

Design Assignment 1

Embedded 'C' Programming Basics

Kevin Cao and Dhruvit Naik

ELC 411-02

Dr. Pearlstein

September 11, 2015

Introduction

The purpose of the lab was to introduce programming for embedded systems using the C language. We used PSOC Creator as our IDE. The lab focused on data types and how they are converted from signed to unsigned and vice versa, identifying registers and defining them as macros, use of hexadecimal numbers within embedded systems, the usefulness of bitwise operations, such as the and, or, and shift operations to manipulate data within the registers, and cycle counting.

Before beginning the actual lab, we were required to review the C language on our own. In addition, we had to understand conversions between smaller unsigned types and larger unsigned types, smaller signed types to larger signed or unsigned types, and conversions between signed and unsigned types of the same size.

The task of the lab was to create a small program that read and write to two registers within the PSOC board, `NVIC_CPUID_BASE` and `PRT6_DR`. After manipulating the data within the registers we were to observe the effects they had on the PSOC board. Before beginning the lab the we were required to review the C language on their own. We were also required to look at the `stdint.h` library in order to familiarize ourselves with the data types we would be working with. Furthermore, the addresses and subfields of `NVIC_CPUID_BASE` and `PRT6_DR` were identified before coming into lab. The students were required to write a program to change the values within the `NVIC_CPUID_BASE` and `PRT6_DR` registers. The subfields of the registers were also recorded along with the bit field locations. The students also needed to review C integer types and sign extension before starting the lab. Specifically the students reviewed the variable types defined in `stdint.h` and reviewed the rules when a conversion is made between signed and unsigned integer types.

The lab also emphasized the importance of using pointers in order to access memory mapped peripherals. To properly read the peripherals the students needed to ensure that the correct bit positions were read.

[31:24] IMPLEMENTER - Implementer code. ARM is 0x41

[23:20] VARIANT - Implementation defined variant number.

[15:4] PARTNO - Number of processor within family

[3:0] REVISION - Implementation defined revision number.

PRT6_DR: The data written to this register specifies the high (Data=1) or low (Data=0) state for the GPIO pin at each bit location of the selected port.

Results/Discussion

At the end of the lab it was found that the addresses of NVIC_CPUID_BASE and PRT6_DR are 0xe000ed00 and 0x40005160 respectively. THE NVIC_CPUID_BASE register is a 32 bit register where bits [31:24] refer the the implementer subfield, bits [23:20] refer to the variant subfield, bits [15:4] refer to the partno subfield, and bits [3:0] refer to the revision subfield. After manipulating the [3:2] bits of PRT6_DR it was observed that the LED lit up when the bits were set as 1 and turned off when they were set back to 0. This is because the bits within the PRT6_DR specify the states for the GPIO pins at each bit location of the selected port.

Conclusion

In the end we were able to successfully create a C program that manipulated the NVIC_CPUID_BASE and PRT6_DR addressed found within the PSOC board. The addresses were accessed by creating macros to store their hexadecimal address value. A pointer was then used to store the address of the macro, and was dereferenced when the values within the pointers were needed. The simplest and most effective way to manipulate the addresses and their values was by using the >>, &, and | operators to change the individual bits.

1. How does “C” handle the following:

- a. converting a smaller unsigned type to a larger signed or unsigned type

When an unsigned type is converted to a larger signed or unsigned type it is converted using zero-extending. This is done by padding the left side with zeroes until it matches the desired length. The values are preserved unless it is converted by a float.

- b. converting a smaller signed type to a larger signed or unsigned type

When converting to an unsigned and the signed type is positive, the value is unchanged. Sign-extending is done, a similar process to zero-extending, except ones or zeroes are padded to the left side depending on the sign bit.

- c. converting signed and unsigned types of the same size

When converting an unsigned type to a signed type of the same size the bit pattern remains the same except for the MSB which becomes the sign bit. When converting a signed bit to an unsigned bit of the same size the pattern is preserved and the MSB loses its function of as a sign bit.

2. The requested register addresses that you identified

The address of the NVIC_CPUID_BASE register is 0xe000ed00 and the address of the PRT6_DR is 0x40005160

3. The values that you observed in the watch window, during single step, along with explanations

- Commented in the ‘C’ Code