**Chapter 2**

**2.13 Without changing their values, convert the following 2's complement**

**binary numbers into 8-bit 2's complement numbers.**

**a. 1010** 11111010

**b. 011001**  00011001

**c. 1111111000** 11111000

**d. 01** 00000001

**2.14 Add the following bit patterns. Leave your results in binary form.**

**a. 1011 + 0001** 1100

**b. 0000 + 1010** 1010

**c. 1100 + 0011** 1111

**d. 0101 + 0110** 1011

**e. 1111 + 0001** 2’s Complement:0000 Unsigned: 10000

**Q:请问在题目没有说明类型时时，需要扩充还是溢出舍去？**

**2.17 Add the following 2's complement binary numbers. Also express the**

**answer in decimal.**

**a. 01 + 1011** 1100 -4

**b. 11 + 01010101**  01010100 84

**c. 0101 + 110** 0011 3

**d. 01 + 10** 11 -1

**2.22 Create two 16-bit 2's complement integers such that their sum causes an**

**overflow.**

0111111111111111 + 011111111111110

**2.24 Create two 16-bit unsigned integers such that their sum causes an**

**overflow.**

1000000000000000 + 1000000000000001

**2.27 The LC-3, a 16-bit machine adds the two 2's complement numbers 0101010101010101 and 0011100111001111, producing 1000111100100100. Is there a problem here? If yes, what is the problem? If no, why not?**

Overflow has occurred when add 2 positive numbers and have the result as a negative number.

**2.37 If *n* and *m* are both 4-bit 2's complement numbers, and *s* is the 4-bit result of adding them together, how can we determine, using only the logical operations described in Section 2.6, if an overflow occurred during the addition? Develop a "procedure" for doing so. The inputs to the procedure are n, m, and s, and the output will be a bit pattern of all zeros (0000) if no overflow occurred and 1000 if an overflow did occur.**

|  |  |  |
| --- | --- | --- |
| **m** | **n** | **s** |
| **0** | **0** | **1** |
| **1** | **1** | **0** |

[((NOT n) AND (NOT m) AND s) OR (n AND m AND (NOT s))] AND 1000

**2.56 Define a new 8-bit floating point format with 1 sign bit, 4 bits of exponent, using an excess-7 code (that is, the bias is 7), and 3 bits of fraction. If xE5 is the bit pattern for a number in this 8-bit floating point format, what value does it have? (Express as a decimal number.)**

1 1100 101

-

1100-7 = 5

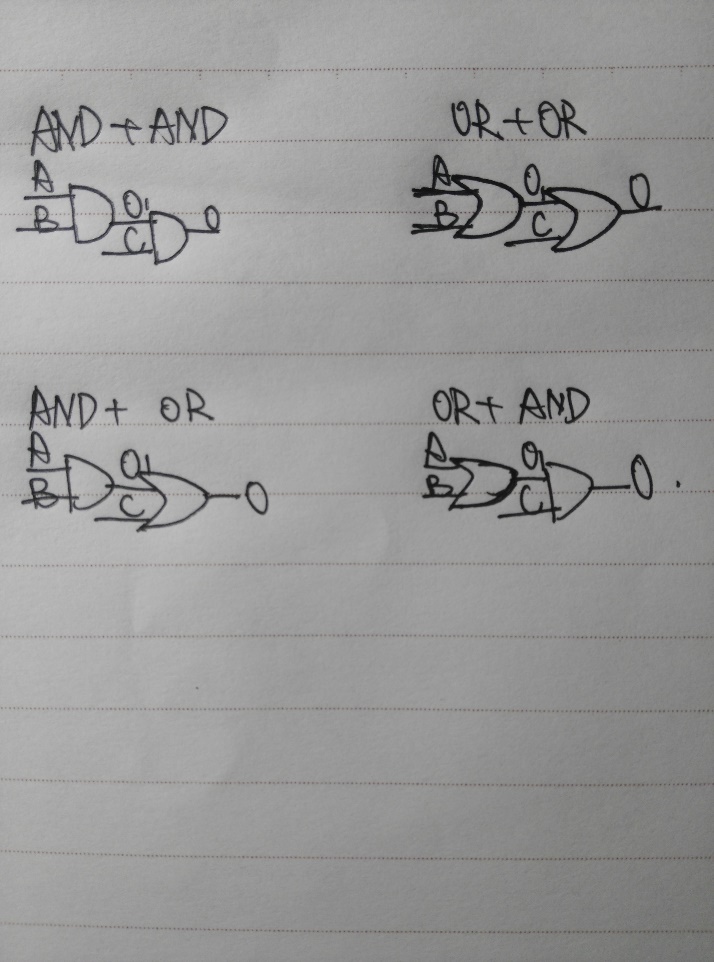
1. 101

-1.101\*2^5 = -52

**-**52

**Chapter 3**

**3.3 A two-input AND and a two-input OR are both examples of two-input logic functions. How many different two-input logic functions are possible?**

****

There are 16 different two-input logic functions.

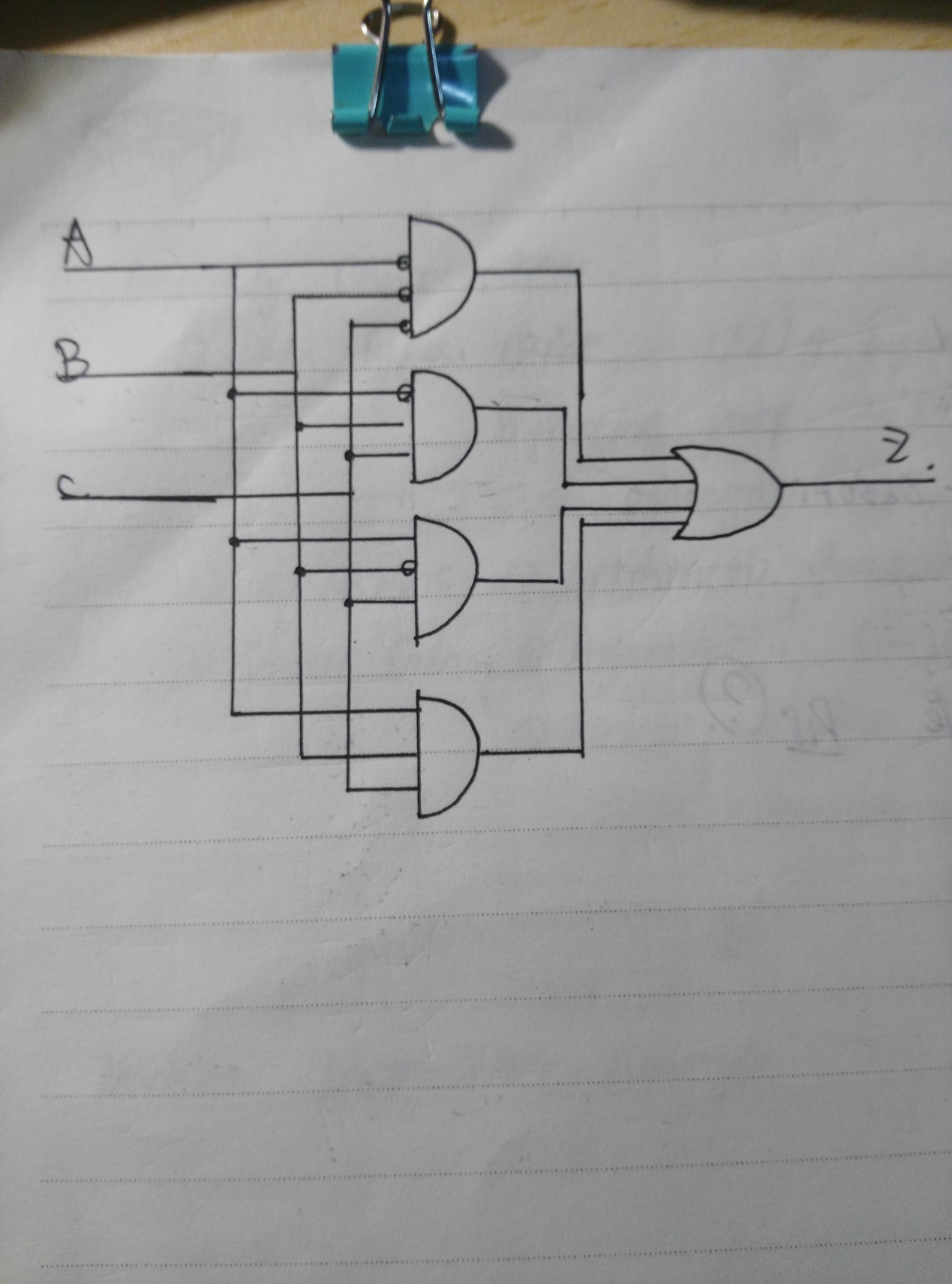
|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | O |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

**3.5 Complete a truth table for the transistor-level circuit in Figure 3.34.**

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | O |
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

**3.16 Given the following truth table, generate the gate-level logic circuit, using the implementation algorithm referred to in Section 3.3.4.**

**需要把所有的与门都画出来，然后输出是1的与门与或门相连e.g.P42图**

****

**3.23 Given the logic circuit in Figure 3.38, fill in the truth table for the output value Z.**

|  |  |  |  |
| --- | --- | --- | --- |
| A | B | C | O |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

**3.27 For this question, refer to the figure below.**

***a.* Describe the output of this logic circuit when the select line *S* is a logical 0. That is, what is the output Z for each value of A?**

When S=0, Z=A

***b.* If the select line *S* is switched from a logical 0 to 1, what will the output be?**

When S=1, Z doesn’t change its value.

***c.* Is this logic circuit a storage element?**

Yes, the circuit is a storage element since even though the value of S changes, the value of Z retains.

**3.29 A 16-bit register contains a value. The value x75A2 is written into it. Can the original value be recovered?**

No. The original value can’t be recovered once the new value is written into it.