

G51CSA Lab Exercise TWO: Addressing Modes

Steven R. Bagley

Introduction

This exercise is designed to get you familiar with the different addressing modes available on the ARM CPU. To do this, you'll implement the string copy routine in a variety of different ways. The C code for a string copy routine is presented below:

```
char *dest, *src;

while(*src != 0)
{
    *dest++ = *src++;
}

*dest++ = 0;
```

You will need to translate this into ARM assembler. In terms of what memory to use for the destination, I'd just place a label with a single `DEFB` after your ARM code and use the address of that. This will end up being the address of the rest of the computer's memory which won't be used for anything else, e.g. the code might look like this:

```
        B main
srcmsg  DEFB "Vworp! Vworp! "

main    ; Your string copy code goes here

        ; Print out the copied string
        ADR R0, dest
        SWI 3
        SWI 2
dest    DEFB 0
```

The code for this exercise should be relatively straight forward, although remember you'll need to make sure the destination string is terminated with a zero-byte. I suggest you start by implementing the copy routine by using straight-forward register indirect memory access and some simple `ADD` instructions to change the addresses accessed. Once you have that version working, make a copy of it and then modify the copy to use indirect with a register index access instead. Finally, you might want to try using post-indexed indirect addressing with writeback to implement the routine.