

1. Convert the number 8 into a 5-bit binary number. Use unsigned, signed-magnitude, 1's complement, and 2's complement.
2. Convert the number -8 into a 5-bit binary number, using signed-magnitude, 1's complement, and 2's complement.
3. Convert the number -11 into signed-magnitude, 1's complement, 2's complement.
4. For the 5-bit 2's complement binary number; what is 11011 in decimal?
5. For the 5-bit 2's complement binary number; what is 01011 in decimal?
6. Add the 2's complement 5-bit numbers 00101 and 11011 together; show your work. What is the result in decimal? Is it correct? If not, why?
7. Add the 2's complement 5-bit numbers 01111 and 00001 together; show your work. What is the result in decimal? Is it correct? If not, why?
8. Add the 2's complement 5-bit numbers 11111 and 00011 together; show your work. What is the result in decimal? Is it correct? If not, why?
9. Suppose you have two unsigned binary numbers 110 and 101. Multiply them together (in binary), and show the result (in binary).
10. Sign extend the 5-bit binary number 00101 to 8 bits.
11. Sign extend the 5-bit binary number 11001 to 8 bits.
12. Hexadecimal puts bits into groups of 4, and then uses the digits 0-9 and A-F to represent the bits. So, for an 8 bit number, you would only need two hex digits. In binary, what is A3 hex?
13. What would 0110 1101 be in hex?
14. Octal groups bits into 3s, and uses the digits 0-7. What would octal 36 be in binary?
15. What would 111 010 be in octal?
16. Write the number 6 and $3/4$ in binary, using simple notation (xxx.xxx).
17. Convert the number above into scientific notation (shift the decimal place, so that your answer is $x.xxxx * 2^{\text{exponent}}$).
18. The IEEE format for floating point numbers is [Sign Bit][8 bits for the exponent][23 bits for the fraction]. The number stored in the 8 bit section is actually your exponent plus 127. Write (in binary) the IEEE format for the number 6 and $3/4$.
19. What does the IEEE floating point number 1 10000001 1010100... represent? You might look at example 2.12 from the book.
20. What does the IEEE floating point number 0 10000011 000000... represent?
21. What does the IEEE floating point number 0 01111110 000000.... represent?
22. Write a truth table for the logical AND function (see section 2.6.1).
23. Write a truth table for the logical OR function.
24. Write a truth table for the logical NOT function.
25. The book gives a number of examples where the AND, OR, and NOT are applied to bit vectors. What is 0101 AND 1100?
26. What is 0101 OR 1100?
27. What is NOT 0101?
28. The XOR or exclusive OR is another handy one. Write the truth table for exclusive or.
29. Suppose A is 1, B is 0, C is 0. What is (A OR B) AND (A AND NOT C)?
30. Draw a P-type transistor (like figure 3.3. Note the bubble on the gate. That's important!).
31. Draw an N-type transistor (no bubble on this one).

32. Draw a CMOS inverter (figure 3.4).
33. If the input to the inverter is 1, circle the transistors that will be conducting (allowing electricity to pass).
34. If the input to the inverter is 0, put a check mark by the transistors that are conducting.
35. Write a truth table for the inverter (it should look just like the NOT truth table).
36. Draw a NOR gate (figure 3.5).
37. If $A=0$, $B=1$, draw a circle around the transistors that are conducting in your NOR gate.
38. If $A=0$, $B=0$, put Xs by the conducting transistors.
39. Draw a NAND gate (figure 3.7).
40. If $A=0$, $B=1$, draw a circle around the transistors that are conducting in your NAND gate.
41. If $A=0$, $B=0$, put Xs by the conducting transistors.
42. Give figure 3.8, put the names below the diagrams.
43. Draw a 3-input NAND gate.
44. Draw a 3-input NOR gate.
45. Write DeMorgan's Law (see section 3.2.4). Do it correctly. It's in the book. Don't make up your own. Use the one in the book. Yes, memorize the one in the book.