Annie Hayes ahayes5@g.clemson.edu

Homework
Due: Saturday, April 10, 2021, midnight
Submit through Canvas

This homework is designed to help you study for the 3rd exam. Complete the problems below. Show all work. Your answers must be in RED. Many of these come directly from the slides. However, if you simply copy them, you will not learn anything and will not do well on the exam.

1. Show the steps needed to multiply the following binary numbers:
   100_2(M = multiplicand) * 101_2(Q = multiplier)
   Step 0: Initialize the data M = 100 C = 0 Acc = 0 Q = 101
   Fill in the remaining steps.

   Step 1: C ACC Q
   0 000 101
   \+ 100 ---- lsb = 1
   ---
   0 100 101 shift >> 1
   0 010 010

   Step 2 C ACC Q
   0 010 010
   \+ 000 ---- lsb = 0
   ---
   0 010 010 shift >> 1
   0 001 001

   Step 3 C ACC Q
   0 001 001
   \+ 100 ---- lsb = 1
   ---
   0 101 001 shift >> 1
   0 010 100

   FINAL: 10100 = 20

2. With respect to dividing by powers 2's complement using right shifting, it is common for rounding errors to occur, which can accumulate causing larger problems. In class, we discussed how "biasing" the value before shifting can correct this problem. Given -12340 = 1100111111001100 complete the chart below. This is the chart directly from the slides. Yes, you can simply copy the answers. However, to help you grasp this I strongly suggest you make sure you understand what this is doing. Such as, what is the bias and how was it determined. Why am I adding the bias? Etc.
   Bias: 2^k - 1

Annie Hayes ahayes5@g.clemson.edu

K	Bias	-12340 + bias (binary)	>> K (binary)	Decimal	$12,340/2^k$
0	(2^k-1) = 0	-12340 + 0 = 1100 1111 1100 1100	1100 1111 1100 1100	-12340	-12340.0
1	2^k-1) = 1	-12340 + 1 = 1100 1111 1100 1101	1110 0111 1110 0110	-6170	-6170.0
4	2^k-1) = 15	-12340 + 15 = 1100 1111 1101 1011	1111 1100 1111 1101	-771	-771.25
8	2^k-1) = 255	-12340 + 255 = 1101 0000 1100 1011	1111 1111 1101 0000	-48	-48.203125

3. Determine the fractional representation of the following positional binary representations. These are not IEEE.
   Show your work.

Again, this is directly off of the slides, you must show your work. If you do not you will not get credit.

Binary representation	Value	Decimal
0.0	0/2	0.0
0.01	1/4	0.25
0.010	2/8	0.25
0.0011	4 bits after decimal 2^4 = 16 0011 = 3 3/16	0.1875

Annie Hayes ahayes5@g.clemson.edu

0.00110	5 bits after decimal 2^5 = 32 110 = 5 5/32	0.1875
0.001101	6 bits after decimal 2^6 = 64 1101 = 13 13/64	0.203125
0.0011010	7 bits after decimal 2^7 = 128 11010 = 26 13/64	0.203125
0.00110011	51/256	0.19921875

4. Using positional notation for fractional binary numbers convert the following binary number to decimal. You must show your work.
   1010.101 This is not IEEE. There are examples of this on the slides.
   1010 = 2 + 8 = 10
   .101 ->(2^3 = 8, 101 = 5) = 5/8 = .625
   10.625

5. For both single and double precision, name the three sets of bit(s) that make up the IEEE 754 floating point number. Also list the number of bits required for each of the three parts that make up the single and double precision.
   Signed bit, exponent bits, mantissa bits
   Single precision # of bits: signed – 1, exponent – 8, mantissa - 23
   Double precision # of bits: signed – 1, exponent – 11, mantissa - 52

6. An IEEE Floating-Point value encoded by a given bit representation can be divided into three different cases with the third having two variants depending on the value of the mantissa (frac). Name and describe each case.
   a. Normalized – when the bit pattern of exponent portion is neither 0s nor 1s
   b. De-Normalized – when the exponent field is 0
   c. Infinity – occur when all the exponents are 1s – if the fractional field are 0s (with s equals 0 as positive infinities and s equals 1 as negative)
   d. NaN – when the fraction is not all 0s (used when there is uninitialized data)

C – Review

Annie Hayes ahayes5@g.clemson.edu

7. What is the purpose of the C function memcpy?
   It copies a certain amount of bytes
8. What is the purpose of the C function memset?
   Used to fill memory byte-by-byte for a specified value
9. What is the purpose of the C function memcmp?
   Compares byte-by-byte of two values to find if the values are equal, less than, or greater than each other

Chapter 3

10. Describe what the compile flag -Og does.
    Tells the compiles to apply a level of optimization that yields machine code that follows the overall structure of the original code
11. Describe the significance of the compile flag -S.
    Shows the assembly code (from the compiler)
12. Describe the significance of the compile flag -c.
    The -c command will tell the compiler to compile and assemble the code
13. If I want to be able to understand the content of an ".o" file I can use a disassembler. We discussed two disassemblers in class. Only one of them can be used with linux and Mac OS. What command would I use to view the content of a ".o" file on a linux architecture.
    Objdump -d <filename>.c
14. Much of the success of computers in the last 60+ years has been due to the use of transistors.
15. How many general-purpose registers does x86-64 have?
    16
16. X86-64, also have 4 special-purpose registers. What are they? How many of these are also considered general-purpose registers?
    %rsp, %rbp, %rip, %rflags.
    There are two that are considered general and special purpose registers
17. What is the register used for the return value of a function?
    %rax
18. How many of the general-purpose registers are used for function parameters? List these registers.
    6
19. What happens when a function has more parameters than the registers set aside for them (where are the excess parameters stored)?
    They are passed into the stack
20. What general purpose-register is use as the stack pointer?
    %rsp
21. What special-purpose register is use as the program counter also known as the instruction pointer?
    %rip
22. We discussed three type of operand specifiers used as source values when performing many assembly operations. Name the three and give an example of each.

Annie Hayes ahayes5@g.clemson.edu

Immediate: $0x1F
Register: %eax
Memory: (%esp)

Annie Hayes ahayes5@g.clemson.edu

23. Similar to what we discussed in class, fill in the table showing the values for the indicated operands.

Address	Value	Register	Value
0x100	0xFF	%rax	0x100
0x104	0xAB	%rcx	0x1
0x108	0x13	%rdx	0x2
0x10C	0x11		

Operand	Value
%rdx	0x2
0x108	0x13
$0x10C	0x10C
(%rax)	0xFF
8(%rax)	0x13
10(%rax,%rdx)	0x11
257(%rcx,%rdx)	0xAB
0xFC(,%rcx,8)	0xAB
(%rax,%rdx,4)	0x13

Annie Hayes ahayes5@g.clemson.edu

24. Move is one of the most used instruction for x86-64.
25. Fill in the chart below. This chart represents the sizes of C data types in x86-64.

C Declaration	Intel Data Type	Assembly-code suffix	Size (bytes)
char	byte	b	1
short	word	w	2
int	double word	l	4
long	quad word	q	8
char*	quad word	q	8
float	single precision	s	4
double	single precision	l	8

26. There are 5 possible combinations of source and destination types for the move instruction set. List the five and give an example of each.
    a. Immediate -> Register: (movl $0x1, %eax)
    b. Register -> Register: movw %bp, %sp
    c. Memory -> Register: movb (%rdl, %rcx), %al
    d. Immediate -> Memory: moveb $0x1, (%esp)
    e. Register -> Memory: %rax, (%rbp)