

A Simple Mixed-Signal Circuit for Driving a Stateful Audiovisual Display

Jonathan Doman (jmdoman@vt.edu)

Zachary Rattner (zrattner@vt.edu)

Virginia Polytechnic Institute and State University
Blacksburg, Virginia, United States of America

April 11, 2011

Abstract

In this paper we present a method for designing a mixed-signal circuit used for presenting synchronized aural and visual messages. This circuit was designed with a particular display medium in mind, namely a series of 17-segment LED displays. However, it would be possible to extend the methods presented to work for other media. The circuit is stateful in that several different messages can be displayed, and the time cycle between each is dynamically configurable. In addition to timing, the intensity of the display and the nature of the generated audio can be changed while the circuit is in operation.

Keywords — PLaSS, beard

1 Introduction

1.1 Motivation

The need for a stateful audiovisual display was made apparent while the author was enrolled in ECE 4534: Embedded System Design. It was determined there was a dearth of levity in the course proceedings which could only be overcome with an interesting, and perhaps humorous, circuit. Additionally, the author was disappointed with the scope and nature of the projects assigned in the aforementioned class, finding them to be too restrictive and unimaginative. And so, casual conversation between the authors led to the idea for the circuit presented below.

1.2 Requirements

It was determined at the outset that the circuit should be able to cycle between two messages, with an audible beep after each. Thus there are three states to the overall display:

1. Message One
2. Message Two
3. Beep (or possibly a Boop).

Additionally, it seemed desirable to have the messages display for a set time, say T_1 , with the beep only being produced for about $T_2 = \frac{T_1}{3}$, or some similar fraction of T_1 .

Because we wanted to keep the state machine simple, this timing requirement meant

that we must have a variable period clock. This is perhaps the most interesting aspect of the circuit, and its details are presented below.

2 Description

2.1 Overview

At a high level, there are three different states of the circuit. A variable period clock is used to transition the state registers. The state code is used to drive three switches that control whether Message One is showing, Message Two is showing, or the Beep is being produced.

2.2 Clock Generation

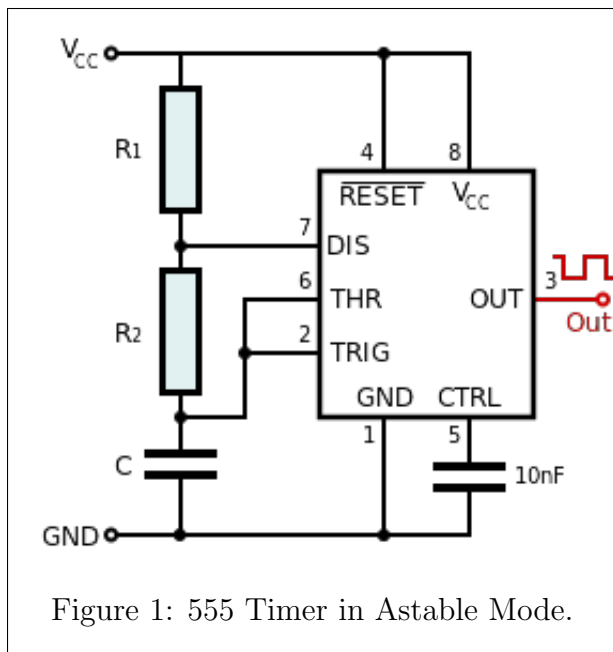


Figure 1: 555 Timer in Astable Mode.

As the basis for a variable period clock, we decided to use a 555 timer configured as an astable multivibrator. As seen in Figure 1, this configuration uses two external resistors and one external capacitor to set the timing characteristics.

The time that the output signal is low will correspond to the sound being made, while

the output signal high will correspond to a message being shown. The aural and visual outputs are designed to be mutually exclusive in pursuit of simplicity. A value of $1\text{ k}\Omega$ was chosen for R_2 , along with a value of $200\ \mu\text{F}$ for C . This yields a constant output low time of $\ln(2) \cdot (1\text{ k}\Omega) \cdot (200\ \mu\text{F}) = 139\text{ ms}$.

We next picked a value of $470\ \Omega$ for R_1 , which yielded an output high time of 204 ms . In order to provide a variable high time, a trim potentiometer (wired as a variable resistor) was added in series with R_1 . This caused the range of R_1 to be $470\ \Omega \leq R_1 \leq 10.47\text{ k}\Omega$ and the corresponding time range to be $204\text{ ms} \leq T_1 \leq 1.59\text{ s}$.

Therefore, these values meet our requirement that the visual messages should always show for longer than the Beep is heard.

2.3 Dual-Edge Triggeration

The signal output by the 555 timer has data encoded in each transition. That is, a rising edge indicates a transition to visual state should occur, while a falling edge indicates a transition to an aural state should occur. This type of signal is obviously incapable of clocking standard 74HC CMOS logic. However, since a 555 timer output was by far the easiest way to produce the desired signal, an intermediate circuit was required to translate to something the flip flops could work with.

2.3.1 Theory

Basically, we needed to clock the flip flops on every transition detected. This type of pulse detection circuit is not widely documented, but it is in fact used by all clocked logic to detect pulses. The idea is to invert and delay the input signal by some small amount, and then *AND* the direct input and the delayed input. On a rising edge, the inverted, delayed signal will not fall until some finite time t_{pd} after the original signal rose. The result of *AND*ing the two will be a high pulse

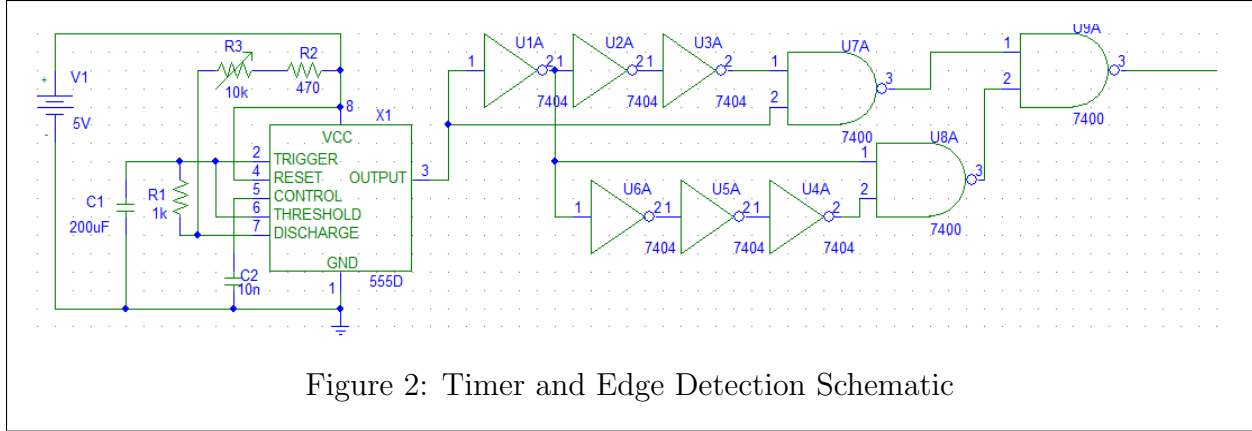


Figure 2: Timer and Edge Detection Schematic

lasting as long as t_{pd} . Performing the same operation on the inverted signal will result in falling edge detection. The two pulse outputs can be *OR*ed together to produce a dual-edge detection signal.

2.3.2 Analysis

The complete circuit so far developed can be seen in Figure 2. Three inline inverters are used to simultaneously invert the signal and provide a delay. On the standard 74HC04 chips which were used, a single inverter provides about 7 ns delay. Therefore, the total t_{pd} is 21 ns. While we were unsure if a delay of this size would be sufficient to reliably clock the flip flops, simulation in PSpice and actual implementation have demonstrated it to be completely sufficient.

2.4 State Machine

2.4.1 Derivation

In order to utilize the clock signal generated by the 555 timer and produce the appropriate control signals to actuate the display with appropriate timing, a hardware state machine was realized. The display was to show one of two distinct messages at a time, and turn off momentarily between each message displaying state.

Given this information, there are three distinct states:

State	Meaning
B	Beep
$M1$	Message 1
$M2$	Message 2

A formal depiction of this state machine follows. Note that the B state occurs twice.

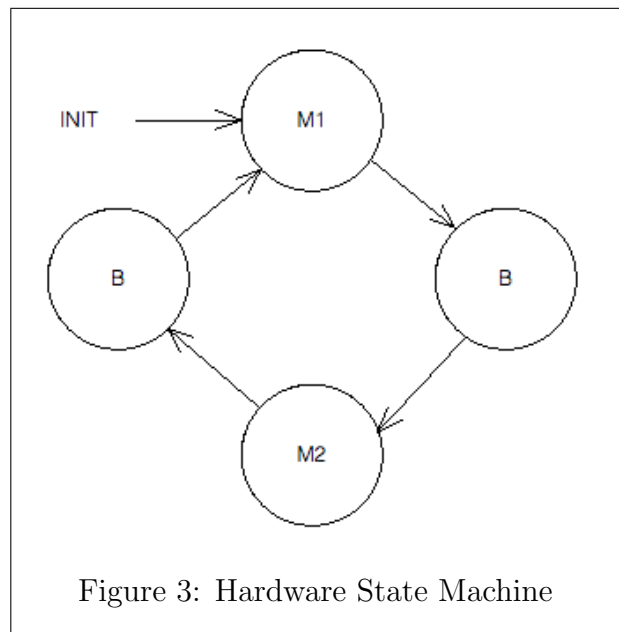


Figure 3: Hardware State Machine

2.4.2 Implementation

Codes were assigned to each state in the machine in Figure 3 using a dense state assign-

ment procedure. The two instances of B transition to $M1$ and $M2$. Due to the discrepancies between their transitions, the two message states are incompatible with each other. As a result, B requires two distinct state codes, so a minimum of $\lceil \log_2(4) \rceil = 2$ D-flip flops were required to realize the state machine. Karnaugh mapping revealed a state assignment which minimized the total number of D-flip flops. The state machine hardware was realized by a two-bit free-running counter with control logic associated with each bit. In the following equations, S_1 refers to the most significant bit, and S_0 refers to the least significant bit:

- $B = S_1$
- $M1 = \overline{S_1} \cdot \overline{S_0}$
- $M2 = S_1 \cdot \overline{S_0}$

The above set of equations was implemented using two 7400 Quad 2-Input NAND chips and one 74175 Quad D-Flip Flop chip. Since the 74175 chip supports an active-low asynchronous reset input, the ANDY board's push button B was used to assert the reset signal.

Since asserting the reset signal caused both state variables to return low, a safeguard was implemented to ensure the state machine stayed synchronized with the timer circuit through a reset. Since $M1 = \overline{S_1} \cdot \overline{S_0}$, the state machine returned to the $M1$ state in the event of a reset. As a result, resets could only occur in the display states— $M1$ or $M2$. To delay servicing the reset during the B states, the reset signal consisted of the push button signal *ANDed* with $\overline{S_0}$.

2.5 LED Display Driver

The three controls signals output by the state machine were used to drive NPN transistor as switches. The control signal was connected

to the base, with +5 V on the collector and the LED segments following on the emitter. The typical setup for driving LED displays requires a separate current limiting resistor for each segment. However, because we decided to only support two static messages (Message One and Message Two), a significant number of simplifications could be made to the driver circuit.

Most notably, any LED segments shared between Message One and Message Two could be partitioned into a common circuit. For the particular messages we chose, there were 36 active segments in Message One and 35 active segments in Message Two, and they had 24 segments in common. Thus, we could use just three resistors - one for each branch.

In addition, a trim potentiometer (configured as a variable resistor) was placed between the common cathode of the 17-segment displays and ground. Thus, by increasing the resistance, the available voltage to drop across the LEDs would decrease and they would become dimmer. By connecting all the common cathodes, the brightness of the entire display can be adjusted at one.

2.5.1 Example

For the Message One circuit, there were 12 LEDs in parallel and we needed to pick one resistor to limit current to all of them. A typical quiescent point at which to operate the LEDs is 20 mA at 2.1 V. Because there are 12 LEDs in parallel, there will be 240 mA through the current limiting resistor and transistor. To be on the safe side, we did not even take into account the collector-emitter voltage, V_{CE} , dropped across the transistor. Assuming that the resistor will have to drop all 2.9 V at 240 mA, we picked a 12 Ω current limiting resistor. This seemed to be sufficient until the circuit was actually built, and the resistor was observed to be getting exceptionally hot. Of course, we were

using $\frac{1}{2}$ -W rated resistors, and they were being used to dissipate (in the worst case) $(2.9V)(240mA) = 0.696W$. This was solved by putting smaller resistors in series so that the power rating for each subcircuit was met.

2.6 Aural Stimulation

In addition to the the three LED subcircuits, a state machine output was used to switch a small speaker. The ANDY board integrated function generator output was put on the collector of an NPN transistor and the control signal was applied to the base. The speaker was placed between the emitter and ground. Thus, when the LED display was not on, a pleasant tone would be emitted. The frequency and wave shape could be adjusted to simulate whatever tone is desired, from a heart rate monitor to the soothing rhythm of a malfunctioning printer.

2.7 Oral Stimulation

None.

3 Conclusion

We have presented a general method and analysis for a means to display audiovisual messages. However it is necessary to decide the specific messages early in the design process, since the 17-segment displays must be physically rewired in order to change messages.

As previously stated, the motivation for this circuit was centered around Embedded System Design. As a nod to the professor's technical acuity and a reference to his most memorable five-letter facial feature, we chose to implement the messages "Plass" and "beard".

A Bill of Materials

555 Timer	1
74HC175 Quad D-Flip Flop	1
74HC00 Quad 2-Input NAND	3
75HC04 Hex Inverter	1
2N2222 NPN Transistor	4
100 μ F Capacitor	2
10 nF Capacitor	1
1 k Ω Resistor	1
470 Ω Resistor	1
5.6 Ω Resistor	2
6.8 Ω Resistor	2
1.8 Ω Resistor	3
10 k Ω Trim Potentiometer	1
100 Ω Trim Potentiometer	1
LTP537G 17-Segment LED Display	5
Beeper (opt. Booper)	1
ANDY Board	1
0.1" Jumper	10
Green wire	0.110 Smoot
Red wire	7.370 in
Black wire	18.720 cm
Gray wire	0.102 fathoms
Orange wire	0.037 rods
Yellow wire	1.251 pAU