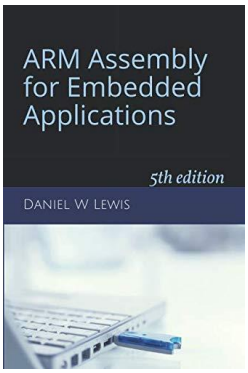

ARM Instructions Worksheet #3

Addressing Modes

Prerequisite Reading: Chapter 4
Revised: March 26, 2020

Objectives: *To use the web-based simulator ("CPUlator") to better understand the four addressing modes:*

1. Immediate Offset Addressing: `[R1]` and `[R1,4]`
2. Register Offset Addressing: `[R1,R2]` and `[R1,R2,LSL 2]`
3. Post-Indexed Addressing: `[R1],4`
4. Pre-Indexed Addressing: `[R1,4]!`

To do offline: Answer the questions that follow the listing below. (Numbers at far left are memory addresses.)

```
                    .syntax      unified
                    .global      _start
                    .skip        0x100

00000100  Array32:  .word  0xBEEFBEEF          // uint32_t Array[4] ;
00000104            .word  0xC0DEC0DE
00000108            .word  0xF00DF00D
0000010C            .word  0xFACEFACE

00000110  _start:   LDR    R1,=Array32         // *** EXECUTION STARTS HERE ***
00000114            LDR    R0,[R1]             // Address provided by R1
00000118            LDR    R0,[R1,4]           // Address = R1 + 4

0000011C            LDR    R2,=8               // R2 = Offset = 8
00000120            LDR    R0,[R1,R2]          // Address = R1 + R2

00000124            LDR    R2,=3               // R2 = Subscript = 3
00000128            LDR    R0,[R1,R2,LSL 2]    // Address = R1 + 4*R2

0000012C            LDR    R0,[R1],4           // Address = R1; Post-Increment
00000130            LDR    R0,[R1,4]!          // Address = R1 + 4; Pre-Increment

00000134  done:     B      done                // infinite loop

                    .end
```

What hex *address* is copied into `R1` by the `LDR` instruction at address 00000110_{16}?

The address of Array32[0] or 00000100

What hex *data* is copied from the address in R1 by the LDR at address 00000114_{16}?

The data of Array32[0] or BEEFBEEF

What hex *data* is copied into `R0` by the LDR instruction at address 00000118_{16}?

The data of Array32[1] or C0DEC0DE

What hex *address* did that value come from?

The address of Array32[1] or 00000104

What hex *data* is copied into `R0` by the `LDR` instruction at address 00000120_{16}?

The data of Array32[2] or F00DF00D

What hex *address* did that value come from?

The address of Array32[2] or 00000108

What hex *data* is copied into `R0` by the `LDR` instruction at address 00000128_{16}? **The data of Array32[3] or FACEFACE**

What hex *address* did that value come from? **The address of Array32[3] or 0000010C**

What hex *data* is copied into `R0` by the `LDR` instruction at address $0000012C_{16}$? **The data of Array32[0] or BEEFBEEF**

What hex *address* did that value come from? **The address of Array32[0] or 00000100**

What hex *address* is left in `R1` by the `LDR` instruction at address $0000012C_{16}$? **00000100+4 or 00000104**

What hex *data* is copied into `R0` by the `LDR` instruction at address 00000130_{16}? **The data of Array32[2] or F00DF00D**

What hex *address* did that value come from? **00000104+4 or 00000108**

What hex *address* is left in `R1` by the `LDR` instruction at address 00000130_{16}? **00000104+4 or 00000108**

Getting ready: Now use the simulator to collect the following information and compare to your earlier answers.

1. Click here to open a browser for the ARM instruction simulator with pre-loaded code.
2. Press Ctrl-M to open the memory display window and drag-n-drop it about halfway to the right.
3. In the "Memory" window, enter `0x100` into the search box and press Enter to highlight that address for easy reference.

Step 1: *Press `F2` exactly **2** times to execute the first two `LDR` instructions. (The 3rd `LDR` should be highlighted in yellow.)*

What hex *address* is copied into `R1` by the `LDR` instruction at address 00000110_{16}? **00000100**

What hex *data* is copied from the address in R1 by the LDR at address 00000114_{16}? **beefbeef**

Step 2: *Press `F2` exactly **once** to execute the `LDR R0,[R1,#4]`*

What hex *data* is copied into `R0` by the LDR instruction at address 00000118_{16}? **c0dec0de**

What hex *address* did that value come from? **00000104**

Step 3: *Press `F2` exactly **2** times to execute the `LDR R2,=8 (MOV R2,#8)` and the `LDR R0,[R1,R2]`*

What hex *data* is copied into `R0` by the `LDR` instruction at address 00000120_{16}? **f00df00d**

What hex *address* did that value come from? **00000108**

Step 4: *Press `F2` exactly **2** times to execute the `LDR R2,=3 (MOV R2,#3)` and the `LDR R0,[R1,R2,LSL #2]`*

What hex *data* is copied into `R0` by the `LDR` instruction at address 00000128_{16}? **faceface**

What hex *address* did that value come from? **0000010c**

Step 5: *Press `F2` exactly **once** to execute the `LDR R0,[R1],#4`*

What hex *data* is copied into `R0` by the `LDR` instruction at address $0000012C_{16}$? **beefbeef**

What hex *address* did that value come from? **00000100**

What hex *address* is left in `R1` by the `LDR` instruction at address $0000012C_{16}$? **00000104**

Step 6: *Press `F2` exactly **once** to execute the `LDR R0,[R1,#4]!` .*

What hex *data* is copied into `R0` by the `LDR` instruction at address 00000130_{16}? **f00df00d**

What hex *address* did that value come from? **00000108**

What hex *address* is left in `R1` by the `LDR` instruction at address 00000130_{16}? **00000108**