**CSCI240 – Computer Organization and Assembly Language Programming**

**Name:**  Jonathan Limpus

**Student ID:** 1429394

**Assignment:** Homework 2

1. 2.1) Given *n* bits, there are 2 to the *n*th power combinations of bits
2. 2.2) You would need 5 bits to represent every letter, as that would allow 32 possible combinations. In order to represent both lower and upper case letters, you would need 6 bits (64 combinations), as you would need 52 different combinations.
3. 2.4) You could represent any unsigned integer with *n* bits, with a range of [0,∞)
4. 2.6) The 6 bit2’s complement representation of – 32 is 10 0000.

* 10 0000 Unsigned representation of 32
* 01 1111 Flip bits
* 01 1111 + 1 Add one
* 10 0000 Answer

1. 2.7)

|  |  |
| --- | --- |
| Decimal | 2’s Complement |
| 7 | 0111 |
| 6 | 0110 |
| 5 | 0101 |
| 4 | 0100 |
| 3 | 0011 |
| 2 | 0010 |
| 1 | 0001 |
| 0 | 0000 |
| -1 | 1111 |
| -2 | 1110 |
| -3 | 1101 |
| -4 | 1100 |
| -5 | 1011 |
| -6 | 1010 |
| -7 | 1001 |
| -8 | 1000 |

1. 2.8)
2. The largest positive number you could represent in 2’s complement code with 8 bits is 0111 1111, which is 127.
3. The greatest magnitude negative number you could represent in 8 bits with 2’s complement is

1111 1111, which is -127

1. 2.10)
2. 1010 = -6
3. 01011010 = 166

01011001 Subtract 1

10100110 Flip bits

2^1 + 2^2 + 2^5 + 2^7 = 166

1. 11111110 = 2

11111101 Subtract 1

00000010 Flip bits

2^1 = 2

d. 0011100111010011 = 50733

0011100111010010 Subtract 1

1100011000101101 Flip bits

2^15 + 2^14 + 2^10 + 2^9 + 2^5 +2^3 +2^2 +2^0 = 50733

1. 2.11)
2. 2.13)
3. 1111 1010
4. 0001 1001
5. 1111 1000
6. 0000 0001
7. 2.14)
8. 1011 + 0001 = 1100
9. 0000 + 1010 = 1010
10. 1100 + 0011 = 1111
11. 0101 + 0110 = 1011
12. 1111 + 0001 = 10000
13. 2.15) It is equivalent to dividing by 2.
14. 2.17)
15. 01 + 1011 = 1100 => -4
16. 11 + 01010101 = 01011000 => 168
17. 0101 + 110 = 1011 => -5
18. 01 + 10 = 11 => -2
19. 2.18)
20. 01 + 1011 = 1100 => 12
21. 11 + 01010101 = 01011000 => 88
22. 0101 + 110 = 1011 => 11
23. 01 + 10 = 11 => 3
24. 2.20)
25. 2.25)
26. 2.30)
    1. 01010111 AND 11010111 = 01010111
    2. 101 AND 110 = 100
    3. 11100000 AND 10110100 = 10100000
    4. 00011111 AND 10110100 = 00010100
    5. (0011 AND 0110) AND 1101 = 0000
    6. 0011 AND (0110 AND 1101) = 0000
27. 2.33)
    1. 01010111 OR 11010111 = 11010111
    2. 101 OR 110 = 111
    3. 11100000 OR 10110100 = 11110100
    4. 00011111 OR 10110100 = 10111111
    5. (0101 OR 1100) OR 1101 = 1101
    6. 0101 OR (1100 OR 1101) = 1101
28. 2.34)
    1. NOT(1011) OR NOT (1100) = 0111
    2. NOT(1000 AND(1100 OR 0101)) = 0111
    3. NOT(NOT(1101)) = 1101
    4. (0110 OR 0000) AND 1111 = 0110
29. 2.43)
    1. x48656c6c6f21 -> Hello!
    2. x68454c4c4f21 -> hELLO!
    3. x436f6d70757465727321 -> Computers!
    4. x4c432d32 -> LC-2
30. 2.44)
31. 2.45)
    1. 1101 0001 1010 1111