

Tutorial 4: Arithmetic and Logic Operations

Problem 1: Data Representation

Consider the number $A = 0xEEEEEDDDD$. What is the value of this number in decimal? What is its value when represented in 8-bit unsigned and signed numbers? What is its value when logical logically shifted to the right by two bits and represented in 8-bit unsigned and signed numbers? What is its value when arithmetically shifted to the right by two bits and represented in 8-bit unsigned and signed numbers?

Problem 2: Shift Operations in C

Consider the C code in Figure 1. Answer the following questions.

What are the outputs of the `printf` statements?

What are the outputs of the `printf` statements if (`a = 0x7722`)?

```
#include <stdio.h>

int main (void)
{
    short a = 0xDEE5;
    char b;

    b = a;
    printf("short = \"%d\"\n\n", a >> 1);
    printf("char = \"%d\"\n\n", b >> 1);

    return 0;
}
```

Figure 1: Program on Shift Operations

Problem 3: Rotate Operation

Consider the ARM Assembly code in Figure 4 that does rotation of bits in a register. Write an equivalent C version of this program. If (V1 = 0xEEBAE213), what would register A1 contain after the execution this instruction?

```
mov a1, v1, ror #8 ;a1 ← v1 >> 8 bits , a1[31:24] ← v1[7:0]
```

Figure 4: Assembly Program on Rotation

