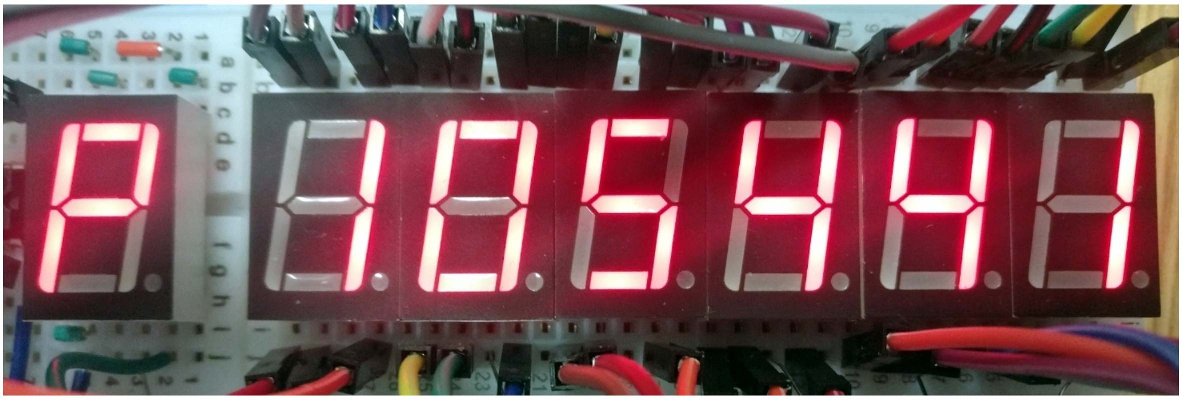
LED Digital Clock

EE111 Electric Circuits Project Final Report



Yifan Cao Yincen Xia

January 15, 2017

Contents

1 Overview Schematics **2**

2 Introduction **3**

3 Schematics **3**

3.1 Block schematics 3

3.2 Pulsing block 4

3.3 Counting block 5

3.4 Displaying block 6

3.5 Buzzing block 7

3.6 Notes for schematics 8

4 Simulation **9**

5 Practical Design **10**

5.1 Components 10

5.2 Practical design 11

6 Our Works & Thoughts **12**

6.1 Our works 12

6.2 Our thoughts 12

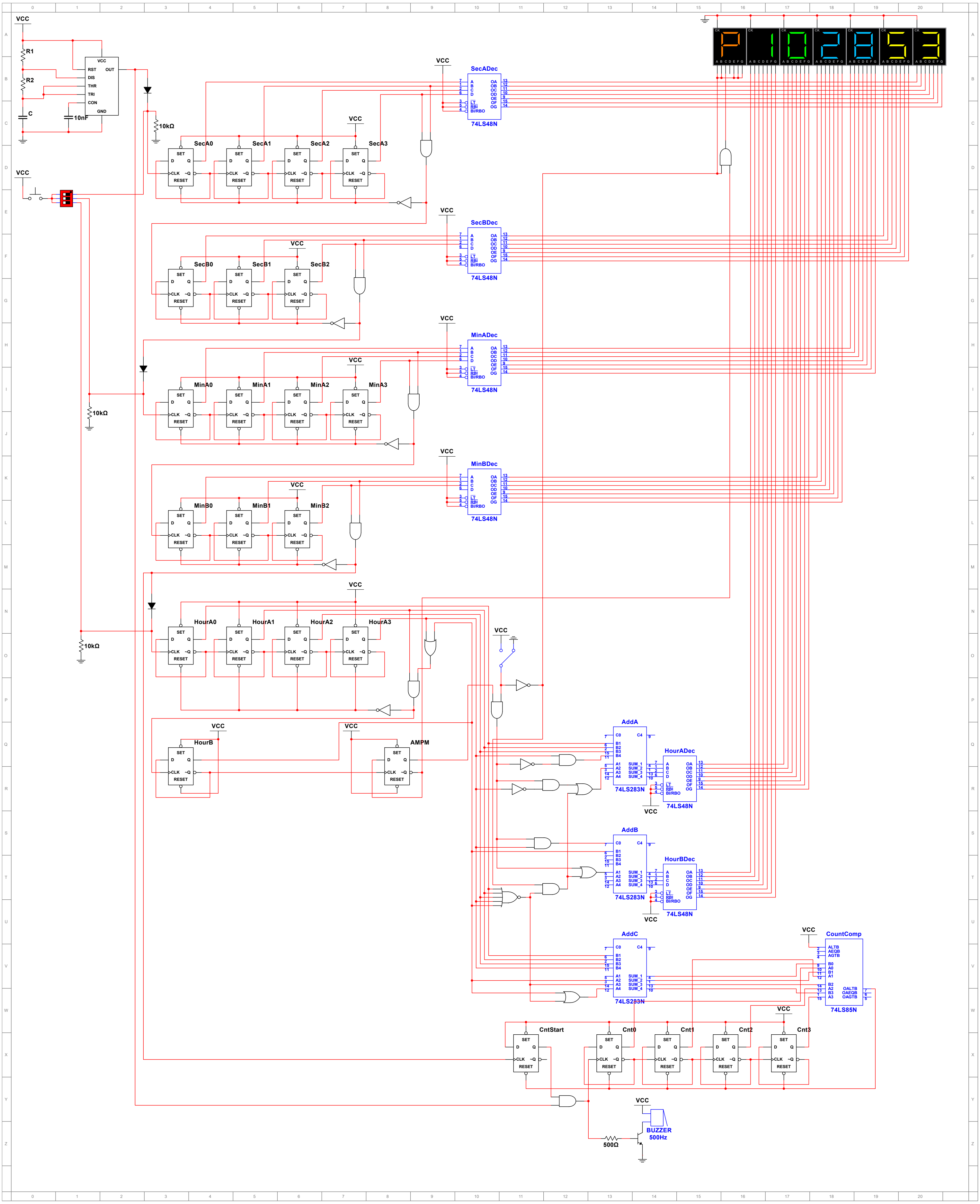
6.2.1 Yifan Cao 12

6.2.2 Yincen Xia 12

Acknowledgments **13**

References **13**

1 Overview Schematics



2 Introduction

A digital clock is a clock which can display the time in digits. Three units are often considered: hour, minute and second. To represent the time, a seven-segment LED is often used, two digits for each unit respectively. To store the time, there are internal counters for the units. The counters work as follows.

- A 1 Hz pulse is generated to make the 'second' counter count.
- Whenever the 'second' counter counts to 60, immediately, it resets to 0, while the 'minute' counter counts once.
- Whenever the 'minute' counter counts to 60, immediately, it resets to 0, while the 'hour' counter counts once.
- For the 'hour' counter:
  - In 12-hour format, there is a status representing AM or PM. Whenever the 'hour' counter counts to 12, immediately, it resets to 0 while the AM/PM status changes.
  - In 24-hour format, whenever the 'hour' counter counts to 24, immediately, it resets to 0.

Many advanced digital clocks may have larger units for date. However, that will not be covered in this project.

3 Schematics

Note: The overview schematics is on page 2.

3.1 Block schematics

In our project, we expect to design a digital clock with three units: hour, minute and second. Generally, there are four blocks: pulse generating, counting, displaying, buzzing.

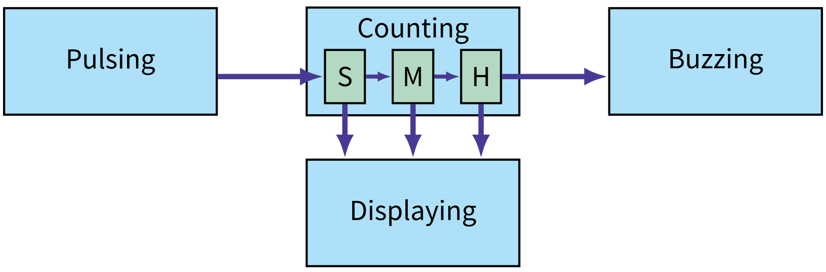


Figure 1 Block schematic: pulsing, counting, displaying, buzzing

In addition, three features are equipped:

Time adjustment. Using a knob, values for three units could be adjusted, one at a time.

Format switching. A switch is used to switch between 12-hour and 24-hour format, and the LED should display different values for hour according to the format we set.

The hour bell. Whenever the 'hour' counts (but not due to the adjustment by the knob), the buzzer buzzes for the same times as the value of 'hour' in 12-hour format, which signifies the hour (like the Big Ben).

3.2 Pulsing block

We expect to generate a 1 Hz pulse in order to make the 'second' count once a second. The 555 timer IC has a so-called astable mode. It works as follows.

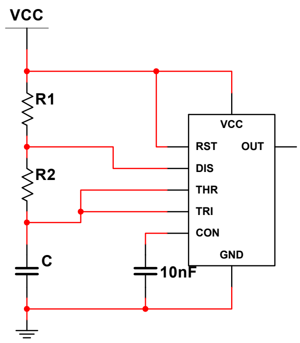


Figure 2 Pulse generating using 555 timer IC

1. Initially, the capacitor C stores no energy, so the trigger and threshold voltages are 0.
2. The trigger voltage is currently lower than $V_{CC}/3$, so effective immediately, the output is set to HIGH (1). The discharge switch opens, and the capacitor C charges through R_1 and R_2. The threshold (and trigger) voltage increases gradually.
3. As long as the threshold voltage increases to $2V_{CC}/3$, effective immediately, the output is set to LOW (0). The discharge switch closes, and the capacitor C discharges through R_2. The trigger (and threshold) voltage decreases gradually.
4. As long as the trigger voltage decreases to $V_{CC}/3$, go to step 2.

From the steps above, we find that there are two statuses: charging (with HIGH output 1) or discharging (with LOW output 0). From the RC circuit features, we yield the voltage on the capacitor C for each status, when the circuit goes to steady state.

- Charging status: From $V_{CC}/3$, to V_{CC} (ignoring the discharge switch).

  $$v(t) = V_{CC} + \left(\frac{1}{3}V_{CC} - V_{CC}\right)\mathrm{e}^{-t/\tau} = V_{CC}\left(1 - \frac{2}{3}\mathrm{e}^{-t/\tau}\right), \quad \text{where } \tau = (R_1 + R_2)C.$$

  This status ends as long as $v(t) = 2V_{CC}/3$, i.e. $t = \ln 2 \cdot (R_1 + R_2)C$, which is the HIGH time of the pulse.

- Discharging status: From $2V_{CC}/3$, to 0 (ignoring the discharge switch).

  $$v(t) = 0 + \left(\frac{2}{3}V_{CC} - 0\right)\mathrm{e}^{-t/\tau} = \frac{2}{3}V_{CC}\mathrm{e}^{-t/\tau}, \quad \text{where } \tau = R_2C.$$

  This status ends as long as $v(t) = V_{CC}/3$, i.e. $t = \ln 2 \cdot R_2C$, which is the LOW time of the pulse.

Thus the astable mode has frequency

$$f = \frac{1}{\ln 2 \cdot (R_1 + 2R_2)C}.$$

Then we decide R_1, R_2 and C to make $f = 1\,\text{Hz}$. Let $C = 100\,\mu\text{F}$ and $R_2 = 5\,\text{k}\Omega$, then $R_1 = 4.427\,\text{k}\Omega$. We can use a $10\,\text{k}\Omega$ potentiometer to represent R_1. However, we do not use these values during simulation, since it could be much slower.

3.3 Counting block

We know that second counts from 0 to 59. We could also consider it as, the second has two digits, one from 0 to 5, and the other from 0 to 9. So we focus on the counter from 0 to 9 first, which is the last digit in the clock.

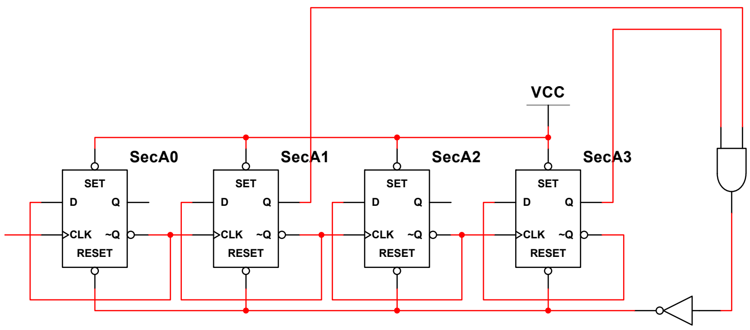


Figure 3 Counter for the second digit of second using D Flip-Flops

We use multiple D Flip-Flops (D-FF below) to achieve this. Initially, all the D-FFs have output $Q = 0$. As long as a pulse is generated, the rising edge of the clock is captured by the first D-FF. Before that, we have $D = \overline{Q} = 1$. So after that, $Q = 1$ and $D = \overline{Q} = 0$. We discover that the rising edge toggles the output each time.

What's more, if the toggle of Q is 1 to 0, that of $\overline{Q}$ is 0 to 1. Since the $\overline{Q}$ signal is sent to the clock input of the second D-FF, we conclude that as long as Q toggles from 1 to 0 for one D-FF, the next D-FF toggles. From right to left, the values of D-FFs could be: 0000, 0001, 0010, 0011, 0100, ... Then the D-FFs work as a 4-bit counter.

For the second digit of second, we want it to count to 9 (1001) instead of 15 (1111). Thus, as long as it counts to 10 (1010), i.e. the second and fourth bit are both 1 for the first time, we reset the D-FFs and a rising-edge signal is sent to the first digit of second (which counts from 0 to 5). An AND gate is used to achieve this. The NOT gate is added at the reset input since 0 is the effective potential.

For the first digit of second, we use only 3 bits. We want it to count to 5 (101), so reset as long as it counts to 6 (110), i.e. the second and third bit are both 1 for the first time, while a rising-edge signal is sent to the second digit of minute.

For the digits of minute, nothing new since the counting principles are the same as that for the digits of second, i.e. from 0 to 5 and from 0 to 9.

For the digits of hour, the counters work a bit differently. Here we store the hour as 12-hour format, i.e. two digits and one AM/PM status. For the second digit of hour, reset as long as it counts to 10 (1010), while a rising-edge signal is sent to the first digit. However, we should also reset as long as the first digit is 1 and the second digit is 2 (0010), while a rising-edge signal is sent to the first digit

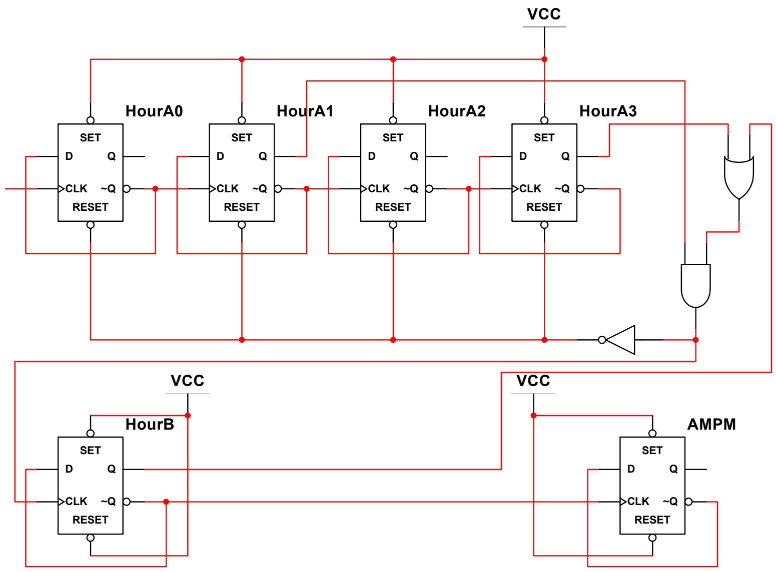


Figure 4 Counter for the digits of hour and AM/PM status

and the AM/PM status D-FFs, since they are both expected to toggle. The AND gate and the OR gate achieve what we expect above.

To adjust the time, we use a 3-bit DIP switch (for hour, minute and second) in series with a rotary encoder which is connected to V_{CC}. As we rotate the rotary encoder, pulses are sent to the clock inputs, which adjust the time.

3.4 Displaying block

It is simple for the display of minute and second. For each digit, we send the 4-bit outputs of D-FFs to a BCD to 7-segment decoder. Then we send the 7-segment outputs to a 7-segment LED. That's it.

However, it's not such simple for the display of hour. We expect that:

- For the 12-hour format, display the AM/PM status as a letter A or P on the 7-segment LED. Also, display 0 as 12. (e.g. we accept 12:00:00 PM, but not 00:00:00 PM!)
- For the 24-hour format, if it is PM, add 12 before displaying. (e.g. 1 PM is 13.)

However, there are two digits for hour. For the second digit, what we actually expect is:

- If the hour is 0 in 12-hour format, or it is PM in 24-hour format and the second digit is not 8 or 9 (this means that the fourth bit is 0), add 2.
- If it is PM in 24-hour format and the second digit is 8 or 9, change 8 to 0 and 9 to 1. In other words, change 1000 to 0000 and 1001 to 0001. Only the fourth bit is changed!

And for the first digit, what we actually expect is:

- If the hour is 0 in 12-hour format, or it is PM in 24-hour format, add 1.
- If it is PM in 24-format and the second digit is 8 or 9, add another 1.

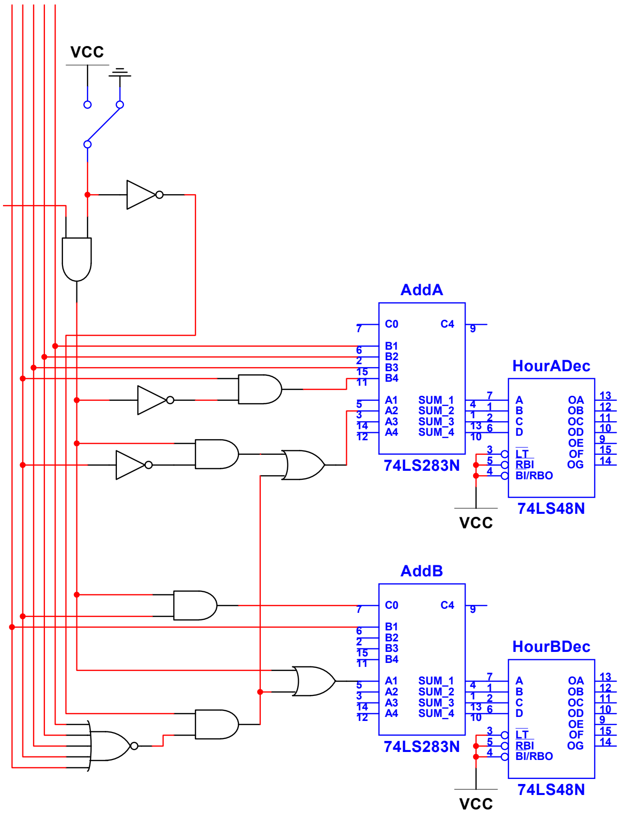


Figure 5 Displaying hour. The five wires on the left represents the first digit, and 4 bits of the second digit of hour. The wire on the AND gate represents PM. The switch is now in the 12-hour format state.

We process the outputs of hour carefully using logical gates and adders. Then we could send the results to the decoders, then to the LEDs. As we mentioned above, a switch is used to switch between 12-hour and 24-hour format.

To display the AM/PM status as a letter A or P for the 12-hour format, for the 7-segment LED, A, B, E, F, G are always 1, D is always 0, and C is 1 only when it is AM (A and P look different only for the C segment). But for the 24-hour format, the LED should not display.

3.5 Buzzing block

We set up another counter, and it counts as the same times as the value of hour in 12-hour format before reset. Actually, we compare the value of this counter and that of hour. There are 5 bits for hour since there are two digits. It is quite annoying, since we could not use only one 4 bit comparator directly. But we do not need 5 bits in this block, since we do not display. The hour counts from 1 to 12, and it definitely can be represented by 4 bits!

So we first transform the 5-bit hour to 4-bit as follows.

- If the highest bit is 1, add 10 (1010) to the last 4 bits.
- If the hour is 0, add 12 (1100) to the last 4 bits.

Notice that 10 and 12 are considered as 4 bits instead of 5. We make the transformation carefully using logical gates and a adder. Then we could send the results to the comparator, which stands for the 4-bit hour.

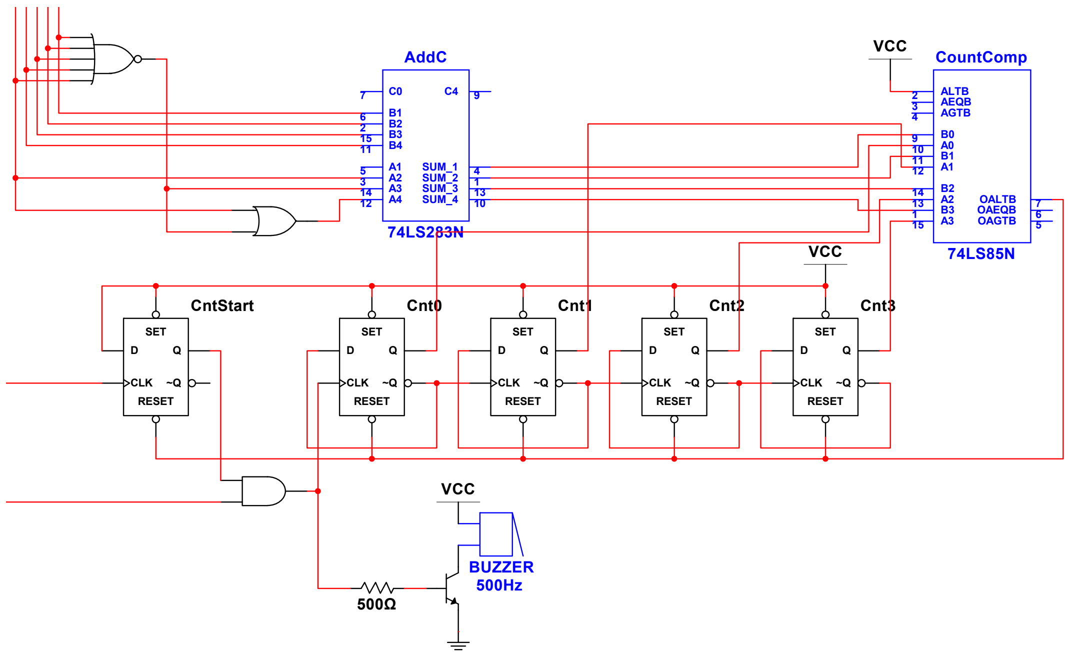


Figure 6 Buzzing block. The five wires on the left represents the first digit, and 4 bits of the second digit of hour. The wire on the input of AND gate is from the output of 555 timer. The wire on the clock input of the D-FF is from the hour, producing a rising edge if the hour counts once.

The 4-bit counter starts to count as long as the hour counts once (the rising edge of the clock could be captured). Effective immediately, the first D-FF has output $Q = D = 1$. An AND gate is applied to this output and that from 555 timer, which shows that a pulse of 1 Hz is used for counting as long as the count starts, and the rising edge makes the counter counts to 1 (0001) immediately.

Next, in a 1-second period, if it is 1 from the 555 timer, the buzzer buzzes. If 0, it does not. The buzzer buzzes only when it is 1 from the 555 timer, so it sounds intermittently, which is what we expect.

The comparator compares the values of counters for the buzzer (A) and hour (B). If $A \leq B$, the buzzer should buzzes, so it buzzes B times. On the comparator, we make $I_{A<B} = 1$, which means that $O_{A<B} = 1$ when $A = B$. As long as $O_{A<B} = 0$, i.e. $A > B$, we reset the counter for the buzzer and end the buzz immediately.

3.6 Notes for schematics

- For simplicity, we do not add resistors for all 7-segment displays. They should be 500 Ω. Also, we do not use all input pins — they should be connected to the GND practically.
- There is a 5-input NOR gate. Practically, we will use 2-input OR gate and NOT gate to represent this.

4 Simulation

We start the clock in 24-hour format and it is 22:00:02. It should display 2 digits for hour, minute and second, respectively. If we switch to 12-hour format, it displays 10:00:02 PM.

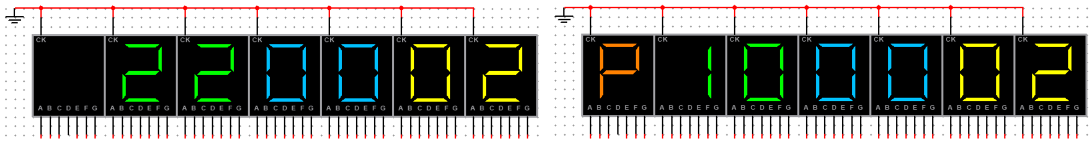


Figure 7 The clock at 22:00:02, which is also 10:00:02 PM

It is 22:00:59. The next second causes a carry to minute, and then it is 22:01:00.

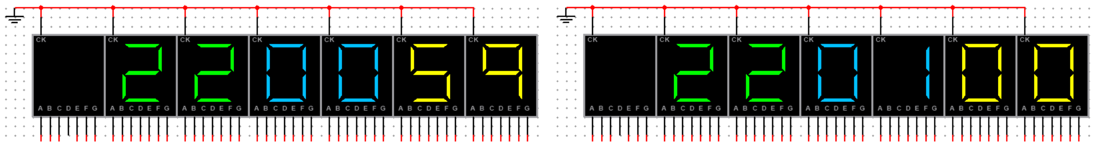


Figure 8 The carry to minute

It is 23:59:59. The next second causes a carry to minute, then to hour. The hour goes back to 0, so it is 00:00:00. The buzzer buzzes 12 times since it's 12 AM.

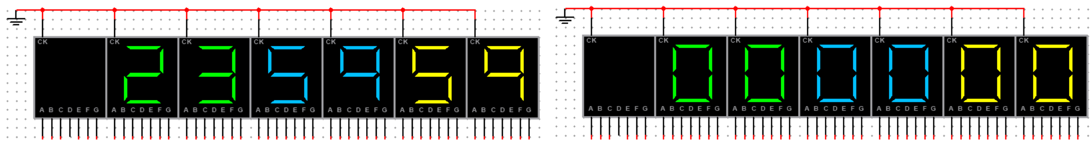


Figure 9 The carry to minute and hour while the hour goes back to 0

If we switch to 12-hour format, it displays 12:00:00 AM.

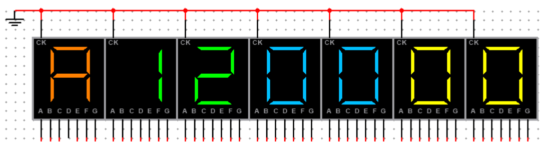


Figure 10 00:00:00 is also 12:00:00 AM

5 Practical Design

5.1 Components

Components that we will use in the practical circuit is listed below. We purchase them on the Internet[1]. For basic components like resistors and capacitors, we use the ones provided in the lab.

Table 1 Components

Name	Model	Quantity
Quad 2-Input NOT Gate	74LS04	3
Quad 2-Input AND Gate	74LS08	4
Quad 2-Input OR Gate	74LS32	3
BCD to 7-Segment Decoder	74LS48	6
Dual D Flip-Flop	74LS74	13
4-Bit Full Adder	74LS83	3
4-Bit Comparator	74LS85	1
555 Timer	NE555	1
7-Segment Display		7
Active Buzzer		1
Breadboard (extensible)		13
Rotary Encoder	EC11	1
Toggle Switch	SPDT On-On	1
DIP Switch	3-Pin	1
Power Supply Module	5 V	1
Resistor	510 Ω	8
Resistor	5 kΩ	5
Capacitor	100 μF	1
Diode		3
NPN BJT	9013	1

[1]The websites are `https://risym.tmall.com`, `https://shop106050244.taobao.com`. For your information, we listed the components we need before we purchased them, which was on Google Sheets: `https://docs.google.com/spreadsheets/d/1yNDflRI-cFFP55Oon6irWy0gpXRfURwcjaKoz-rEdtc/edit?usp=sharing`

5.2 Practical design

We use 13 extensible breadboards to finish our practical design.

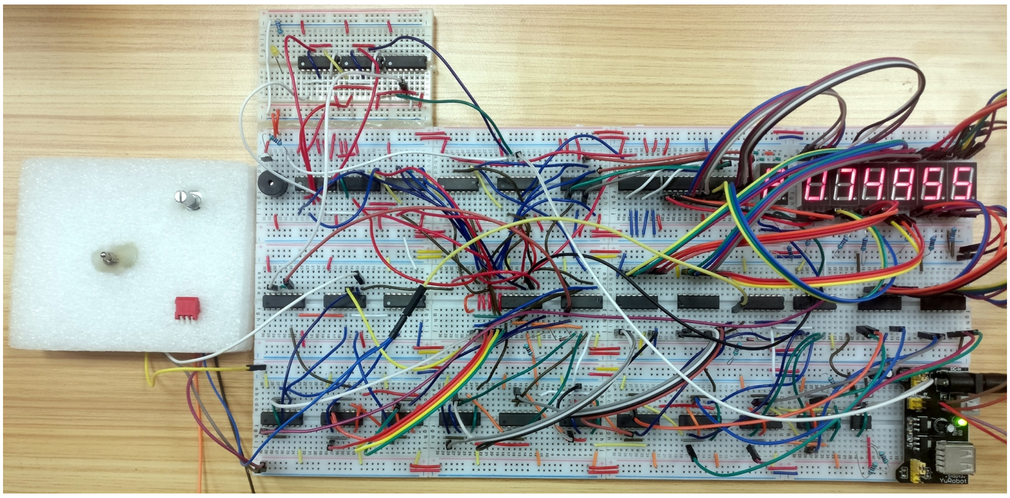


Figure 11 Practical design

6 Our Works & Thoughts

6.1 Our works

We held a discussion, then designed and simulated the pulsing and counting block in a relatively short time. Yifan suggested that two features could be added: format switching (12-hour and 24-hour) and the hour bell (like the Big Ben). Thankfully, Yifan was able to implement (simulate and practically run) the two features in the end, with a quite simple method. These two features were mainly done by Yifan, while Yincen provided many critical opinions during his work.

After the simulation, we listed the components needed and purchased them on the Internet. Yincen recommended a Tmall store called Risym, which was most our components came from.

In the construction of the practical circuit, Yincen built a 'control center' — a piece of foamed plastic, with rotary encoder, toggle switch and 3-pin DIP switch fixed and Dupont line soldered. He also helped to decide the position of the 34 chips, so that we could use wires as short as possible. Yifan helped with the rest of the circuit, especially the wires on the logical gates. He used an LED to detect if a digital signal was HIGH or LOW, which is my major method of debugging.

Yincen helped with the entire design of the poster. For the contents, Yifan derived them from the interim report and made a few modifications.

6.2 Our thoughts

6.2.1 Yifan Cao

At first, I suggested choosing this project, since I think that it could be fun. But I found that it was really difficult to deal with the digital signals (0 or 1). We used 34 chips, and many of them were flip-flops and gates. If an MCU was allowed, that would be many lines of codes instead, which could be much easier, I think.

I would like to say that the wiring scheme was *very* complex, and critical. In other words, a small mistake could cause a bug or burn a chip. To be honest, if a second chance was provided, there was a high probability that I did not choose this project again. However, I did have experienced the complete construction process of a large project, and did realize that how ICs could be important today. Nowadays, when we are enjoying the products with highly integrated chips, we should pay honor to the engineers who built them, since many products we are using today are their masterpieces.

This is my first project. Much stuff was new to me, which encouraged me to think. For many times, I together with Yincen stayed in front of the circuit and neglected meals. One of the most important harvests was the experience I gained during the whole process. Also, thanks to my partner Yincen Xia, who helped me out when I was in trouble every time.

6.2.2 Yincen Xia

Frankly speaking, the difficulty of this project is much higher than we expected when choose this project. Thanks to Yifan, who is so proactive, we finished our project and even added some extra feature into it finally.

Since I'm an IoT developer, I've done several electronic projects before this one, so I'm quite familiar with how to make the real circuit, such as some practical skills about where to choose and buy components. And I'm also a designer for university's magazine, so it is not hard for me to design a poster for the project with the help of Yifan, who provided the contents.

Although this project cost lots of time and far more than 100 yuan, I think it is worthwhile. Yifan have done lots of work on simulate the circuit and connect components together, and I tried my best to assist him. It's really a nice experience for me to cooperate with such a diligent teammate. I not only learn from the project itself, but also learn a lot from him.

Acknowledgments

This project requires a large amount of work, and we didn't finish our work until the day before the presentation. Thanks to the Teaching Assistant Shuo Ni, who provided necessary guidance for us. Meanwhile, thanks to the Professor Pingqiang Zhou, who delivered dozens of high quality lectures, which leads us to the wonder zone of electric circuits.

Yifan Cao & Yincen Xia

January 15, 2018

References

[1] Wikipedia. *555 timer IC*. `https://en.wikipedia.org/wiki/555_timer_IC`

[2] Wikipedia. *D flip-flop*. `https://en.wikipedia.org/wiki/Flip-flop_(electronics)#D_flip-flop`

[3] Wikipedia. *Seven-segment display*. `https://en.wikipedia.org/wiki/Seven-segment_display`

[4] Learn About Electronics. *Digital Counters*. `http://www.learnabout-electronics.org/Digital/dig56.php`