# ECE 450: Digital System Design

Lab 7 Prelab
Subway Signal Control Logic II

Joseph Mesches

November 21, 2025

## 1 Introduction

This prelab report presents the design of a sequential logic circuit for subway signal control. The system tracks train direction using two photocells ($P_1$ and $P_2$) and generates a direction signal $D$ that alternates between allowing two left-to-right trains followed by one right-to-left train.

## 2 State Transition Diagram

The finite state machine is modeled as a Moore machine with three state variables $(A, B, C)$. Each state is labeled with its name, binary encoding, and Moore output $D$. Every possible input combination $P_1P_2$ for each state is explicitly drawn and labeled.

### 2.1 State Descriptions

- **State 0 ($ABC = 000$):** Initial state, no train present. $D = 1$ (allowing left-to-right).
- **State 1 ($ABC = 001$):** Train entering from left (P2 blocked first).
- **State 2 ($ABC = 010$):** Train fully between sensors or exiting right.
- **State 3 ($ABC = 011$):** Train completing a passage before direction change.
- **State 4 ($ABC = 100$):** After two left-to-right trains. $D = 0$ (requiring right-to-left).

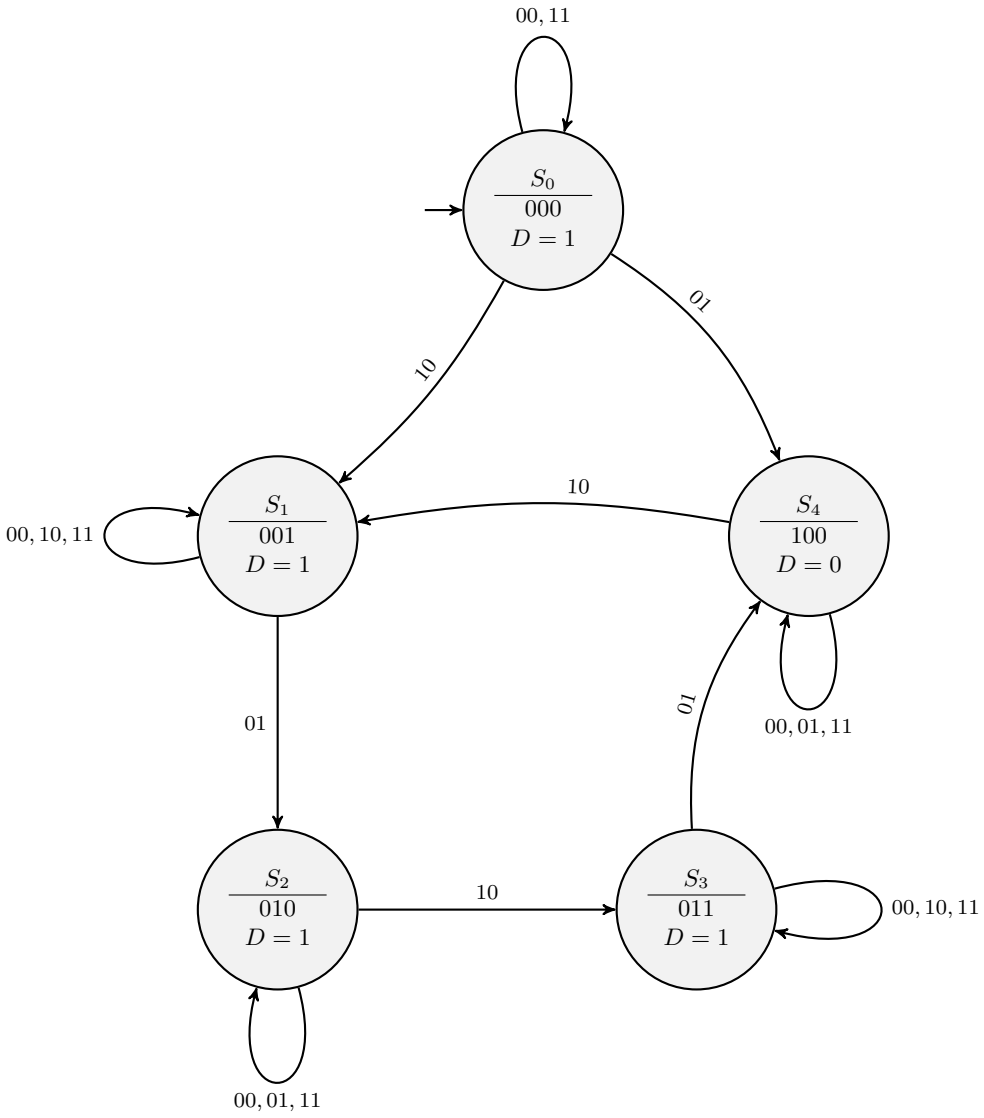


**Figure 1:** Complete Moore finite state machine for subway signal control. Each edge is labeled with input $P_1P_2$.

# 3 Truth Table

The complete state transition table uses the given state assignments and next-state values. States 5–7 are unreachable and treated as don't care conditions.

**Table 1:** State Transition Truth Table

| **S** | $A$ | $B$ | $C$ | $P_1$ | $P_2$ | $A^+$ | $B^+$ | $C^+$ | $D$ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 |
| | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 2 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |
| | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 |
| | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| 3 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 | 0 | X | X | X | X |
| | 1 | 0 | 1 | 0 | 1 | X | X | X | X |
| | 1 | 0 | 1 | 1 | 0 | X | X | X | X |
| | 1 | 0 | 1 | 1 | 1 | X | X | X | X |
| 6 | 1 | 1 | 0 | 0 | 0 | X | X | X | X |
| | 1 | 1 | 0 | 0 | 1 | X | X | X | X |
| | 1 | 1 | 0 | 1 | 0 | X | X | X | X |
| | 1 | 1 | 0 | 1 | 1 | X | X | X | X |
| 7 | 1 | 1 | 1 | 0 | 0 | X | X | X | X |
| | 1 | 1 | 1 | 0 | 1 | X | X | X | X |
| | 1 | 1 | 1 | 1 | 0 | X | X | X | X |
| | 1 | 1 | 1 | 1 | 1 | X | X | X | X |

# 4 Boolean Equations

This section records the minimized Boolean equations for the next-state bits and output, based on the state transition table.

## 4.1 Derivation Table

**Table 2:** Minterms, Don't Cares, and Boolean Equations

| Function | Minterms | Don't Care Terms | Minimized Boolean Equation |
|---|---|---|---|
| $A^+$ | 0, 12, 16, 17, 19 | 20–31 | $\overline{B}\,\overline{C}\,\overline{P_1}P_2 + BC\overline{P_1}P_2 + A\overline{P_1} + AP_2$ |
| $B^+$ | 5, 8, 9, 10, 11, 12, 14, 15 | 20–31 | $\overline{B}C\overline{P_1}P_2 + B\overline{C} + B\overline{P_2} + BP_1$ |
| $C^+$ | 2, 4, 6, 7, 10, 12, 14, 15, 18 | 20–31 | $P_1\overline{P_2} + C\overline{P_2} + CP_1$ |
| $D$ | 0–15 | 20–31 | $\overline{A}$ |

## 4.2 Simplified Boolean Equations

$$A^+ = \overline{B}\,\overline{C}\,\overline{P_1}\,P_2 + BC\overline{P_1}P_2 + A\overline{P_1} + AP_2 \tag{1}$$

$$B^+ = \overline{B}C\overline{P_1}P_2 + B\overline{C} + B\overline{P_2} + BP_1 \tag{2}$$

$$C^+ = P_1\overline{P_2} + C\overline{P_2} + CP_1 \tag{3}$$

$$D = \overline{A} \tag{4}$$

# 5 Design Considerations

## 5.1 State Encoding

The design uses a 3-bit state encoding $(A, B, C)$ which provides 8 possible states. Only 5 states (0–4) are used in normal operation; states 5–7 are treated as don't care conditions to enable logic minimization.

## 5.2 Train Detection Logic

- $P_1 = 0$, $P_2 = 1$: Train entering from left (blocks P2 first).
- $P_1 = 1$, $P_2 = 0$: Train entering from right (blocks P1 first).
- $P_1 = 1$, $P_2 = 1$: Train between sensors or both beams blocked.
- $P_1 = 0$, $P_2 = 0$: No train present (both beams clear).

### 5.3 Direction Control

The system implements the required alternating pattern by counting completed left-to-right passages. After two left-to-right trains (reaching state 4), the direction signal $D$ switches to 0, forcing the next train to travel right-to-left before resetting to the initial pattern.

## 6 Conclusion

This prelab presents a complete finite state machine design for the subway signal control logic. The Moore state diagram explicitly shows all input-labeled transitions, the truth table records the exact next-state and output values, and the Boolean equations provide an implementable realization of the next-state and output logic.