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, 2003

Section 001

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 numeric equations is fine too.

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."

DEC - Decrement

Usage: DEC dest

Modifies flags: AF OF PF SF ZF

Description: Unsigned binary subtraction of one from the destination.

INC - Increment

Usage: INC dest

Modifies flags: CF AF OF PF SF ZF

Description: Adds one to destination unsigned binary operand.

Jxx - Jump Instructions Table

Mnemonic	Meaning	Jump Condition
JE	Jump if Equal	ZF=1
JG	Jump if Greater (signed)	ZF=0 and SF=OF
JGE	Jump if Greater or Equal (signed)	SF=OF
JL	Jump if Less (signed)	SF != OF
JLE	Jump if Less or Equal (signed)	ZF=1 or SF != OF
JMP	Unconditional Jump	unconditional
JNB	Jump if Not Below	CF=0
JNE	Jump if Not Equal	ZF=0
JNG	Jump if Not Greater (signed)	ZF=1 or SF != OF
JNL	Jump if Not Less (signed)	SF=OF
JZ	Jump if Zero	ZF=1

MOV - Move Byte or Word

Usage: MOV dest,src

Modifies flags: None

Description: Copies byte or word from the "src" operand to the "dest" operand.

NOT - One's Compliment Negation (Logical NOT)

Usage: NOT dest

Modifies flags: None

Description: Inverts the bits of the dest operand forming the 1s complement.

POP - Pop Word off Stack

Usage: POP dest

Modifies flags: None

Description: Transfers word at the current stack top (SS:SP) to the destination then increments SP by two to point to the new stack top. CS is not a valid destination.

PUSH - Push Word onto Stack

Usage: PUSH src

Modifies flags: None

Description: Decrements SP by the size of the operand (two or four, byte values are sign extended) and transfers one word from source to the stack top (SS:SP).

SAL/SHL - Shift Arithmetic Left / Shift Logical Left

Usage: SAL dest,count SHL dest,count

Modifies flags: CF OF PF SF ZF (AF undefined)

Shifts the destination left by "count" bits with zeroes shifted in on right. The Carry Flag contains the last bit shifted out.

SAR - Shift Arithmetic Right

Usage: SAR dest,count

Modifies flags: CF OF PF SF ZF (AF undefined)

Shifts the destination right by "count" bits with the current sign bit replicated in the leftmost bit. The Carry Flag contains the last bit shifted out.

Answer questions 1 through 9 based on the following settings of the 8086 registers.

AX = 2345h	SP = 0123h	CS = 1000h
BX = 8080h	IP = 1234h	DS = 2000h
CX = 0055h	DI = 2345h	SS = 3000h
DX = 9876h	SI = 3456h	ES = 4000h

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

   DH is the upper half (byte) of DX. Therefore, the value in DH is **98h**.

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

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

   DS = 2000h → the segment address is 2000**0**h (notice the added zero)

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

   ```
    20000
   + 2345
   -------
    22345
   ```

   Therefore, the physical address pointed to by DS:DI (2000:2345) is **22345h**.

3. 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)

   TRUE: The physical address of the next instruction to execute is determined from the values contained in CS and IP (CS:IP). Since both of those values are present above, then we can calculate the physical address of the next instruction to execute.

4. True or false: The physical address of the top of the stack can be calculated from the above data? (2 points)

   TRUE: The physical address of the top of the stack is determined from the values contained in SS and SP (SS:SP). Since both of those values are present above, then we can calculate the physical address of the top of the stack.

5. What is the value of SP after the execution of the instruction **`POP AX`**? (2 points)

   If you go to the list of instructions on the previous page, you'll see that executing a POP instruction "increments SP by two to point to the new stack top". Therefore, add two to SP to get **0125h**.

6. What is the value of SS after the execution of the instruction **`POP AX`**? (2 points)

   If you go to the list of instructions on the previous page, you'll see that executing a POP instruction does not affect SS. Therefore, SS remains the same as shown in the above table, i.e., **3000h**.

7. Assume that the instruction **DEC CH** is executed. How would the following flags be set? ***Write "N/A" if the flag was not affected.*** (3 points)

ZF = ______ CF = ______ SF = ______

Referring to the list of instructions on the second page, we see that the DEC subtracts 1 from the operand and affects the flags AF, OF, PF, SF, and ZF. (Note that CF is not affected, so we automatically put N/A into the CF slot.) Since CH equals 00 before the instruction is executed, it equals -1 or FF after the instruction is executed. Therefore, since it is not equal to zero, the zero flag (ZF) is cleared or false. Since it is a negative value (the MSB of the result is set to 1), then the sign flag (SF) equals 1.

ZF = __0__ CF = __N/A__ SF = __1__

8. Assume that the instruction **SAL AH,2** is executed. How would the following flags be set? ***Write "N/A" if the flag was not affected.*** (3 points)

ZF = ______ CF = ______ SF = ______

Referring to the list of instructions on the second page, we see that the SAL AH,2 will shift the 8-bit AH register 2 places to the left with zeros shifted in on the right. In addition, the carry flag contains the last bit shifted out. So what does AH equal before the instruction?

AH = 23h = 00100011b

After the instruction, the two zeros on the left side are shifted out, (the last one remaining in the carry flag, CF) and zeros are shifted in from the right.

new AH = 10001100b = 8Ch

SAL modifies the flags CF, OF, PF, SF, and ZF. Since the new value of AH is not equal to zero, the zero flag (ZF) is cleared or false. Since it is a negative value (the MSB of the result is set to 1), then the sign flag (SF) equals 1.

ZF = __0__ CF = __0__ SF = __1__

9. Assume that the instruction **SAR BL,3** is executed. What would the new value of BL be? (2 points)

Referring to the list of instructions on the second page, we see that the SAR BL,3 will shift the 8-bit BL register 3 places to the right with the most significant bit (the sign bit) replicated to fill in the bits left empty on the left. So what does BL equal before the instruction?

BL = 80h = 10000000b

After the instruction, the three zeros on the right side are shifted out, and the one in the most significant position is duplicated three times to fill in for the blank bits created during the shift.

new BL = 11110000b = F0h

10. List the two benefits of segmented addressing. (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

11. List one benefit of using assembly language. (2 points)

- full access to all processor resources
- the ability to make much faster code
- the ability to make far more compact code

12. List one drawback of using assembly language. (2 points)

- complicated to learn and use
- hard to debug
- more time consuming to write
- harder to decipher if the programmer is unfamiliar with a particular program

13. For each of the following registers, identify how many bits they contain. (4 points)

BP = __**16**__ bits IP = ___**16**__ bits DI = ___**16**__ bits DL = ___**8**__ bits

14. What character/symbol is used to indicate the start of a comment in assembly language for the assembler we used in class? (2 points)

A semicolon (;)

15. For each of the assembly language commands below, what is the boolean value for the active low signals ^MRDC, ^MWTC, ^IORC, and ^IOWC. (4 points)

Remember that ^MRDC goes low when reading from memory, ^MWTC goes low when writing to memory, ^IORC goes low when reading from an I/O port, and ^IOWC goes low when writing to an I/O port.

	^MRDC	^MWTC	^IORC	^IOWC
`mov ah,[5674h]`	X			
`in bh,1234h`			X	
`mov [ax],bx`		X		
`out 4af5h,bh`				X

16. Assume the register BX contains the value 2000h and the table to the right represents the contents of a short portion of memory. Indicate what value AL contains after each of the following MOV instructions. (2 points each)

Address	Value
DS:2000	17h
DS:2001	28h
DS:2002	39h
DS:2003	4Ah
DS:2004	5Bh
DS:2005	6Ch

MOV AL, DS:[BX] AL = **17h**

MOV AL, DS:[BX+1] AL = **28h**

MOV AX, BX AL = **20h**

MOV AX, 2003 AL = **20h**

17. If a processor takes 3 cycles to execute any instruction (fetch, decode, execute), how many cycles would a non-pipelined processor take to execute 5 instructions? (3 points)

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

number of cycles = 3 x 5 = 15 cycles

18. If a processor takes 3 cycles to execute any instruction (fetch, decode, execute), how many cycles would a pipelined processor take to execute 5 instructions? (3 points)

A pipelined processor overlaps 2 cycles for each instruction. Therefore, it will take 2 cycles to fill the pipeline, then one cycle per instruction to execute each one.

number of cycles = 2 + 5 = 7 cycles

19. On an 80x86 processor with 32 address lines, what is the maximum number of I/O ports? (2 points)

Regardless of the number of address lines available on the processor, 80x86 architectures ALWAYS have 16 address lines going to the I/O ports. Therefore, there is always a maximum of

2^{16} = 65,536 (64K) ports

20. True or false: Interrupt-driven I/O is faster than non-interrupt-driven I/O because the processor does not need to constantly be checking with the device to see if it needs serviced. (2 points)

TRUE: Interrupt driven I/O relies on the device to interrupt the processor when it needs serviced. Therefore, this burden is removed from the processor.

21. True or false: DMA is faster than non-DMA because the data does not have to be read from the I/O device into the processor then copied to memory. It is simply copied directly from the device to memory. (2 points)

TRUE: In a typical non-DMA data transfer, the data is read by the microprocessor so that it can be stored into memory. In this case, every piece of data must pass through the processor. In DMA, the DMA controller takes care of the transfer directly to memory and only interrupts the processor to tell it the transfer is complete.

22. Of the following jump instructions, indicate which ones will jump to the address LOOP, which ones will simply execute the next address (i.e., not jump), and which ones you don't have enough information to tell.

Instruction		Current Flags	Jump to LOOP	Not jump to LOOP	Cannot be determined	
JE	LOOP	SF=0, ZF=1, OF=0	☒	□	□	(2 points)
JNB	LOOP	SF=1, ZF=0, CF=0	☒	□	□	(2 points)
JMP	LOOP	SF=0, ZF=0, OF=0	☒	□	□	(2 points)
JGE	LOOP	ZF=0, SF=0, OF=1	□	☒	□	(2 points)

23. Using an original value of 11000011_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
11000011_2	AND	00001111_2	**00000011_2**
11000011_2	OR	00001111_2	**11001111_2**
11000011_2	XOR	00001111_2	**11001100_2**

24. Assume AX=1000h, BX=2000h, and CX=3000h. After the following code is executed, what would AX, BX, and CX contain? (3 points)

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

Place your answers in space below:

AX = **1000h**

BX = **3000h**

CX = **2000h**

For the next 5 problems, indicate whether the statement describes DRAM or SRAM

25. __**SRAM**_____ is the fastest. (2 points)

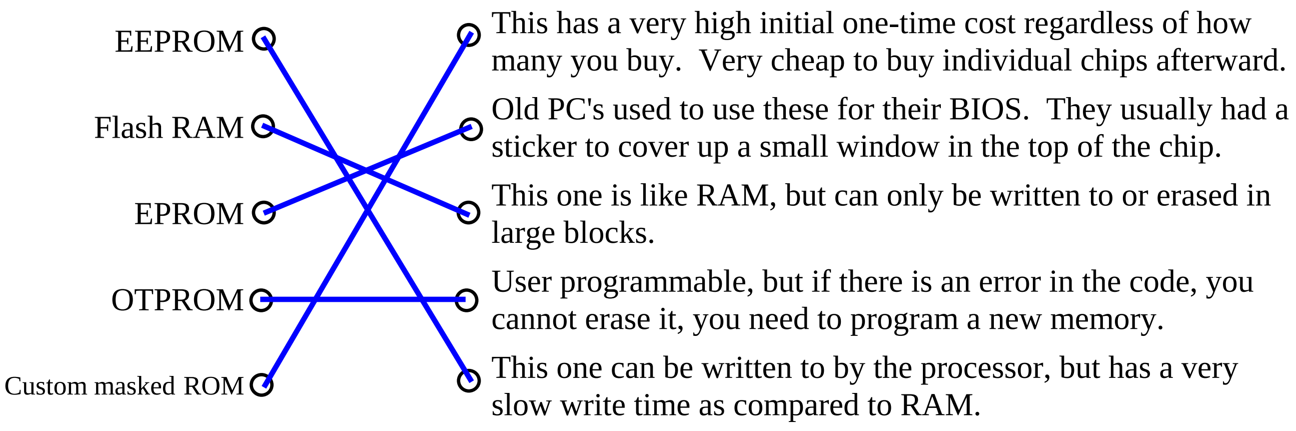
26. __**DRAM**_____ is the cheapest. (2 points)

27. __**SRAM**_____ is the uses flip-flops or D-latches to store data. (2 points)

28. __**DRAM**_____ is used for main memory. (2 points)

29. __**SRAM**_____ is used for cache memory. (2 points)

30. Draw a line between the memory type on the left and its most appropriate characteristic on the right. (1 point each)



EEPROM

Flash RAM

EPROM

OTPROM

Custom masked ROM

This has a very high initial one-time cost regardless of how many you buy. Very cheap to buy individual chips afterward.

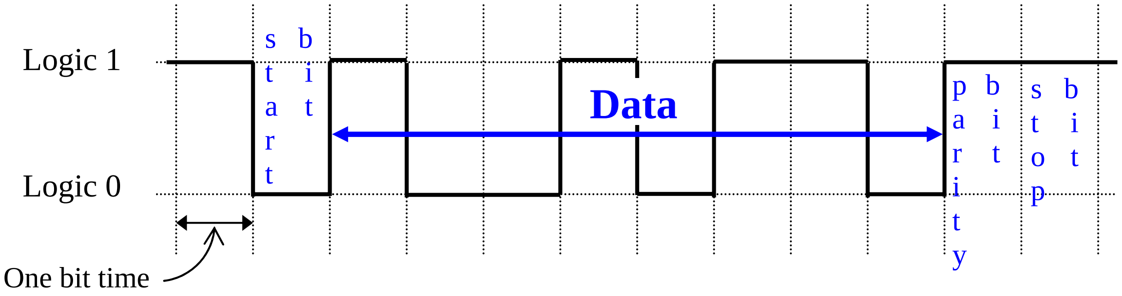
Old PC's used to use these for their BIOS. They usually had a sticker to cover up a small window in the top of the chip.

This one is like RAM, but can only be written to or erased in large blocks.

User programmable, but if there is an error in the code, you cannot erase it, you need to program a new memory.

This one can be written to by the processor, but has a very slow write time as compared to RAM.

Questions 31 through 32 are based on the following RS232 serial signal sent with 8 data bits.



31. What is the binary value being transmitted in this signal? (4 points)

    The bits are received as 10010110, but the value is being sent least significant bit first. Therefore, the binary value being transmitted is **01101001**.

32. Assuming the parity had been set to even, indicate whether an error has occurred. (2 points)

    a.) No error b.) Error c.) Cannot be determined

    Remember that the parity is a count of the number of ones for both the data and the parity bit. In this case, the data has 4 ones and the parity bit is a 1. Therefore, there are 5 ones → an odd number. If the parity has been defined as even, then an error has occurred and **the answer is B**.

33. True or false: a parity bit can only detect if an odd number of bit errors has occurred. (2 points)

    TRUE: If an even number of errors has occurred, then the parity bit appears to be correct.

34. What is the maximum number of devices that can be connected to a single RS232 serial connection? (2 points)

    Two: RS232 is a point-to-point protocol only allowing for two devices total. Therefore, only a single RS232 device can be connected to each RS232 port of your PC (the PC being the second device in the connection).