Lab report

Electronic devices and circuits lab - Assignment 3

EE23BTECH11066[1] **and EE23BTECH11054**[2]

[1] *Yakkala Amarnath Karthik*
[2] *Sai Krishna Shanigarapu*

August 23, 2024

Abstract

In this lab experiment, we present the digital clock using asynchronous circuits, with the input clock signal generated using Arduino. This project explores the use of asynchronous circuits to build a functional digital clock. In this experiment, Arduino serves as the clock pulse generator. In this project, key concepts such as gates, decoders, flip flops, counters are applied. The successful implementation of clock demonstrates the effectiveness of the combinational and sequential logic, and use of arduino to build a digital clock.

1. Introduction

In this lab report, we have designed a digital clock using Combinational and sequential circuits - decoders, flip flops, logic gates and other components. The goal of this project is to make a functional clock with hours, minutes and seconds till 8 hours. Instead of using the traditional IC 555-timer, we have used Arduino to give a 1Hz signal as a clock to the first D flip flop. The type of circuit we used is asynchronous[1] circuit.

Materials Required

- D flip flops
- Arduino
- Decoders
- 7-segment display
- jumping wires
- Nand gate
- Or gate

2. 1Hz clock

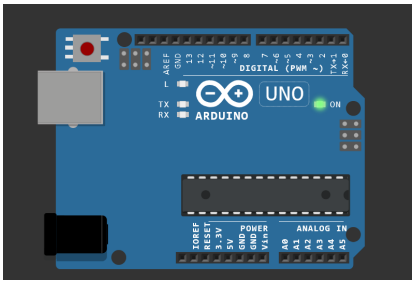

Figure 1

```
#define CLK 7
#define PRE 6
#define Q_0 8
#define Q_1 9
#define Q_2 10
#define Q_3 11

void setup() {
  Serial.begin(115200);
  pinMode(CLK, OUTPUT);
  pinMode(PRE, OUTPUT);
  pinMode(Q_0, INPUT);
  pinMode(Q_1, INPUT);
  pinMode(Q_2, INPUT);
  pinMode(Q_3, INPUT);

  digitalWrite(PRE, LOW);
  delay(100);
  digitalWrite(PRE, HIGH);
}

void loop() {
  Serial.print(digitalRead(Q_3));
  Serial.print(digitalRead(Q_2));
  Serial.print(digitalRead(Q_1));
  Serial.println(digitalRead(Q_0));
  digitalWrite(CLK, LOW);
  delay(0.1);
  digitalWrite(CLK, HIGH);
  delay(0.1);
}
```

The output is a 1Hz clock which is given to the first D flip-flop.

3. 7 segment display

A 7 segment display is a basic display tool with LED's - generally used to display the numbers. We have 4 input lines A, B, C, D. The 7 segment display is of 7 output lines a, b, c, d, e, f and g.
Here, we use a BCD to 7 segment decoder.

Digit	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Table 1. Truth table for BCD to 7 segment decoder

[1] Asynchronous circuits takes less hardware than that of Synchronous circuits

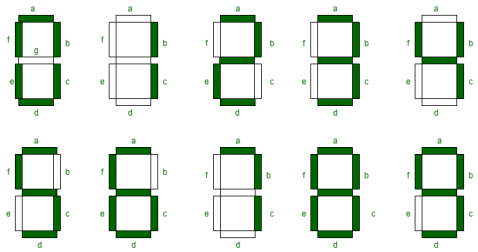

Figure 2. 7 segment output

The output of the Table 1 is shown in Figure[2] 2

4. Seconds

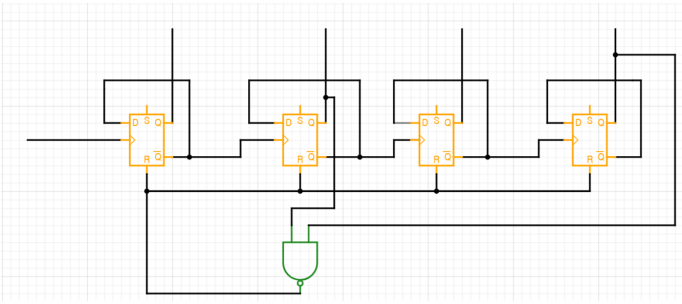

Figure 3. Schematic circuit for seconds

In the above circuit[3] the first D flip flop has the output of Q_0. Similarly, the other flip flops have Q_1, Q_2 and Q_3, where Q_3 is the most significant digit. The input clock is given from the Arduino.

4.1. Working

The output we will be getting is $Q_3\ Q_2\ Q_1\ Q_0$. Since the seconds should reset at 9, we need to give reset to each and every flip flop.
In BCD form 10 is 1010. Thus we can introduce a NAND gate for Q_3 and Q_1.
Since 1 NAND 1 is zero, the whole gets reset and starts from the beginning.

4.2. Seconds Ones Digit Counter

Design Overview

The seconds ones digit counts from 0 to 9, requiring a modulo-10 (MOD-10) counter. This can be implemented using four D flip-flops representing four bits ($Q_3Q_2Q_1Q_0$).

State Table

The state table for the MOD-10 counter is as follows:

Present State				Next State			
Q_3	Q_2	Q_1	Q_0	Q_3^+	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1

Table 2. State Table for Seconds Ones Digit MOD-10 Counter

Next State Equations

Using the state table, we derive the next state equations for each flip-flop.

$$D_0 = \overline{Q_0}$$
$$D_1 = Q_0 \oplus Q_1$$
$$D_2 = Q_2 \oplus (Q_1 \cdot Q_0)$$
$$D_3 = Q_3 \oplus (Q_2 \cdot Q_1 \cdot Q_0)$$

Reset Mechanism

The counter resets to 0 after reaching 9 (1001 in binary). The reset condition can be implemented using a `NAND` gate detecting when the count reaches 10 (1010 in binary), forcing all flip-flops back to 0.

$$\text{Reset Condition} = \overline{Q_1\, Q_3}$$

This condition can be connected to the asynchronous reset inputs of the flip-flops.

4.3. Seconds Tens Digit Counter

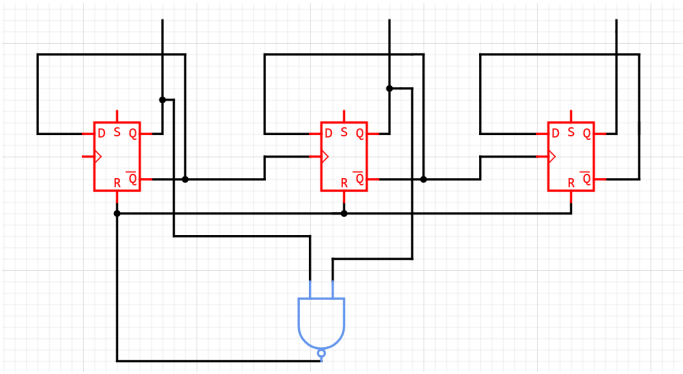

Figure 4. Schematic circuit of seconds tens digit

Design Overview

The seconds tens digit counts from 0 to 5, requiring a modulo-6 (MOD-6) counter. This can be implemented using three D flip-flops representing three bits ($Q_2Q_1Q_0$).

State Table

The state table for the MOD-6 counter is as follows:

[2]source: https://www.geeksforgeeks.org/seven-segment-displays/
[3]Website used to draw the circuit: https://www.circuit-diagram.org

Present State			Next State		
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	0	0	0
0	0	0	0	0	1

Table 3. State Table for Seconds Tens Digit MOD-6 Counter

Next State Equations

Using the state table, we derive the next state equations for each flip-flop.

$$D_0 = \overline{Q_0}$$
$$D_1 = (Q_0) \oplus Q_1$$
$$D_2 = (Q_2 \cdot \overline{Q_1} \cdot \overline{Q_0}) + (\overline{Q_2} \cdot Q_1 \cdot Q_0)$$

Reset Mechanism

The counter resets to 0 after reaching 5 (101 in binary). The reset condition can be implemented using a `AND` gate detecting when the count reaches 6 (110 in binary), forcing all flip-flops back to 0.

$$\text{Reset Condition} = \overline{Q_2 Q_1}$$

This condition can be connected to the asynchronous reset inputs of the flip-flops.

4.4. Integration and Working Mechanism

Clock Pulse Input

Both counters receive clock pulses:

- **Ones Counter**: Receives clock pulses every second.
- **Tens Counter**: Receives a pulse every time the ones counter resets from 9 to 0.

Reset Synchronization

When the ones counter resets after reaching 9, it triggers an increment in the tens counter. Similarly, when the tens counter reaches 5 and the ones counter resets, both counters reset to 0.

5. Minutes

Minutes is similar to the seconds. It requires the same conditions bcox it needs to reset at 60, just as seconds. We can refer Section 4 for its complete detailing.

6. Hours

6.1. Hour Ones Digit Counter

Design Overview

The hour ones digit counts from 0 to 9, requiring a modulo-10 (MOD-10) counter. This can be implemented using four D flip-flops representing four bits ($Q_3Q_2Q_1Q_0$).

State Table

The state table for the MOD-10 counter is as follows:

Present State				Next State			
Q_3	Q_2	Q_1	Q_0	Q_3^+	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1

Table 4. State Table for Hour Ones Digit MOD-10 Counter

Next State Equations

Using the state table, we derive the next state equations for each flip-flop.

$$D_0 = \overline{Q_0}$$
$$D_1 = Q_0 \oplus Q_1$$
$$D_2 = Q_2 \oplus (Q_1 \cdot Q_0)$$
$$D_3 = Q_3 \oplus (Q_2 \cdot Q_1 \cdot Q_0)$$

Reset Mechanism for Hour Ones Digit

The hour ones digit counter needs to reset at two different conditions:

- When the counter reaches 10 (1010 in binary), it should reset to 0.
- When the overall hour count reaches 24 (0010 0100 in binary, where tens digit is 2 and ones digit is 4), the counter should reset both digits (tens and ones) to 0.

Overall Reset Condition: The overall reset condition can be implemented using an `OR` gate that combines both reset conditions.

Let q and Q denotes for unit and tens digit respectively. Then the reset condition is

$$\overline{Q_1 q_2 q_0 \; + \; q_1 q_3}$$

6.2. Hour Tens Digit Counter

Design Overview

The hour tens digit counts from 0 to 2, requiring a modulo-3 (MOD-3) counter. This can be implemented using three D flip-flops representing three bits ($Q_2Q_1Q_0$).

State Table

The state table for the MOD-3 counter is as follows:

Present State			Next State		
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	0	0

Table 5. State Table for Hour Tens Digit MOD-3 Counter

Next State Equations

Using the state table, we derive the next state equations for each flip-flop.

$$D_0 = \overline{Q_0}$$
$$D_1 = Q_0 \oplus Q_1$$
$$D_2 = Q_2 \oplus (Q_1 \cdot Q_0)$$

Reset Mechanism for Hour Tens Digit

The tens digit counter needs to reset when the hour count reaches 24. This is detected as part of the overall reset condition for the hour counter.

Reset Condition: The reset condition for the hour tens digit is included in the overall reset condition for the hour counter, as discussed in the hour ones digit section.

$$\text{Reset condition: } \overline{Q_1 q_2}$$

7. Conclusion

This lab experiment successfully demonstrates the working of an asynchronous 24 hour clock using D flip flops and Arduino for clock generation (1 Hz). By using decoders, logic gates, flip flops, this project highlights the practical use of combinational and sequential logic in circuit design.

8. More Information

- The working of 24 hour clock is provided in our github link.
  Github: https://github.com/ssk1504/24-Hour-Digital-Clock/blob/main/24_hr_clk.mp4
- Data sheet for flip flops used.
  TI: https://www.ti.com/lit/ds/symlink/sn74ls74a.pdf

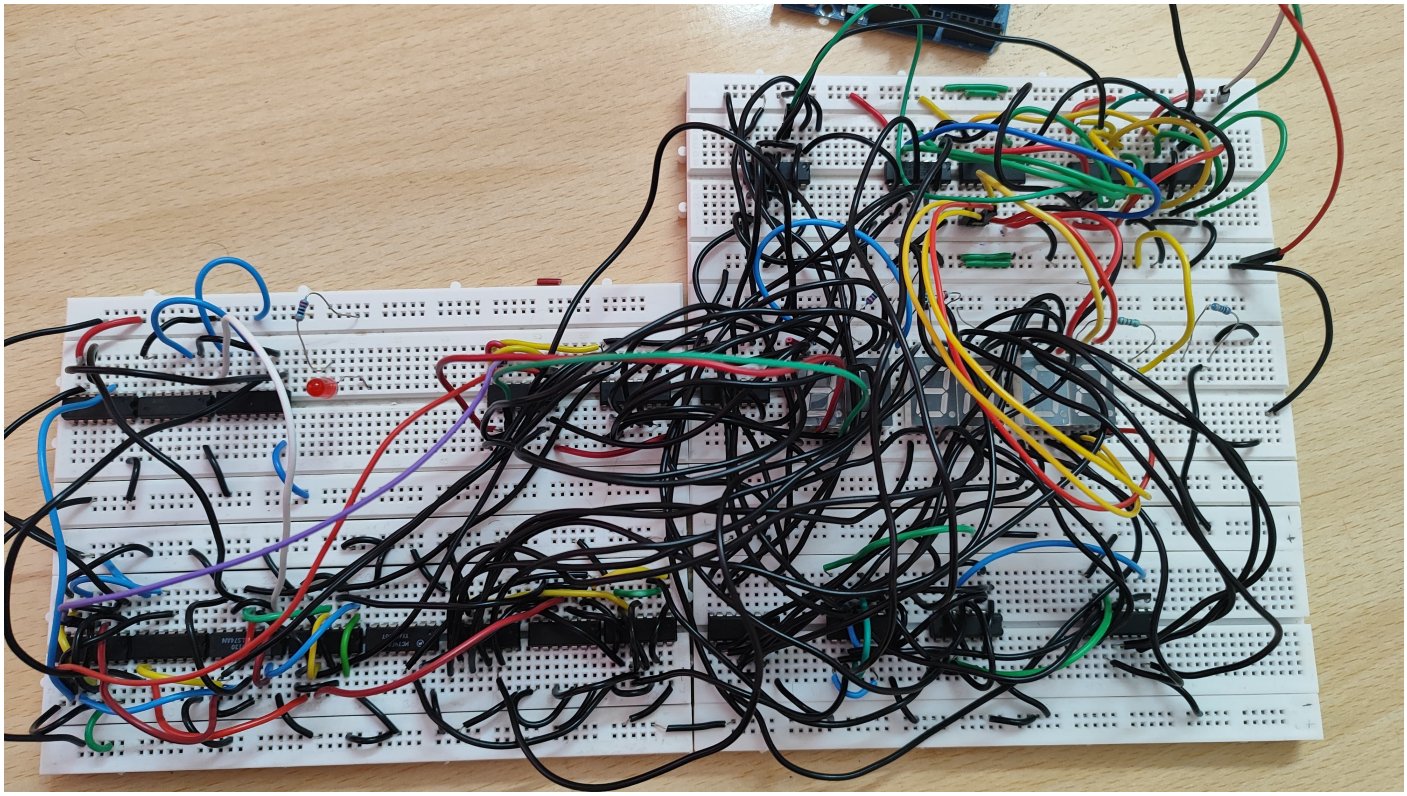

Figure 5. Circuit