Laboratory assignment five
A loose approximation of cardiac pacemaker design
EECS 203

Lab due on 14 May
Prepared by Robert Dick after interviewing Alan Sahakian

Please carefully review lecture four before starting this assignment. If you make catastrophic wiring mistakes, this could result in be exploding integrated circuits sending chunks of plastic into your forehead.

In this laboratory assignment, you will be building an approximation of the control finite state machine for a cardiac pacemaker.

Please show your work in your lab report.

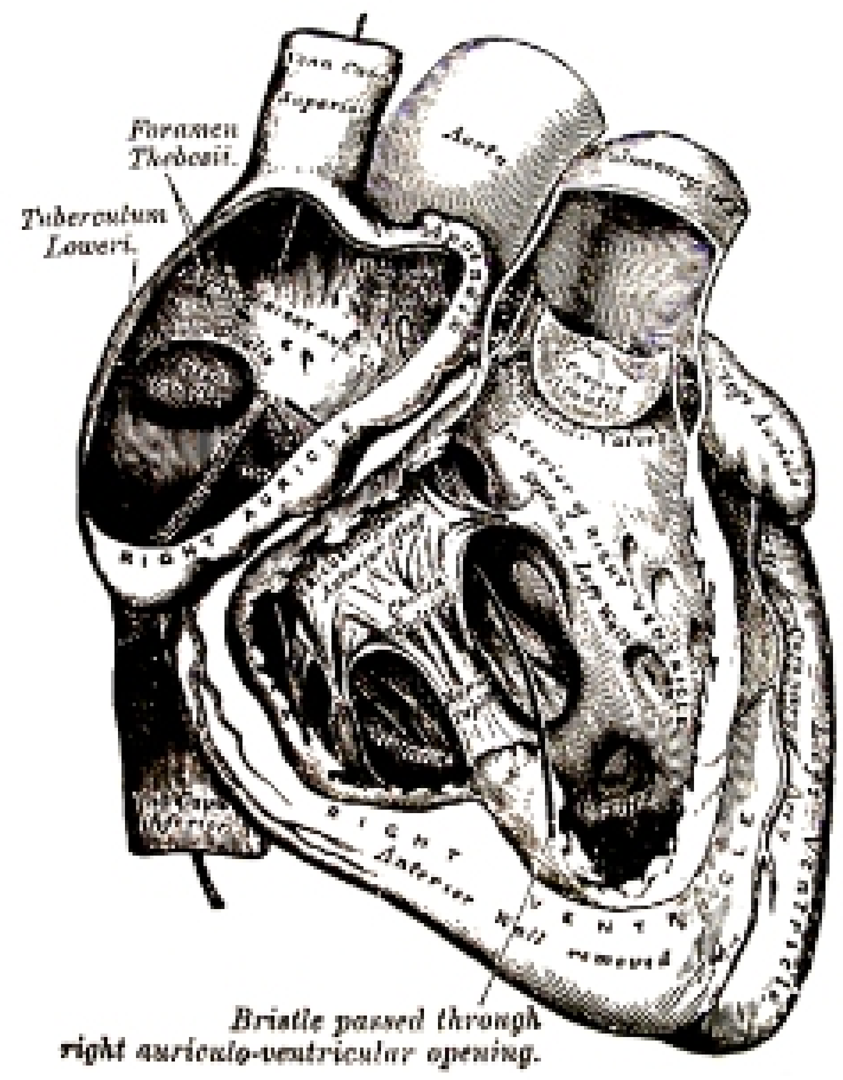

Figure 1: The human heart (from Henry Gray's *Anatomy of the Human Body*, 1918)

1 Assignment

Healthy human hearts contain electrical circuitry that periodically triggers and controls contraction of their upper chambers, atria or auricles, and lower chambers, ventricles (please see Figure 1). In some hearts, this control circuitry fails. Fortunately, we can replace it with a cardiac pacemaker.

Please build a finite state machine based controller for a dual-chamber cardiac pacemaker. This device will periodically electrically stimulate the heart in order to trigger contraction. Your pacemaker will sit under the patient's collar-bone. One of its barbed electrodes will be lodged in fibrous material (trabeculae) lining the inside of the right atrium. The other will be lodged in the right ventricle. In order to trigger contraction of the atria or ventricles, a 3 V potential is generated between the appropriate electrode's tip and a ring farther back along the electrode.

Your prototype will have a clock input, E, an accelerometer input, M, and an asynchronous clear input, C. In the finished pacemaker, the clock signal will be generated by a built-in oscillator. For your prototype, you will generate the clock signal with a debounced pushbutton. The accelerometer input (generated by a switch in your prototype) will be used to control the pacemaker's period. Frequent acceleration of the

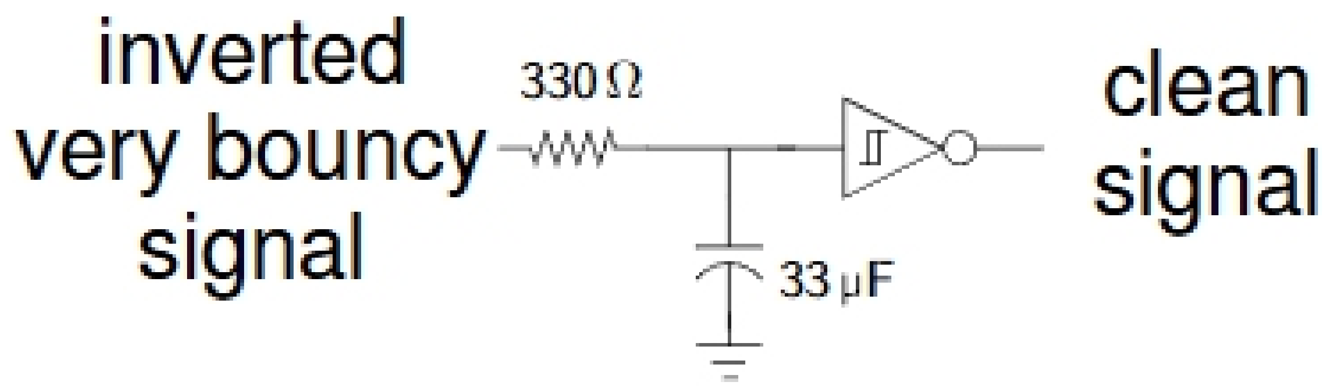

Figure 2: Long time constant debouncer

patient indicates physical activity, e.g., running or dancing. The pacemaker should cause the heart to beat faster when the patient is physically active and slower when the patient is resting.

2 Debouncing

Mechanical switches and buttons bounce, allowing spurious signal transitions to be generated. This is a problem for clock signal generation, as multiple pulses can unintentionally be generated by one press of a button. You may use the hysteresis of a 74LS14 Schmitt trigger inverter to clean up the output. It has the same pin-out as a 74LS04.

Design a circuit, based on the pushbutton, that will output a high signal when the button is not pressed and a low signal when the button is pressed. Pass this output through a low-pass filter into the input of one of the 74LS14's, as shown in Figure 2, and use the Schmitt trigger inverter's output to clock your pacemaker prototype.

3 Design guidelines

A real pacemaker controller would require something on the order of 1024 states in order to precisely control signal timing. In this assignment, to limit implementation complexity, we will be using only eight states. The same basic idea we're using in this assignment would allow the precise control required in a real pacemaker. However, a lot more wiring would be necessary.

Please implement a synchronous finite state machine with three state variables, J, K, and L. The first state should be assigned the value 000. Activating the asynchronous clear input, C, should return the machine to state 000. The machine should have two outputs, A and V, that will control the stimulation of atria and ventricles, respective. When the accelerometer signal, M, is low, the machine should repeatedly cycle through the following outputs:

A	V
1	0
0	0
0	1
0	0
0	0
0	0
0	0
0	0

When the M is high, the machine should repeatedly cycle through the following outputs:

A	V
1	0
0	1
0	0
0	0

Please put LEDs (with current-limiting resistors) on J, K, L, A, and V.

4 Hints

I have gone to great lengths to find a design that may be fully implemented using only five integrated circuit packages from your lab kits.

Please consider the following facts during design:

1. You may use left-over Schmitt trigger inverters as normal inverters.
2. Note that you may leave unused set inputs floating high when prototyping with TTL. This is must not be done for CMOS, or for final products.
3. Think about how you can manipulate your state variable functions to make them easy to implement with your lab kits. In addition to the flip-flops and the 74LS14, my design ended up using seven two-input NAND gates and a couple of three-input ANDs.
4. Some counting orders are easier to implement than others. Try coming up with something clever on your own. I found the following sequences to work well:

   001, 010, 011, 110, 111, 100, 101, 000 (for $M = 0$)

   and

   001, 011, 111, 101 (for $M = 1$)
5. Note that none of my state variable functions had more than four literals. It's fine if you have a somewhat less efficient design. However, if you end up with something dramatically more complicated, consider alternative designs before you start wiring.

5 Requirements

Prepare a laboratory report. This report should contain the following information.

- A problem statement or objective for the laboratory assignment
- Anything you used in achieving this objective, e.g., truth tables or algebraic simplification, etc.
- A list of the parts required for the circuits you implemented
- Schematic diagrams of the circuits you implemented.
- A brief discussion of how you verified that the implementation meets the requirements
- A circuit floorplan (optional)
- Comments and observations

The lab will be graded as follows: