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, 2006

# Section 001

**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, 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, 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. For each of the following types of signals connected to a memory chip, identify whether it is an input going into the memory device from the processor or bidirectional, i.e., signals go both directions between the memory device and the processor. (4 points)

| Input to memory | Bidirectional | |
|---|---|---|
| ☐ | ☑ | - data lines |
| ☑ | ☐ | - address lines |
| ☑ | ☐ | - $\overline{R}$ |
| ☐ | ☐ | - chip select |

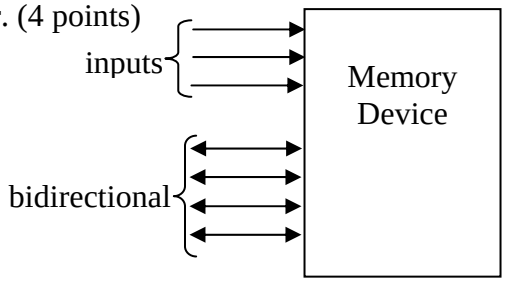
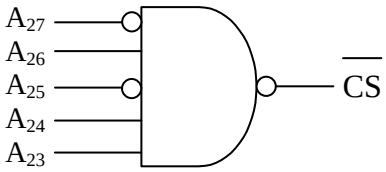



*Diagram included only as reference*

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 cache RAM h.) is faster than an DRAM

3. 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 28 address lines. This is found by noting that the highest address line has a subscript of 27 and therefore, since we begin counting at 0, we know that there are 28 address lines. Address lines 23 to 27 go to the chip select circuitry while the remaining lines, 0 through 22 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: $A_{27}$=0, $A_{26}$=1, $A_{25}$=0, $A_{24}$=1, and $A_{23}$=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. (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: 0101 1000 0000 0000 0000 0000 $0000_2$ = $5800000_{16}$
High address: 0101 1111 1111 1111 1111 1111 $1111_2$ = $5FFFFFF_{16}$

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

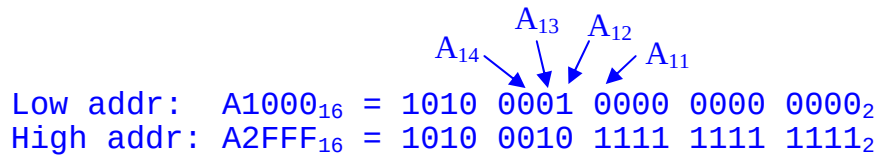
There are 23 address lines that go to the address inputs of the memory chip. Therefore, there are $2^{23}$ possible addresses. This means that the memory chip has $2^{23} = 2^3 \times 2^{20}$ = 8M memory locations.

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

There are 28 address lines coming out of the processor. Therefore, there are $2^{28}$ possible addresses that the processor can address, i.e., the memory space is $2^{28} = 2^8 \times 2^{20}$ = 256Meg.

6. True or false: The address range $A1000_{16}$ to $A2FFF_{16}$ is a valid range for a single memory. (2 points)

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



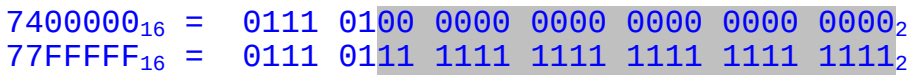

   There is no way to divide this set of addresses into the most significant bits defining the chip select (the bits that stay constant) and the bits that go to the memory chip's address lines (the bits that go from all zeros to all ones). If you draw the line between $A_{11}$ and $A_{12}$, then both $A_{12}$ 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**.

7. Chip selects are typically active low making the NAND gate the gate of choice for their operation. What gate would be the best for an ***active high*** chip select? (2 points)

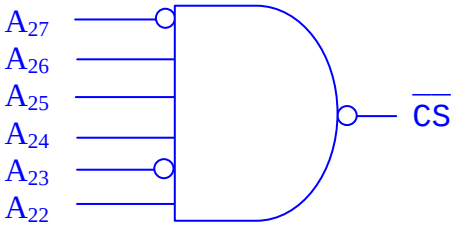
   a.) AND  b.) NAND  c.) OR  d.) NOR  e.) XOR  f.) Exclusive-NOR  g.) NOT

8. 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.




   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.




9. For each of the four following groups of information, put a check mark next to the ones for which there is enough information to correctly make the chip select logic for a memory device. Assume that you know the number of address lines coming from the processor. (4 points)

   - [√] The ending (high) address and the size of the memory device.
   - [√] The starting (low) address and the size of the memory device.
   - [√] The any valid address for the memory device and the size of the memory device.
   - [√] The high and low addresses for the memory device's address range.

10. Name two characteristics of storage devices that ***improve*** as you move ***down*** through the memory hierarchy away from the processor? (3 points)

    Capacity and cost per byte. Also, the lowest level of the hierarchy is non-volatile.

11. Frequency modulation (FM) magnetic encoding changes the magnetic polarity between every bit position and in the middle of bit positions where 1's are stored. Modified frequency modulation (MFM) encoding changes the magnetic polarity only between consecutive zeros and in the middle of a bit position where a 1 is stored. How much more data can be stored on a drive using MFM encoding than on an identical drive using FM encoding? (2 points)

    a.) No difference (b.) Twice as much c.) 4 times as much d.) It depends on the data stored

12. A gap is left between sectors within a track on a hard drive. This is to: (2 points)

    (a.) provide synchronization, i.e., help the hard drive controller know where it is on a track
    b.) prevent data from "bleeding over" from one sector to the next.
    c.) provide better data density since the write head can be smaller
    d.) none of the above

13. True or (false:) The platters/disks on ***multiple zone recording*** hard drives must turn faster as the read/write head moves toward the outer tracks. (2 points)

    The rotational speed of the platters/disks remains constant regardless of whether the drive is constant angular velocity or multiple zone recording.

14. The number of sectors per track on a ***multiple zone recording*** hard drive _____ as you go closer to the center of the disk. (2 points)

    a.) increases (b.) decreases c.) stays the same

15. Describe how the FIFO 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 first-in first-out (FIFO) replacement algorithm deletes/clears the line in the cache that has been in the cache the longest, i.e., the one that was loaded before all of the others.

*The table below represents a small section of a cache that uses fully associative mapping. Refer to it to answer questions 16 through 20.*

| Tags (binary) | Word within the block | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| 01101110010110110 | $A0_{16}$ | $01_{16}$ | $62_{16}$ | $00_{16}$ | $BB_{16}$ | $CC_{16}$ | $89_{16}$ | $9A_{16}$ |
| 00110010101011100 | $6B_{16}$ | $71_{16}$ | $D7_{16}$ | $11_{16}$ | $AA_{16}$ | **$DD_{16}$** | $67_{16}$ | $AB_{16}$ |
| 01010110111001011 | $C0_{16}$ | $21_{16}$ | $82_{16}$ | $22_{16}$ | $99_{16}$ | $EE_{16}$ | $56_{16}$ | $BC_{16}$ |
| 11001010010100110 | $3D_{16}$ | $93_{16}$ | $F9_{16}$ | $33_{16}$ | $88_{16}$ | $FF_{16}$ | $45_{16}$ | $CD_{16}$ |
| 01101100110111001 | $E0_{16}$ | $31_{16}$ | $02_{16}$ | $44_{16}$ | $77_{16}$ | $01_{16}$ | $34_{16}$ | $EF_{16}$ |
| 11001010010011101 | $5F_{16}$ | $B5_{16}$ | $2A_{16}$ | $55_{16}$ | $66_{16}$ | $12_{16}$ | $23_{16}$ | $F0_{16}$ |

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

    The processor address is broken into two parts when storing using fully associative mapping. The lower bits identify the word position within the block while the remaining upper bits become the tag. Therefore, since the tag has 17 bits and the word id has 3 bits, the processor's address has 20 bits.

17. What is the block size (in number of memory locations) for the cache shown above? (2 points)

    The block size is determined by the number of bits in the word id, i.e., how many words are in a block. Since three bits are used for the word id, the number of unique word id's for a block is $2^3 = 8$. This makes sense since there are 8 columns in the table in which to store the data of a block.

18. From what address in main memory did the value $DD_{16}$ (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. 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 $0011001010101110 0_2$ and a word id of $101_2$, then the physical address is $0011001010101110 0101_2$ **$= 32AE5_{16}$**.

19. A copy of the data from memory address $6CDCB_{16}$ is contained in the portion of the cache shown above. What is the value stored at that address? (2 points)

    Dividing the physical address $6CDCB_{16} = 01101100110111001011_2$ into its tag and 3-bit word id gives us a tag of $0110110011011100 1_2$ and a word id of $011_2$. Searching through the visible lines shows us that the second line from the bottom has the same tag, i.e., it contains the block which contains the data from the physical address $6CDCB_{16}$. A word id of $011_2$ points us to the data in the fourth column, i.e., the value of **$44_{16}$**.

20. ***If*** the block containing memory address $3C245_{16}$ were to be loaded into the cache described above, what would the ***binary*** tag be? (Note: it is not represented in the data shown above.) (2 points)

    Dividing the physical address $3C245_{16} = 00111100001001000101_2$ into its block id (tag) and 3-bit word id **gives us a tag of $0011110000100100 0_2$** and a word id of $101_2$.

21. True or false: The method used to identify a tag for a block to be stored in an associative cache is the same as that used to identify a subnet ID in a TCP/IP network. (2 points)

    This is **TRUE**. An address, regardless of whether it is an IP address or a memory address can be divided into two parts. The most significant bits identify the group or block in which the address is included while the least significant bits identify the specific item within the group. This is also the same method as is used with chip selects identifying a memory device on a bus.

22. 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 5 instructions? (2 points)

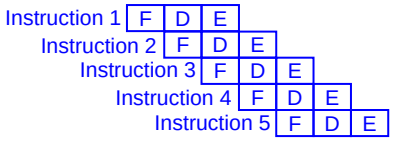
A non-pipelined processor simply executes the instructions one at a time with no overlap. Therefore, the number of cycles equals 3 cycles/instruction times the number of instructions:

number of cycles = 3 × 5 = 15

b. How many cycles would a ***pipelined*** processor take to execute 5 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 5 cycles to execute the 5 instructions.




number of cycles = 2 + 5 = 7 cycles

23. What are the settings of the zero flag, the sign flag, the carry flag, and the overflow flag after a processor performs the addition shown to the right? (4 points)

```
1  1  11
  10100011
+ 10110110
  01011001
```

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

Remember that 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.

24. Immediately after a processor performs a compare, the flags can be checked to see if the two values are equal or to see if one is greater than the other. Which flag is used to check equality? (2 points)

(a.) ZF b.) SF c.) CF d.) OF e.) None of these

A compare instruction such as CMP A, B compares the two values A and B by performing a "virtual subtract" A – B. The settings of the flags determines the relationship of the magnitudes of A and B. The table below shows the settings of the different flags based on A's relationship to B.

| | | | |
|---|---|---|---|
| A = B | ZF = 1 | and | SF = 0 |
| A < B | ZF = 0 | and | SF = 1 |
| A ≤ B | ZF = 1 | or | SF = 1 |
| A > B | ZF = 0 | and | SF = 0 |
| A ≥ B | ZF = 1 or 0 | and | SF = 0 |

25. What is the purpose of the instruction decoder? (2 points)

The instruction decoder receives a single machine code instruction (the numeric version of an assembly language command), runs it through a decoder to identify the instruction, then energizes the appropriate execution circuits in the CPU to have the instruction executed. In other words, it decodes the instructions.

26. Direct Memory Access is an improvement over interrupt driven I/O because (select best) (2 points)

a.) the processor is completely removed from the I/O process
b.) the processor does not have to perform the transfer of data from the I/O device to memory
c.) the processor does not have to check back with the I/O device to see if it is ready
d.) the memory bus will always be available for access by the processor

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

Place your answers in space below:

AX = **$2000_{16}$**

BX = **$3000_{16}$**

CX = **$1000_{16}$**

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

- Swap register values (as in the previous problem)
- Pass parameters to a function
- Store the return address from a function
- Temporary storage when an application runs out of registers

29. Which single gate can be used to quickly compute a parity bit? (2 points)

a.) AND b.) NAND c.) OR d.) NOR e.) XOR f.) A single gate cannot be used

30. Which bitwise operation can be used to clear all bits but the LSB of an integer value to determine if a number is odd? (2 points)

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

31. 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 | $00001111_2$ | **$00110011_2$** |

32. In a 1's complement checksum scheme, the receiving processor adds all of the words of data, then adds the received checksum to the resulting datasum. 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

33. Describe how a CRC is a significant improvement over a datasum-based checksum. (2 points)

With a datasum-based checksum, it is easy to have two errors in equal and opposite directions (i.e., positive and negative) cancel each other out. For example, if the numbers 32 and 9 are contained in a message, and the $2^3 = 8$ bit of 32 gets set and the same bit of 9 gets cleared, their values become 40 and 1 respectively. This is a problem. Unfortunately, the error cancel each other out and the message looks as if it is error free.

A CRC on the other hand uses a system a lot like that of the remainder function. The goal is that a small error creates a large difference in the error checking value. This is much harder to be canceled out by a second error.

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

- A typical long division requires that both the dividend and the divisor are loaded into registers. This is impossible if you're trying to send a message with more than 64 bits. An Ethernet message, for example, could have up to $1500 \times 8 = 12000$ bits.
- An XOR subtraction is much faster than a standard subtraction.

35. True or false: When using a CRC for error checking, the entire transmitted message must be received before computation of the CRC can begin. (2 points)

This is **FALSE**. The pseudo long division is performed as the message is being received.

36. 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 | |
|---|---|---|---|
| ☑ | ☐ | ☐ | Only used within a single network, i.e., doesn't cross into other networks. |
| ☐ | ☑ | ☑ | Uses a datasum-based checksum for error detection. |
| ☑ | ☐ | ☐ | Uses a preamble of alternating 1's and 0's to synchronize all receivers. |
| ☐ | ☑ | ☑ | Only has a header, i.e., no frame trailer is used. |
| ☐ | ☑ | ☐ | Uses a logical address defined by a network administrator for its addressing. |
| ☐ | ☑ | ☐ | Includes a “time to live” field so that it can be removed from the network(s) in case it cannot find its destination. |