CLOCK WITH KMAPS

EE1003 : Scientific Programming for Electrical Engineers
Indian Institute of Technology Hyderabad

Yellanki Siddhanth (EE24BTECH11059)

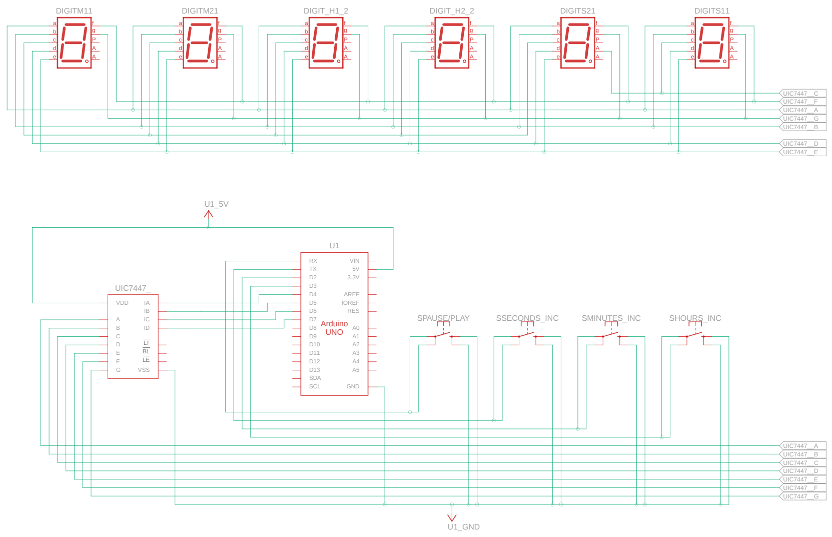
1 Introduction

The digital clock system described here utilizes Karnaugh maps (K-Maps) for incrementing time units and a multiplexing technique to display time on six seven-segment displays using only a single BCD. This report outlines the system's architecture, operation, and implementation details.

2 Circuit Description

The circuit consists of the following key components:

1) Arduino Atmel328p Microcontroller
2) Six Seven-Segment Displays
3) IC7447 BCD for the Seven-Segment Displays
4) Wires
5) 220Ω Resistors for the Seven-Segment Displays



Title: Clock

Made with Tinkercad®

Date: 3/24/2025, 9:23:08 PM

Sheet: 1/1

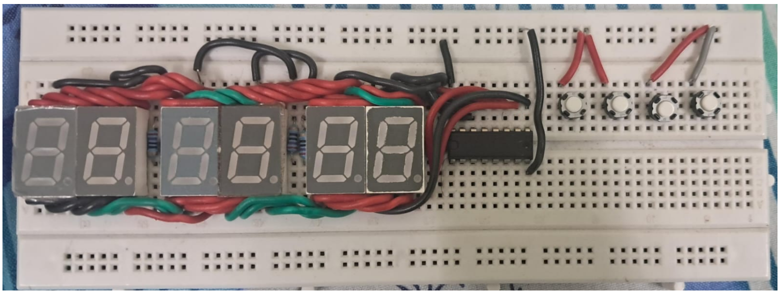


Fig. 5: Circuit Diagram

3 Multiplexing Technique

Multiplexing is achieved by connecting all inputs of the seven-segment displays to a single BCD. Digital pins are connected to the common of each display, allowing selective display of the BCD output. The displays are alternated with a 2μs time gap, creating the illusion of simultaneous operation.

4 K-Map Incrementing Logic

The incrementing logic for each display is implemented using decade counters. For the unit's place of the seconds, the logic is as follows:

$$A_1 = \overline{W_1}; \tag{1}$$
$$B_1 = (W_1 \land \overline{X_1} \land \overline{Z_1}) \lor (\overline{W_1} \land X_1); \tag{2}$$
$$C_1 = (\overline{X_1} \land Y_1) \lor (\overline{W_1} \land Y_1) \lor (W_1 \land X_1 \land \overline{Y_1}); \tag{3}$$
$$D_1 = (\overline{W_1} \land Z_1) \lor (W_1 \land X_1 \land Y_1). \tag{4}$$

For the ten's place of the seconds, which varies from 0 to 5:

$$A_2 = \overline{W_2}; \tag{5}$$
$$B_2 = (\overline{Y_2} \land \overline{X_2} \land W_2) \lor (\overline{W_2} \land X_2); \tag{6}$$
$$C_2 = (\overline{W_2} \land Y_2) \lor (X_2 \land W_2); \tag{7}$$
$$D_2 = 0. \tag{8}$$

To synchronize the ten's place increment with the unit's place reaching 9, an additional variable C is used:

$$C = W_1 \land \overline{X_1} \land \overline{Y_1} \land Z_1 \tag{9}$$
$$A_2 = (A_2 \land C) \lor (W_2 \land \overline{C}) \tag{10}$$
$$B_2 = (B_2 \land C) \lor (X_2 \land \overline{C}) \tag{11}$$
$$C_2 = (C_2 \land C) \lor (Y_2 \land \overline{C}) \tag{12}$$
$$D_2 = (D_2 \land C) \lor (Z_2 \land \overline{C}). \tag{13}$$

Now, using the above logic, the ten's digit of seconds only updates when the unit's digit previously is 9. This logic can be reapplied for the next display, i.e., the unit's digit of the minutes:

$$A_3 = \overline{W_3}; \tag{14}$$

$$B_3 = (W_3 \wedge \overline{X_3} \wedge \overline{Z_3}) \vee (\overline{W_3} \wedge X_3); \tag{15}$$

$$C_3 = (\overline{X_3} \wedge Y_3) \vee (\overline{W_3} \wedge Y_3) \vee (W_3 \wedge X_3 \wedge \overline{Y_3}); \tag{16}$$

$$D_3 = (\overline{W_3} \wedge Z_3) \vee (W_3 \wedge X_3 \wedge Y_3). \tag{17}$$

The value of C for this case is:

$$C = W_2 \wedge \overline{X_2} \wedge Y_2 \wedge \overline{Z_2} \wedge W_1 \wedge \overline{X_1} \wedge \overline{Y_1} \wedge Z_1. \tag{18}$$

5 Control Implementation

Four buttons connected to PORTD control the clock:

1) Pressing the first button will pause/play the clock.
2) Pressing the second button while paused will increment the seconds.
3) Pressing the third button while paused will increment the minutes.
4) Pressing the fourth button while paused will increment the hours.

To increment the minutes, the incrementing logic is run 60 times. Similarly, incrementing hours requires running the loop 3600 times.

6 Software Implementation

EEPROM persistence is implemented for all variables $(W_i, X_i, Y_i, Z_i, A_i, B_i, C_i, D_i)$, ensuring the clock retains its time when powered off.

7 Conclusion

This digital clock system effectively utilizes Karnaugh maps and multiplexing to display time on multiple seven-segment displays. The system's design and implementation provide a practical example of integrating logical incrementing techniques with efficient display methods.

Code Requirements

The code is located in the following directory:

```
/codes/final/code.c
```

It can be compiled and uploaded directly using the Makefile.