Midterm Exam #2 Answer Key

Name:_______________________________________

Student ID #:__________________________________________

I have read and understand Washington State University’s policy on academic dishonesty and cheating.

Signed:__________________________________________

Problem 1) Consider the following fragment of assembly code:

```assembly
src db 1,2,3,5,7,11,13,17,19,23,29,31
dst db 06Fh,06Dh,06Bh,069h,057h,045h,
       043h,041h,03Dh,03Ah,038h,034hh
count dw 4
.label1:
    add count,3   17+6+8
    mov cx,count  8+6+4
    xor bx,bx     3
    mov di,offset dst 4
    mov dh,src[bx+1] 8+9+4
    xor dh,dst[bx] 9+9+4
    mov [bx+di],dh 9+8+4
    inc bx         2
    loop label1    17/5

label2:
```

a) (10 Points): What will be the value in DX when control reaches label2?

______52??h___________

b) (10 Points) How many clock cycles will this code take to execute on an 8088? Instruction execution times are given in Appendix B, Page 641 of the text.

31+18+3+4 + 7*(21+22+21+2+17) + 6*17 + 5

_____744___________

c) (5 Points) Assuming a CPU clock speed of 8.0 Mhz, how long will this code take to execute?

93 usec
**Problem 2)** Consider the following logical diagram of a decoder.

a) (15 points) Fill in the following table with 1’s, 0’s and X’s to indicate the partial state of this decoder’s truth table.

<table>
<thead>
<tr>
<th>(2A_1)</th>
<th>(2A_0)</th>
<th>(1E)</th>
<th>(2E)</th>
<th>(2Y_3)</th>
<th>(2Y_2)</th>
<th>(2Y_1)</th>
<th>(2Y_0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>X</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>X</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>X</td>
<td>X</td>
<td>X</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

b) (5 points) What is a common identity of this kind of device in the 74LSxxx nomenclature?

_______74LS139_________

c) (5 points) If a 74LS128 is commonly described as a 3 x 8 decoder, what would the description of this device be?

__A pair of 2 x 4 decoders______________
Problem 3) The following is a brief section of an assembly language program.

```
label1:   add    ax, [bx]
           dec    cx
           jnz label1

label2:
```

a) (10 points) Assemble the assembly language code into its corresponding machine code. The address for ‘label1’ in memory will be 3E20:032Ch. Your answer may be in either binary or hex.

**ANSWER:**

```
0000 0011 1000 0111     or  03 87h
0100 1001        or  49h
0111 0101 0010 1100 0000 0011  or  75 2C 03h
```

b) (5 points) At what address in memory will ‘label2’ reside? Please write your answer in CS:IP format.

**ANSWER:** Since we are still in the same code segment, the CS component is unchanged. The number of instruction bytes preceding ‘label2’ is 6. 6 + 032Ch = 0332h.

**Hence:** 3E20:0332h
Problem 4) (40 Points)
You are designing an 8088 based computer for a control application, and are designing part of the memory section. For availability purposes, the system was designed with four 2KB 8-bit RAMs & one 8KB 8-bit ROM—are identical in design to the 62256 memory chips except for the difference in capacity and the corresponding addressing.

Draw the logic necessary to decode the correct addresses and hook up all necessary address, data, and control signals to the memory chips. You may include either 74LS138 or 74LS139 decoders in your solution. Be sure to correctly distinguish I/O cycles from memory cycles. Remember, the CS, RD, and WR inputs to the device are active low signals.
Problem 5) (40 Points) Write a procedure in assembly language to count the number of times that EACH of the DOS Function Calls (INT 21h) are performed. You are to “hook” into the existing interrupt so that the operation of the DOS function call occurs prior to your update of the various counters.

Include the following:

- Data segment definition for “fnccounts” as a vector where the total count for each DOS function can be stored.
- lines of code at the beginning of the main routine required to modify the interrupt vector table to point to your procedure
- lines of code at the end of the main routine for restoring the interrupt vector table to its original status.
- The complete requested procedure

You may assume the following:

- There are memory locations labeled “intaddr” where the code segment and offset of the original interrupt service routine may be stored.
- Each DOS function will not be called more than $2^{16}-1$ times by the main routine.
- Calls to INT 21h for the getting/setting of the interrupt vector table at the beginning and end of the main routine may be ignored in the counts.

Answer: …

```
  fnccounts dw 255 dup (?)
  ...
  mov ah,35h    ; get int vector
  mov al,21h    ; int vector index
  int 21h
  mov intaddr,ds   ; save cs of int vector
  mov intaddr+2,dx   ; save ip of int vector
  mov es, cs    ; load current code seg.
  mov bx, offset fcncntr  ; load offset of procedure
  mov ah, 25h    ; set int vector
  int 21h
  ...

  fcncntr proc far
  pushf
  call word ptr intaddr  ; perform call to original ISR
  push bx
  xor bx,bx
  mov bl,ah    ; get DOS function number
  add bx,bx    ; set up offset
  inc fnccounts[bx]  ; count the function call
  pop bx
  iret     ; return to interrupted routine

  fcncntr endp
```