Christopher Martin, Jennifer Wolfe, Geoffrey Corey
April 29, 2013
CS 472 HW 4

Chapter 3 Homework Questions

(3.8) **Compare general purpose registers versus separate data and address registers.**

(Advantage) The general-purpose registers hold all the information needed for the address and then registers to keep the data.
(Disadvantage) This could make the opcode/instruction more complicated.

(3.9) **What is a misaligned operand? Why are misaligned operands such a problem in programming?**

When you complete shifts where the byte overlaps into the next byte. It causes something like overflow and messes up results.

(3.24) **What is the meaning of each of the P, U, B, W, and L bits in the encoding of an ARM memory reference instruction?**

P - pre/post index increment point
U - up/down - increments the pointer or decrements the pointer
B - byte/word access
W - write back - write or don't write back adjusted pointer
L - load/store - load into register or store in memory

(3.26/) **What is the effect of LDR r0, [r5, r6, LSL r2] ?**

Shift r6 left by the value of r2. Then offset r5 by the new value of r6 and load into r0.

(3.30) **What is meaning of sign-extension in the context of copying data from one location to another?**

For example, when moving an 8 bit into 32 bit, the sign is extended across the remaining unfilled bits.

(3.33) **Most RISC processors do not include a block move instruction. What are the advantages and disadvantages of the ARM' LDM and STRM instructions?**

(Advantage) You can move large amounts of data at once.
(Disadvantage) The calls only work for r0-r7 registers.

(3.34) **What is the effect of executing STMB r13! r0-r2,r4? Draw a picture of the state of the stack pointed at by r13 before and after this operation.**

Copies registers r0-r2 and r4 into sequential locations using r13 that is incremented before each transfer.

(3.36) **Without using the ARM's multiplication instruction, write one or more instructions (using ADD, SUB and shifting) to multiply by the following integers.**

A. 33
LSL r0, #6
ADD r0, #1

B. 1025
LSL r0, #11
ADD r0, #1

C. 4095
LSL r0, #13
SUB r0, #1

(3.44) **What does the following code do?**

TEQ r0, #0 ;tests as, EOR, to see if r0 is equal to 0.
RSBMI r0, r0, #0 ;if N flag is 1 (set) than the reverse subtract of 0-r0 is calculated in stored in r0

(3.48) **What in the context of assembly language is a pseudo-operation?**

It is shorthand instructions that won't be a part of ISA. Examples, LDR and ADR (loads data and loads address, respectively, into the register)

(3.54) **Explain what this fragment of code does instruction by instruction and what purpose it achieves (assuming that register r0 is the register of interest). Note that data in r0 must not be 0 on entry.**

MOV r1, #0
loop MOVS r0, r0, LSL #1
ADDCC r1, r1, #1
BCC loop

(3.60) **A computer has three eight-element vectors in memory. Va, Vb, Vc. Each element of a vector is a 32-bit word. Write the code to calculate all elements of Vc if the ith element is given by VCi = 1/2(Cai + Vbi)**

(3.61) **Register r15 is the program counter. You can use it with certain instructions such as MOV (e.g., MOV pc, r14). However r15 cannot be used in conjunction with most data processing instructions. Why?**