ENIB Semester S3P:
Electronics - Digital Circuits

Design and Analysis of State Machines: Implementation of a Trunk Opening System and Analysis of an Acoustic Rangefinder

Florian PASCO

Version 1.0: June 4, 2022

Contents

List of Figures

1: Design of the Trunk Opening Control System

I. Specifications

The system is fully synchronous, and all pulses last longer than the clock period. It has the following characteristics:

4 Inputs:

- 3 Push buttons ("BP0", "BP1", "BP2") used to enter a code.
- 1 Trunk lock input ("V") – when $V = 1$, the trunk locks again.
  - Note: This input has no effect if the trunk is not open.

1 Output:

- 1 trunk opening signal ($S = 1$ opens the trunk).

4 States:

- E_0: No button pressed.
- E_1: "BP0" pressed.
- E_2: "BP0 BP1" pressed.
- E_3: "BP0 BP1 BP2" pressed / Trunk opened.

Important: If two buttons are pressed simultaneously, the code entry resets (the system detects this as an error).
Note: Due to a labeling mistake, the buttons are "BP0", "BP1", and "BP2" instead of "BP1", "BP2", and "BP3" as stated in the original specifications.

II. State Diagram (Moore Machine Design)

The following figure represents the system's state diagram:

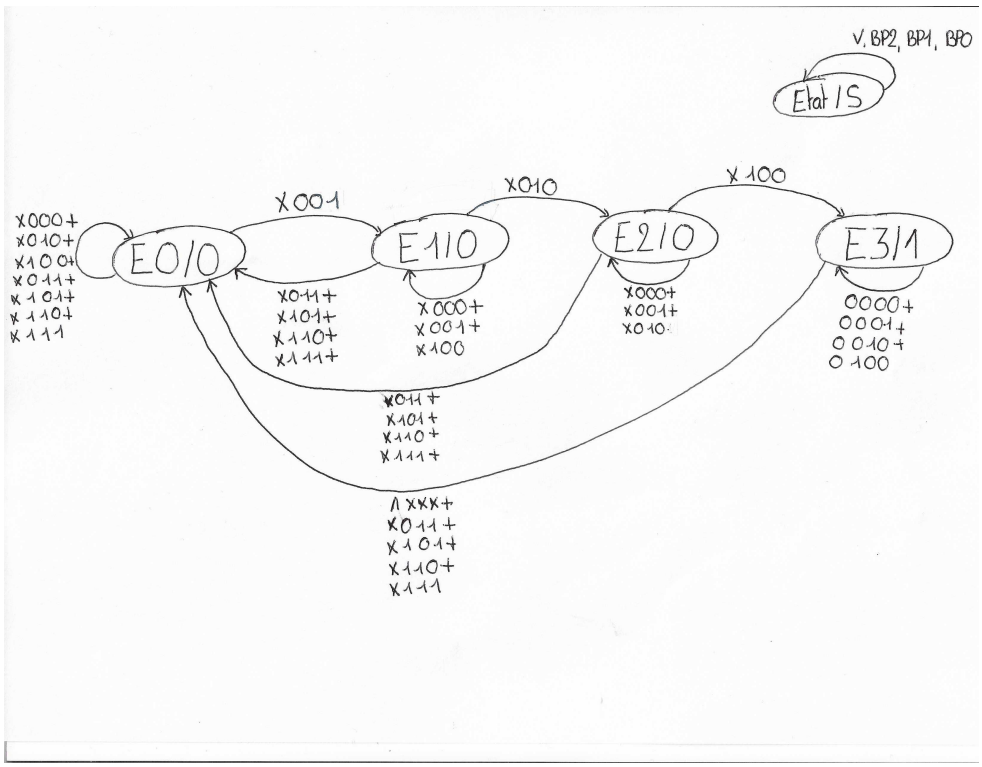

Figure 1: State diagram of the system

III. Deriving Flip-Flop Inputs and Output Expressions

1. State Transition Table

From the previous state diagram, the following transition table is obtained:

Etat actuel	bit Etat		Entrée				bit Etat futur		Etat futur	Sortie
	e_1	e_0	V	BP2	BP1	BP0	e_1^+	e_0^+		S
E0	0	0	X	0	0	0	0	0	E0	0
E0	0	0	X	0	1	0	0	0	E0	0
E0	0	0	X	1	0	0	0	0	E0	0
E0	0	0	X	0	0	1	0	1	E1	0
E1	0	1	X	0	0	0	0	1	E1	0
E1	0	1	X	0	0	1	0	1	E1	0
E1	0	1	X	1	0	0	0	1	E1	0
E1	0	1	X	0	1	0	1	0	E2	0
E2	1	0	X	0	0	0	1	0	E2	0
E2	1	0	X	0	0	1	1	0	E2	0
E2	1	0	X	0	1	0	1	0	E2	0
E2	1	0	X	1	0	0	1	1	E3	1
E3	1	1	0	0	0	0	1	1	E3	1
E3	1	1	0	0	0	1	1	1	E3	1
E3	1	1	0	0	1	0	1	1	E3	1
E3	1	1	0	1	0	0	1	1	E3	1
E3	1	1	1	X	X	X	0	0	E0	0
X	X	X	X	0	1	1	0	0	E0	0
X	X	X	X	1	0	1	0	0	E0	0
X	X	X	X	1	1	0	0	0	E0	0
X	X	X	X	1	1	1	0	0	E0	0

Figure 2: State transition table of the system

2. Karnaugh Maps

a. Karnaugh Map for e_0^+ Using the transition table, we derive the Karnaugh map for e_0^+:

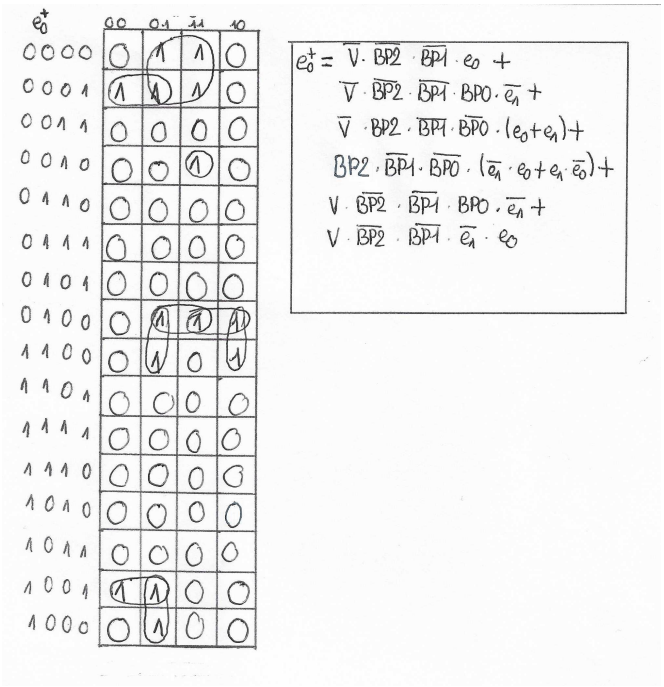

Figure 3: Karnaugh map for e_0^+

b. Karnaugh Map for e_1^+ Using the transition table, we derive the Karnaugh map for e_1^+:

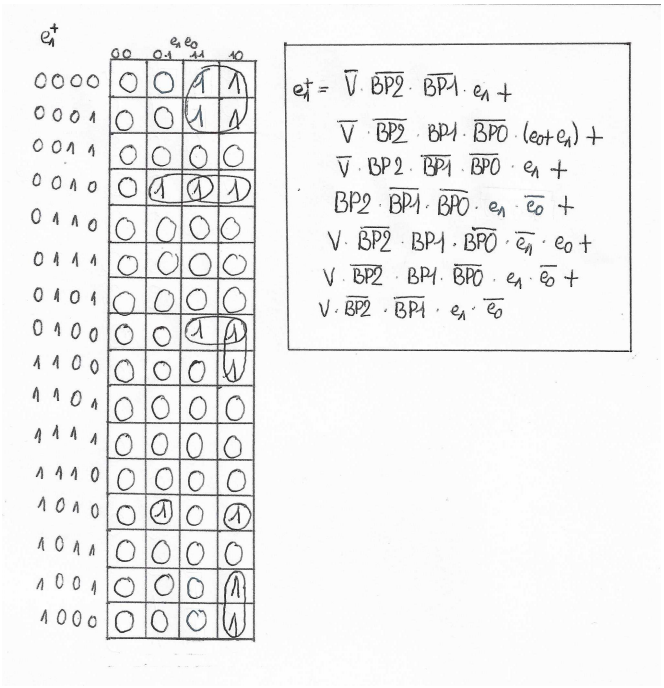

Figure 4: Karnaugh map for e_1^+

c. Output Expression Derivation From the transition table, the output equation is directly obtained as:

$$S = e_0 \cdot e_1$$

IV. System Design with D Flip-Flops

Based on the previous equations, the following circuit is implemented:

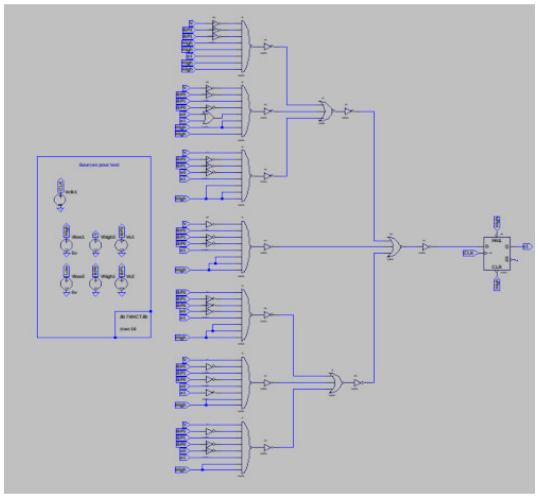

Figure 5: Circuit section for generating e_0

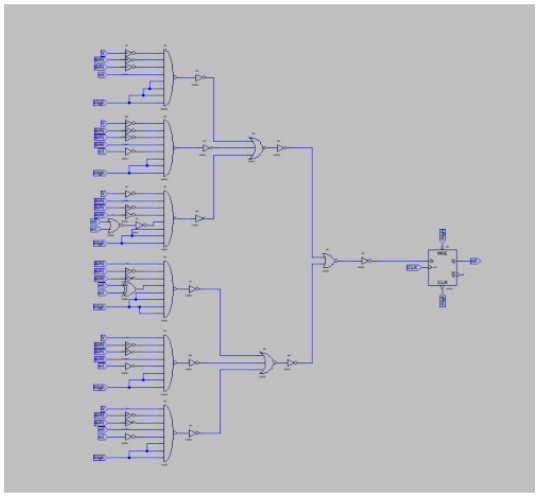

Figure 6: Circuit section for generating e_1

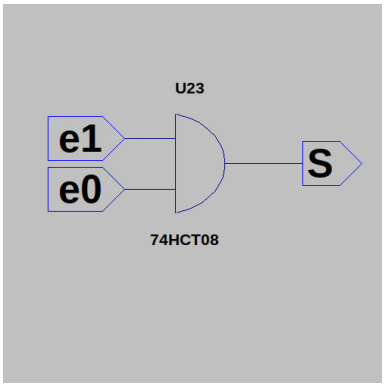

Figure 7: Circuit section for generating S

V. System Simulation and Testing in LTSpice

To validate the circuit, a test scenario is created, leading to the following simulation:

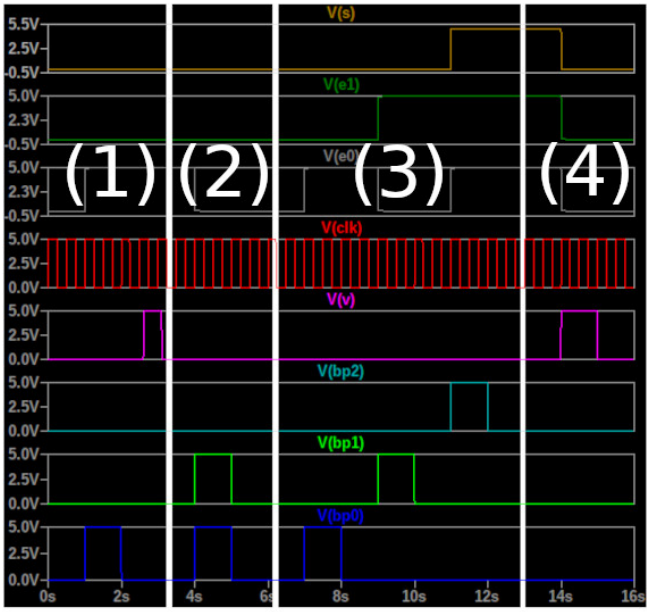

Figure 8: Circuit test using a simulated scenario

Test Scenario (corresponding steps are labeled in the simulation):

- **Step 1:** The operator presses "BP0", transitioning to E_1 ($e_1 = 0$, $e_1 = 1$). Then, the operator presses "V" – nothing happens as expected, since "V" should have no effect unless the trunk is open.
- **Step 2:** Still in E_1 ($e_1 = 0$, $e_1 = 1$). The operator presses "BP0" and "BP1" simultaneously, resetting the state to E_0 ($e_1 = 0$, $e_1 = 0$). The system detects an error, requiring the code to be re-entered.
- **Step 3:**
  - The operator presses "BP0" $\rightarrow$ transitions to E_1 ($e_1 = 0$, $e_1 = 1$).
  - The operator presses "BP1" $\rightarrow$ transitions to E_2 ($e_1 = 1$, $e_1 = 0$).
  - The operator presses "BP2" $\rightarrow$ transitions to E_3 ($e_1 = 1$, $e_1 = 1$), and the trunk opens ($S = 1$).
- **Step 4:** "V" is set to high, returning the system to E_0 ($e_1 = 0$, $e_1 = 0$), closing the trunk ($S = 0$).

2: Study of an Acoustic Rangefinder

The following state diagram needs to be corrected:

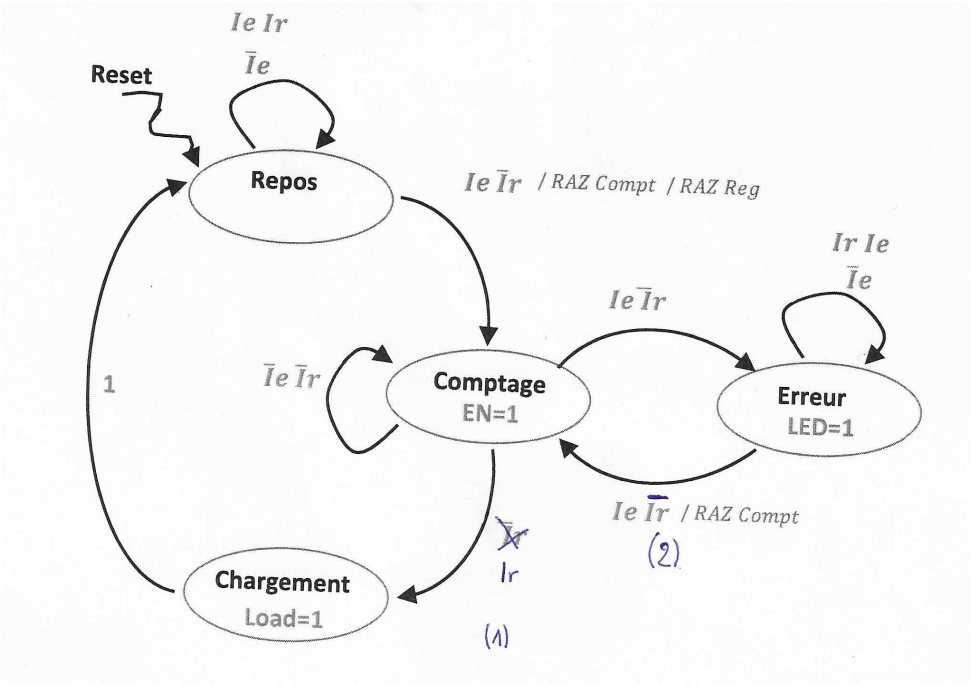

Figure 9: State diagram to be corrected

There are two errors in this system:

- **Error (1):** "$\overline{Ir}$" instead of "Ir" – The loading should occur when receiving Ir.
- **Error (2):** "IeIr" instead of "Ie$\overline{Ir}$" – The counting should resume when an Ie pulse is sent and no Ir pulse is received.