Computer Science

1. (3 pts.) Write an UNIX command to make the file "hello" in the directory "opencv" executable by owner?

2 chmod opencv/hello chmod u+x opencv/hello

94

2. (3 pts.) Write an UNIX command to create a directory "opencv" in your home directory.

2 mkdir opencv mkdir ~/opencv

3. (3 pts.) Write an UNIX command to remove the directory "work".

3 rm -rf work

95

4. (3 pts.) Write an UNIX command to list contents of the current directory.

3 ls -al.

5. (10 pts.) Given a 15-bit Hamming codeword 101100011011001 with odd parity, extract the data bits, determine bit error if any, and if so, correct it.

101100011011001

10 8 9 10 11 12 13 14 15 0 ✓

1 1 0 1 1 0 0 1

1 3 5 7 9 11 13 15 0 ✓

1 1 0 0 1 1 0 1

2 3 6 7 10 11 14 15 0 ✓

0 1 0 0 0 1 0 1

4 5 6 7 12 13 14 15 0 ✓

1 0 0 0 1 0 0 1

all correct #

6. (10 pts.) Given an 4-bit data word 0111, compute its 7-bit Hamming codeword with odd parity.

10

1 3 5 7

1 0 1 1

2 3 6 7

1 0 1 1

4 5 6 7

0 1 1 1

1100111 #

7. (4 pts.) Convert the following hexadecimal numbers to decimal. Show your calculation. a. $(C1F)_{16}$ b. $(19F)_{16}$ c. $(0.01)_{16}$ d. $(16.16)_{16}$

4

a. $(C1F)_{16} = 12\times16^2+1\times16+15 = (3103)_{10}$

b. $(19F)_{16} = 1\times16^2+9\times16+15 = (415)_{10}$

c. $(0.01)_{16} = 1\times16^{-2} = (\frac{1}{256})_{10}$

d. $(16.16)_{16} = 1\times16+6+\frac{1}{16}+\frac{6}{256} = (22\frac{11}{128})_{10}$ #

8. (4 pts.) Convert the following hexadecimal numbers to decimal. a. $(B2D)_{16}$ b. $(16)_{16}$ c. $(E1E)_{16}$ d. $(3.14)_{16}$

4

a. $(B2D)_{16} = 11\times16^2+2\times16+13 = (2861)_{10}$

b. $(16)_{16} = 16\times1+6 = (22)_{10}$

c. $(E1E)_{16} = 14\times16^2+16+14 = (3614)_{10}$

d. $(3.14)_{16} = 3+\frac{1}{16}+\frac{4}{256} = (3\frac{5}{64})_{10}$ #

9. (4 pts.) Convert the following decimal numbers to binary. a. 1234 b. 612.25 c. 1.3125 d. 65535

3

a. $(10011010010)_2$

b. $(1001100100.01)_2$

c. $(1.0101)_2$

d. $(1111111111111111)_2$

10. (4 pts.) Convert the following binary numbers to hexadecimal. a. $(1111111111111011)_2$ b. $(1010101)_2$ c. $(01010.01)_2$ d. $(1.101)_2$

4

a. $(FFFB)_{16}$

b. $(55)_{16}$

c. $(A.4)_{16}$

d. $(1.A)_{16}$ #

c. A+0.4

d. 1+0.A

$\frac{1}{4} = \frac{4}{16}$

$\frac{1}{2}+\frac{1}{8} = \frac{8+2}{16} = \frac{10}{16}$

11. (4 pts.) Change the following 8-bit two's complement numbers to decimal. a. 01010101 b. 10110101 c. 01111111 d. 10000000

4

a. $(1010101)_2 = 2^6+2^4+2^2+1 = 64+16+4+1 = 85$

b. 10110101 $(1001011)_2 = 2^6+2^3+2^1+1 = 64+8+3 = 75$

c. $(1111111)_2 = 2^6+2^5+2^4+2^3+2^2+2^1+1 = 127$

a. 85

b. -75

c. 127

d. -128 #

12. (4 pts.) Change the following decimal numbers to 8-bit two's complement integers. a. -128 b. -1 c. 99 d. 142

4

a. 10000000

b. 11111111

c. 01100011

d. overflow #

13. (10 pts.) Show the following decimal numbers in 32-bit IEEE format. (Hint: 32-bit IEEE format

exponent size is 8 bits

mantissa size is 23 bits

bias is 127

a. 1000000010000000000000000000000

b. 111100101111111111111111111111111

#

5

a. $-1 \times 2^{-126}$

b. $2 - 2^{-23} \times 2^{127}$

a. $-126+127=1$

b. $2^1 - 2^{104}$

1

$= -2^{104} + 2^1$

$= -(2^{104} - 2) = -(2^{104} \times 1.0 - 2^{104} \times 0.\underbrace{\dots 1}_{104})$

round-off error (we have 23 bits M)

14. (2 pts.) We need to unset (force to 0) the five leftmost bits of an 8-bit pattern. Show the mask and the operation.

2

XXXXXXXX
mask and 00000111
00000XXX

15. (2 pts.) We need to flip the two rightmost and the four leftmost bits of a pattern. Show the mask and the operation.

2

XXXX · · · · · XX
mask xor 111100 · · · 0011
$\bar{X}\bar{X}\bar{X}\bar{X}$XX · · · XX$\bar{X}\bar{X}$

16. (2 pts.) Which of the following operation creates an overflow if numbers and the result are represented in 8-bit two's complement representation? a. $11000011_2 + 01111111_2$ b. $01000011_2 + 00111111_2$

2

⊖ ⊕ overflow

A: a.

1000011
0111111
10000010

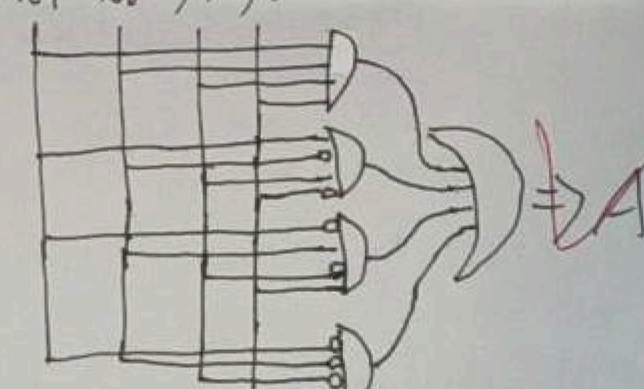
17. (15 pts.) The notation $x_1x_0$ represents a two-bit binary number that can have any value (00, 01, 10, or 11); for example, when $x_1 = 1$ and $x_0 = 0$, the binary number is 10, and so on. Similarly, $y_1y_0$ represents another two-bit binary number. Design a logic circuit, using $x_1, x_0, y_1$, and $y_0$ inputs, whose output will be HIGH only when the two binary numbers $x_1x_0$ and $y_1y_0$ are equal. Construct the truth table, the sum-of-products expression for the output, and the circuit

15

| $x_1$ | $x_0$ | $y_1$ | $y_0$ | A |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | [illegible] | 0 | 1 | 0 |

$x_1 \cdot x_0 \cdot y_1 \cdot y_0 + x_1 \cdot \overline{x_0} \cdot y_1 \cdot \overline{y_0} + \overline{x_1} \cdot x_0 \cdot \overline{y_1} \cdot y_0 + \overline{x_1} \cdot \overline{x_0} \cdot \overline{y_1} \cdot \overline{y_0} = A$

$x_1$ $x_0$ $y_1$ $y_0$




18. (15 pts.) The full-adder has three inputs: an A bit, a B bit, and $C_i$ bit. The result of the addition of these three bits produces two bits: a *sum* bit S, and a *carry* bit $C_o$. Design a logic circuit that will perform the full-adder. Construct the truth table, the sum-of-products expression for the S and $C_o$, and the circuit.

15

| A | B | $C_i$ | S | $C_o$ |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

$A \cdot \overline{B} \cdot \overline{C_i} + \overline{A} \cdot B \cdot \overline{C_i} + \overline{A} \cdot \overline{B} \cdot C_i + A \cdot B \cdot C_i = S$

$A \cdot B \cdot \overline{C_i} + \overline{A} \cdot B \cdot C_i + A \cdot \overline{B} \cdot C_i + A \cdot B \cdot C_i = C_o$

A B $C_i$