Points missed: _____ Student's Name: ___________________________________

Total score: _____ /100 points

East Tennessee State University
Department of Computer and Information Sciences
CSCI 4717 – Computer Architecture
TEST 1 for Fall 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.***
- You may use one sheet of scrap paper that you will turn in with your test.
- **When possible, indicate final answers by drawing a box around them. This is to aid the grader. Failure to do so might result in no credit for answer. Example:**

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

- 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.

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
2^4	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. For each of the following traits of system component design, indicate which implementation method is the best, hardware (HW), software (SW), or firmware (FW), and which is the worst.

Best	Worst	Characteristic	
_HW_	_SW_	Least expensive during development	(2 points)
_SW_	_HW_	Least expensive during manufacturing	(2 points)
_SW_	_HW_	Ease of upgrade	(2 points)
_HW_	_SW_	Reliability	(2 points)

Since the performance characteristics of firmware fall between hardware and software, you usually wouldn't see it in one of these "best"/"worst" ratings.

2. For the following items, indicate whether the characteristic more closely describes a top-down (TD) or a bottom-up (BU) design method. (1 point each)

_BU_ Best for small production quantities, i.e., we're not going to sell a lot of these

_TD_ Doesn't have any excess components, e.g., one goal is to reduce chip count for size

_BU_ New design is closely related to long-standing, existing product line

_TD_ Cost is not a concern

_TD_ There are critical performance goals that need to be met

3. Name an important performance measurement for a video conferencing system? Justify your answer. (3 points)

The primary concern with video conferencing is getting huge amounts of data from one point to the other and vice versa. Therefore, throughput is of utmost concern. The CODEC (compression/decompression algorithm) requires a great deal of processing power, therefore MIPS may also be a good measurement.

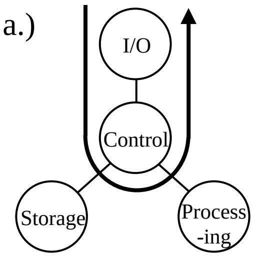
4. Name an important performance measurement for a PDA? Justify your answer. (3 points)

The biggest concerns with PDA's are portability and "rugidization." Portability affects power consumption (directly related to heat dissipation) because of its effects on battery life. Rugidization is important because we need to worry about things like dropping the PDA on the floor and hoping it still works.

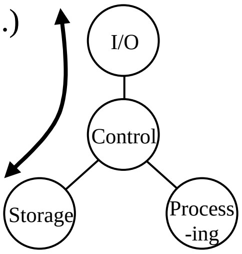
Some people wrote MIPS as a measurement. This is not nearly as important since MIPS usually relates directly to frequency, and higher frequency means higher power consumption. If, however, you said processing power as it relates to things like handwriting recognition, that would be okay. You don't necessarily need MIPS for handwriting recognition if your instruction set is designed to support it specifically. Speed is also found through compact applications.

For problems 5 through 8, identify which of the following system functional diagrams, a, b, c, or d, best describes the operation of the example system or application. (2 points each)

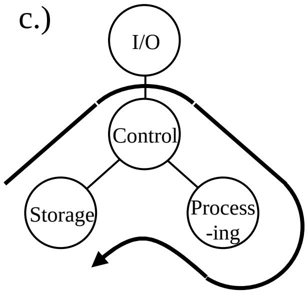
a.)



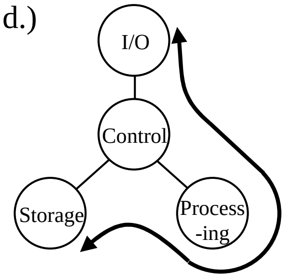
b.)



c.)



d.)



5. An inventory control system checking inventory in and out of a database: __b or d____

6. A hard drive controller buffering data between hard drives and the processor: __b______

   By using the word "buffering," it should have been apparent that no processing would be needed for the data flow. This would have made answer 'd' not as good as 'b'.

7. A lighting controller reading switch inputs to control the lighting in a room: ____a______

8. Hand writing recognition (processor must identify character written before storing it): ____d_____

   The statement "processor must identify character written before storing it" was intended to show that processing would be needed before the character was sent to storage. Therefore, 'b' would not have made as good an answer as 'd'.

9. Name three of the five effects discussed in class that Moore's law has had on the contemporary application of computers. (4 points)

   - Costs have fallen dramatically since chip prices have not changed substantially since Moore made his prediction
   - Tighter packaging has allowed for shorter electrical paths and therefore faster execution
   - Smaller packaging has allowed for more applications in more environments
   - There has been a reduction in power and cooling requirements which also helps with portability
   - Solder connections are not reliable. Therefore, with more functions on a single chip, there are fewer unreliable solder connections

10. Name two of the four solutions discussed in class for improving DRAM transfer rates. (3 points)

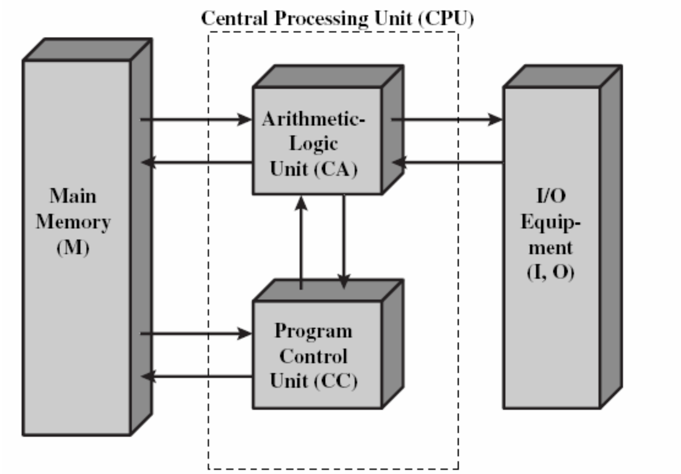
- Increase number of bits retrieved at one time
  - Make DRAM "wider" rather than "deeper"
- Change DRAM interface
  - Add third level of cache
- Reduce frequency of main memory access
  - More complex cache and cache on chip
- Increase interconnection bandwidth
  - High speed buses
  - Hierarchy of buses

11. Circle ***all*** of the following traits that are identified as key concepts of the von Neumann architecture. (3 points)

[✓ a.)] Control unit interprets instructions and causes them to be executed
b.) Memory divided into blocks as a way to group related data or instructions
[✓ c.)] Sequential execution of code except in the case of “jumps”
d.) The use of transistors instead of vacuum tubes for the implementation of circuits
[✓ e.)] Addressable memory contents without consideration to type of content

12. In the space below, sketch the basic von Neumann architecture. Be sure to label each component and include each of the interconnections. (4 points)

Basically, this is Figure 2.1 from our text book.


Central Processing Unit (CPU)
Main Memory (M)
Arithmetic-Logic Unit (CA)
Program Control Unit (CC)
I/O Equip-ment (I, O)

13. Which cache mapping function does not require a replacement algorithm? (2 points)

[✓ a.)] Direct mapping b.) Set associative mapping c.) Fully associative mapping

14. Which cache mapping function does not relate a block address to a specific line or set of lines in the cache? (2 points)

a.) Direct mapping b.) Set associative mapping [✓ c.)] Fully associative mapping

15. Which cache mapping function is the most expensive to implement? (2 points)

a.) Direct mapping b.) Set associative mapping [✓ c.)] Fully associative mapping

16. Which cache mapping function is most likely to thrash, i.e., two blocks contending with each other to be stored in the same line of the cache? (2 points)

[✓ a.)] Direct mapping b.) Set associative mapping c.) Fully associative mapping

17. Which replacement algorithm is ***easier*** to implement with a 2-way set associative cache, Least Frequently Used or Least Recently Used? (2 points)

Least Recently Used – Two-way set associative simply uses a USE bit. When one block is referenced, its USE bit is set while its partner's in the set is cleared.

18. Describe how the Least Frequently Used cache replacement algorithm works. (3 points)

A counter is associated with each line in the cache. Every time a block is loaded, the counter is cleared. The counter is then incremented each time the block is accessed. If the cache needs to decide between multiple lines which one to delete, the one with the lowest "hit" count gets replaced.

19. Assume the processor needs to check the cache for data from the 20-bit memory address $3C45D_{16}$ = 00111100010001011101_2. The cache is a fully associative cache with 2^{10} = 1024 lines and a block size of 2^3 = 8. Derive the tag from the address that will be used to compare to the tags in the cache. (3 points)

In a fully associative cache, no line numbers are used. Therefore, the address is broken into only two parts: the tag and the word id. Since an 8-word block requires 3 word id bits, the lower three bits of the address are the word id. Everything else is the tag.

tag	word id
00111100010001011	101

20. Repeat the previous problem for a direct cache instead of fully associative. (3 points)

In a direct cache, the middle section of the address identifies the line number in which to store the block. Since there are 2^{10} = 1024 lines, then 10 bits are needed to identify the line number. These bits are immediately to the left of the 3 word id bits. Everything else is the tag.

tag	line number	word id
0011110	0010001011	101

The table below represents a small section of a cache that uses direct mapping. Refer to it to answer questions 21 and 22. Assume the processor's memory bus uses 24 bits for an address.

Tag (binary)	Line number (binary)	Word within block				
		00	01	10	11	
100111011010	0101001011	08_{16}	19_{16}	$2A_{16}$	$3B_{16}$	Row a
110110011000	0101001100	$4C_{16}$	$5D_{16}$	$6E_{16}$	$7F_{16}$	Row b
101101010101	0101001101	50_{16}	61_{16}	72_{16}	83_{16}	Row c
010010111101	0101001110	94_{16}	$A5_{16}$	$B6_{16}$	$C7_{16}$	Row d
111001110011	0101001111	$D7_{16}$	$E9_{16}$	FA_{16}	$0B_{16}$	Row e
011001111001	0101010000	$1C_{16}$	$2D_{16}$	$3E_{16}$	$4F_{16}$	Row f
		Col 0	Col 1	Col 2	Col 3	

21. What is the size of the cache in words (bytes)? (2 points)

The size of the cache in words is equal to the number of lines times the number of words per line (i.e., block). Since the number of bits in a line number is 10, the number of lines in the cache is 2^{10}. Since the number of words per block is 2^2 = 4, then the size of the cache is:

$$2^{10} \times 4 = 4K$$

22. A block containing the address $8B453F_{16}$ = $100010110100010100111111_2$ is not contained in the cache. When loaded, which row (a-f) and column (0-3) will its values be stored in? (3 points)

Begin by dividing the address into its tag, line id, and word id. Ten bits are needed for the line id and 2 bits are needed for the word id. This gives us:

tag	line id	word id
100010110100	0101001111	11

Matching the line id with the lines from the above cache segment reveals its destination to be ***row e***. The word id of $11_2 = 3_{10}$ shows us that it will be stored in ***column 3***.

23. Assume a ***2-way set-associative cache*** with 2^{11} = 2048 lines is being used in a system with a 24-bit address using a 16-word block size. If a block containing the address $7BE453_{16}$ = $011110111110010001010011_2$ is stored in the cache, what would the tag be? (2 points)

The set associative cache uses a set id instead of a line id to figure out where to store a block. In other words, a block maps to a specific line within a set instead of mapping to a unique line. And just line the line id, the set id comes from the middle section of the address. Since there are 2^{11} = 2048 lines, and since there are 2 lines per set, then there are 2^{10} = 1024 sets. This means that 10 bits are needed to identify the set number. These bits are immediately to the left of the word id bits. Since a block is 2^4 = 16 words, the right most 4 bits serve as the word id. Everything else is the tag.

tag	line number	word id
0111101111	1001000101	0011

24. Assume a memory access to main memory on a cache "miss" takes 40 ns and a memory access to the cache on a cache "hit" takes 10 ns. If 75% of the processor's memory requests result in a cache "hit", what is the average memory access time? (2 points)

[✓ a.)] 17.5 nS b.) 32.5 nS c.) 10 nS d.) 40.0 nS e.) 7.5 nS f.) 2.5 nS

First, since we know that not all of the memory accesses are going to the cache, the average access time must be greater than the cache access time of 10 ns. This eliminates answers 'c' and 'f'. Similarly, not all accesses are going to the main memory, so the average must be less than 40 ns eliminating 'd' as an answer. For the exact answer, you need to see that 75% of the time, the access will be 10 ns while the rest of the time (25%) the access time will be 40 ns. This gives us the following equation.

$$\begin{aligned}(0.75 \times 10 \text{ ns}) + (0.25 \times 40 \text{ ns}) &= 7.5 \text{ ns} + 10 \text{ ns}\\ &= 17.5 \text{ ns}\end{aligned}$$

25. For each of the following statements, identify which represent write-back (WB) and which represent write-through (WT) cache writing policies? (2 points each)

_WB_ When using this write policy with multiple cache-equipped processors sharing a single RAM, it is possible for RAM to contain an incorrect value for a significant period.

_WT_ This write policy generates higher write traffic to DRAM.

26. Why does a DRAM use column and row addressing? ***Circle all that apply.*** (2 points)

[✓ a.)] It reduces the number of pins needed to interface with the DRAM
b.) It's necessary in order to store data to capacitors.
[✓ c.)] It can be used to speed up transfers of sequential data within the same row by using page mode.
d.) It allows for simultaneous writing data to and reading data from the DRAM.
e.) The cache uses the columns and rows to organize its own data.

27. True or [✓ false:] The processor may still read from the DRAM while the DRAM is being refreshed. (2 points)

28. True or [✓ false:] A time critical interrupt should allow nested interrupts. (2 points)

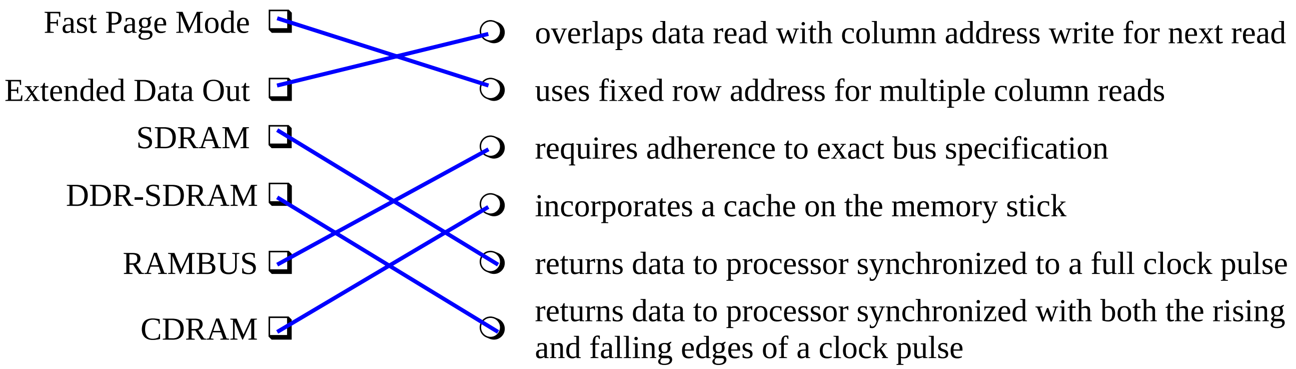
Any process that is critical, especially one that is time critical, should not be interrupted until it is finished. Yes, there might be interrupts that are of a higher priority, and if those exist, the designer should take precautions to allow those interrupts to nest, but in general, no nesting should occur here.

29. Name and describe one of the four types of interrupts, i.e., what types of interrupt triggers are there, that we discussed in class. (2 points)

The following is copied directly from the notes:

- **Program** – Something that occurs as a result of program execution such as illegal instructions, arithmetic overflow, divide by zero, or memory handling error
- **Timer** – Generated by one of the processor's internal timers so that the processor can perform some time-scheduled task
- **I/O** – Generated by an I/O controller to request service from the processor such as keyboard, mouse, NIC, disk drive
- **Hardware failure** – signifies some error condition with the hardware

30. Match the DRAM type with its primary characteristic. (5 points)



Fast Page Mode □
Extended Data Out □
SDRAM □
DDR-SDRAM □
RAMBUS □
CDRAM □

○ overlaps data read with column address write for next read
○ uses fixed row address for multiple column reads
○ requires adherence to exact bus specification
○ incorporates a cache on the memory stick
○ returns data to processor synchronized to a full clock pulse
○ returns data to processor synchronized with both the rising and falling edges of a clock pulse

31. For single error correction, what is the minimum number of check bits required for 12 data bits? (2 points)

a.) 4 [✓ b.)] 5 c.) 6 d.) 7 e.) 8 f.) 9 g.) 10 h.) 11 i.) 12

For single error correction, the number of data bits, M=12, and the number of check bits, K must satisfy the expression $M + K \le 2^k - 1$. The following works this expression out for a few of the values of K to show where the minimum one is.

K = 4: $12 + 4 \le 2^4 - 1$

$16 \le 15$ ← **FALSE!**

K = 5: $12 + 5 \le 2^5 - 1$

$17 \le 31$ ← **TRUE!**

Therefore, 4 is not enough, but 5 is.

32. If it takes a minimum of 9 bits to correct a single error in a 256-bit data word, then how many bits will it take to do single-error correction with double-error detection? (2 points)

a.) 8 b.) 9 [✓ c.)] 10 d.) Cannot be determined e.) None of the above

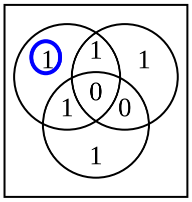
All that's needed to add double-error detection to a single-error correction scheme is a single universal parity bit. Therefore, for the 9 check bits, a tenth bit is all that's needed.

33. What is the total number of ***possible*** syndrome word values for 4 data bits and 3 check bits if we only allow for the cases of no errors or single bit errors? (2 points)

a.) 6 b.) 7 [✓ c.)] 8 d.) 16 e.) 32 f.) 64 g.) 128 h.) 256 i.) 512

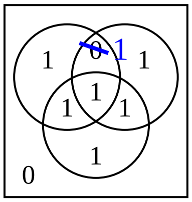
Since the syndrome word is simply the bitwise exclusive OR of the check bits, the syndrome word is also 3 bits long. The total number of possible values of 3 bits is $2^3 = 8$.

34. The graphic to the right depicts the digits of a 4-bit Hamming code where a single bit error has occurred. Circle the bit that has flipped. (2 points)



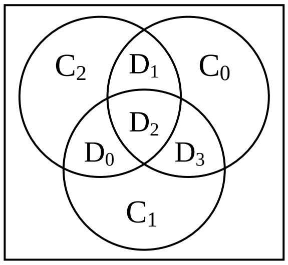
The goal here is to find the intersection of the circles with incorrect (odd) parity. The only circle with odd parity is the top left circle. The only bit in the diagram that is contained in this circle and none of the others is the top left bit.

35. [✓ True] or false: The graphic to the right depicts the digits of a 4-bit Hamming code with parity where a double bit error has occurred. (2 points)



Begin by using the same process from problem 34 to find and correct the bit that is in error. The two top circles have odd parity meaning that the single zero that they share should be in error. Correct this by changing it to a 1. Once the suspected bit is corrected, parity is then computed for the entire diagram including the parity bit in the lower left corner. 1 + 1 + 1 + 1 + 1 + 1 + 0 = 7 which is odd. Therefore, a double error has occurred.

36. Assume the figure below shows the grouping of four data bits, D_0, D_1, D_2, and D_3, with three check code bits, C_0, C_1, and C_2. In the space provided, write the expressions used to generate the check code bits for this system. (4 points)



$C_2 = D_0 \oplus D_1 \oplus D_2$

$C_1 = D_0 \oplus D_2 \oplus D_3$

$C_0 = D_1 \oplus D_2 \oplus D_3$