Akshat Keshan

304031039

Homework 1

2.66)

int leftmost\_one(unsigned x) {

x = x | x>>1;

x = x | x>>2;

x = x | x>>4;

x = x | x>>8;

x = x | x>>16;

x = x | x>>32;

x = x + 1;

x = x >> 1;

return x;

}

2.71)

A. The function does not sign extend the byte to a 32-bit int as the right shift is executed on an unsigned number. After the operation is performed, the sign bit of the int is always 0.

B.

int xbyte(packed\_t word, int bytenum)

{

return ( (int) word << (3 – (bytenum << 3))) >> 24;

}

2.72)

A. sizeof(val) returns a value of type size\_t which is unsigned. After the difference is taken between the int and the unsigned value in the original if statement, the result is always greater than or equal to zero explaining why the code always succeeds.

B. You can rewrite the conditional test by changing the if condition to:

void copy\_int(int val, void \*buf, int maxbytes){

if(maxbytes-((int)sizeof(val)) >= 0)

memcpy(buf, (void \*) &val, sizeof(val));  
}

2.81)

A. No. Consider x = TMIn and y = 0.

B. Yes. Ring properties of two’s compliment arithmetic

C. No. The sum of the compliments is not equal to the compliment distributed into the sum as proven by the integer representation.

D. Yes. Unsigned integers are related in this way to two’s compliment arithmetic.

E. Yes. Due to rounding.