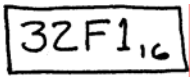
Points missed: _____ Student's Name: _____

Total score: _____ /100 points

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

Read this before starting!

- The total possible score for this test is 100 points.
- This test is ***closed book and closed notes***
- ***Please turn off all cell phones & pagers during the test.***
- 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:



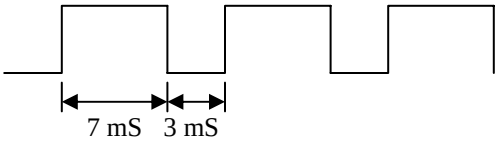
- **1 point will be deducted** per answer for missing or incorrect units when required. **No** assumptions will be made for hexadecimal versus decimal, so you should always include the base in your answer.
- If you perform work on the back of a page in this test, indicate that you have done so in case 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:	$\overline{(AB)} = (\overline{A} + \overline{B})$	$\overline{(A + B)} = (\overline{A} \cdot \overline{B})$

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)



frequency = $\frac{1}{\text{period}} = \frac{1}{7\text{ mS} + 3\text{ mS}} = \frac{1}{10\text{ mS}} = 100\text{ Hz}$

You could have left your answer as $1/(10\times10^{-3})$ 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?

Duty cycle = $\frac{t_h}{T} * 100\% = \frac{7\text{ mS}}{10\text{ mS}} * 100\% = 70\%$

3. If a processor is sampling an analog signal at 6,000 Hz, what is the upper limit of the frequency range it will be able to capture?

a.) 2,000 Hz (b.) 3,000 Hz c.) 4,000 Hz d.) 9,000 Hz e.) 12,000 Hz f.) 18,000 Hz

The Nyquist Theorem says that you need to sample a signal at more than twice the maximum frequency you wish to capture. Therefore, the upper limit of the frequency range you'll capture with a 6,000 Hz sampling rate is 3,000 Hz.

4. How many combinations of one's and zeros can be represented with 9 bits?

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}

The number of patterns of 1's and 0's that can be represented with n bits is 2^n. Substituting 9 for n gives us 2^9. The tricky part is that the maximum value that can be represented with unsigned binary notation is $2^9 - 1$. You need to count $0_{10} = 000000000_2$ in the total giving you 2^9.

5. What is the most negative value that can be stored using a 9-bit signed magnitude representation?

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

The most negative value in 9-bit signed magnitude representation is 111111111_2. The first bit is the negative sign. The rest of the number, 11111111_2, is magnitude. This is the maximum value for an 8-bit unsigned value, i.e., $2^8 - 1$. Therefore, the most negative value is $-(2^8 - 1)$

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

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

The easiest way to do this is just figure out what the maximum values are for each number of bits:

8 bits → $2^8 - 1 = 255$
9 bits → $2^9 - 1 = 511$
10 bits → $2^{10} - 1 = 1023$
11 bits → $2^{11} - 1 = 2047$
12 bits → $2^{12} - 1 = 4095$
13 bits → $2^{13} - 1 = 8191$

It looks like 9 is not enough, but 10 is.

7. 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 Inputs from a multiple choice test, e.g., pick an answer 1 through 5

   BCD Phone numbers that need to be stored quickly and efficiently

   FP Number of e-mails sent in a year (approximately 10 trillion worldwide)

8. True or False: The 32-bit IEEE 754 floating-point number 11000110100100111101101101011001 is greater than 1, i.e., the exponent is greater than or equal to zero.

   Let's begin by dividing up the floating-point number into its components.

S	E	F
1	10001101 = 128 + 8 + 4 + 1 = 141	00100111011011010011001

   Since E is greater than 127, the exponent in the expression shown below will be greater than 0.

$$\pm 1.F \times 2^{(E-127)}$$

9. Write the complete truth table for a 2-input NAND gate.

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

10. Multiply the 16-bit value 0000011010100000_2 by 32. Leave your answer in binary. (Hint: Remember the shortcut!) (3 points)

    Since 32 is a power of two, i.e., $32 = 2^5$, then the multiplication can be performed by simply shifting the binary number left 5 positions. This is the same as adding five zeros to the right side of the number.

$$0000011010100000\mathbf{00000}_2$$

11. Which of the following gates is the fastest?

    a.) AND b.) OR c.) XOR **d.) NAND** e.) NOR f.) XNOR

12. In the boolean expression below, circle the operation that would be performed first.

$$A \cdot B \cdot \overline{(C \oplus D)}$$

Medium-ish Answer (4 points each unless otherwise noted)

13. Convert the floating-point number 01000000101101100000000000000000 to its binary exponential format, e.g., 1.1010110×2^{-12}, (which, by the way, is not even close to correct).

    Once again, begin by dividing up the floating-point number into its components.

S	E	F
0	10000001 = 128 + 1 = 129	01101100000000000000000

    Substituting into the expression $\pm 1.F \times 2^{(E-127)}$ gives us our answer.

$$+1.F \times 2^{(E-127)} = +1.011011 \times 2^{(129-127)} = +1.011011 \times 2^2 = 101.1011$$

14. Convert 101.0001 to decimal. (You may leave your answer in expanded form if you wish.)

Remember that binary digits to the right of the point continue in descending integer powers relative to the 2^0 position. Therefore, the powers of two are in order to the right of the point $2^{-1} = 0.5$, $2^{-2} = 0.25$, $2^{-3} = 0.125$, and $2^{-4} = 0.0625$.

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

Therefore, the answer is:

$$2^2 + 2^0 + 2^{-4} = 4 + 1 + 1/16 = 5.0625$$

You could have left your answer in any of these forms in order to receive full credit.

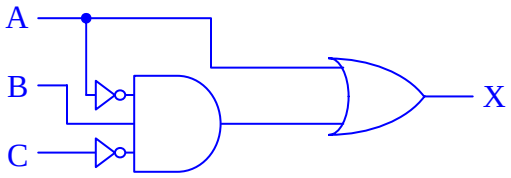
15. What does the expression A⊕A simplify to? (⊕ is the XOR or exclusive-OR) Hint: make the truth table and see if you recognize the output.

Remember that the XOR circuit counts the number of ones at the input. If there is an even number of ones, the XOR outputs a 0. An odd number of ones will force a 1 to be output. Using this information, we can fill out the truth table.

A	A⊕A
0	0 ⊕ 0 = 0
1	1 ⊕ 1 = 0

Therefore, A ⊕ A = 0.

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



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

First, the rule can be proven using Boolean algebra.

A + A·B	= A(1 + B)	Distributive Law
	= A·1	Rule 2: A + 1 = 1
	= A	Rule 4: A·1 = A

Or, you can use a truth table.

A	B	A·B	A + A·B
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

18. Convert 100001100110111100_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:

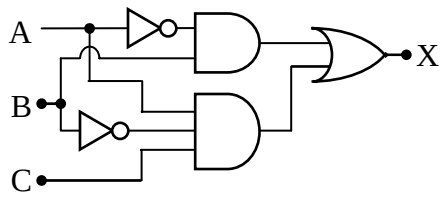
0010 0001 1001 1011 1100

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.

$219BC_{16}$

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

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



A	B	C	$\overline{A}$	$\overline{B}$	$\overline{A}\cdot B$	$A\cdot\overline{B}\cdot C$	$\overline{A}\cdot B + A\cdot\overline{B}\cdot C$
0	0	0	1	1	0	0	0
0	0	1	1	1	0	0	0
0	1	0	1	0	1	0	1
0	1	1	1	0	1	0	1
1	0	0	0	1	0	0	0
1	0	1	0	1	0	1	1
1	1	0	0	0	0	0	0
1	1	1	0	0	0	0	0

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

$$\overline{A}\cdot B + A\cdot\overline{B}\cdot C$$

21. If a 10-bit binary number is used to represent an analog value in the range from 100 to 900, 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

$= (900 - 100)/(2^{10} - 1)$

$= 800/1023$

$= 0.78201$ 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 $\overline{\overline{(A + B)} + C}$ to the individual input terms. ***Do not simplify!***

$\overline{\overline{(A + B)} + C}$	First, distribute the inverse across the OR outside the parenthesis. This places an inverse over the already inverted (A + B) and an inverse over the C while changing the OR to an AND.
$\overline{\overline{(A + B)} \cdot \overline{C}}$	The two inverses over (A + B) cancel each other out.
$(A + B) \cdot \overline{C}$	This gives us the final answer. Note the importance of the parenthesis.

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.) $A \cdot \overline{B} \cdot \overline{C} + A \cdot \overline{B} \cdot D + A \cdot \overline{B} \cdot \overline{D} + A \cdot \overline{B} \cdot C = A \cdot \overline{B}$ Answer: **True**

$A \cdot \overline{B} \cdot (\overline{C} + D + \overline{D} + C)$	Apply Distributive Law
$A \cdot \overline{B} (\overline{C} + C + \overline{D} + D)$	Rearrange using Associative
$A \cdot \overline{B} (1 + 1)$	Anything OR'd w/inverse = 1
$A \cdot \overline{B} \cdot 1$	1 + 1 = 1
$A \cdot \overline{B}$	Anything AND'd w/1 = itself

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

$A + \overline{B} \cdot \overline{C} + \overline{B} \cdot C$	Apply DeMorgan's Theorem
$A + \overline{B} \cdot (\overline{C} + C)$	Pull B out of last two terms
$A + \overline{B} \cdot 1$	Anything OR'd w/inverse = 1
$A + \overline{B}$	Anything AND'd w/1 = itself

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

$A \cdot B + A \cdot \overline{A} + B \cdot B + B \cdot \overline{A}$	F-O-I-L
$A \cdot B + B \cdot B + B \cdot \overline{A}$	$A \cdot \overline{A} = 0$ and drops out of OR
$A \cdot B + B + B \cdot \overline{A}$	$B \cdot B = B$
$B \cdot (A + 1 + \overline{A}) = B \cdot 1 = B$	Anything OR'd w/1 = itself

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". (7 points per row)

Decimal	2's complement binary	Signed magnitude binary	Unsigned binary	Unsigned BCD
140	**X**	**X**	**10001100**	**X**
98	**01100010**	**01100010**	**01100010**	**10011000**
–19	**11101101**	**10010011**	**X**	**X**

First row:

- Decimal: We simply need to figure out which powers of two are represented by the unsigned value 10001100_2 in order to calculate the decimal value. This gives us $2^7 + 2^3 + 2^2 = 128 + 8 + 4 = 140_{10}$. 2's Complement: Since the MSB is a 1, its magnitude as a positive value is too great for either of the signed representations, 2's complement and signed magnitude. Therefore, both of those representations get an 'X'.
- Signed Magnitude: See above.
- BCD: BCD is a bit different. Using the hexadecimal to binary conversion table shown earlier in this document, we see that $1_{10} = 0001$, $4_{10} = 0100$, and $0_{10} = 0000$. This means that the BCD value is 0001 0100 0000 which requires 12 bits → too many. Therefore, the BCD column gets an 'X' too. (I also accepted 000101000000)

Second row:

- Decimal: Since the MSB of the 2's complement representation is 0, this is a positive value and has the same pattern of ones and zeros as unsigned binary. Therefore, we simply need to figure out which powers of two are represented by the unsigned value 01100010_2 in order to calculate the decimal value. This gives us $2^6 + 2^5 + 2^1 = 64 + 32 + 2 = 98_{10}$.
- Signed Magnitude: Since 2's complement value is positive, the representation is exactly the same, i.e., 01100010.
- Unsigned Binary: Since 2's complement value is positive, the representation is exactly the same, i.e., 01100010.
- BCD: Using the hexadecimal to binary conversion table shown earlier in this document, we see that $9_{10} = 1001$ and $8_{10} = 1000$. This means that the BCD value is 10011000.

Third row:

- 2's Complement: To convert –19 to 2's complement, we begin by converting the positive value 19 to unsigned binary. Do this by breaking it into its powers of two components: 16, 2, and 1.

$19 = 16 + 2 + 1 = 2^4 + 2^1 + 2^0$ →

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	0	1	0	0	1	1

Using the shortcut to convert this to 2's complement, we get (red indicates inverted bits):

19:	0	0	0	1	0	0	1	1
–19:	1	1	1	0	1	1	0	1

- Signed Magnitude: Simply flip the MSB of the unsigned binary value 00010011 to get 10010011.
- Unsigned Binary and BCD: Both are positive values only, so these columns get X's.