16.070 Exam #2

April 10, 2002

Answer the questions in the spaces provided on the question sheets. If you run out of room for an answer, continue on the back of the page.

Name:  _______________________________________________________________________________

**Question 1: (14%)**

You are given the following 8 bit hex number: BA\textsubscript{16} (like a sheep says)

A) Convert this number to binary.

\[1011\ 1010_2\]

B) Left shift the binary number by 2.

\[1110\ 1000_2\]

C) Right shift the result from (B) by 3.

\[0001\ 1101_2\]

D) Convert the result from (C) to octal.

\[35_8\]

E) What is 16\textsubscript{10} * 16\textsubscript{10} in base 16?

\[100_{16}\]

F) Does FAD\textsubscript{16} have even parity?

\[\text{FAD}_{16} = 1111\ 1010\ 1101_2; \text{ There are 9 "one's", which is odd parity, so the answer is NO.}\]

G) Explain how you would set bit number 3 to a "0" in any given byte. Assume the other bits have to remain unchanged. Answer by explaining the operations, or by writing a line of code.

Use the logical AND operator. AND the byte with a mask of 1111 0111\textsubscript{2}. When you AND something with a 0, the result is 0. When you AND something with a 1, the result will be whatever the original bit value was, so the remaining bits will remain unchanged.
**Question 2: (12%)**

A) Design the continuous state transition matrix equation that describes the motion of the minute hand of a circular wall clock. Recall that $\dot{X} = AX + Bu$. Make sure you fill in the A and B matrices.

*Hints:*  
- Use two states... Make one of them angular position.  
- There are no inputs ($u$) to the system.

\[
X = \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix}
\]
\[
\dot{X} = \begin{bmatrix} \dot{\theta} \\ \dot{\dot{\theta}} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]

B) Discretize the continuous equation derived in part A. You may consider $\Delta t$ to be the amount of time elapsed between $X_N$ and $X_{N+1}$. Express your final answer by filling in values or variables for the matrices: $X_{N+1}$, $X_N$, A, B, and u. Place the matrices in a discretized state transition equation.

\[
\theta_{N+1} = \theta_N + \Delta t \dot{\theta}_N
\]
\[
\dot{\theta}_{N+1} = \dot{\theta}_N
\]

\[
\begin{bmatrix} \theta_{N+1} \\ \dot{\theta}_{N+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_N \\ \dot{\theta}_N \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\]
Question 3: (13%)

A) Draw a truth table for an OR gate. Now, draw another truth table to prove that the circuit diagram below simplifies to \( C = A \ OR \ B \).

*Hint: all 3 gates are of the NAND variety.*

\[
\begin{array}{c|c|c|c|c|c|c}
A & B & \text{A OR B} & \text{A NAND A} & \text{B NAND B} & \text{(A NAND A) NAND (B NAND B)} & \text{A OR B} \\
0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 \\
1 & 0 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 0 & 0 & 1 & 1 \\
\end{array}
\]

B) The 8 bit character variable MysteryByte, expressed in bits, is \([u \ v \ w \ 0 \ 0 \ x \ y \ z]\). We want to place the two bit number represented by \([v \ w]\) in the integer variable BugBits. Describe your approach and write a single line of C code followed by a semicolon to accomplish this.

\[
\text{BugBits} = ((\text{MysteryByte } <<1) >>6); \\
or \\
\text{BugBits} = (\text{MysteryByte} \ & \ 96) / 32; \\
or \\
\text{BugBits} = (\text{MysteryByte} \ * \ 2) / 64;
\]

What would the be the value of the C expression \((\text{BugBits} \mid 10)\)?

\[
\text{BugBits is } 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ v \ w_2 \\
\text{Logical OR BugBits with } 10_{10}:
0 \ 0 \ 0 \ 0 \ 0 \ v \ w_2 \ \text{OR} \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0_2 = 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ w_2 = 10_{10}+w
\]
**Question 4: (18%)**

You are writing a program for a telescope control system. The compiler you choose uses 2 bytes for integers and IEEE standards for float and double as shown in the table below:

<table>
<thead>
<tr>
<th>Type</th>
<th>Smallest positive value</th>
<th>largest value</th>
<th>precision</th>
<th>size</th>
</tr>
</thead>
<tbody>
<tr>
<td>float</td>
<td>$1.17 \times 10^{-38}$</td>
<td>$3.40 \times 10^{38}$</td>
<td>6 digits</td>
<td>32 bits</td>
</tr>
<tr>
<td>double</td>
<td>$2.22 \times 10^{-308}$</td>
<td>$1.79 \times 10^{308}$</td>
<td>15 digits</td>
<td>64 bits</td>
</tr>
</tbody>
</table>

For the following data, choose the most efficient data type to use, i.e. choose a data type that will hold the information but does not waste memory. Choose one data type from char (8 bits), int, float, double or Boolean (1 bit) for each of the following purposes:

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>store the counting variable for a loop that runs 350 times.</td>
<td>int (char can represent only up to 256 values)</td>
</tr>
<tr>
<td>store the status of a protective cap that protects the lens. Possible</td>
<td>char (Boolean can represent only up to 2 values)</td>
</tr>
<tr>
<td>values are: open, closed, and in-transition.</td>
<td></td>
</tr>
<tr>
<td>store the status of the lock that locks the protective cap once it is</td>
<td>Boolean</td>
</tr>
<tr>
<td>closed. Possible values are: locked and unlocked.</td>
<td></td>
</tr>
<tr>
<td>store the pitching angle for a telescope. The telescope can pitch from</td>
<td>char, since there are 200 possible values to represent</td>
</tr>
<tr>
<td>–25 to 25 in 0.25 increments.</td>
<td></td>
</tr>
<tr>
<td>store data a light sensor will detect and transmit to the program in</td>
<td>Char, since 00-AF\textsubscript{16} is 176 values to represent</td>
</tr>
<tr>
<td>HEX (base 16) format from 00 to AF.</td>
<td></td>
</tr>
</tbody>
</table>
Question 5: (15%)

A) Circle the correct answer. After the definition:

```c
struct foo
{
    int part1;
    int part2;
};
```

[a.] foo is a new data type which can be used to declare variables as pointers to a structure  
[b.] foo is a new data type which can be used to declare variables of this type of structure  
[c.] foo is a variable which can hold a pointer to a structure  
[d.] foo is a variable which can hold a structure  
[e.] none of the above

B) Given the structure definition below, declare a structure variable "p" of type "part." Write a `scanf()` statement which reads in the "quantity" member of "p."

```c
struct part
{
    int partNum;
    char description[30];
    int quantity;
    char type;
};
```

```c
struct part p = {0, '\0', 0, '0'};
scanf("%d", &p.quantity);
```

C) Declare a pointer to the structure variable "p" declared in part B). Then use this pointer to set the "partNum" member to the number 5.

```c
struct part *ptr_p = &p;
ptr_p -> partNum = 5;
```

or

```c
(*ptr_p).partNum = 5;
```
Question 6: (10%)  

On the lines below, identify 3 of the 5 highlighted portions of the Handyboard schematic shown on the next page.

A) _______data bus__________

B) _______static RAM, or memory__________

C) _______NAND gate__________ (what could this chip do?)

D) _______LED__________

E) _______Serial comm port__________

Fill in the blanks:

F) This register receives the result of ALU operations: ______Accumulator__________

G) Name the three busses found inside a CPU________Address__________,
    ______Data__________, ______Control__________.

H) Identify the component(s) that provide(s) fast temporary storage inside the CPU
    ______Registers (or RAM)__________

I) A diode is an example of what type of junction? ______p-n__________

J) A computer system has a CPU with 32 address lines. What is the maximum memory capacity of this system? $2^{32}$
Question 7: (20%)

Write a program function, `make_identity`, that will generate an N x N identity matrix ("ones" along the diagonal and "zeros" elsewhere). You are given the following code segment with which to interface. **Fill in the blanks** below and **write the function** to create the matrix.

The output of your final program should be

```
1 0 0 0
0 1 0 0
0 0 1 0
0 0 0 1
```

Note: Your `make_identity` function should work for a square matrix of any size N.

```c
#include <stdio.h>
#define N 4

/* Function prototype */
void make_identity(int size, int *ptr_matrix);

/* MAIN */
int main(void)
{
    int identity_matrix[N][N] = {0}; /* array to hold identity matrix */
    int i = 0, j = 0; /* iteration variables */
    make_identity(N, &identity_matrix[0][0]);
    /* display matrix */
    for(i=0; i<N; i++)
    {
        for(j=0; j<N; j++)
        {
            printf("%d ",identity_matrix[i][j]);
        }
        printf("\n");
    }
    return 0;
}

/* Write your "make_identity" function here! */
/* make_identity */
void make_identity (int size, int *ptr_matrix)
{
    int row = 0, col = 0;
    for(row=0; row<size; row++)
    {
        for(col=0; col<size; col++)
        {
            if(row == col)
            {
                *(ptr_matrix + row*size + col) = 1;
            }
            else
            {
                *(ptr_matrix + row*size + col) = 0;
            }
        }
    }
    return;
}
```

Page 8 of 9
**Question 8: (6%)**
You have just measured 5 points of sampled analog data. Each measurement was taken 1 second after the previous measurement. Every measurement is 3 Volts. Draw the sampled data points and two curves on the axes below that could have resulted in the data provided.

This is an example of aliasing.