Embedded System Course – ECOM 3322  
By Eng. Husam Y. Alzaq  
Homework (2)  
(Due to Saturday, the 14th of April, 2010)

**Question 1:**
A small sensor is used to monitor some falcons in their environments. For the designed sensor, the scientists want to have one input X and one output Z such that Z is 1 if the last three inputs are the completely different (010 or 101) and 0 if the last three inputs are partial similar (000, 001, 011, 100, 110, 111). For example, if the inputs are 00100100101 then the outputs would be 010010011 (the first 0 is the first bit). The output will be sent via external wireless model. Assume that there is one input per clock cycle.

**What are the functional requirements and non-functional requirements for the system?**

**Draw a state diagram for your finite state machine.**

**Question 2:**

**TUTMAC Application Description**
In this problem, a custom WLAN terminal application will be built. The application is a custom Medium Access Control (TUTMAC) protocol of a proprietary WLAN (TUTWLAN). The TUTMAC protocol should be modeled with UML in order to meet the TUT-Profile.
The Tutmac_Protocol class is the top-level class of the application. The class is composed of five classes. The Management, RadioManagement and RadioChannelAccess classes are functional components containing behavior, and can be instantiated as application processes. The UserInterface and DataProcessing classes are structural components having composite structure without behavior.

**Draw the TUTMAC class diagram**

**Question 3:**
The Microwave oven has a **Run push button** to start (apply the power) and a **Timer** that determines the cooking period. Cooking can be interrupted at any time by **opening the oven Door**. After closing the Door cooking continues. Cooking is terminated when the **Timer elapses**. When the Door is opened, a Lamp inside the oven is switched on; when the Door is closed the Lamp goes off.
The control system has the following inputs:
1. **Run push button** - when activated starts cooking,
2. **Timer** - while this runs the cooking is enabled (time a cooking run up to 59 min and 59 seconds in 1-s increment),
3. **Door sensor** - can be true (door closed) or false (door open).

And the following outputs:
1. **Power** - can be true (power on) or false (power off),
2. **Lamp** - can be true (lamp on) or false (lamp off).

The knobs to set the power level and timeout values are irrelevant for the control algorithm structure. The behavior of the microwave oven control is determined by the Run push button, Timer and Door sensor.

1. **Draw the class diagram for the classes required in a basic oven.**
2. **Draw the finite state machine for the cooking process**
3. **Write the C code for it.**

**Question 4:**

For the basic block given below, draw the data flow graph for that form.

\[
\begin{align*}
\lambda_1 &= \frac{-b + \sqrt{b^2 - 4ac}}{2a} \\
\lambda_2 &= \frac{-b - \sqrt{b^2 - 4ac}}{2a}
\end{align*}
\]

**Question 5:**

For the basic block given below,
A. draw the data flow graph for that form.
B. determine the minimum number of registers required to perform the operations when they are executed in the order shown in the code. (You can assume that all computed values are used outside the basic blocks, so that no assignments can be eliminated.)
C. determine the order of execution of operations that gives the smallest number of required registers.

\[
\begin{align*}
w &= a - b + c; \\
x &= w - d *4; \\
y &= x - 2*a; \\
w &= a + b - c; \\
z &= y + d; \\
y &= b * c * d;
\end{align*}
\]