**1. (4.1 Largest radix-r number)**

Since X0=1, we can define a function which has the value of a string of decimal digits.

(dN−1, … ,d0)10 = (N−1)∑(i=0) di ∗ 10i

For n = 0, r^0 – 1 = 0

For n =1, r^1 -1 = r -1

If n = k, the largest number to present is r^k -1

For (k+1) digits, (r^k-1)\*r+(r-1)=(r^(k+1)-r+r-1)=r^(k+1)-1

Then we can prove that the largest number we can represent with

N digits of radix r is rN – 1

**2. (4.2 Carry bits)**

First, the carry bits are always 0 or 1.

The largest number of 1 digit is r-1, the carry is 1

The smallest number 0, 0+0=0 with carry 0.

**3. (4.3 Complement number range)**

Given the formal definition, derive the minimum and maximum two’s complement numbers that can be represented in N bits.

The largest is (2^(i), i<=n-1)

The smallest is: (-)(2^(n))

We can get (2^(n))-1+(2^(n))=(2^(n+1)-1)

**4. (4.4 2's complement operation)**

For a number B with magnitude less than 2N-2, show that if B is represented by a 2’s complement number with N bits bN-1…b0

**5. (4.5 Sign extension)**

Prove that “sign-extension” is value preserving

For positive numbers, 0s on the left would not affect the result. A negative number will convert to decimal with inverting each bit and add one to get the result, 0s also have no effects to the result. So the sign-extension is value preserving.

**6. Using the technique presented in section 4.1, convert the following decimal number to binary**

107

107 53 26 13 6 3 1

1 1 0 1 0 1 1

107 to decimal is: 1101011

2312

2312 1156 578 289 144 72 36 18 9 4 2 0

0 0 0 1 0 0 0 0 1 0 0 1

2312 to decimal is: 100100001000

31333

31333 15666 7833 3916 1958 979 489 244 122 61 30 15 7 3 1

1 0 1 0 0 1 1 0 0 1 0 1 1 1 1

31333 to decimal is: 111101001100101

97

97 48 24 12 6 3 1

1 0 0 0 0 1 1

97 to decimal is: 1100001

**7. Perform the following subtraction operations using complements as described in section 4.2**

103-92

103

+ 8

1. 1

Result is: 11

1027-11

1027

+ 89

1016

Result is: 1016

129-33

129

+ 67

96

Result is: 96

2222-222

2222

+ 778

2000

Result is: 2000

**8. Convert the following decimal numbers to 8-bit two's complement**

-91

91 45 22 11 5 2 0

1 1 0 1 1 0 1

8-bit: 01011011

Invert: 10100100

+ 1

Result is: 10100101

-96

96 48 24 12 6 3 1

0 0 0 0 0 1 1

8-bit: 01100000

Invert: 10011111

+ 1

Result is: 10100000

-126

126 63 31 15 7 3 1

0 1 1 1 1 1 1

8-bit: 01111110

Invert: 10000001

+ 1

Result is: 10000010

101

101 50 25 12 6 3 1

1 0 1 0 0 1 1

Result of 8-bit is: 01100101

78

78 39 19 9 4 2 0

0 1 1 1 0 0 1

Result of 8-bit is: 01001110