1. 4.1 Largest radix-r number

**Exercise 4.1: Largest radix-r number**

**Prove, using induction, that for radix-r, the largest number that can be**

**represented with *N* digits is .**

Proposition: For radix-r, the largest number that can be represented with *N* digits is .

Proof: For any radix-r system, a number is represented by

where .

We can expand this summation as follows:

By cancelling out terms, we get the final result of:

Thus, we have proved that the maximum value that can be represented by any radix-r with N digits is .

**2. 4.2 Carry bits**

**Prove that for radix-r addition, the carry bits are always 0 or 1.**

Proof: In any radix-r system the maximum sized number we can represent in base r with N digits is . We can show that there is no way to have a carry greater than 1 when adding two max sized radix-r numbers if we can show that . The reason for this is that means the carry would be greater than or equal to 2 and not 1. An example of this can be shown using our decimal number system. If we add 9 + 9, the largest number we can use in radix-10, we see the result is 18 and has a carry of 1. This is illustrated below:

|  |  |
| --- | --- |
| 1 | 0 |
|  | 9 |
| + | 9 |
| 1 | 8 |

Addition of :

We also need to prove that the carry can never be less than zero for any radix-r. We can do this by adding the smallest digits in any radix-r system. Since we are only working with positive radix, the smallest digit will always be zero. Since 0 + 0 = 0, the carry is zero and never less than zero.

**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.**

Maximum two’s complement:

The number will start with ‘0’ followed by ‘1’s. For an 8-bit two’s, , the maximum number would be 0111 1111. By collapsing the formal definition, we can simplify to find the largest positive number two’s complement with *N* bits being represented by .

Minimum two’s complement:

The number will start with ‘1’ followed by ‘0’s. For an 8-bit number, , the minimum two’s complement would be 1000 0000. By collapsing the formal definition, we can simplify to find the smallest negative number two’s complement with *N* bits being represented by .

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

**For a number B with magnitude less than , show that if B is represented by a 2’s complement number with N bits then**

**5. 4.5 Sign extension**

**Prove that “sign-extension” is value preserving.**

I will show that sign extension is value preserving for both positive and negative numbers by showing some examples.

Positive Numbers

We start with the positive number 3 in the decimal system. When converted to binary, the resulting two’s complement number is 0011 when using 4 bits. Since this is a positive binary number, the leading number is always zero to denote positivity. We can recognize the binary number 0011 as the following equation:

We can show that sign extension is value preserving for both adding and removing leading zeros. We can remove a zero and make 0011 a 3-bit binary number; 011. We can do this only because the number proceeding the leading zero in 0011 is also zero. If the number we were working with was 0111, we couldn’t remove a zero because the number proceeding the leading zero is a 1.

We can show 011 is equivalent to 0011 and thus 3 by the following equation:

We can show 0 0011 is equivalent to 0011 and thus 3 by the following equation:

Thus, I have shown that sign extension is value preserving for positive numbers.

Negative Numbers

We start with the negative number 3 in the decimal system. When converted to binary, the resulting two’s complement number is 1101 when using 4 bits. Since this is a negative binary number, the leading number is always 1 to denote negativity. We can recognize the binary number 1101 as the following equation:

We can show that sign extension is value preserving for both adding and removing leading 1’s. We can remove a 1 and make 1101 a 3-bit binary number; 101. We can do this only because the number proceeding the leading 1 in 1101 is also 1. If the number we were working with was 1011, we couldn’t remove a zero because the number proceeding the leading 1 is a zero.

We can show 101 is equivalent to 1101 and thus -3 by the following equation:

We can show 1 1101 is equivalent to 1101 and thus -3 by the following equation:

Thus, I have shown that sign extension is value preserving for negative numbers.

**6. Using the technique presented in section 4.1, convert the following decimal numbers to binary (show your work):**

**a. 121**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Repeated Division (mod 2) | | | | | | | |
| Quotients | 121 | 60 | 30 | 15 | 7 | 3 | 1 |
| Remainders | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

**b. 1537**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Repeated Division (mod 2) | | | | | | | | | | | |
| Quotients | 1537 | 768 | 384 | 192 | 96 | 48 | 24 | 12 | 6 | 3 | 1 |
| Remainders | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |

**c. 31333**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Repeated Division (mod 2) | | | | | | | | | | | | | | | |
| Quotients | 31333 | 15666 | 7833 | 3916 | 1958 | 979 | 489 | 244 | 122 | 61 | 30 | 15 | 7 | 3 | 1 |
| Remainders | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |

**d. 97**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Repeated Division (mod 2) | | | | | | | |
| Quotients | 97 | 48 | 24 | 12 | 6 | 3 | 1 |
| Remainders | 1 | 0 | 0 | 0 | 0 | 1 | 1 |

**7. Perform the following subtraction operations using complements as described in section 4.2 (show your work):**

**a. 121 - 41**

Complement of 41

The complement of 41 is as follows: .

Complement Addition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 0 | 0 | 1 |  |
|  |  | 1 | 2 | 1 |
|  |  |  | 5 | 8 |
| + |  |  |  | 1 |
|  |  | 1 | 8 | 0 |

By discarding the final carry out, we get our final result of 80.

**b. 1022 – 35**

Complement of 35

The complement of 35 is as follows: .

Complement Addition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 |  |
|  | 1 | 0 | 2 | 2 |
|  |  |  | 6 | 4 |
| + |  |  |  | 1 |
|  | 1 | 0 | 8 | 7 |

By discarding the final carry out, we get our final result of 987. Since the number right after the first number is zero, we discard the leading 1 and change the zero to a 9.

**c. 151 – 90**

Complement of 90

The complement of 90 is as follows: .

Complement Addition

|  |  |  |  |
| --- | --- | --- | --- |
|  | 0 | 1 |  |
|  | 1 | 5 | 1 |
|  |  | 0 | 9 |
| + |  |  | 1 |
|  | 1 | 6 | 1 |

By discarding the final carry out, we get our final result of 61.

**d. 2120 – 101**

Complement of 101

The complement of 101 is as follows: .

Complement Addition

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 1 | 0 |  |
|  | 2 | 1 | 2 | 0 |
|  |  | 8 | 9 | 8 |
| + |  |  |  | 1 |
|  | 3 | 0 | 1 | 9 |

By discarding the final carry out, we get our final result of 2019.

**8. Convert the following decimal numbers to 8-bit two's complement (show your work):**

**a. -121**

Convert 121 to binary using textbook technique

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Repeated Division (mod 2) | | | | | | | |
| Quotients | 121 | 60 | 30 | 15 | 7 | 3 | 1 |
| Remainders | 1 | 0 | 0 | 1 | 1 | 1 | 1 |

Convert 121 to hex then to binary

Negate the number (by inverting the bits) and adding 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
|  | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 |
| + |  |  |  |  |  |  |  | 1 |
|  | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

So, as an 8-bit two’s complement

**b. -51**

Convert 51 to binary using textbook technique

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Repeated Division (mod 2) | | | | | | |
| Quotients | 51 | 25 | 12 | 6 | 3 | 1 |
| Remainders | 1 | 1 | 0 | 0 | 1 | 1 |

Convert 51 to hex then to binary

Negate the number (by inverting the bits) and adding 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |  |
|  | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| + |  |  |  |  |  |  |  | 1 |
|  | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 1 |

So, as an 8-bit two’s complement if we discard the final carry out.

**c. -104**

Convert 104 to binary using textbook technique

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Repeated Division (mod 2) | | | | | | | |
| Quotients | 104 | 52 | 26 | 13 | 6 | 3 | 1 |
| Remainders | 0 | 0 | 0 | 1 | 0 | 1 | 1 |

Convert 104 to hex then to binary

Negate the number (by inverting the bits) and adding 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |  |
|  | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| + |  |  |  |  |  |  |  | 1 |
|  | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

So, as an 8-bit two’s complement.

**d. 115**

Convert 115 to binary using textbook technique

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Repeated Division (mod 2) | | | | | | | |
| Quotients | 115 | 57 | 28 | 14 | 7 | 3 | 1 |
| Remainders | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

Convert 115 to hex then to binary

**e. 127**

Convert 127 to binary using textbook technique

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Repeated Division (mod 2) | | | | | | | |
| Quotients | 127 | 63 | 31 | 15 | 7 | 3 | 1 |
| Remainders | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

Convert 127 to hex then to binary