Course Name:Computer Architecture and Assembly Lab

Course Number and Section: 14:332:333:2A

Experiment: [Experiment # [1] – Introduction, GitHub tutorial, Number representation]

Lab Instructor: Jalal Abdulbaqi

Date Performed:9/12/18

Date Submitted:9/26/18

Submitted by: [Samuel Cho 171002966]

Number Representation

1. Unsigned Integers

If we have an *n*-digit unsigned numeral $d_{n-1}d_{n-2}...d_0$ in radix (or base) r, then the value of that numeral is $\sum_{i=0}^{n-1} r^i d_i$, which is basically saying that instead of a 10's or 100's place we have an r's or r^2's

place. For binary, decimal, and hex r equals 2, 10, and 16, respectively.
Just a reminder that in order to write down large number we typically use the IEC prefixing system: Ki = 2^{10}, Mi = 2^{20}, Gi = 2^{30}, Ti = 2^{40}, Pi = 2^{50}, Ei = 2^{60}, Zi = 2^{70} , Yi = 2^{80} .

1.1 Conversions

a. Convert the following from their initial radix to the other two common radices:

0b10001110,
2+4+8+128=142
1000=8 1110=E 0x8E

0xC3BA,
12*16^3+3*16^2+11*16^1+10*16^0=49152+768+176+10=50106
C=1100 3=0011 B=1011 A=1010 0b1100001110111010

81,
81/2=40 1
40/2=20 0
20/2=10 0
10/2=5 0
5/2=2 1
2/2=1 0
1/2=0 1
0b1010001

81/16=1 5
1/16=0 1
0x51

0b100100100,
4+32+256=292
0001=1 0010=2 0100=4
0x124

0xBCA1,

11*16^3+12*16^2+10*16^1+1*16^0 = 45056+3072+160+1 = 48289
B=1011 C=1100 A=1010 1=0001
0b1011110010100001

0,
0x0
0b0

42,
42/2 21 0
21/2 10 1
10/2 5 0
5/2 2 1
2/2 1 0
1/2 0 1
0b101010
42/16 2 A
2/16 0 2
0x2A

0xBAC4
11*16^3+10*16^2+12*16^1+4*16^0 = 45056+2560+192+4 = 47812
B=1011 A=1010 C=1100 4=0100
0x1011101011000100

a. Write the following using IEC prefixes:
2^{14}, 2^4 * 2^10 16 Ki
2^{43}, 2^3 * 2^40 8 Ti
2^{23}, 2^3 * 2^20 8 Mi
2^{58}, 2^8 * 2^50 256 Pi
2^{64}, 2^4 * 2^60 16 Ei
2^{42} 2^2 * 2^40 4 Ti

c. Write the following as powers of 2:
2 Ki, 2^1 *2^10 = 2^11
512 Pi, 2^9 *2^50 = 2^59
256 Ki, 2^8 *2^10 = 2^18
32 Gi, 2^5 *2^30 = 2^35
64 Mi, 2^6 *2^20 = 2^26
8 Ei 2^3 *2^60 = 2^63

2. Signed Integers

Unsigned binary numbers work for natural numbers, but many calculations use negative numbers as well. To deal with this, a number of different schemes have been used to represent signed numbers, but we will focus on two’s complement, as it is the standard solution for representing signed integers.

2.1 Two’s complement

- Most significant bit has a negative value, all others are positive. So, the value of an n-digit two’s complement number can be written as: $\sum_{i=0}^{n-2} 2^i d_i - 2^{n-1} d_n$
- Otherwise exactly the same as unsigned integers.
- A neat trick for flipping the sign of a two’s complement number: flip all the bits and add 1.
- Addition is exactly the same as with an unsigned number.
- Only one 0, and it’s located at 0b0.

2.2 Exercises

For questions 1 – 3, assume an 8-bit integer and answer each one for the case of a two’s complement number and unsigned number, indicating if it cannot be answered with a specific representation.

1.What is the largest integer? The largest integer + 1?
two’s complement 01111111 = 127
+1 makes it 10000000 = -128
unsigned number 11111111= 255
+1 makes it 00000000 = 0

2. How do you represent the numbers 0, 3, and -3?
two’s complement 0=00000000 3=00000011 -3=11111101
unsigned 0=00000000 3=00000011 -3=can’t be represented

3. How do you represent 42, -42?
two’s complement 42=00101010 -42=11010110
unsigned 42=00101010 -42=can’t be represented

4. What is the largest integer that can be represented by any encoding scheme that only uses 8 bits?
11111111=255 for unsigned integer
but if the encoding means that each bit can represent any number, then any of the 8 bits could represent infinitely large integers meaning that there is no definite largest integer that can be represented by 8 bits if there is no specified encoding scheme.

5. Prove that the two’s complement inversion trick is valid (i.e. that x and x’+ 1 sum to 0).
01010101 = 64+16+4+1 = 85
10101011 = -128+32+8+2+1=-85
85+-85=0
Inverting a binary number and adding it will just give you all 1’s as the previous 0’s will now fill in as they are inverted. Adding 1 to a full 8 bits of 1’s will cause it to overflow and you will get

all 0’s with the non existing 9th bit being 1.

6. Explain where each of the three radices shines and why it is preferred over other bases in a given context.
Decimal system is useful in counting because we have 10 fingers and it the universal system for math and everyday use. The binary system is useful for computers and electronics as they work on a systems of switches that can be on or off. Representing values as either 1 or 0 is more efficient for computers to hold data. Using hexadecimal numbers is for a more condensed and easier way to represent binary numbers.

3. Counting

Bitstrings can be used to represent more than just numbers. In fact, we use bitstrings to represent everything inside a computer. And, because we don’t want to be wasteful with bits it is important that to remember that n bits can be used to represent 2^n distinct things. For each of the following questions, answer with the minimum number of bits possible.

3.1 Exercises

1. How many bits do we need to represent a variable that can only take on the values 0, π or e?
2 bits to represent these values.
00 for 0
01 for π
10 for e
and 11 is extra

2. If we need to address 2TiB of memory and we want to address every byte of memory, how long does an address need to be?
2TiB is 2^41 bytes
an address needs to be 41 bits long to represent 2^41 different bytes.

3. If the only value a variable can take on is e, how many bits are needed to represent it.
1 bit to represent e.