Digital Design of an Automated Irrigation System

ENGG 121.02-A
Final Project

Andreas Josef C. Diaz
Simon Fredrick J. Fernan

Background

The creators have been interested in creating an automatic irrigation system as they have taken up gardening as a hobby. They have learned that it is essential to be regular in watering plants or never grow well. However, there are times when people have no time to water them, especially in busy times such as a "finals week." The creators then thought of implementing this system for them to have more time to do other works and still fulfill the needs of their plants.

The creators thought that this would be the gist of the system:
Create an automatic irrigation system that would turn on two times a day. The user can set the times for watering. In addition to this, the irrigation system would not start through the user-set times if it had already rained that day. However, the user can also begin the automated irrigation system manually. Lastly, if the plants' soil is super dry, the system should open (considered an emergency).

This system utilized logic circuits in its implementation in Multisim and is discussed further in this document.

Limitations and Assumptions

During the study, several limitations and assumptions were found and considered. First, the soil moisture sensors were assumed to return a digital 5 Volt signal to the super dry soil input. Moreover, the sensor is capable of getting measurements of the whole area. Second, the user can only set two irrigation times through the user interface. Furthermore, the activation period of the pump is limited to 1 minute. Third, the clock circuit is assumed to synchronize with the user's timezone. Fourth, the frequency used for testing will be faster than the actual time to have more time-efficient testing. Notably, it should be four times quicker than a second. Fifth, the rain checker will disable the comparator system for the whole day, assuming that there will be heavy rain every time it rains.

Block Diagram

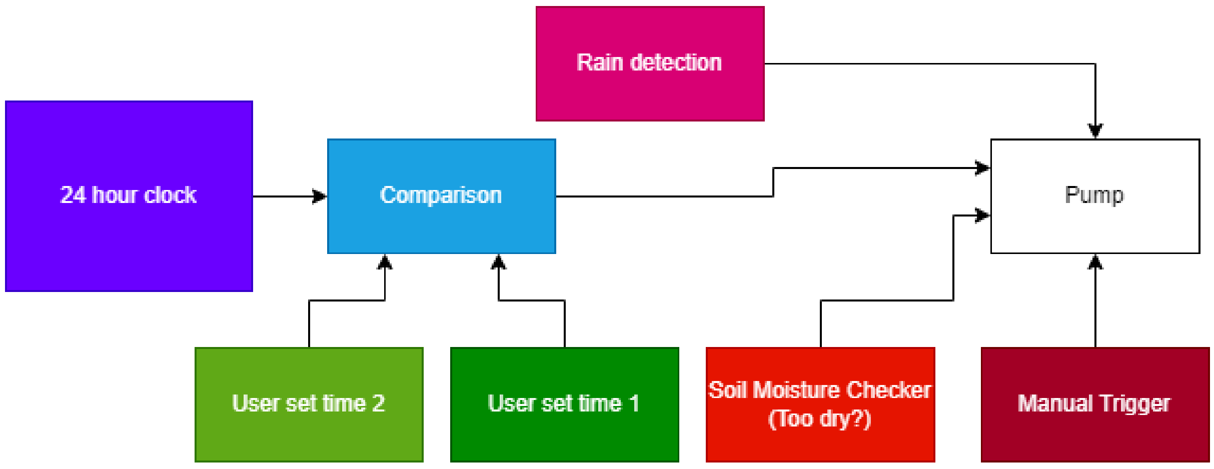


Fig. 1 Block Diagram for the Automated Irrigation System

ASM Diagram

Given the assumptions and limitations set for the system,

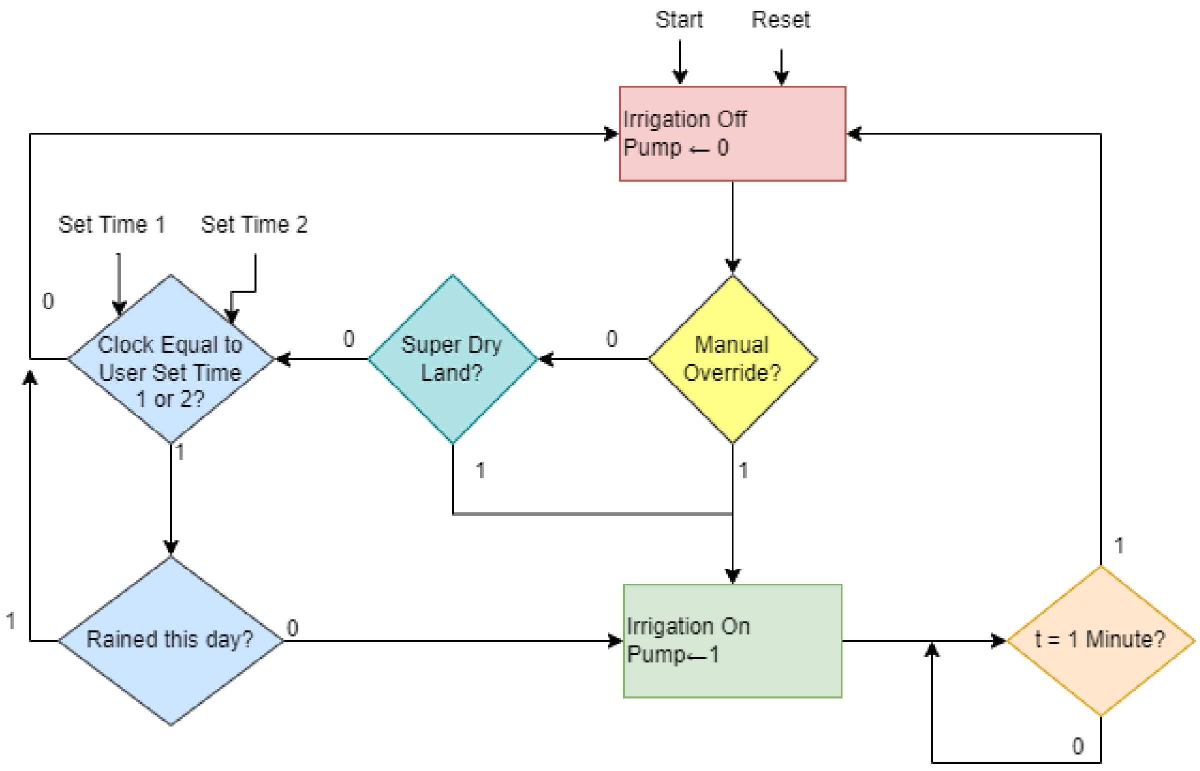


Fig. 2 ASM Diagram for the Automated Irrigation System

Discussion for the Circuit

The whole circuit comprises the pump system, the clock comparator, and the rain checker.

- **Pump System and Countdown Subsystem**
  The pump system is the brain of the whole system, taking all of the inputs considered in the ASM diagram (Fig. 2). Notably, the pump should open either when the manual override button is on, when the super dry sensor turns on, or when the clock is equal to the user set times and when it did not rain that day. In addition to this, the pump should turn off after a minute.

  Flip-flops would be used to maintain the present state of the pump system. However, there are different types of flip-flops. Therefore, the creators focused on implementing a D or a JK flip-flop.

  Table I summarizes the transition of the pump system. Note that the system's state is equal to its output.

TABLE I. General Transition Table for the Pump System

Present State	Inputs					Next State	Flip-Flop Inputs		
	Manual Override	Super Dry?	Time Equal?	Did it Rain Today ?	Minute passed ?				
A	B	C	D	E	F		D	J	K
0	1	x	x	x	x	1	1	1	x
0	0	1	x	x	x	1	1	1	x
0	0	0	1	0	x	1	1	1	x
0	0	0	1	1	x	0	0	0	x
1	x	x	x	x	0	1	1	x	0
1	x	x	x	x	1	0	0	x	1

  Based on Table I and the K-map in Fig. 3, the equation for the D flip-flop input is:

$$D_{Pump} = A'DE' + A'B + A'C + AF'$$

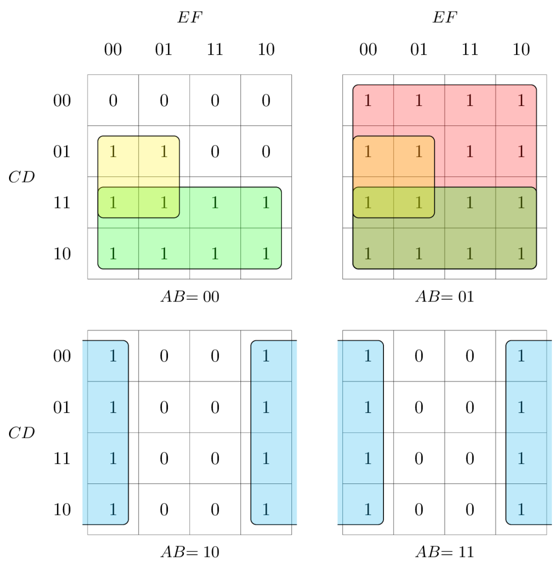


Fig. 3 K-Map for the D Flip-Flop in the Pump System

Based on Table I and the K-maps in Fig. 4 and Fig. 5, the equations for the JK flip-flop inputs are:

$$J_{Pump} = C + B + DE'$$

$$K_{Pump} = F$$

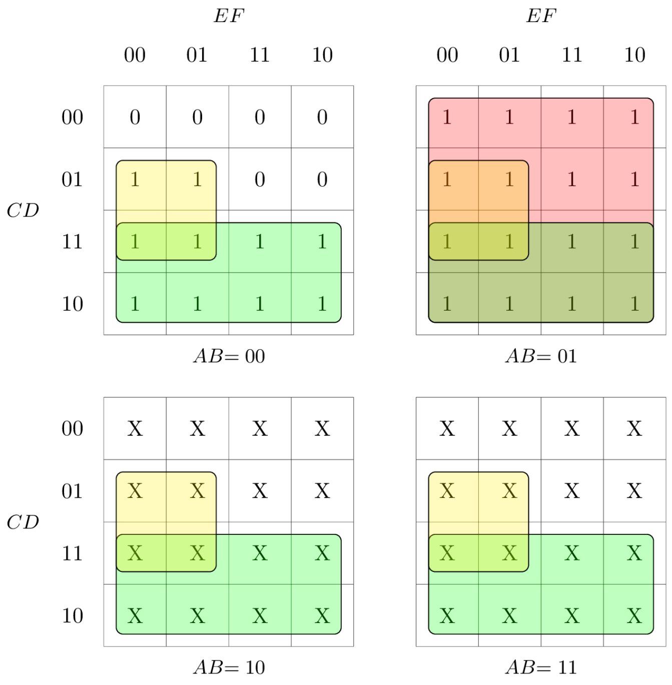


Fig. 4 K-Map for the J Input in the Pump System

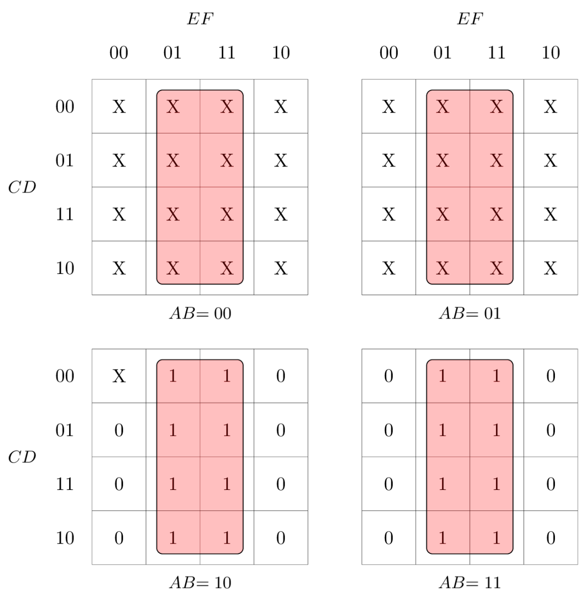


Fig. 5 K-Map for the K Input in the Pump System

In addition to this pump system utilizing a flip-flop, a subsystem should count how long the irrigation would turn on. Counters would be appropriate for this application, specifically the 74190 IC, as this specific IC can count down. Furthermore, the datasheet of the 74190 IC from Motorola states that multiple counter ICs can utilize the RCO' pin to cascade them [1]. Therefore, the subsystem would have to use two 74190 ICs.

Once the counter goes to 00, it should output logic 1 to the pump system and then reset it back to logic 0, turning the irrigation pump off. Additionally, the counter should reset to the minute (60) after 00.

The subsystem should utilize a flip-flop to latch on to a state, preferably logic 1 so that it would not always reset the counter and would only go to state 0 when the counter goes to 00. State 0 would turn off the irrigation pump and reset the countdown subsystem. Moreover, the RCO' pins of the two 74190 ICs can help tell the flip-flop of the countdown that the counter is already 00. Table II shows the general transition of the counter flip-flop.

TABLE II. General Transition Table for the Countdown Subsystem

Present State	Inputs			Next State	Outputs		Flip-Flop Inputs		
	Pump System	RCO' Left Counter	RCO' Right Counter		Q	Q'			
G	H	I	J				D	J	K
0	0	x	x	1	0	1	0	0	x
0	1	1	x	1	0	1	0	0	x
0	1	x	1	1	0	1	0	0	x
0	1	0	0	0	0	1	1	1	x
1	x	x	x	1	1	0	0	x	1

Based on Table II and the K-map in Fig. 6, the equation for the D flip-flop input is:

$$D_{Countdown} = G'HI'J'$$

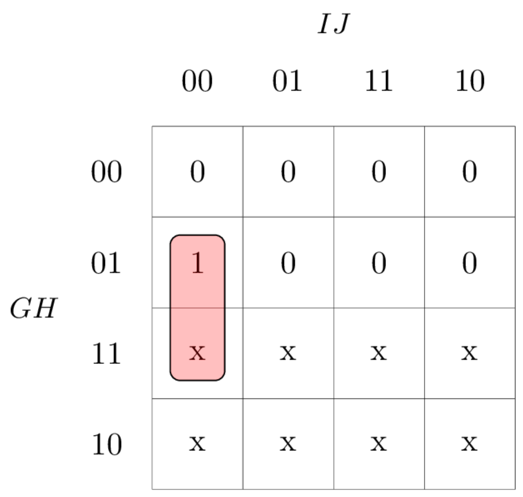
$$\begin{array}{cc|cccc}
 & & & IJ & & \\
 & & 00 & 01 & 11 & 10 \\
\hline
 & 00 & 0 & 0 & 0 & 0 \\
GH & 01 & \boxed{1} & 0 & 0 & 0 \\
 & 11 & 0 & 0 & 0 & 0 \\
 & 10 & 0 & 0 & 0 & 0
\end{array}$$

Fig. 6 K-Map for the D Input in the Countdown Subsystem

Based on Table II and the K-map in Fig. 7 and Fig. 8, the equations for the JK flip-flop input are:

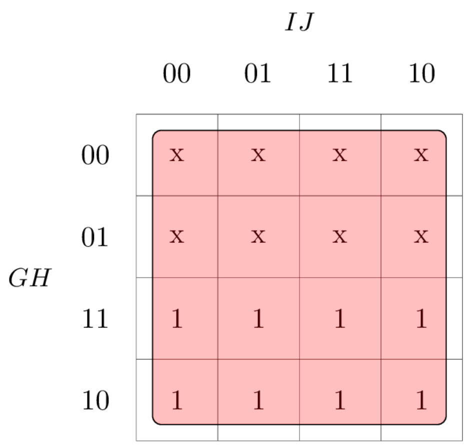
$$J_{Countdown} = HI'J'$$

$$K_{Countdown} = 1$$



GH \ IJ	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	x	x	x	x
10	x	x	x	x

Fig. 7 K-Map for the J Input in the Countdown Subsystem



GH \ IJ	00	01	11	10
00	x	x	x	x
01	x	x	x	x
11	1	1	1	1
10	1	1	1	1

Fig. 8 K-Map for the K Input in the Countdown Subsystem

Implementing the whole pump and countdown system used the equations found above. Fig. 9 shows the D flip-flop implementation of the systems, while Fig. 10 shows the JK flip-flop implementation. Notice that additional SPDT switches connect to the ~CLR pins of the flip-flops. In Fig. 2, the ASM diagram, the user can reset the system and stop the irrigation pump. The SPDT switches (which have the hotkey 9) ensure that both the irrigation and countdown would reset to state 0 for the irrigation flip-flop while the counters should show “60.”

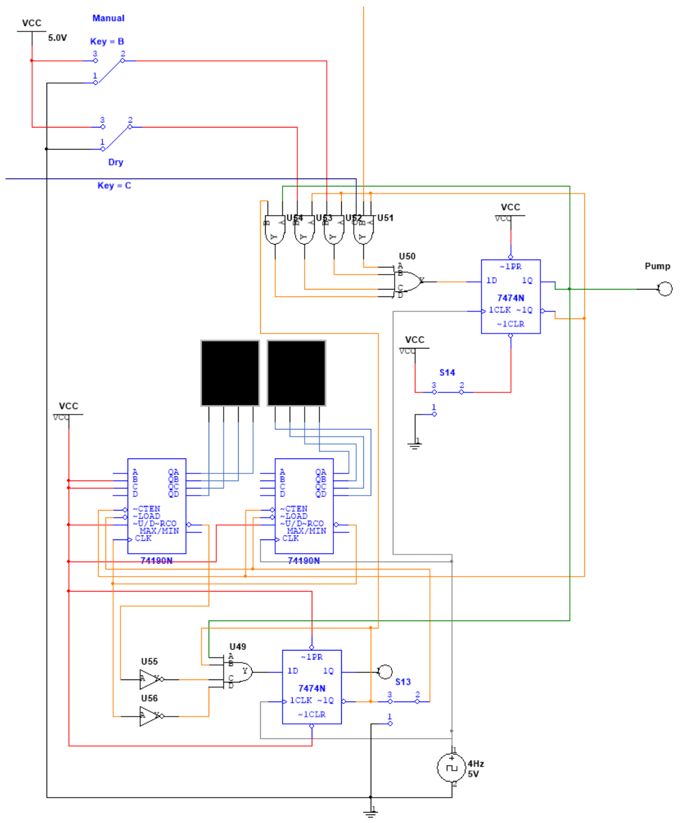


Fig. 9 D Flip-Flop Implementation of the Pump System and Countdown Subsystem

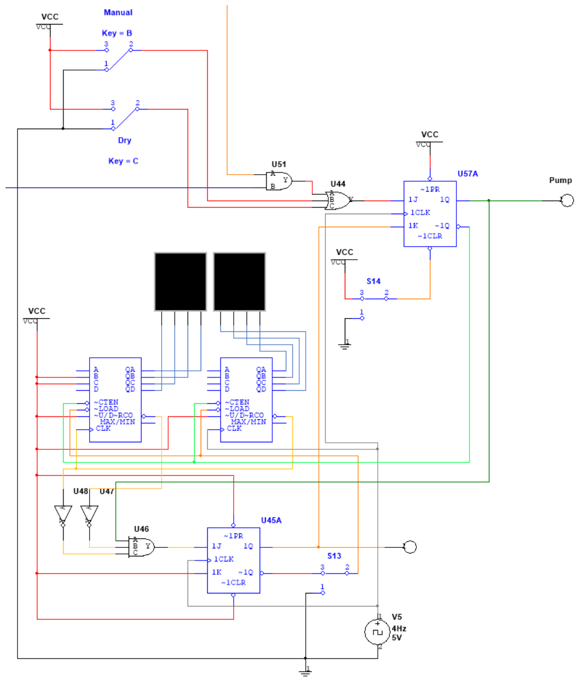


Fig. 10 JK Flip-Flop Implementation of the Pump System and Countdown Subsystem

- **Time Comparator**

  The time comparator is the sum of three different parts. First, the clock circuit keeps track of the time. Second, the user input circuit stores the two times selected by the user. Lastly, the comparator network constantly compares the user set times to the clock.

  To begin with, both the clock circuit and the user interface bear similarities in the way each works. For instance, Fig. 11 and Fig. 12 show both components using counter ICs going through 0-60 and 0 -24, which are the minutes and hours, respectively. However, how the counter ICs are enabled differs in that a square wave function generator triggers the clock circuit in a chain. At the same time, the user interface has the button triggers of the hours and minutes individually via the clock pins. Moreover, upon startup of the system, the counters open up to either 01:01 or 01:00. This is due to the RCO' pin starting as low because it begins at 00. After that, it will move to 01, which then turns RCO' to high. To remedy the issue, reset buttons turn all counters back to 0. In addition, the clock circuit also has a pause switch used during development and can be used to pause the clock to set up the time and temporarily disable the comparator system.

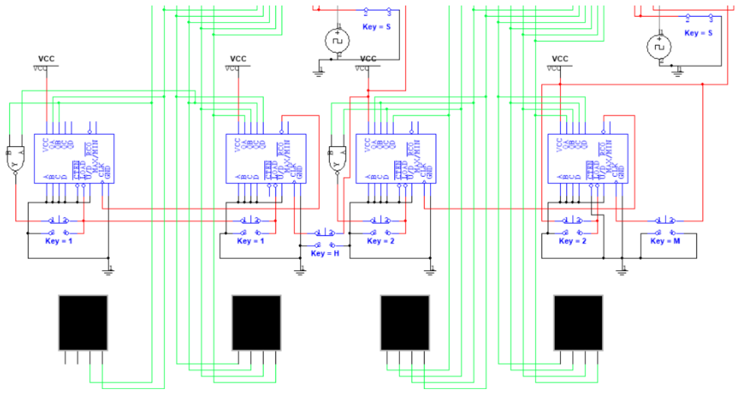


Fig. 11 User Input Circuit of the Time Comparator Component

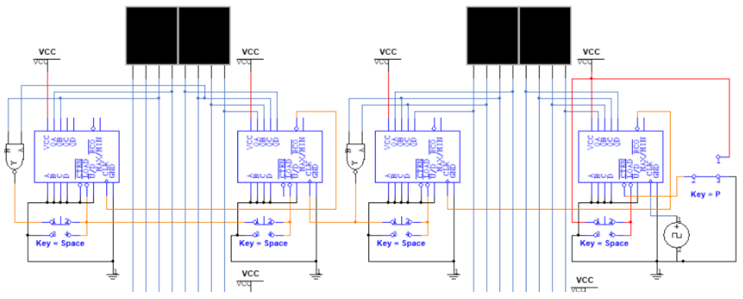


Fig. 12 Clock Circuit of the Time Comparator Component

The 16-bit output of the clock is compared to the two 16 bit outputs of the user interface. This comparison between the clock and user set times is made with the help of several 4585B comparator ICs that have been set to “OAEQB,” which means that the circuit will output a positive signal when both A and B are equal to each other [2]. For instance, 1 in A and 1 in B, as well as 0 in A and 0 in B, will both output a positive signal in the comparator. However, one thing to note is that though the system uses 16 bits, only 14 are utilized. This is because two bits from the second digit of the hour counter are only used. These two bits make up the only three possible numbers 0, 1, and 2. Falling back towards the comparators, the original plan made by the students was in the application of several XNOR logic gates as Table III shows that the truth table for XNOR is similar to comparator ICs in equal mode. However, the plan ran into software limitations in the number of components that could be simulated. As a result, the students switched from Electronic Workbench to Multisim. In addition, the discovery of 4-bit comparator ICs made the final push into adopting the new method of comparing the 16-bit values.

TABLE III. XNOR Truth Table

Inputs		Output
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1

It can be seen in Fig. 12 that there is another part than just the user input part. These other ICs are composed of two 8 bit shift registers used to remember one of the user set times. This is done by connecting each of the set pins to the outputs of the user input clocks. From there, a switch turns on and off the circuit’s ability to retain the time set. This results in storing the first set time, and then the counter ICs mentioned above will save the second user set time.

- **Rain Checker**

  The last part of the system is the rain checker. As the name suggests, the primary function of this part is to check if it rained within the day.

  Most of the parts are simple when looked at individually. Firstly, the day checker is just a set of comparators to see if the clock is already at 00:00. Once that condition is met, it will send a pulse towards the logic circuit seen before the D flip-flop. This logic circuit checks the current states of the D flip-flop, rain switch trigger, and the set of comparators. If these inputs and previous states match certain conditions, it will trigger the D flip-flop. The D flip-flop is the final step of the rain checker. Its only function is to toggle between outputting high or low depending on the inputs. Table IV summarizes the general behavior of the rain checker.

TABLE IV. General Transition Table for the Rain Checker

Present State	Inputs		Next State	Flip-Flop Inputs		
	Rained this day?	New Day? (00:00)				
L	M	N		D	J	K
0	0	x	0	0	0	x
0	1	x	1	1	1	x
1	x	0	1	1	x	0
1	x	1	0	0	x	1

Based on Table IV and the K-map in Fig. 13, the equation for the D flip-flop input is:

$$D_{RainChecker} = L'M + LN'$$

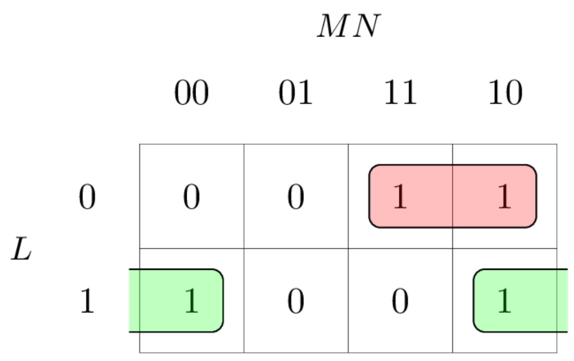


Fig. 13 K-Map for the D Input in the Rain Checker

Based on Table II and the K-map in Fig. 14 and Fig. 15, the equations for the JK flip-flop input are:

$$J_{RainChecker} = M$$

$$K_{RainChecker} = N$$

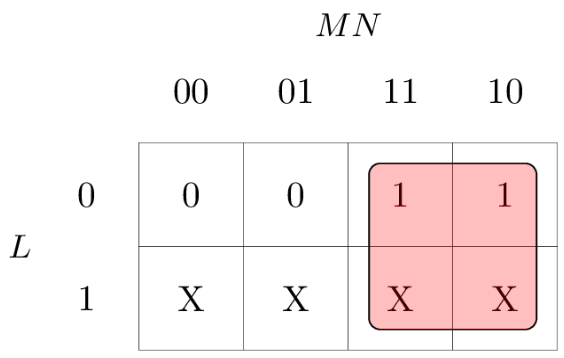


Fig. 14 K-Map for the J Input in the Rain Checker

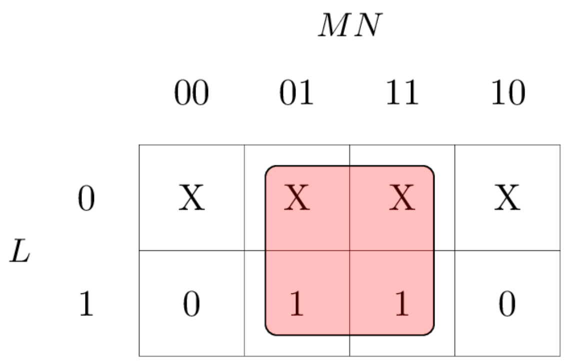


Fig. 15 K-Map for the K Input in the Rain Checker

Fig. 16 shows the D flip-flop implementation of the rain checker, while Fig. 17 shows the JK flip-flop implementation.

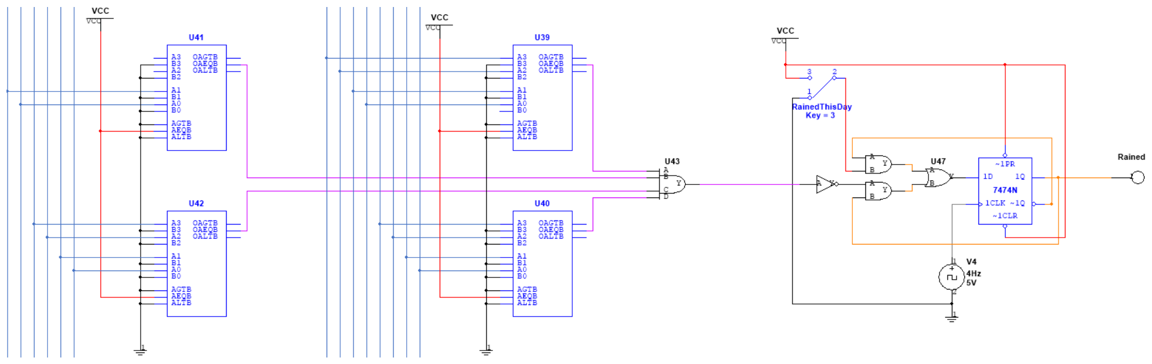


Fig. 16 D Flip-Flop Implementation of the Rain Checker

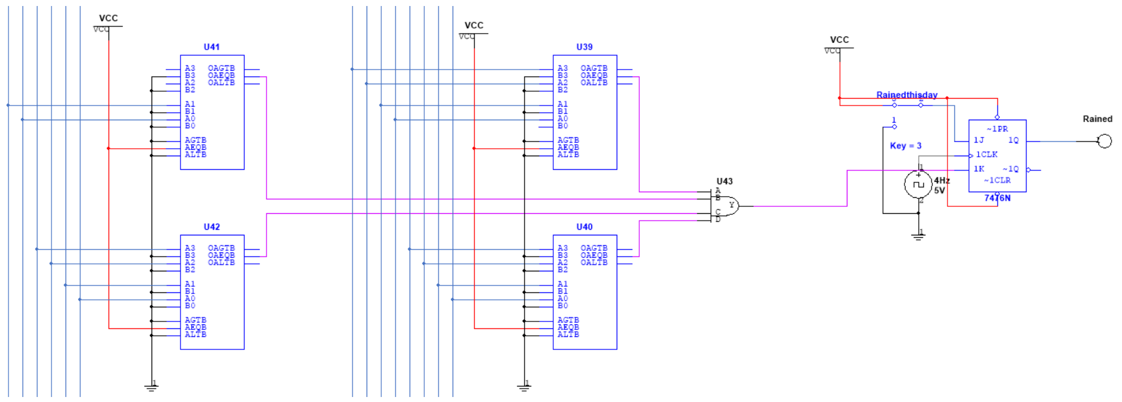


Fig. 17 JK Flip-Flop Implementation of the Rain Checker

Summary of Components

Tables V and VI show all the number of components that the whole circuit requires and the number of possible ICs used as the simulation in Multisim did not utilize the equivalent ICs on logic gates. Notice that overall, the JK Flip-Flop implementation uses fewer components and ICs. Furthermore, note that the implementation can reduce the number of ICs by utilizing the unused pins of other ICs with the same logic.

TABLE V. List of Components used for the D Flip-Flop Implementation

Component	**No. of items**	**No. of possible ICs**	**Description**
74LS190D	10	10	Counter IC
74198N	2	2	8-bit shift register
7474N	3	2	D flip-flop with

			set and reset
4585BD_5V	12	12	4-bit comparator
AND gate (2-input)	5	2	
AND gate (3-input)	1	1	
AND gate (4-input)	4	4	
OR gate (2-input)	2	1	
OR gate (4-input)	1	1	
NOT gate	3	1	
NAND gate (2-input)	4	1	
TOTAL	**47**	**37**	

TABLE VI. List of Components used for the JK Flip-Flop Implementation

Component	**No. of items**	**No. of possible ICs**	**Description**
74LS190D	10	10	Counter IC
74198N	2	2	8-bit shift register
7474N	3	2	D flip-flop with set and reset
4585BD_5V	12	12	4-bit comparator
AND gate (2-input)	1	1	
AND gate (3-input)	1	1	
AND gate (4-input)	3	3	
OR gate (2-input)	2	1	

OR gate (3-input)	1	1	
NOT gate	3	1	
NAND gate (2-input)	4	1	
TOTAL	**42**	**35**	

D Flip-Flop vs. JK Flip-Flop

As mentioned previously, the system can use flip-flops to maintain the state of the pump. The students kept in mind the two flip-flops that the students kept in mind were the D flip-flop and JK flip-flop. Siding with the former, D flip-flops have only one input. Ideally, it would mean that the necessary components needed to use the device must be simple. But, in contrast, that was not the case, as seen in Fig. 9.

On the other hand, a JK flip-flop has two inputs: a set and the other as a reset. But, looking at Fig. 10, it can be seen that though there are more necessary input pins, the flip-flop requires fewer preprocessing logic gates than the D flip-flop.

Additionally, Fig. 16 and Fig. 17 show the massive difference between D flip-flops and JK flip-flops. Notice that Fig. 16 uses four logic gates all in all; however, Fig. 17 uses no logic gates. This is further supported by the summary of components wherein there is a difference between two ICs in favor of JK flip-flops. Therefore, it is better to implement this system using JK flip-flops.

Complete Circuit using JK Flip-Flops

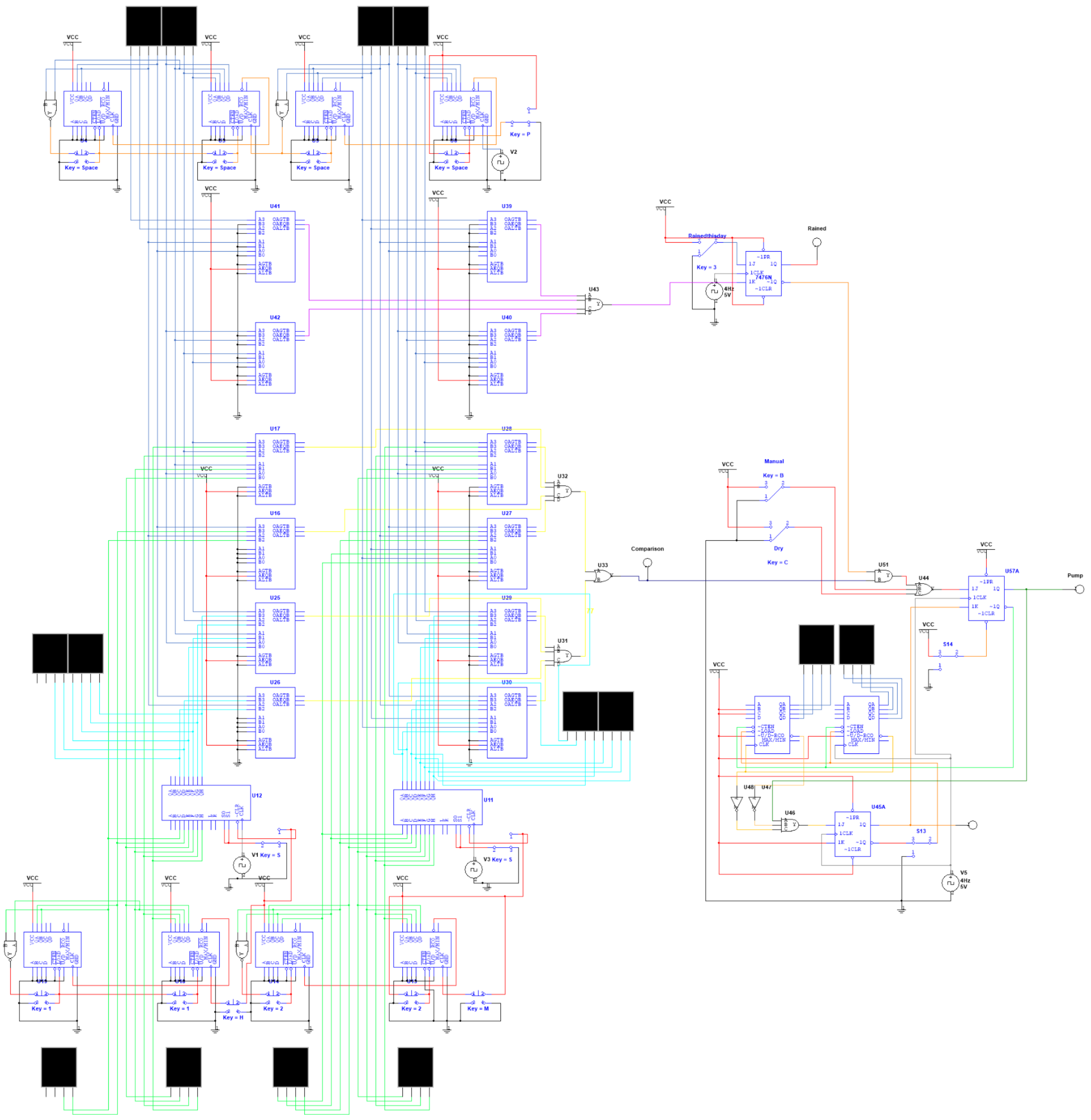


Fig. 18 Complete Circuit Using JK flip-flops

List of Button/Switch Hotkeys

TABLE VII. List of Circuit Hotkeys

Hotkey	Function
Spacebar	Reset Clock
1	Reset Hour of User-Set Time
2	Reset Minute of User-Set Time
9	Reset Pump System
P	Pause Clock
3	Turn on Rain
H	Increment Hour of User-Set Time
M	Increment Minute of User-Set Time
B	Manual Override
C	Land Super Dry

Recommendations for Further Improvement

There are three recommendations to improve this system through digital design.

1. *Make Countdown Subsystem user-settable.*

   Creating a user set countdown time will accommodate every user since everyone will not experience the same water pressure. The users should calculate how long they have to turn the pumps on to irrigate their land thoroughly.

   The countdown subsystem must have additional counters cascaded to the current countdown system. This would also affect the equations for the counter flip-flop. Therefore the students would recommend solving for it again. Additionally, the students recommend using additional counters to accommodate the user set time, similar to the process seen in the time comparator component. The outputs of these counters would connect to the parallel data input pins to the countdown timers. This would change the starting time of the countdown timers once they reset.

2. *Make user wait 5 seconds before setting off on "Manual Override."*

   There are always times when people can set off something accidentally, and this system can be one of them. In addition, there may be times when the manual override switch is accidentally turned on and would start the irrigation process unintentionally. Therefore, implementing this function would lessen unintentional water pumping and result in less water wasted.

3. *Use an alarm clock IC.*

   The whole system uses a lot of components. Notably, a lot is used in the time comparator component. This increases the financial costs in implementing the system and may not be feasible.

   However, there are integrated circuits that have the same function as the time comparator component. For example, the LM8363D IC can show the current time and can accommodate two alarm times according to the datasheet from Sanyo [3]. This means that the current system can be simplified better and used by many people.

References

1. Presettable BCD/Decade Up/Down Counters, Presettable 4-Bit Binary Up/Down Counter. (n.d.). Motorola. Retrieved December 13, 2021, from https://pdf1.alldatasheet.com/datasheet-pdf/view/5679/MOTOROLA/74LS190.html
2. CMOS 4-Bit Magnitude Comparator (n.d.). R&E International, Inc. Retrieved December 13, 2021, from https://html.alldatasheet.com/html-pdf/253561/RANDE/4585B/148/1/4585B.html.
3. Dual-Alarm Digital Clock (n.d.) SANYO. Retrieved December 13, 2021, from https://pdf1.alldatasheet.com/datasheet-pdf/view/80325/SANYO/LM8363D.html