

Lab Assignment 3

E. Mihir Divyansh^a and Manoj Kumar Ambatipudi^b

^aEE23BTECH11017, BTech 2nd Year, Electrical Engineering

^bEE23BTECH11040, BTech 2nd Year, Electrical Engineering

Dr. Gajendranath Chaudury

Contents

1	Introduction	1
2	Equipment Used	1
3	Theory and Design	1
3.1	The D Flip-Flop	1
3.2	Modulo N counters	1
	The BCD counter • Mod 6 Counter	
3.3	Making an 8 Hour clock	3
	Seconds Display • Minutes Display • 8 Hours	
4	Derivation of Segment Expressions for Seven-Segment Display	3
4.1	Truth Table for Seven-Segment Display	3
4.2	Karnaugh Maps for a the Segment	3
4.3	Final Boolean Expressions	4
5	Components Used	4
6	Troubleshooting and Sources of Error	4
7	Final Result	5
8	Conclusion	5

1. Introduction

This lab report presents the design and construction of a digital clock for Assignment 3, without using pre-made counter ICs. Instead, the counters were designed from scratch using D flip-flops, and some combinatorial circuitry.

2. Equipment Used

The following items were the core components used in building the circuit

- Dual D Flip-Flop - IC 7474 $\times 10$
- BCD to 7 Segment Decoder - IC 7447 $\times 6$
- 7 Segment Common Anode Displays $\times 6$
- Dual Quad-Input AND gates - IC 7421 $\times 4$
- Hex Inverter - IC 7404 $\times 5$
- Arduino UNO Board
- Breadboard $\times 7$

3. Theory and Design

A digital clock runs on the basic principle of frequency division, which is essentially a process of counting. At its core, a digital clock takes a high-frequency input signal, typically from a crystal oscillator, and divides it down to produce time intervals of seconds, minutes, and hours. This division is achieved through a series of counters that systematically reduce the frequency of the input signal by counting the pulses.. For this project, we chose to use a 1Hz clock as the input clock.

3.1. The D Flip-Flop

D flip-flops are fundamental components in digital electronics and are used as building blocks for counters. A D flip-flop is bistable, meaning it has two stable states and can store a single bit of data. The output state of a D flip-flop changes in response to the input data (D) at the rising or falling edge of the clock pulse. In a typical frequency

divider circuit, D flip-flops are connected in a series configuration where the output of one flip-flop serves as the clock input for the next. This arrangement effectively divides the input clock frequency by a factor of 2 for each flip-flop stage.

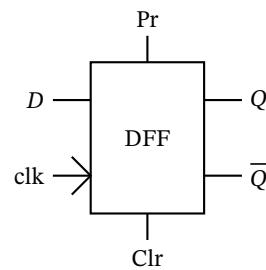


Figure 1. A D Flip-Flop

3.2. Modulo N counters

Counters which count from 0 to $N - 1$, and thus posses N distinct states are called Modulo N counters.

3.2.1. The BCD counter

A Modulo-10 counter, also known as a BCD (Binary-Coded Decimal) counter, is a type of digital counter that counts from 0 to 9 and then resets to 0 on the next clock pulse, thus making it a counter with ten distinct states. BCD is a form of binary encoding where each digit of a decimal number is represented by its equivalent 4-bit binary number. The counter operates by using 4 flip-flops, as frequency dividers to count from 0 to 15, and then use the 'Clr' pin to reset the circuit to '0000', when the counter reaches 10. So, it effectively counts from '0000' to '1001', i.e., 0 to 9. The Timing Diagram for a mod 10 counter is given in Figure 3.

Working of the BCD counter

- The input of the D flip-flop is tied to \overline{Q}_3 , this will alternate 0 and 1 on clock edge (As Q follows D , \overline{Q} is \overline{D} , so D becomes \overline{D} each clock cycle.)
- The clock of the second flip flop is \overline{Q}_3 . This will alternate with Half the frequency of the input clock signal.
- This process continues for 4 flip flops.
- The Output $Q_3Q_2Q_1Q_0$ is constantly checked by the 4 input NAND gate, when it becomes 1010, a reset signal is sent to the clear pin of DFF.

3.2.2. Mod 6 Counter

A mod 6 counter is similar to a BCD counter, but counts from 0 to 5 instead. The principles of the counting remain same. The design is shown in Figure 4

The timing diagrams shown in Figures 3 and 5 are generated by my verilog code modelling the circuit. It was done by writing a single D Flip-Flop, then using it 4 times to make a mod 10 counter, 3 times for a mod 6 counter. These 2 counters were then connected together in a mod 60 module. This counter was called twice for minutes and seconds.

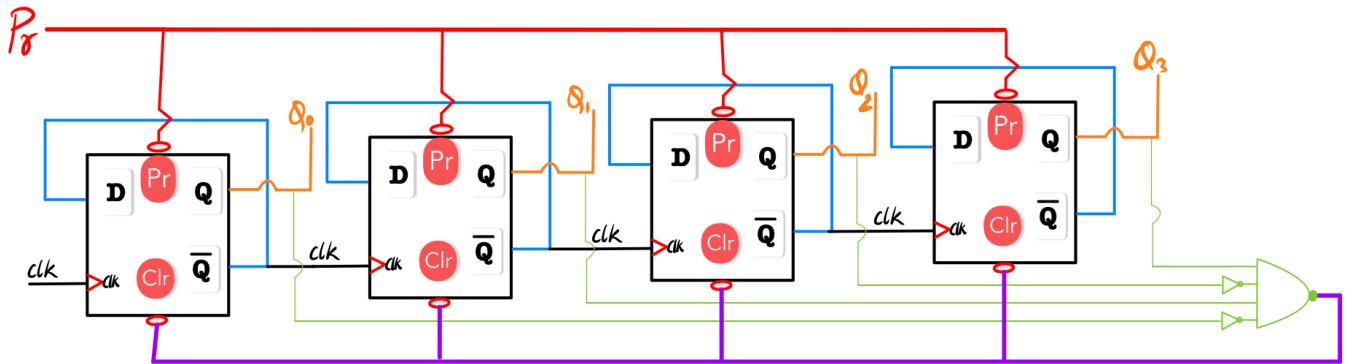


Figure 2. Design of a Mod 10 Asynchronous Counter

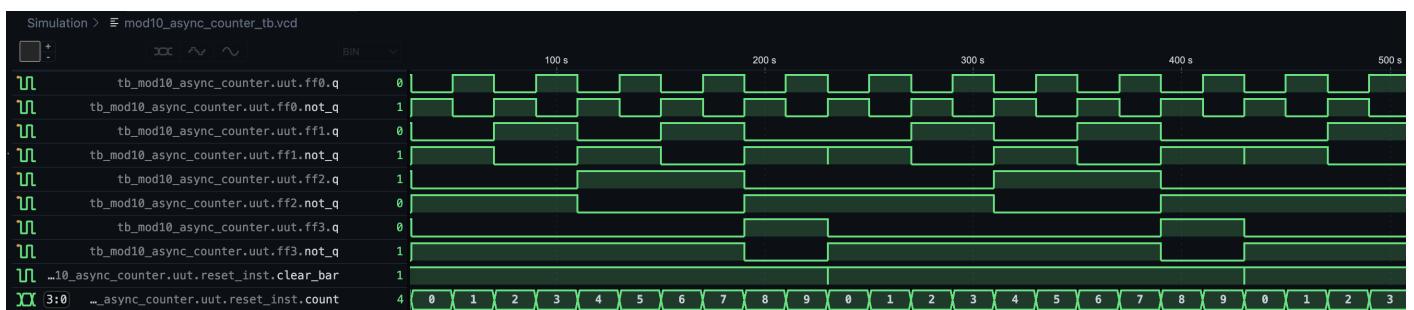


Figure 3. Timing Diagram of a Mod 10 Asynchronous Counter

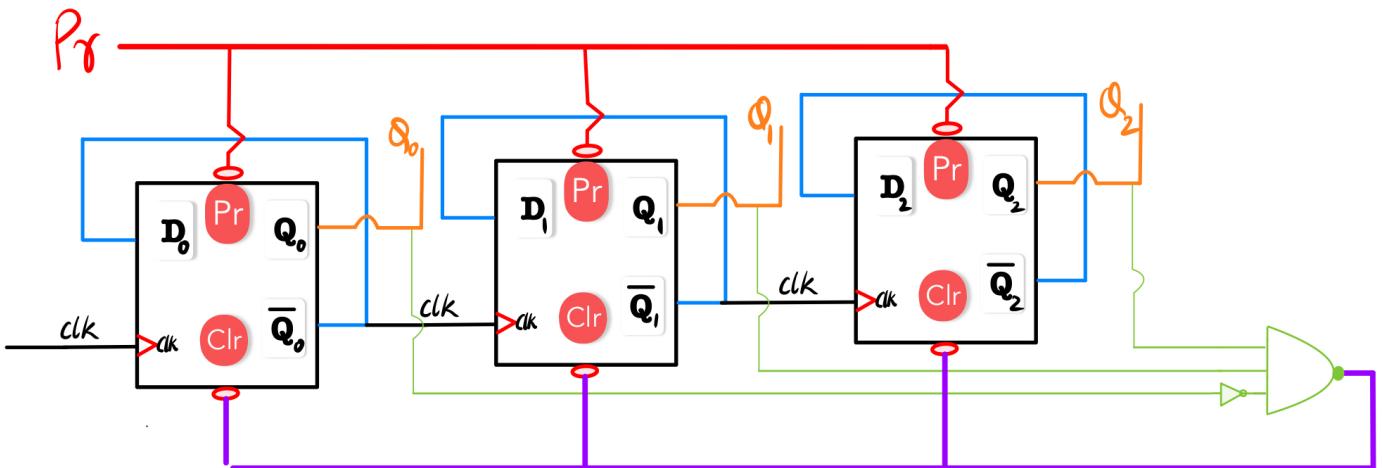


Figure 4. Design of a Mod 6 Asynchronous Counter



Figure 5. Timing Diagram of a Mod 6 Asynchronous Counter

3.3. Making an 8 Hour clock

3.3.1. Seconds Display

Now that a Mod-6 and Mod-10 counter are built, Seconds can be built by feeding 1Hz clock to the BCD counter, and then feeding \overline{Q}_3 to input clock of Mod-6 counter. Q_3 takes values 0, 0, 0, 0, 0, 0, 0, 1, 1 from 0-9. \overline{Q}_3 takes 1, 1, 1, 1, 1, 1, 1, 0, 0. This gives a rising edge during the transition from 9 to the next 0. Since the counters are posedge triggered, the Mod-6 counter increments only when the BCD transitions from 9 to 0.

The outputs of the 10 and 6 counters are taken to a decoder and then connected to a seven segment display. The displays look like Figure 6.

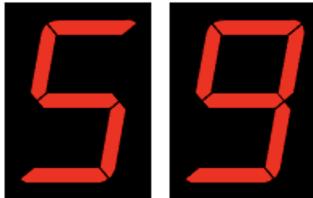


Figure 6. Seven segment display

The Mod-6 counter completes one cycle when 1 minute has passed. Hence we use the output from the Mod-6 counter as clock for the next signal.

3.3.2. Minutes Display

Same logic as the seconds display from 3.3.1. With the input clock as the inverted \overline{Q}_2 of the first Mod-6 counter

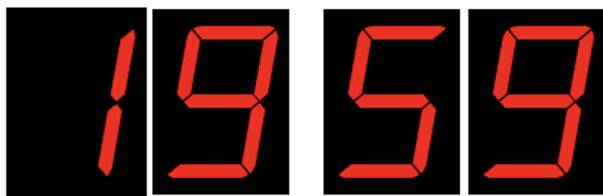


Figure 7. Seven segment display showing minutes

3.3.3. 8 Hours

Counting Mod-8 is made really simple due to the nature of the counters we build. They are always Mod- 2^n as each flip-flop divides by 2. So a counter made from 3 flip flops, without any reset circuitry will suffice for this purpose. Imagine Figure 4, without the reset. $Q_2 Q_1 Q_0$ counts from 0 to 7. All that is left is to trigger it appropriately. We feed \overline{Q}_2 from Mod-6 of the minutes counter as clock to the hours counter.



Figure 8. Seven Segment Display showing Hours : Minutes : Seconds

4. Derivation of Segment Expressions for Seven-Segment Display

In a seven-segment display, each segment (a, b, c, d, e, f, g) is controlled by a specific combination of the BCD input, which corresponds to the digits 0-9. We can derive the Boolean expressions for each segment by using Karnaugh maps (K-maps).

4.1. Truth Table for Seven-Segment Display

The truth table for a seven-segment display with BCD input (D, C, B, A) is shown below:

BCD	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
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 Seven-Segment Display with BCD Input

4.2. Karnaugh Maps for a the Segment

1. Segment a:

DC/BA	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	1	1	X	X
10	1	1	X	X

Simplified Expression:

$$a = \overline{AC} + D + B + AC$$

2. Segment b:

DC/BA	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	X	X	X	X
10	1	1	X	X

Simplified Expression:

$$b = \overline{C} + \overline{BA} + AB$$

3. Segment c:

DC/BA	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	X	X	X	X
10	1	1	X	X

Simplified Expression:

$$c = \overline{B} + A + C$$

4. Segment d:

DC/BA	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	X	X	X	X
10	1	1	X	X

Simplified Expression:

$$d = \overline{AC} + C\overline{BA} + D + BA$$

5. Segment e:

<i>DC/BA</i>	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	X	X	X	X
10	1	0	X	X

Simplified Expression:

$$e = \overline{AC} + BA$$

6. Segment f:

<i>DC/BA</i>	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

Simplified Expression:

$$f = \overline{AB} + C\overline{BC} + C\overline{A} + D$$

7. Segment g:

<i>DC/BA</i>	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	X	X	X	X
10	1	1	X	X

Simplified Expression:

$$g = B\overline{C} + C\overline{B} + C\overline{A} + D$$

4.3. Final Boolean Expressions

The final Boolean expressions for each segment of the seven-segment display based on the BCD input (W, X, Y, Z) are as follows:

$$a = \overline{AC} + D + B + AC$$

$$b = \overline{C} + \overline{BA} + AB$$

$$c = \overline{B} + A + C$$

$$d = \overline{AC} + C\overline{BA} + D + B\overline{A}$$

$$e = \overline{AC} + BA$$

$$f = \overline{AB} + C\overline{BC} + C\overline{A} + D$$

$$g = B\overline{C} + C\overline{B} + C\overline{A} + D$$

These expressions can be implemented in digital logic circuits to drive the segments of a seven-segment display. But this circuitry is in-built in the decoder IC (7447) that was used.

5. Components Used

The 7 segment display used is shown in Figure 12. The seven segment display is Common Anode. This means, High (V_{cc}) has to be given to one of the common pins.

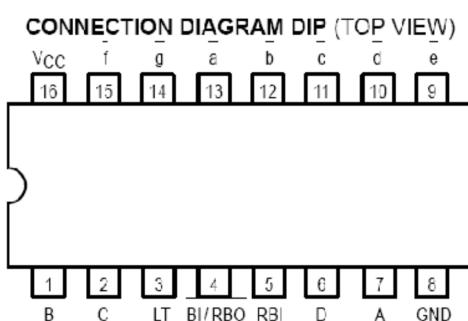


Figure 9. Pinout of 7447 - BCD to Seven Segment Decoder

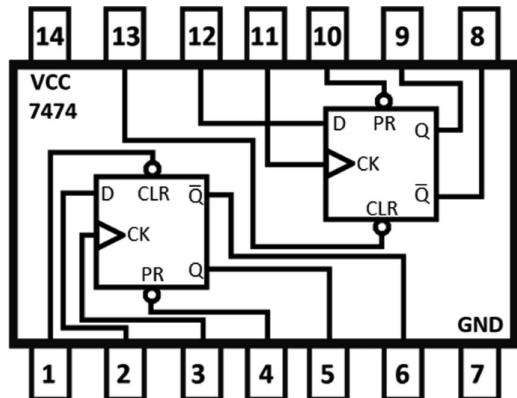


Figure 10. Pinout of 7474 - A Dual D flip flop

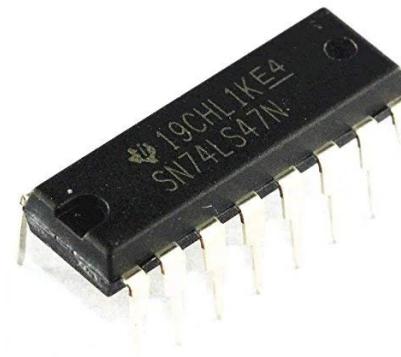


Figure 11. BCD to Seven Segment decoder

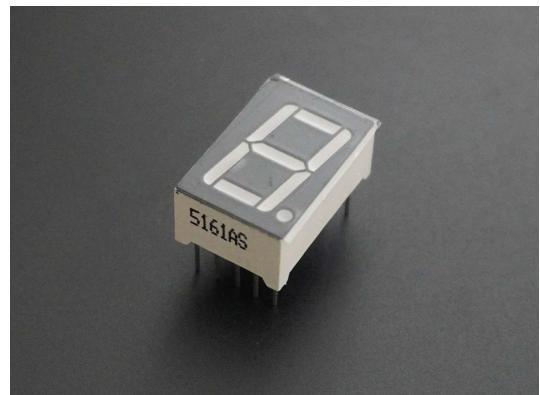


Figure 12. Seven Segment Display

Along with the above, the reset circuitry required combinational logic. The following components were used.

- AND gate IC - 7421
- Hex Inverter - 7404

Figure 16 contains the block diagram of the circuit.

6. Troubleshooting and Sources of Error

There were many issues encountered during the construction of this clock. Just to note a few,

- The ground for one of the breadboards was disturbed at some point in transit. Took us 1 day to figure out the issue.

- The reset circuitry was causing the circuit to start from 1 instead of 0 only at the beginning of the circuit.
- Some of the connections between the flip-flops were loose, resulting in intermittent clock signal propagation. This caused erratic behavior in the counters, leading to incorrect counting sequences.
- Inadequate testing of individual modules before integration led to compounding errors that were difficult to diagnose once the entire circuit was assembled. This meant we had to reassemble the circuit multiple times.

7. Final Result

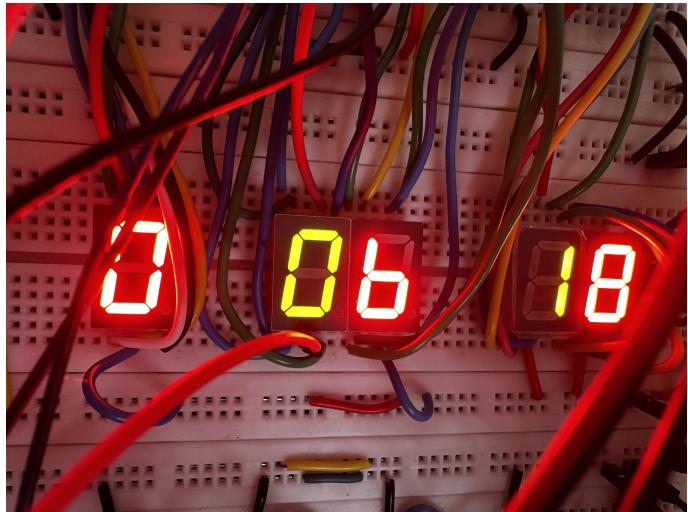


Figure 13. 8 Hour Clock - Close Up

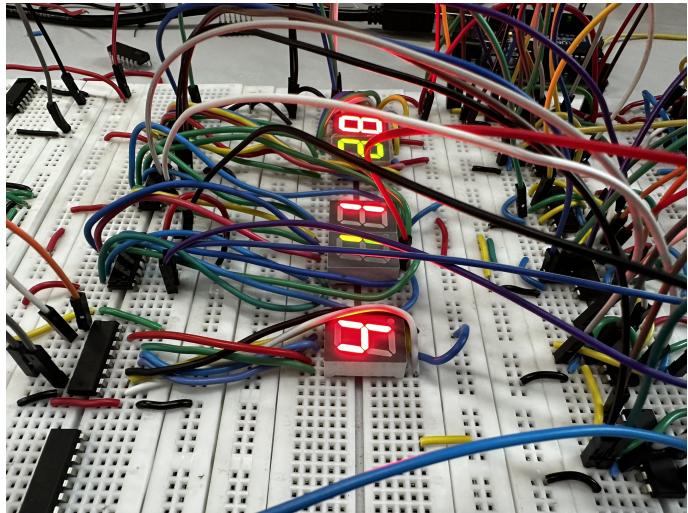


Figure 14. 8 Hour Clock - Hours

avoiding race conditions, and ensuring stable signal integrity across the circuit. These issues necessitated careful attention to wiring, component placement, and signal timing.

Through iterative testing and troubleshooting, we were able to address these challenges, though not without significant effort. The importance of a robust reset mechanism, proper signal grounding, and the use of debounce circuits became evident during the project. These experiences underscore the complexity of asynchronous circuits compared to their synchronous counterparts and emphasize the need for meticulous planning and verification in digital design.

Overall, the project was a success in achieving the intended functionality of the clock, and it provided a deeper understanding of the practical aspects of digital circuit design. The lessons learned in troubleshooting and error mitigation will be invaluable for future projects involving complex digital systems.

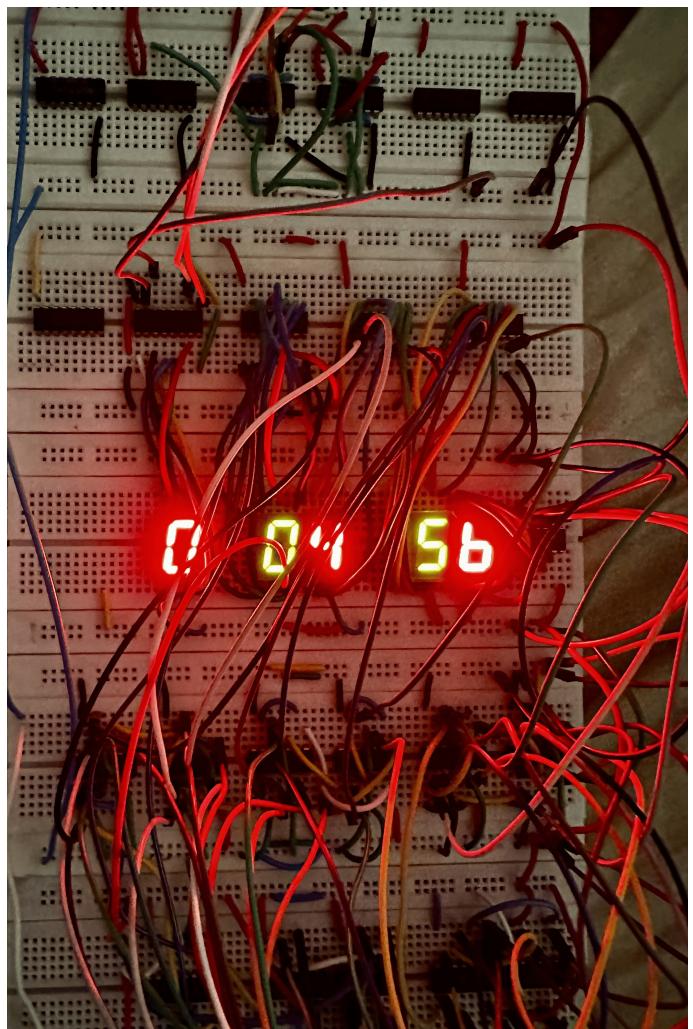


Figure 15. 8 Hour Clock - Top View

8. Conclusion

The construction of an asynchronous digital clock using D flip-flops provided valuable hands-on experience with digital logic design, timing analysis, and circuit troubleshooting. The project successfully demonstrated the operation of cascaded counters to keep time, with a Mod-60 counter for seconds and minutes. By leveraging D flip-flops in an asynchronous configuration, we were able to design a functioning clock circuit that effectively counts and displays time in a reliable manner.

However, the process also highlighted several challenges inherent to asynchronous design, such as managing propagation delays,

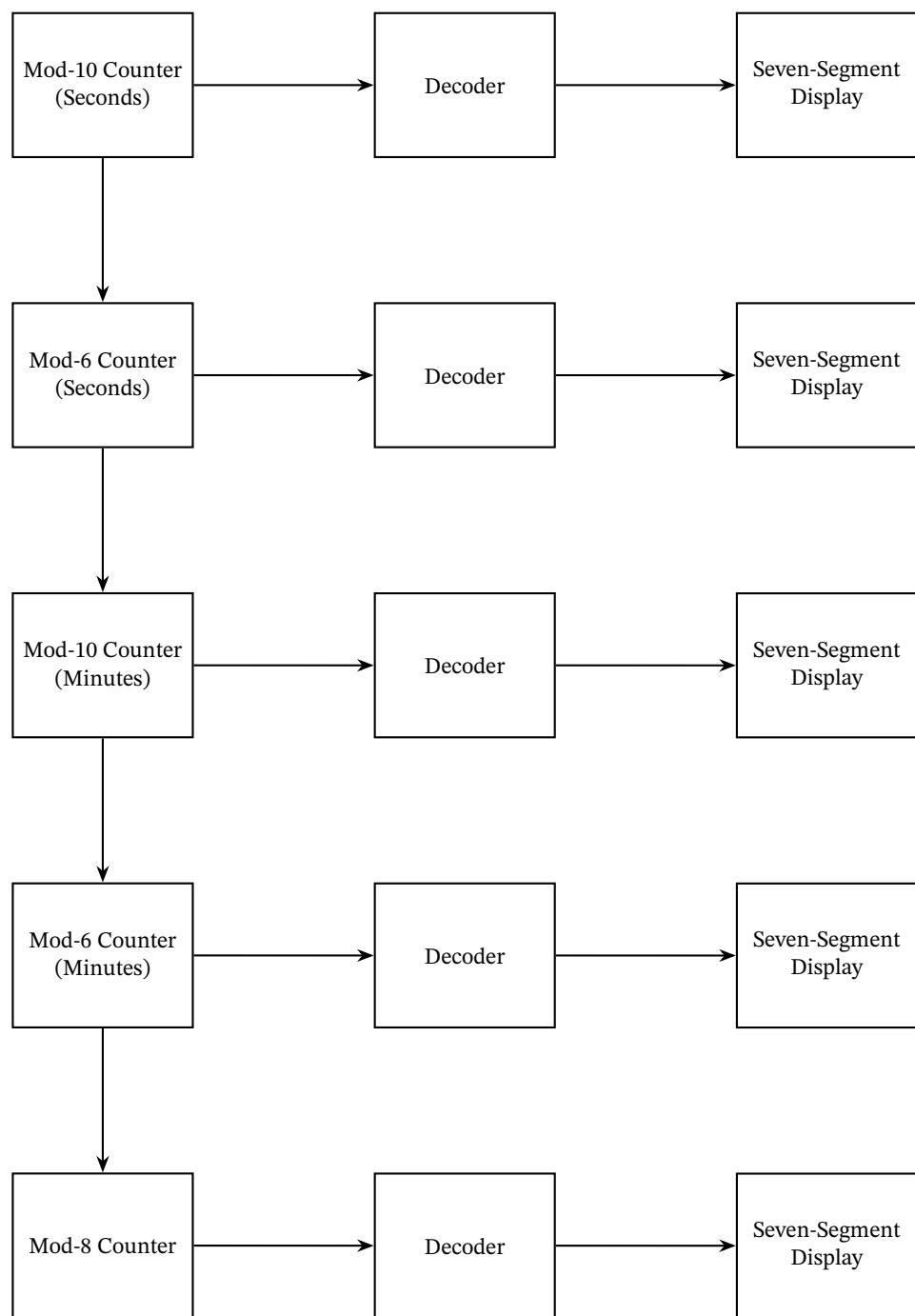


Figure 16. Block diagram of circuit