Drake Bridgewater & Ryan Phillips
April 26, 2014
CS 472 HW 4

(3.8)

What are the relative advantages and disadvantages of general-purpose registers compared to separate address and data registers?

(3.9)

What is a misaligned operand? Why are misaligned operands such a problem in programming?

A misaligned operand is just what it sounds like an operand that is not aligned causing the operand to span two cache lines either slowing the system down substantially or causing a bus error. From some articles online typically x86 and x64 will happily work with unaligned data but will tend to be slower where as on other processors the system will just error. From the ARM website "Non word-aligned load and store multiple, double, semaphore, synchronization, and coprocessor accesses always signal Data Abort with an Alignment fault status code when the U bit is set." [1]

(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 Do you want to adjust pointer before using?
U Do you want to decrement the pointer instead of incrementing it?
B Is this word access?
W Do you want to update the pointer after use?
L Do you want to store data in memory?

(3.26)

What is the effect of LDR r0, [r5, r6, LSL r2] ?

LDR is the load with offset. It will load a double word from R5 and R6 with an immediate offset specified by the left shift of R2 into R0

(3.30)

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

(3.33)

Most RISC processors do not include a block move instruction. What are the advantages and disadvantages of the ARM's LDM and STM instructions?

(3.34)

What is the effect of executing SMBIB r13,{r0-r2,r4}? Draw a picture of the state of the stack pointed at by r13 before and after this operation.

(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

B 1025

(3.44)

What does the following code do?

```
TEQ     r0, #0
RSBMI   r0, r0, #0
```

(3.48)

What, in the context of assembly language, is a psuedo-operation?

(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 the 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, and 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:

$Vc_i = \frac{1}{2}(Va_i + Vb_i)$
note: This is a repeat of the one of the exercises we did for lab 3.

(3.61)

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

References

[1] ARM Ltd. The architecture for the digital world, 2014.