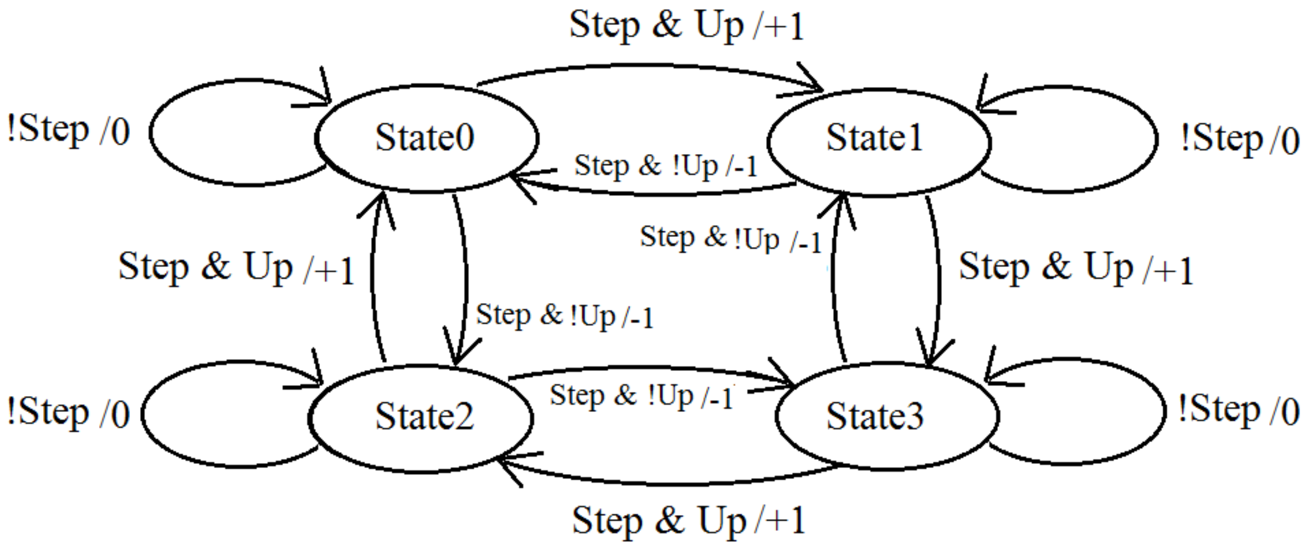
ECE 241 **Home Study 3** **Fall 2016**

1) Draw the State Transition Diagram for the program in Appendix A.

2) Write a system that will implement the following STD. The system should consist of an "enum" defining the states, and a function to be called every 100 milliseconds (the prototype of which is shown below). The STD shows the transitions and the inputs that cause them and what value should be returned in each case.

```
        int SystemNextState( int Step, int Up );
```



Appendix A: Code for Part 1.

```
// Variable of actions and states
// for Serial Transmitter.
enum Actions {
        Nothing, RTS_Count,
        StartBit, Send, StopBit,
        Reset
};
enum States  { Idle, Pause, Transmit, Stop1, Stop2 };
int State = Idle;

// Transmit variables
int Count = 0;
char OutputByte = 0;

// Function that manages the states.
int NextState(int Start, int CTS, int CountEqual0)
{
        int ReturnValue = Nothing; // Default action is nothing.
        switch (State)
        {
        case Idle:      // Waiting for data
                if (Start) // Data Ready
                {
                        State = Pause;  // Set RTS and pause.
                        ReturnValue = RTS_Count;
                }
                break;
        case Pause: // Waiting on Clear To Send
                if (!CTS) // Clear To Send low
                {
                        State = Transmit; // Go to transmit
                        ReturnValue = StartBit; // Send Start bit.
                }
                break;
        case Transmit:  // Transmitting
                if (CountEqual0) // Done transmitting
                {
                        State = Stop1;  // Go to First Stop Bit.
                        ReturnValue = StopBit;
                }
                else
                        ReturnValue = Send; // Send next bit.
                break;
        case Stop1:  // First Stop Bit.
                State = Stop2;
                break;
        case Stop2:  // Second Stop bit.
                State = Idle;
                ReturnValue = Reset; // Clear RTS.
                break;

        } // end of state switch

        return ReturnValue;

} // End of NextState
```