1. Adapt the basic-internalflash-project to measure the read and write power consumption. The basic-internalflash-project is available as an example program from the IAR Startup Screen (Help → Startup Screen → Example applications).
2. Compare the results to each other and to the previous results. Comment on the findings.

TA Initial Lab07 Box 6

2.8 Compete against your classmates

1. Write a program to consume as much power as possible (Don't worry about batteries – we have plenty).
2. Measure the power consumption of this monster and try to consume more power than the rest of the class.
3. We will reward the winner with a fitting prize once all results have been tabulated.

TA Initial Lab07 Box 7

3 Lab Report

Your lab report should contain the following items:

1. Submit ALL power measurements with respective comparisons, comments, and answers to questions asked.
2. Answer the following questions
   - Why are Ibat and Icore different?
   - It takes a small while for the current to reach steady state right after pressing the "Wake Up" button on the board. Why?
   - How could code optimizations (Lab07) impact the power consumption of a program? Give 3 examples.
   - The AT91SAM7L offers several features for lowering power consumption. Refer to the datasheet, and describe at least 3 of these features in a few sentences.
   - What were the most power consuming devices / software routines? Why might these consume more power than the others?
   - What were some surprising results from this lab? Can you offer an explanation for these happenings?