* This homework is due on printed paper at the start of the second lecture in Week 6.
* Your solution for each problem must show your work to receive partial credit. Failure to show your thought process will not earn points for incorrect answers.

1. Two instructions have been stored in instruction memory using little-endian format. These binary numbers were created by the assembler from assembly language statements typed by the programmer. Determine the assembly language statements typed by the programmer by disassembling the binary numbers back into their text form.

Address 4-7: CMP R7, #448

Address 0-3: SUB R7, R1, R9

|  |  |
| --- | --- |
| ADDRESS | HEX BYTE |
| 0 | 09 |
| 1 | 70 |
| 2 | 41 |
| 3 | E0 |
| 4 | 07 |
| 5 | 0D |
| 6 | 57 |
| 7 | E3 |

1. Implement the following Java code in ARM assembly language. You can use the MUL instruction shown on your ARM reference card for this problem. You must use the assigned registers noted in the comments. Your solution must be well-commented ARM assembly code and the Keil uVision4 simulation showing the final values after executing to the infinite loop. Write hand-written comments on the simulation that describe how you know that R5 contains the correct answer.

|  |  |  |
| --- | --- | --- |
| CE1921 HOMEWORK HW5 | SPRING 2019 | NAME: Julian Singkham |

// s is stored in ARM saved value register R5

// iterator i is stored in ARM temp register R12 // multiplication destination is saved value register R6 int s=0; for(int i=0; i<=10; i++)

{

s = s + i\*i;

}

while(1);



Dr. Russ Meier, Milwaukee School of Engineering, Last Update: April 4, 2019

1. Determine the rotation and immediate fields required to create these constants. An answer of “not possible” is allowed.

|  |  |  |
| --- | --- | --- |
| ROTATION | IMMEDIATE | RESULTS IN |
| 0011 | 00101001 | 0x0000A400 |
| Not | Possible | 0x000007D8 |
| 1011 | 0101 1101 | 0x00017400 |
| 1101 | 01100110 | 0x00001980 |

|  |  |  |
| --- | --- | --- |
| CE1921 HOMEWORK HW5 | SPRING 2019 | NAME: Julian Singkham |

1. Describe the error in each of these ARM instructions.

|  |  |  |
| --- | --- | --- |
| INSTRUCTION | | ERROR |
| ADD | R3,R7,#1023 | #1023 is not a possible immediate value. |
| SUB | R1,R3,R18 | R18 does not exist. Arm only has registers R0-R15 |
| EOR | R3,#0xFF,R2 | Incorrect Syntax, #0xFF must be after R2 |
| MOV R2,R3,R5 | | Too many inputs. Move can only have 2 inputs. Rd and RM/IMM |

1. ARM allows the second ALU source operand to be shifted by the barrel shifter before an arithmetic operation. Logical shifts discard the sign bit and direction is represented by the mnemonics LSL and LSR for left and right respectively. The sign bit can be retained when right shifting by using the mnemonic ASR. Determine the equation calculated by this ARM instruction.

ADD R4,R5,R2, LSL #4

R4=R5+(R2\*16)



Dr. Russ Meier, Milwaukee School of Engineering, Last Update: April 4, 2019

