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 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.***
- **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 - 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}	1 kilo (K)
2^{20}	1 mega (M)
2^{30}	1 giga (G)
2^{40}	1 tera (T)
2^{50}	1 peta (P)

"Fine print"

Academic Misconduct:

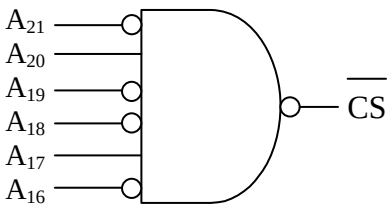
Section 5.7 "Academic Misconduct" of the East Tennessee State University Faculty Handbook, October 21, 2005:

"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. Circle ***all*** that apply. A memory cell in an DRAM: (4 points)

a.) is cheaper than a cell of an SRAM	b.) is volatile	c.) is refreshed to avoid losing data
d.) is faster than a cell of an SRAM	e.) is a D latch	f.) is used for main memory
g.) is smaller than a cell in a SRAM	c.) uses a charge on a capacitor to represent a logic 1	

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



There are 22 address lines. This is found by noting that the highest address line has a subscript of 21 and therefore, since we begin counting at 0, we know that there are 22 address lines. Address lines 16 to 21 go to the chip select circuitry while the remaining sixteen lines, 0 through 15, go to the address inputs of the memory device.

Looking at the inputs to the NAND gate, we see that to set ^CS to zero, their values must be: A_{21}=0, A_{20}=1, A_{19}=0, A_{18}=0, A_{17}=1, and A_{16}=0. (Inverted inputs need a zero input in order to send a 1 into the NAND gate.) Therefore, the address lines have the following values for the high and low address. (Note that the shaded areas represent the bits that go into the memory device's address lines and range from all 0's for the low address to all 1's for the high address.)

Low address: $01\ 0010\ 0000\ 0000\ 0000\ 0000_2 = 120000_{16}$
High address: $01\ 0010\ 1111\ 1111\ 1111\ 1111_2 = 12FFFF_{16}$

3. For the chip select in problem 2, how big is the memory chip that uses this chip select? (3 points)

There are 16 address lines that go to the address inputs of the memory chip. Therefore, there are 2^{16} possible addresses. This means that the memory chip has $2^{16} = 2^6 \times 2^{10}$ = ***64K memory locations.***

4. For the chip select in problem 2, how big is the memory space of the processor whose address lines are used for the chip select? (3 points)

There are 22 address lines coming out of the processor. Therefore, there are 2^{22} possible addresses that the processor can address, i.e., the memory space is $2^{22} = 2^2 \times 2^{20}$ = ***4Meg.***

5. True or false: The address range 68000_{16} to $6DFFF_{16}$ is a valid range for a single memory. (2 points)

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

A_{14} A_{13} A_{12}

Low addr: $68000_{16} = 0110\ 1000\ 0000\ 0000\ 0000_2$
High addr: $6DFFF_{16} = 0110\ 1101\ 1111\ 1111\ 1111_2$

There is no way to divide this set of addresses into the most significant bits defining the chip select (the bits that stay constant) and the bits that go to the memory chip's address lines (the bits that go from all zeros to all ones). If you draw the line between A_{12} and A_{13} then A_{14} flips making chip select design impossible. If you draw the line between A_{14} and A_{15}, you can make the chip select,

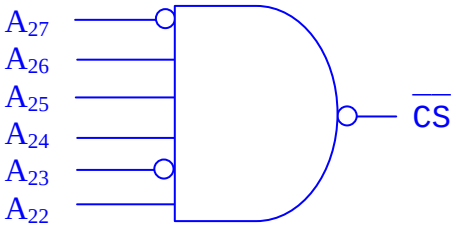
but A_{13} messes things up by not going across the full range, i.e., from all zeros to all ones for the memory chip address inputs. Therefore, since we can't partition this address set into chip-select and memory chip address lines, the answer is **FALSE**.

6. Using logic gates, design an active low chip select for a memory device placed in a 256 Meg memory space with a low address of 7400000_{16} and a high address of $77FFFFF_{16}$. ***Label all address lines used for chip select.*** (5 points)

Since 256 Meg = $2^8 \times 2^{20} = 2^{28}$, the processor must have 28 address lines coming out of it. (A_0 through A_{27}). 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.

7400000_{16} = 0111 0100 0000 0000 0000 0000 0000$_2$
$77FFFFF_{16}$ = 0111 0111 1111 1111 1111 1111 1111$_2$

This shows that the lower 22 address lines (A_0 through A_{21}) go to the memory chip, and the upper 6 address lines (A_{22} through A_{27}) go to the chip select. Also from this diagram, we see that $A_{27} = 0$, $A_{26} = 1$, $A_{25} = 1$, $A_{24} = 1$, $A_{23} = 0$, and $A_{22} = 1$. By inverting the inputs that are to be recognized as zeros, we get the NAND circuit for the chip select shown to the right.



7. A 4 Meg memory can have a starting address of $5D00000_{16}$. (2 points)

a.) True (b.) False c.) Not enough information given

Begin by converting the starting addresses to binary.

A_{19}

Low addr: $5D00000_{16}$ = 0101 1101 0000 0000 0000 0000 0000$_2$

All of the address lines going to the memory chip must be set to zero for the lowest address. Since the low address $5D00000_{16}$ has contiguous zeros from bit A_0 to bit A_{19}, then all 20 of these address lines can go to the memory chip. This means that the largest memory that can start at this address is one of size 2^{20} = 1Meg. Unfortunately, this won't be big enough, so the answer is **FALSE**.

8. A chip select can be designed for a memory device with a starting address of $2C000_{16}$ for a processor with a 256 Meg memory space. (2 points)

a.) True b.) False (c.) Not enough information given

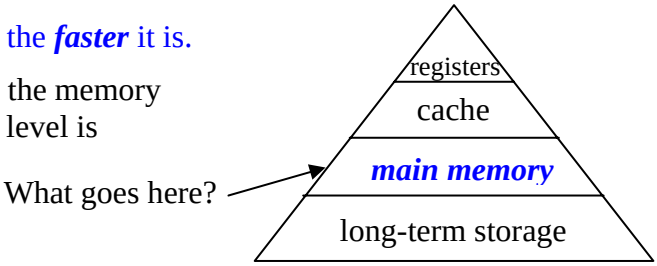
In order to design a chip select, need to know the number of address lines from the processor, the number of address lines going to the memory device, and an address within the memory device's address range. From this problem, we have the address and the size of the memory space, but we don't have the size of the memory. Therefore, there is not enough information.

9. Name the primary characteristic of storage devices that ***improves*** as you move ***closer*** to the processor through the memory hierarchy? (2 points)

   The closer the storage device is to the processor, the ***faster*** it is.

10. To the right you should see a figure representing the memory hierarchy with one of the levels missing. Which level is missing? (2 points)

    Between the cache and the long-term storage, i.e., the hard drive, is the ***main memory***.



11. List one of the two reasons discussed in class why data encoding is necessary to store data on a hard drive, i.e., why must a pattern of polarity changes be used to store data instead of simply having one polarity direction represent 1's while the other direction represents 0's. (2 points)

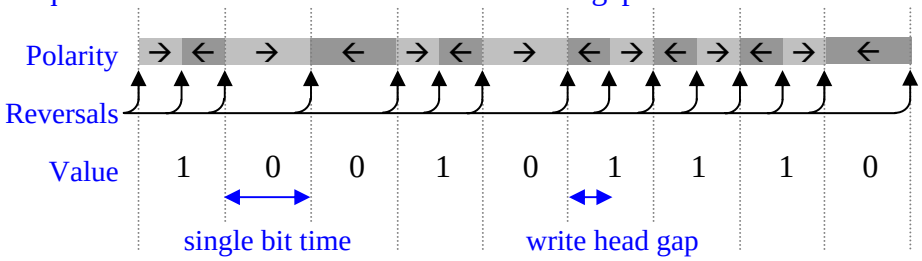
    Encoding must be used to identify ones and zeros instead of the two opposing polarity directions because:

    - Good encoding design can be used to achieve data compression.
    - The controllers only detect changes in magnetic direction, not the direction of the field itself.
    - Large blocks of data that are all 1's or all 0's would be difficult to read because eventually the controller might lose track of where one bit ended and the next began.

12. FM encoding has a magnetic polarity change at the beginning of every bit time and in the middle of a bit time representing a logic 1. Therefore, the width of 1 bit is equal to _____ times the width of the gap in the hard drive's write head, i.e., the minimum length of a polarity change. (2 points)

    a.) ½ b.) ¾ c.) 1 d.) 1.5 e.) 2 f.) 3 e.) varies

    Remember that the gap of the write head must be equal to or smaller than the minimum length of a polarity change. Since the smallest polarity in FM places a polarity change at the beginning and the middle of a bit time, the write head gap must be equal to half the size of a bit. Therefore, the width of 1 bit is equal to 2 times the width of the write head gap.



13. Circle ***one***: A gap is left between tracks on a hard drive. This is to: (2 points)

    a.) provide flexibility in case a new encoding algorithm is used
    b.) provide synchronization, i.e., help the hard drive controller know where the head is positioned
    c.) prevent data from "bleeding over" from one track to the next.
    d.) none of the above

14. Circle ***one***: Which of the following statements best describes Multiple Zone Recording? (2 points)

a.) The rate at which data is read from the disks remains constant regardless of head position.
b.) The hard drive controller may change the rotational speed of the platters/disks.
c.) Special encoding is used on the platters/disks to identify the position of the disks.
d.) Outer tracks have a greater number of sectors to take advantage of the capabilities of the head.
e.) More than one head is used per side of a platter/disk.

15. True or false: The drawback of ***multiple zone recording*** hard drives is that the controller is more complex than that of constant angular velocity hard drives. (2 points)

16. Describe how the LRU replacement algorithm for the fully associative mapping algorithm works. (2 points)

When the cache needs to free up a line in order to store a new block, the least recently used (LRU) replacement algorithm deletes/clears the line in the cache for which the longest period of time has passed since it was last used. It requires a timer for each line of the cache. The timer for a line is reset to zero each time that line is accessed. The line with the highest value in its timer is deleted.

17. True or false: In a properly operating fully associative cache, it is possible to have two lines with identical tags. (2 points)

The table to the right represents a small section of a cache that uses fully associative mapping. Refer to it to answer questions 18 through 22.

	Word ID			
Tags (in binary)	00	01	10	11
01101101100101100000110	$A0_{16}$	01_{16}	62_{16}	00_{16}
10011011010011011010000	$6B_{16}$	71_{16}	$D7_{16}$	11_{16}
00001111011010011010001	$C0_{16}$	21_{16}	82_{16}	22_{16}
10110010011001101111110	$3D_{16}$	93_{16}	**$F9_{16}$**	33_{16}
10010011011010101101101	$E0_{16}$	31_{16}	02_{16}	44_{16}
01001010110100110101010	$5F_{16}$	$B5_{16}$	$2A_{16}$	55_{16}
00111111100011001100111	BB_{16}	CC_{16}	89_{16}	$9A_{16}$
10100111000100110100011	AA_{16}	DD_{16}	67_{16}	AB_{16}
11111111100000000110011	99_{16}	EE_{16}	56_{16}	BC_{16}
01011011000000000011101	88_{16}	FF_{16}	45_{16}	CD_{16}
01011001010011111111111	77_{16}	01_{16}	34_{16}	EF_{16}
	col A	col B	col C	col D

18. Assuming the tags shown to the right do ***not*** delete leading zeros, how many address lines does the processor that uses this cache have? (2 points)

Each tag has 22 bits and each word ID has 2 bits. Since the original address is made from concatenating the tag and word ID, the processor has 22 + 2 = ***24 address lines***.

19. What is the block size (in number of memory locations) for the cache shown to the right? (2 points)

The block size is the same as the number of words that can be contained in a line of the cache. In the case of this example, there are ***4 words to a block***. Another way of looking at it is that the block size is determined by the number of bits in the word id, i.e., how many words are in a block. Since two bits are used for the word id, the number of unique word id's for a block is $2^2 = 4$.

20. From what address in main memory did the value $F9_{16}$ (the value in bold) come from? Leave your answer in binary. (3 points)

Fully associative mapping divides the physical address into two pieces, the block id (which is used as the tag) and the word id. In this case, the word id is the last 2 bits of the address. Since the $F9_{16}$

has a tag of $10110010011001101111 10_2$ and a word id of 10_2, then the physical address is $10110010011001101111 1010_2$ **= $B266FA_{16}$.**

21. A copy of the data from memory address $A71345_{16}$ is contained in the portion of the cache shown above. What is the value stored at that address? (3 points)

Dividing the physical address $A71345_{16} = 101001110001001101000101_2$ into its tag and 2-bit word id gives us a tag of $10100111000100110100 01_2$ and a word id of 01_2. Searching through the visible lines shows us that the fourth line from the bottom has the same tag, i.e., it contains the block which contains the data from the physical address $A71345_{16}$. A word id of 01_2 points us to the data in the second column, i.e., the value of **DD_{16}**.

22. ***If*** the block containing memory address $13C249_{16}$ were to be loaded into the cache described above, which column, A, B, C, or D would the value be loaded into? (Note: This value it is not represented in the data shown above.) (2 points)

Dividing the physical address $13C249_{16} = 000100111100001001001001_2$ into its tag and 2-bit word id gives us a tag of $00010011110000100100 10_2$ and a word id of 01_2. The word ID of 01 corresponds to the second column which is ***column B***.

23. True or false: The primary reason discussed in class for forcing a processor's pipeline to be flushed is when a branch occurs. This can be caused by things like an if-statement or a loop. (2 points)

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

a. How many cycles would a ***non-pipelined*** processor take to execute 7 instructions? (2 points)

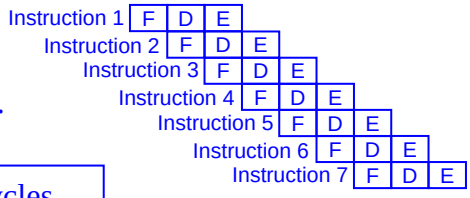
A non-pipelined processor simply executes the instructions one at a time with no overlap. Therefore, the number of cycles equals 3 cycles/instruction times the number of instructions:

number of cycles = $3 \times 7 = 21$ cycles

b. How many cycles would a ***pipelined*** processor take to execute 7 instructions? (2 points)

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

Therefore, it will take 2 cycles to fill the pipe, then one cycle to execute each instruction. This means 2 cycles to fill the pipeline plus 7 cycles to execute the 7 instructions.



number of cycles = 2 + 7 = 9 cycles

25. What are the settings of the zero flag, the sign flag, the carry flag, the overflow flag, and the parity flag after a processor performs the addition shown to the right? (5 points)

```
1  11
  11110001
+ 10010010
  10000011
```

ZF = ___**0**___ SF = ___**1**___ CF = ___**1**___ OF = ___**0**___ PF = ___**1**___

The zero flag (ZF) is set to a 1 only if the result equals 0, which in this case it does not. The sign flag (SF) is set to a one for negative values and follows the most significant bit of the result. The

carry flag (CF) contains the carry out of the most significant bit of the addition. The overflow flag (OF) is set to one when there is a two's complement overflow, i.e., two positive numbers are added together resulting in a negative number or two negative numbers are added together resulting in a positive number. There is no 2's complement overflow in this addition. Lastly, the parity flag (PF) is set to a 1 if the number of ones in the resulting binary value is odd. There are three ones in the result for this case, therefore, the parity flag contains a 1.

26. Remember that a compare is basically a "virtual subtract", i.e., CMP A, B is the same thing as setting the flags after the operation A – B. What would the values of ZF and SF be if A is less than B? (2 points)

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

A – B would result in a non-zero negative value if A is less than B.

27. What is the purpose of the ALU? (2 points)

The ALU performs the arithmetic and logic for the processor. It is in essence the processor's calculator.

28. 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 AX
PUSH CX
PUSH BX
POP BX
POP AX
POP CX
```

Place your answers in space below:

AX = **3000_{16}**

BX = **2000_{16}**

CX = **1000_{16}**

29. Name one of the three purposes discussed in class for a stack. (2 points)

There are a number of purposes for the stack including:

- a temporary storage of register values when the processor runs out of registers;
- a place to swap register values as in the previous problem;
- a location in which to pass values to a function; and
- a location to store the return address from a function.

30. How does the processor determine what to put in the sign flag (SF)? In other words, what part of the result is used to determine what goes in the sign bit? (2 points)

The sign flag uses ***the most significant bit of a result*** to determine whether the result is negative or not.

31. Remember that a signed magnitude binary value uses the first bit as the sign bit. If we flip it, we change it from a positive to a negative number or vice versa. Which bitwise operation could be used to flip the sign of a signed magnitude binary value? (2 points)

a.) AND b.) OR (c.) XOR d.) This function is not possible with a bitwise operation

32. Which bitwise operation could be used to find the absolute value of a signed magnitude binary value, i.e., force the sign bit to a 0? (2 points)

a.) AND b.) OR c.) XOR d.) This function is not possible with a bitwise operation

33. Using an original value of 10100101_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
10100101_2	AND	00001111_2	**00000101_2**
10100101_2	OR	00001111_2	**10101111_2**
10100101_2	XOR	00001111_2	**10101010_2**

34. In a 2's complement checksum scheme, the receiving processor adds all of the words of data, then adds the received checksum to the resulting datasum. Disregarding any carry, the final result should be: (2 points)

a.) a binary number with all 1's
b.) a binary number with all 0's
c.) a binary number equal to the checksum
d.) none of the above

35. There are two ways of handling the carries that occur when generating the datasum for a checksum. One way is to simply discard all carries. What is the other way? (2 points)

Add the carries back into the datasum.

36. A CRC is analogous to what result from what mathematical operation? Be specific. (3 points)

The remainder from a long division.

37. For each of the following statements, place a checkmark in the column(s) identifying which protocol(s) the statement describes. Some statements have more than one checkmark. (6 points)

Ethernet	IP	TCP	
☒	☐	☐	Uses addresses that are hard-wired into the network interface cards
☐	☐	☒	Is used to manage the partitioning of a large message into smaller messages
☐	☒	☐	Uses a logical address defined by a network administrator for its addressing.
☒	☐	☐	Uses a 4-byte CRC for error checking
☐	☐	☒	Uses a pseudo-header in addition to its frame header to calculate checksum
☐	☒	☐	Includes a “time to live” field so that it can be removed from the network(s) in case it cannot find its destination.