# CS-5009: Lab 1 – Worksheet

**Roll No: 111601017 Date:**  9-8-2019

**Name:** Rajendra Singh

Part 1: Memory access and moving data

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **Value after execution of instruction** | | | | | | | |
| LDR R1,const\_val | R1 = 0xDEADBEEF | | | | | | | |
| LDR R0,=const\_val | R0 = 0x00000208 | | | | | | | |
| First 8 bytes of memory starting at address in R0 | | | | | | | |
| EF | BE | AD | DE | 08 | 02 | 00 | 00 |
| LDR R1,[R0] | R1 =0xDEADBEEF | | | | | | | |
| LDRH R1,[R0] | R1 =0x0000BEEF | | | | | | | |
| LDRB R1,[R0] | R1 =0x000000EF | | | | | | | |
| LDR R0,=equate\_val | R0 =0x8BADF00D | | | | | | | |
| LDR R1,=const\_val | R1 =0x00000208 | | | | | | | |
| MOV R2,R0 | R0 =0x8BADF00D | | | | R2 =0x8BADF00D | | | |
| MOVS R2,#0 | R2 =0x00000000 | | | | | | | |

Question:

*LDR R0,=const\_val* has different behavior than *LDR R0,=equate\_val*.

Why? *Because declared using different method(i.e. EQU & DCD)*

Part 2: Arithmetic and logic operations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Instruction** | **Value after execution of instruction** | | | | | |
| MSR APSR,R0  (first) | R1 =0x00000032 | | R2 =0x0000007B | | R3 =0xFFFFFFF0 | |
| N =0 | Z =0 | | C =0 | | V =0 |
| ADDS R2,R1 | R1 =0x00000032 | | | R2 =0x000000AD | | |
| N =0 | Z =0 | | C =0 | | V =0 |
| SUBS R2,R1 | R1 =0x00000032 | | | R2 =0x0000007B | | |
| N =0 | Z =0 | | C =1 | | V =0 |
| ADDS R3,R1 | R1 =0x00000032 | | | R3 = 0x00000022 | | |
| N =0 | Z =0 | | C =1 | | V =0 |
| SUBS R3,R1 | R1 = 0x00000032 | | | R3 =0xFFFFFFF0 | | |
| N =1 | Z =0 | | C =0 | | V =0 |
| MSR APSR,R0  (second) | R1 =0x00000032 | | R2 =0x0000007B | | R3 =0xFFFFFFF0 | |
| N =0 | Z =0 | | C =0 | | V =0 |
| ADD R3,R1 | R1 =0x00000032 | | | R3 =0x00000022 | | |
| N =0 | Z =0 | | C =0 | | V =0 |
| CMP R1,R2 | R1 =0x00000032 | | | R2 =0x0000007B | | |
| N =1 | Z =0 | | C =0 | | V =0 |
| CMP R2,R1 | R1 =0x00000032 | | | R2 =0x0000007B | | |
| N =0 | Z =0 | | C =1 | | V =0 |
| CMP R1,R1 | R1 =0x00000032 | | |  | | |
| N =0 | Z =1 | | C =1 | | V =0 |
| CMP R1,#0x40 | R1 =0x00000032 | | |  | | |
| N =1 | Z =0 | | C =0 | | V =0 |
| CMP R2,#0x40 | R1 =0x00000032 | | |  | | |
| N =0 | Z =0 | | C =1 | | V =0 |
| CMN R1,R3 | R1 =0x00000032 | | | R3 =0x00000022 | | |
| N =0 | Z =0 | | C =1 | | V =0 |
| CMN R1,R3 | R1 =0x00000032 | | | R3 =0x00000022 | | |
| N =0 | Z =0 | | C =0 | | V =0 |

Part 3: Unconditional Branches

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Instruction** | | **Value after execution of instruction** | | |
| B | spot3 | R15 | (PC) | =0x000001F8 |
| B | spot4 | R15 | (PC) | =0x000001FA |
| B | spot2 | R15 | (PC) | =0x000001F6 |
| B | spot1 | R15 | (PC) | =0x000001F4 |

Question:

*What is the address of:*

spot1: 0x000001F4

spot2: 0x000001F6

spot3: 0x000001F8

spot4: 0x000001FA

Part 4: Conditional Branches

***BNE Instruction***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Loop** | **Instruction** | | **Value after execution of instruction** | | | |
| 1 | SUBS R0,#1 | | R0 =0x00000002 | | | |
| N =0 | Z =0 | C =1 | V =0 |
| BNE | dec\_cnt | R15 (PC) =0x000001F6 | | | |
| 2 | SUBS R0,#1 | | R0 =0x00000001 | | | |
| N =0 | Z =0 | C =1 | V =0 |
| BNE | dec\_cnt | R15 (PC) =0x000001F6 | | | |
| 3 | SUBS R0,#1 | | R0 =0x00000000 | | | |
| N =0 | Z =1 | C =1 | V =0 |
| BNE | dec\_cnt | R15 (PC) =0x000001FA | | | |
| 4 | SUBS R0,#1 | | R0 =0x00000002 | | | |
| N =0 | Z =0 | C =1 | V =0 |
| BNE | dec\_cnt | R15 (PC) =0x000001F6 | | | |

***BGE Instruction***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Loop** | **Instruction** | | **Value after execution of instruction** | | | |
| 1 | SUBS R0,#1 | | R0 =0x00000001 | | | |
| N =0 | Z =0 | C =1 | V =0 |
| BGE | dec\_cnt | R15 (PC) =0x000001F6 | | | |
| 2 | SUBS R0,#1 | | R0 =0x00000000 | | | |
| N =0 | Z =1 | C =1 | V =0 |
| BGE | dec\_cnt | R15 (PC) =0x000001FA | | | |
| 3 | SUBS R0,#1 | | R0 =0x00000002 | | | |
| N =0 | Z =0 | C =1 | V =0 |
| BGE | dec\_cnt | R15 (PC) =0x000001F6 | | | |
| 4 | SUBS R0,#1 | | R0 =0xFFFFFFFF | | | |
| N =1 | Z =0 | C =0 | V =0 |
| BGE | dec\_cnt | R15 (PC) =0x000001FA | | | |

Part 5: Subroutines with Linked Branches

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Loop** | **Instruction** | | **Value after execution of instruction** | | | |
| 1 | LDR | R0,=value1 | R0 =0x1FFFF000 | | | |
| First 4 bytes of memory starting at address in R0 | | | |
| 00 | 00 | 00 | 00 |
| 1 | BL | change\_value | R14 (LR) =0x00000217 | | | |
| R15 (PC) =0x0000021E | | | |
| 1 | LDR R0,=value2 | | R0 =0x1FFFF004 | | | |
| First 4 bytes of memory starting at address in R0 | | | |
| 00 | 00 | 00 | 00 |
| 1 | BL | change\_value | R14 (LR) =0x0000021D | | | |
| R15 (PC) =0x0000021E | | | |
| 2 | LDR R0,=value1 | | R0 =0x1FFFF000 | | | |
| First 4 bytes of memory starting at address in R0 | | | |
| 01 | 00 | 00 | 00 |
| 2 | BL | change\_value | R14 (LR) =0x00000217 | | | |
| R15 (PC) =0x0000021E | | | |
| 2 | LDR R0,=value2 | | R0 =0x1FFFF004 | | | |
| First 4 bytes of memory starting at address in R0 | | | |
| 05 | 00 | 00 | 00 |
| 2 | BL | change\_value | R14 (LR) =0x0000021D | | | |
| R15 (PC) =0x0000021E | | | |