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 3 for Spring Semester, 2005

Read this before starting!

- The total possible score for this test is 100 points.
- This test is closed book and closed notes.
- **All** answers **must** be placed in space provided. Failure to do so may result in loss of points.
- **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.
- ***Calculators are not allowed.*** Use the tables below for any conversions you may need. Leaving an answer as a numeric expression is acceptable.

Binary	Hex
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7

Binary	Hex
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Power of 2	Equals
2^3	8
2^4	16
2^5	32
2^6	64
2^7	128
2^8	256
2^9	512
2^{10}	1K
2^{20}	1M
2^{30}	1G

"Fine print"

Academic Misconduct:

Section 5.7 "Academic Misconduct" 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."

1. How many latches (memory cells holding one bit of data) does an SRAM with 24 address lines and 16 data lines have? Leave your answer in the form of an equation with numeric values. (2 points)

   An SRAM with 24 address lines has $2^{24} = 2^4 \times 2^{20} = 16$ Meg memory locations. Since we know that there are 16 data bits per location because of the 16 data lines, then this 16 Meg SRAM contains $2^{24} \times 16 = 2^{28} = 268{,}435{,}456$ latches. (Note: you are not responsible for any of the mathematical calculations. Simply putting $2^{24} \times 16$ would have been sufficient.)

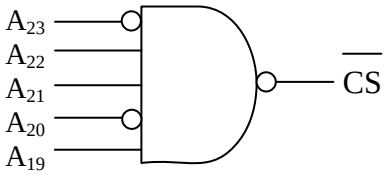
2. Circle ***all*** that apply. A storage cell in an SRAM: (4 points)

   a.) is volatile b.) is a capacitor c.) is cheaper than cells in a DRAM
   d.) is a latch e.) must be refreshed regularly f.) is smaller than cells in a DRAM
   g.) is typically used for main memory h.) is faster than an DRAM

3. Match each of the settings of the bus control signals $\overline{R}$ and $\overline{W}$ on the left with the bus operation on the right. (3 points)

$\overline{R}$	$\overline{W}$	Operation of the bus
0	0	Illegal setting
0	1	The bus is idle
1	0	Processor reads from memory
1	1	Processor writes to memory

4. What are the high and low addresses (in hexadecimal) of the memory range defined with the chip select shown to the right? (6 points)



   There are 24 address lines. This is found by noting that the highest address line has a subscript of 23 and therefore, since we begin counting at 0, we know that there are 24 address lines. Looking at the inputs to the NAND gate, we see that to set ^CS to zero, their values must be: A_{23}=0, A_{22}=1, A_{21}=1, A_{20}=0, and A_{19}=1. (Inverted inputs need a zero input to put a 1 into the NAND gate.) Therefore, the address lines have the following values for the high and low address: (The shaded areas represent the bits that go into the memory device's address lines.)

   Low address: $0110\ 1000\ 0000\ 0000\ 0000\ 0000_2 = 680000_{16}$
   High address: $0110\ 1111\ 1111\ 1111\ 1111\ 1111_2 = 6FFFFF_{16}$

5. For the chip select in problem 4, how big is the memory chip that uses this chip select? (3 points)

   There are 19 address lines that go to the address inputs of the memory chip. Therefore, there are 2^{19} possible addresses meaning that the memory chip has $2^{19} = 2^9 \times 2^{10} = 512$K memory locations.

6. For the chip select in problem 4, how big is the memory space of the processor whose address lines are used for the chip select? (3 points)

   There are 24 address lines coming out of the processor. Therefore, there are 2^{24} possible addresses that the processor can address, i.e., the memory space is $2^{24} = 2^4 \times 2^{20} = 16$Meg.

7. True or false: The address range 65000_{16} to $66FFF_{16}$ is a valid range for a single memory. (2 points)

   Begin by converting the low and the high addresses to binary.

	A_{19}	A_{18}	A_{17}	A_{16}	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
Lo addr =	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Hi addr =	0	1	1	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1

   There is no way to divide this set of addresses into the most significant bits defining the chip select (i.e., the bits that say constant) and the bits that go to the memory chip's address lines, (i.e., the bits that go from all zeros to all ones). If you draw the line between A_{11} and A_{12}, then both A_{12} and A_{13} flip making chip select design impossible. If you draw the line between A_{14} and A_{13}, You can make the chip select, but A_{12} messes it up for the memory chip's address lines. Therefore, since we can't partition this address set into chip-select and memory chip address lines, the answer is **FALSE**.

8. What is the largest memory that can have a starting (lowest) address of $8C000_{16}$? (3 points)

   Remember that the lowest address must have all zeros going to the address lines of the memory chip. Therefore, if we can determine the number of bits equal to zero starting with the least significant, A_0, and going left, then we know how many address lines can be used for the memory chip. Begin by converting $8C000_{16}$ to binary.

$$8C000_{16} = 1\ 0\ 0\ 0\ \ 1\ 1\ 0\ 0\ \ 0\ 0\ 0\ 0\ \ 0\ 0\ 0\ 0\ \ 0\ 0\ 0\ 0$$

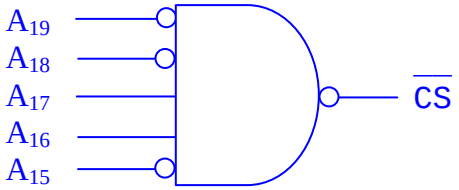
   There are fourteen zeros before you get to the first 1 in the binary value of $8C000_{16}$. Therefore, up to fourteen address lines can go to the memory chip. This gives us an answer of $2^{14} = 2^4 \times 2^{10} =$ 16K.

9. Using logic gates, design an active low chip select for a RAM placed in a 1 Meg memory space with a low address of 30000_{16} and a high address of $37FFF_{16}$. ***Label all address lines used for chip select.*** (6 points)

   Since 1 Meg = 2^{20}, the processor must have 20 address lines coming out of it. (A_0 through A_{19}) Converting the high and low addresses shows us where to draw the line separating the address lines that go to the chip select from the address lines that go to the memory chip.

$$30000_{16} = 0011\ 0000\ 0000\ 0000\ 0000_2$$
$$37FFF_{16} = 0011\ 0111\ 1111\ 1111\ 1111_2$$

   This shows that the lower 15 address lines (A_0 through A_{14}) go to the memory chip, and the upper 5 address lines (A_{15} through A_{19}) go to the chip select. Also from this diagram, we see that $A_{19} = 0$, $A_{18} = 0$, $A_{17} = 1$, $A_{16} = 1$, and $A_{15} = 0$. By inverting the inputs that are to be recognized as zeros, we get the NAND circuit for the chip select shown to the right.



10. True or false: The rotational speed of the platter(s) measured in rotations per minute (RPM) of a ***multiple zone recording*** hard drive varies depending on the position of the head. (2 points)

    FALSE: In multiple zone recording, the disk speed is constant. It is only the number of sectors per track that vary depending on where the head is positioned along its axis.

11. True or false: A small gap is left between the ends of adjacent sectors on a single track of a hard drive disk in order to avoid data bleeding over from one sector to the other. (2 points)

    FALSE: The gap between tracks is meant to prevent data from bleeding from one track to another. The gap between adjacent sectors is for synchronization.

12. True or false: When storing data to the magnetic material coating the surface of a hard drive platter, one direction of magnetic polarization represents the logic ones while the other direction represents the logic zeros. (2 points)

    FALSE: If the direction of magnetic polarization dictated ones and zeros, then a long stream of all zeros or a long stream of all ones would cause the hard drive controller to lose where it was in the bit stream, i.e., it might lose its synchronization. In addition, the heads cannot read directions in polarity, only changes in polarity. That is why we needed to use the different encoding methods described in class.

13. Describe how the LRU replacement algorithm for the fully associative mapping algorithm works. (3 points)

    When a new line needs to be stored in the cache, but there is no room for it, an existing line must be discarded. The least recently used (LRU) replacement algorithm discards the line for which the longest period of time has passed since it was last accessed by the processor.

The table below represents a small section of a cache that uses fully associative mapping. Refer to it to answer questions 14, 15, 16, and 17. Assume the processor's memory bus uses 20 bits for an address.

Tag (binary values)	Word within the block							
	000	001	010	011	100	101	110	111
01001001101010010	00_{16}	61_{16}	$C2_{16}$	23_{16}	84_{16}	$E5_{16}$	46_{16}	$A7_{16}$
11001100101110100	10_{16}	71_{16}	$D2_{16}$	33_{16}	94_{16}	$F5_{16}$	**56_{16}**	$B7_{16}$
00011101100110101	20_{16}	81_{16}	$E2_{16}$	43_{16}	$A4_{16}$	05_{16}	66_{16}	$C7_{16}$
10110011110011010	30_{16}	91_{16}	$F2_{16}$	53_{16}	$B4_{16}$	15_{16}	76_{16}	$D7_{16}$
01011001111001101	40_{16}	$A1_{16}$	02_{16}	63_{16}	$C4_{16}$	25_{16}	86_{16}	$E7_{16}$
01001010110010010	50_{16}	$B1_{16}$	12_{16}	73_{16}	$D4_{16}$	35_{16}	96_{16}	$F7_{16}$
	col 0	col 1	col 2	col 3	col 4	col 5	col 6	col 7

14. From what address in main memory did the value 56_{16} (the value in bold) come from? Leave your answer in binary. (3 points)

    Fully associative mapping divides the physical address into two pieces, the tag and the word id. The word id is the last n-bits of the address where memory is divided into blocks of size 2^n. In case of the fully associative cache shown above, n=3. Since the value has a tag of 11001100101110100_2 and a word id of 110_2, then the physical address is **11001100101110100110_2 = $CCBA6_{16}$**.

15. A copy of the data from memory address $59E6E_{16}$ is contained in the portion of the cache shown above. What is the value stored at that address? (3 points)

Dividing the physical address $59E6E_{16} = 01011001111001101110_2$ into its tag and 3-bit word id gives us a tag of 0101100111100110_2 and a word id of 110_2. Searching through the visible lines shows us that the fifth line from the top has the same tag, i.e., it contains the block which contains the data from the physical address $59E6E_{16}$. A word id of 110_2 points us to the data in the second to the last column, i.e., the value of **86_{16}**.

16. If the block containing memory address $A46FD_{16}$ were to be loaded into the cache described above, what would the tag be? (2 points)

Dividing the physical address $A46FD_{16} = 10100100011011111101_2$ into its tag and 3-bit word id **gives us a tag of 10100100011011111_2** and a word id of 101_2.

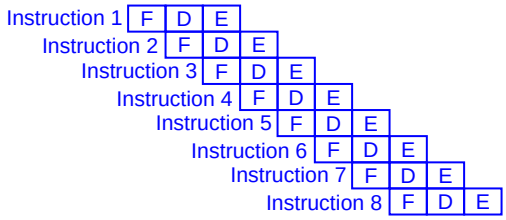
17. What column (0 through 7) would the data item from the previous problem be stored in? (2 points)

The word id of 101_2 corresponds to the sixth column from the left, i.e., **column 5**.

18. Assume a processor takes 3 cycles to execute any instruction (fetch, decode, execute)

a. How many cycles would a ***pipelined*** processor take to execute 8 instructions? (2 points)

A 3-stage pipelined processor overlaps 2 cycles for each instruction as shown in the figure below.



Therefore, it will take 3 cycles to execute the first full instruction, then one cycle per instruction to execute the remaining instructions which in this case is 7 instructions..

number of cycles = 3 + 7 = 10 cycles

b. How many cycles would a ***non-pipelined*** processor take to execute 8 instructions? (2 points)

A non-pipelined processor simply executes the instructions one at a time with no overlap. Therefore, the number of cycles equals 3 cycles/instruction times the number of instructions:

number of cycles = 3 × 8 = 24

19. If a processor compares two values, what are the settings of the zero flag and sign flag if the two values are equal? (2 points)

ZF = ___**1**___ SF = ___**0**___

20. Which 80x86 pointer/index register points to the function parameters stored in the stack? (2 points)

The base pointer (BP)

21. Which two 80x86 pointer/index registers are used for string functions. (3 points)

The source index (SI) and the destination index (DI)

22. Name the two benefits of the segment/pointer addressing system of the 80x86. (3 points)

- It allows us to use 16 bit registers (word-length) to access larger (20-bit) address spaces
- It allows for relocatable code in memory

Answer questions 23 through 26 using the following settings of some of the 80x86 registers.

$AX = 1234_{16}$	$SP = 3021_{16}$	$CS = 1000_{16}$
$BX = 4321_{16}$	$SI = 4356_{16}$	$SS = 2000_{16}$
$CX = 0101_{16}$	$DI = 1423_{16}$	$DS = 3000_{16}$
$DX = FEDC_{16}$	$BP = 5987_{16}$	$ES = 4000_{16}$

23. What is the value contained in the register DH? (2 points)

DH is the upper half (byte) of DX. Therefore, the value in DH is **FE_{16}**.

24. What is the physical address pointed to by ES:DI? (3 points)

ES contains the segment address and DI contains the pointer address. To figure out the physical address, begin by converting the 16-bit value in ES to the 20 segment address by adding a hex 0 to the end of the segment value (4 binary 0's).

ES = 4000_{16} → the segment address is $4000\mathbf{0}_{16}$ (notice the added zero)

The pointer value (DI in this case) can then be added as an offset to the segment address.

```
 40000
+ 1423
-------
 41423
```

Therefore, the physical address pointed to by ES:DI (4000:1423) is **41423_{16}**.

25. True or false: The physical address of the next instruction to be executed by the processor can be calculated from the above data? (2 points)

FALSE: The physical address of the next instruction to execute is determined from the values contained in CS and IP (CS:IP). CS is available from the table above, but IP is not.

26. True or false: The physical address of the last item to be stored to the stack can be calculated from the above data? (2 points)

TRUE: The physical address of the last item to be stored to the stack can be calculated from the values contained in SS and SP (SS:SP). Since both of those values are present above (SS:SP = 2000:3021), then we can calculate the physical address of the stack.

27. Assume AX=1000_{16}, BX=2000_{16}, and CX=3000_{16}. After the following code is executed, what would AX, BX, and CX contain? (3 points)

```
PUSH CX
PUSH BX
PUSH AX
POP BX
POP CX
POP AX
```

Place your answers in space below:

AX = **old CX = 3000_{16}**

BX = **old AX = 1000_{16}**

CX = **old BX = 2000_{16}**

28. Using an original value of 00111100_2 and a mask of 00001111_2, calculate the results of a bitwise AND, a bitwise OR, and a bitwise XOR for these values. (2 points each)

Original value	Bitwise operation	Mask	Result
00111100_2	AND	00001111_2	**00001100_2**
00111100_2	OR	00001111_2	**00111111_2**
00111100_2	XOR	00001111_2	**00110011_2**

29. For each of the following binary bit patterns, set the parity bit for odd parity, i.e., the same parity settings as used by the 80x86 processor parity flag.

Binary value	Parity	
1 0 1 0 1 0 1 1	**0**	(1 point)
1 0 1 1 0 1 1 1	**1**	(1 point)
1 0 0 1 1 1 0 1	**0**	(1 point)

30. If the datasum calculated from a sequence of values is a decimal 95, what would the decimal value of the 2's complement checksum be? (2 points)

Since the 2's complement checksum is equal to the negative of the datasum, then the decimal value of the 2's complement checksum would be **–95**.

31. Describe the primary drawback discussed in class of parity checks. (2 points)

By itself, parity makes a poor choice for error detection. Specifically, this is due to the fact that if an even number of bit errors occurs, the parity will still appear to be correct. For example, if you were to count the number of ones in the value 01101010_2, you would find that there are an even number of ones. Using an XOR parity generator, this would give us a parity of 0, i.e., an even number of 1's always outputs a 0. If two bit errors occurred, e.g., 01101010_2 turned into 11111010_2, we would have 6 ones, still an even number, and a parity of 0.

32. Describe one of the two reasons discussed in class for using an XOR "borrow-less" subtraction in the calculation of a CRC. (2 points)

First, an XOR "borrow-less" subtraction is very fast. Second, an XOR "borrow-less" subtraction only needs to operate in a small window meaning you don't have to have the entire dividend to begin doing a long division. There is sort of a third benefit in that an XOR "borrow-less" subtraction has the same result as an XOR "borrow-less" addition meaning that if we "add" the remainder generated from a CRC to the dividend, it's the same as subtracting it thereby creating a dividend that the divisor divides evenly.

33. True or false: The order of operands in an XOR "borrow-less" subtraction does not matter, e.g., A – B = B – A in a borrow-less subtraction. (2 points)

TRUE: Because a borrow-less subtraction is simply a bitwise XOR, the result does not depend on the order of the operands. For example:

```
  0110          1100
⊕ 1100        ⊕ 0110
  1010          1010
```

34. Circle ***all*** of the following statements that are true for the use of a CRC as a method of data verification in serial communications. (6 points)

a. They out perform basic checksums by translating minor changes in the data into significant changes in the check value.
b. The number of bits in the divisor must be exactly one bit longer than the number of bits per data item being transmitted.
c. A zero result from performing a CRC on received data indicates that the data was error free.
d. Both the transmitting device and the receiving device must use the same polynomial (divisor) for calculation of the CRC.
e. The transmitting device must wait until it is finished computing the CRC before sending any of the serial data stream.
f. The quotient resulting from the long-division used to calculate the CRC is of no importance to the verification of the data. Only the remainder is used.