Points missed: _____ Student's Name: ___________________________________

Total score: _____ /100 points

East Tennessee State University – Department of Computer and Information Sciences
CSCI 2150 (Tarnoff) – Computer Organization
TEST 1 for Spring Semester, 2006

Read this before starting!

- The total possible score for this test is 100 points.
- This test is ***closed book and closed notes***
- You may ***NOT*** use a calculator. Leave all numeric answers in the form of a formula.
- You may use one sheet of scrap paper that you must turn in with your test.
- All answers must have a box drawn around them. This is to aid the grader (who might not be me!) Failure to do so might result in no credit for answer. Example:

$$\boxed{32F1_{16}}$$

- **1 point will be deducted** per answer for missing or incorrect units when required. **No** assumptions will be made for hexadecimal versus decimal, so you should always include the base in your answer.
- If you perform work on the back of a page in this test, indicate that you have done so in case the need arises for partial credit to be determined.
- Statement regarding academic misconduct from Section 5.7 of the East Tennessee State University Faculty Handbook, June 1, 2001:

  "Academic misconduct will be subject to disciplinary action. Any act of dishonesty in academic work constitutes academic misconduct. This includes plagiarism, the changing of falsifying of any academic documents or materials, cheating, and the giving or receiving of unauthorized aid in tests, examinations, or other assigned school work. Penalties for academic misconduct will vary with the seriousness of the offense and may include, but are not limited to: a grade of 'F' on the work in question, a grade of 'F' of the course, reprimand, probation, suspension, and expulsion. For a second academic offense the penalty is permanent expulsion."

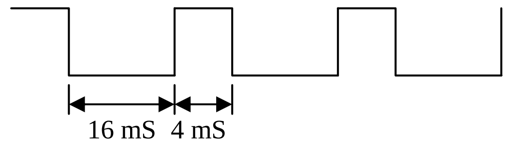
Basic Rules of Boolean Algebra:

1. $A + 0 = A$	**7.** $A \cdot A = A$
2. $A + 1 = 1$	**8.** $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	**9.** $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	**10.** $A + AB = A$
5. $A + A = A$	**11.** $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	**12.** $(A + B)(A + C) = A + BC$

DeMorgan's Theorem: $\overline{(AB)} = (\overline{A} + \overline{B})$ $\qquad$ $\overline{(A + B)} = (\overline{A} \bullet \overline{B})$

Short-ish Answer (2 points each unless otherwise noted)

1. What is the frequency of the periodic signal in the figure shown to the right? ***Be sure to include your units!*** (Note: mS means milliseconds = 10^{-3} seconds)



$$\text{frequency} = \frac{1}{\text{period}} = \frac{1}{16 \text{ mS} + 4 \text{ mS}} = \frac{1}{20 \text{ mS}} = 50 \text{ Hz}$$

You could have left your answer as $1/(20\times10^{-3})$ Hertz if you wanted to. I just needed to see the units of Hertz to be sure that you knew what frequency was.

2. What is the duty cycle of the periodic signal from problem 1?

$$\text{Duty cycle} = \frac{t_h}{T} * 100\% = \frac{4 \text{ mS}}{20 \text{ mS}} * 100\% = 20\%$$

3. True or False: The frequency of a periodic signal can be calculated from its duty cycle alone.

The duty cycle is the ratio of the duration of a logic one to the duration of the period. This ratio removes the units of time from both the numerator and denominator leaving a unit-less ratio and therefore, the frequency cannot be calculated. If the frequency changes, then the period changes. If the period changes, then the duration of the logic one must change.

4. What is the most negative value that can be stored using a 10-bit 2's complement representation?

The most negative 10-bit 2's complement value is 1000000000_2. To figure out its value, convert it to its positive counterpart by doing the bit-flipping thing. This gives us 1000000000_2. Using the standard unsigned binary method for converting to decimal gives us $2^9 = 512$. This means that 1000000000 in 2's complement form represents $-2^9 = -512$.

a.) $-(2^{10} - 1)$ b.) $-(2^9 - 1)$ c.) $-(2^8 - 1)$ d.) $-(2^{10})$ e.) $-(2^9)$ f.) $-(2^8)$

5. What is the most positive value that can be stored using a 9-bit unsigned binary representation?

The most positive 9-bit unsigned binary value is 111111111_2. This is one less than the 10-bit unsigned binary value 1000000000_2 which equals 2^9. Therefore, 111111111_2 equals $2^9 - 1$.

a.) $2^{9-1} - 1$ b.) $2^9 - 1$ c.) 2^9 d.) 2^{9-1} e.) $2^{9+1} - 1$ f.) 2^{9+1}

6. What is the minimum number of bits needed to represent 800_{10} using unsigned binary representation?

Since the basic formula for the maximum value of an unsigned binary value is $2^n - 1$ for n bits, then all we should need to do is substitute a bunch of values for n to figure out the minimum number of bits to represent 800_{10}. Starting at n=8, we get:

n	$2^n - 1$
8	255
9	511
10	1023

Therefore, 10 bits is the minimum needed.

a.) 8 b.) 9 c.) 10 d.) 11 e.) 12 f.) 13

7. What is the minimum sampling rate needed in order to successfully capture frequencies up to 20,000 Hz in an analog signal?

The minimum sampling rate is twice the highest frequency that is supposed to be captured.

a.) 10,000 Hz b.) 20,000 Hz c.) 25,000 Hz d.) 30,000 Hz e.) 35,000 Hz f.) 40,000 Hz

8. For each of the following applications, what would be the optimum (best) binary representation, unsigned binary (UB), 2's complement (TC), IEEE 754 Floating Point (FP), or binary coded decimal (BCD)? Identify your answer in the blank to the left of each application. (2 points each)

UB or BCD Student ID

_UB or FP___ Exam grade (Integers from 0 to 100 possibly added for average)

____FP_____ Weight of an atom in grams

9. True or False: The 32-bit binary floating-point number 11011011010010011101101101011001 is negative.

MSB is the sign bit, and since it equals 1, it is a negative number.

10. Write the complete truth table for a 2-input NOR gate (3 points).

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

Remember that a NOR is an OR gate with an inverted output.

11. Divide 00110100_2 by 4. Leave your answer in binary. (Hint: There is a shortcut.) (3 points)

Remember that a multiplication by 2 can be accomplished with a single left shift while a division by two can be accomplished with a single right shift. We're dividing, so each shift right is equivalent to a single division by 2. Therefore, since $4 = 2^2$, you would need to **shift right 2 times**. We can test this by converting 00110100_2 to decimal and then shifting 00110100 right by 2 bit positions and converting it too to decimal. If the second value equals one fourth the first value, then it must have worked.

$00110100_2 = 32 + 16 + 4 = 52_{10}$

$00001101_2 = 8 + 4 + 1 = 13_{10}$

$52 \div 4 = 13_{10}$

It worked!

12. Circle the function that would first be performed in the following expression.

$$A \cdot B \cdot (C \cdot D + E)$$

__Medium-ish Answer (4 points each unless otherwise noted)__

13. Convert the floating-point number 10111111101100101100000000000000 to its binary exponential format, e.g., 1.1010110 x 2^{-12}, (which, by the way, is not even close to the right answer).

Sign bit	Exponent	Fraction
1	01111111 = 127_{10}	01100101100000000000000

$-1.01100101100000000000000 \times 2^{127-127} = -1.011001011 \times 2^0 = -1.011001011$

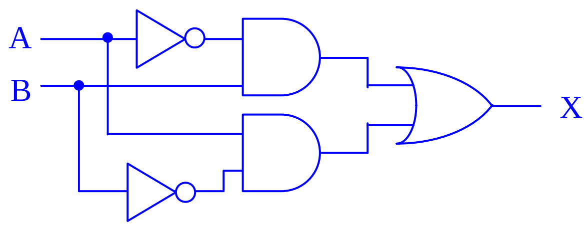
14. Convert 10100.1100 to decimal. (You may leave your answer in expanded form if you wish.)

Point

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
1	0	1	0	0	1	1	0

$2^4 + 2^2 + 2^{-1} + 2^{-2} = 16 + 4 + 0.5 + 0.25 =$ **20.75**

15. Draw the circuit ***exactly*** as it is represented by the Boolean expression $(\overline{A} \cdot B) + (A \cdot \overline{B})$.



16. Convert $010010001111001010101011_2$ to hexadecimal.

First, let's create the conversion table between binary and hex. That table is shown to the right.

Once the table has been created, divide the number to be converted into nibbles. The result is shown below:

0001 0010 0011 1100 1010 1011

Notice that two leading zeros needed to be added. Each of these nibbles corresponds to a pattern from the table to the right. Now it just becomes a 1 to 1 conversion.

$123CAB_{16}$

Binary	Hexadecimal
0 0 0 0	0
0 0 0 1	1
0 0 1 0	2
0 0 1 1	3
0 1 0 0	4
0 1 0 1	5
0 1 1 0	6
0 1 1 1	7
1 0 0 0	8
1 0 0 1	9
1 0 1 0	A
1 0 1 1	B
1 1 0 0	C
1 1 0 1	D
1 1 1 0	E
1 1 1 1	F

17. Use any method you wish to prove rule 10: $A + \overline{A} \cdot B = A + B$. Show all steps.

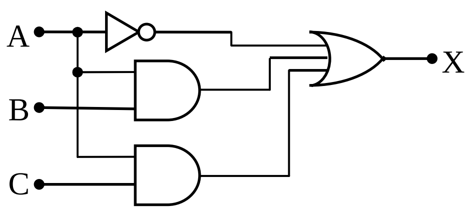
The easiest way is to prove it using a truth table:

A	B	$\overline{A}$	$\overline{A}$·B	A + $\overline{A}$·B	A + B
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

18. List two benefits of a digital circuit that uses fewer gates.

Lower chip count results in
- cheaper circuit
- faster performance
- lower power consumption
- higher reliability (fewer things to break)
- smaller circuit board

19. In the space to the right, create the truth table for the circuit shown below. (5 points)



A	B	C	^A	A·B	A·C	X = ^A + A·B + A·C
0	0	0	1	0	0	1
0	0	1	1	0	0	1
0	1	0	1	0	0	1
0	1	1	1	0	0	1
1	0	0	0	0	0	0
1	0	1	0	0	1	1
1	1	0	0	1	0	1
1	1	1	0	1	1	1

20. Write the Boolean expression for the circuit shown in the previous problem. ***Do not simplify!***

$$\boxed{X = \overline{A} + A \cdot B + A \cdot C}$$

21. If an 8-bit binary number is used to represent an analog value in the range from -25 to 250, how large an analog value does a single binary increment represent? In other words, if the binary number is incremented by one, how much change in the analog range is represented? (Leave your answer in the form of a fraction.)

A single increment = range/number of increments

= $(250 – (–25))/(2^8 – 1)$

= 275/255

= 1.078 units/increment

You could have left your answer in any of the last three forms.

22. Use DeMorgan's Theorem to distribute the inverse of the expression $\overline{(A\bullet B) + C + D}$ to the individual input terms. ***Do not simplify!***

$\overline{(A\cdot B) + C + D}$	First, distribute inverse across OR. This can work across all three inputs, (A·B), C, and D.
$\overline{(A\cdot B)}\cdot\overline{C}\cdot\overline{D}$	Next, distribute inverse across the A·B product.
$(\overline{A} + \overline{B})\cdot\overline{C}\cdot\overline{D}$	This gives us the final answer. Note the importance of the parenthesis. Without them, the order of precedence would AND the inverses of B, C, and D first which would be wrong.

Longer Answers (Points vary per problem)

23. Mark each boolean expression as ***true*** or ***false*** depending on whether the right and left sides of the equal sign are equivalent. Show all of your work to receive partial credit for incorrect answers. (3 points each)

a.) $\overline{(A\cdot B)}\cdot A = A\cdot\overline{B}$ Answer: **True**

$\overline{(A\cdot B)}\cdot A$	Apply DeMorgans
$(\overline{A} + \overline{B})\cdot A$	Apply Associative Law
$A\cdot(\overline{A} + \overline{B})$	Apply Distributive Law
$A\cdot\overline{A} + A\cdot\overline{B}$	Anything AND'd w/inverse = 0
$0 + A\cdot\overline{B}$	Anything OR'd w/0 = itself
$A\cdot\overline{B}$	Answer is TRUE

b.) $\overline{A}\cdot(A + B) + (A\cdot C + A\cdot\overline{C}) = \overline{A} + B$ Answer: **False**

$\overline{A}\cdot(A + B) + (A\cdot C + A\cdot\overline{C})$	Pull A from last two products
$\overline{A}\cdot(A + B) + A\cdot(C + \overline{C})$	Anything OR'd w/inverse = 1
$\overline{A}\cdot(A + B) + A\cdot 1$	Anything AND'd w/1 = itself
$\overline{A}\cdot(A + B) + A$	Apply Distributive Law
$\overline{A}\cdot A + \overline{A}\cdot B + A$	Anything AND'd w/inverse = 0
$0 + \overline{A}\cdot B + A$	Anything OR'd w/0 = itself
$\overline{A}\cdot B + A = A + B$	Apply Rule 11

c.) `A·B + B·C(B + C) = B·(A + C)`   Answer: **True**

```
A·B + B·C(B + C)          Multiply B·C through (B + C)
A·B + B·C·B + B·C·C       Apply Associative Law to B·C·B
A·B + B·B·C + B·C·C       Anything AND'd w/itself = itself
A·B + B·C + B·C           Anything OR'd w/itself = itself
A·B + B·C = B·(A + C)     Pull out the B
```

24. Fill in the blank cells of the table below with the correct numeric format. ***For cells representing binary values, only 8-bit values are allowed!*** If a value for a cell is invalid or cannot be represented in that format, write "X". Use your scrap paper to do your work. (7 points per row)

Decimal	2's complement binary	Signed magnitude binary	Unsigned binary	Unsigned BCD
60	**00111100_2**	**00111100_2**	**00111100_2**	**01100000**
98	**01100010**	**01100010_2**	**01100010_2**	**10011000**
–45	**11010011_2**	**10101101_2**	**X**	**X**

For the first row, begin by breaking 60 into its power-of-two components. To do this, we successively pull out the largest powers of 2 that we can from 60 beginning with $2^5 = 32$. The result we get is:

$$60 = 32 + 16 + 8 + 4 = 2^5 + 2^4 + 2^3 + 2^2$$

Therefore, the unsigned binary representation of 60 has bits 2, 3, 4, and 5 set:

$$74_{10} = 00111100_2$$

Since 60 is a positive number, the 2's complement and signed magnitude representations are the same.

To convert from decimal to BCD, simply take each decimal digit, i.e., '6' and '0', and convert it to the associated BCD nibble which is the same as the hexadecimal nibbles from 0 to 9.

'6' = 0110 and '0' = 0000

Therefore, the BCD value is 01100000.

For the second row, since the MSB of the 2's complement value 01100010_2 equals 0, then 01100010_2 is positive. Since this is a positive value, the unsigned binary and signed magnitude representations are the same.

To convert 01100010_2 to decimal, add up the powers of two represented by the positions containing ones.

$$2^6 + 2^5 + 2^1 = 64 + 32 + 2 = 98_{10}$$

To convert from decimal 98 to BCD, simply take each decimal digit, i.e., '9' and '8', and convert it to the associated BCD nibble which is the same as the hexadecimal nibbles from 0 to 9.

'9' = 1001 and '8' = 1000

Therefore, the BCD value is 10011000.

In the last row, we can put an X's in the unsigned binary column and the BCD column since neither of these representations allow for negative values in eight bits. To convert from decimal to any of the other signed binary representation, we must first figure out what the unsigned binary representation of the positive value of +45 is. To do this, we successively pull out the largest powers of 2 that we can from 45 beginning with $2^5 = 32$. The result we get is:

$$45 = 32 + 8 + 4 + 1 = 2^5 + 2^3 + 2^2 + 2^0$$

Therefore, the unsigned binary representation of 45 has bits 0, 2, 3, and 5 set:

$$74_{10} = 00101101_2$$

To convert this to –45 in 2's complement, we can use the short cut which says starting on the right side, copy all digits up to and including the first 1 we come to. After that, invert the remaining bits to the left. The work below shows what happens with the values in red being copied straight down and the values in blue inverted.

45:	0	0	1	0	1	1	0	1
–45:	1	1	0	1	0	0	1	1

Therefore, the 2's complement representation of –45 is 11010011_2.
To convert to signed magnitude, simply take the original value for +45 and flip the MSB. This gives us the signed magnitude representation of –45 is 10101101_2.