DSM LAB REPORT-6

Group 14: Aaryan Nakul Shah (2024113014)

Lab 6: Latches, Flip-flops and Counters

Experiment-1

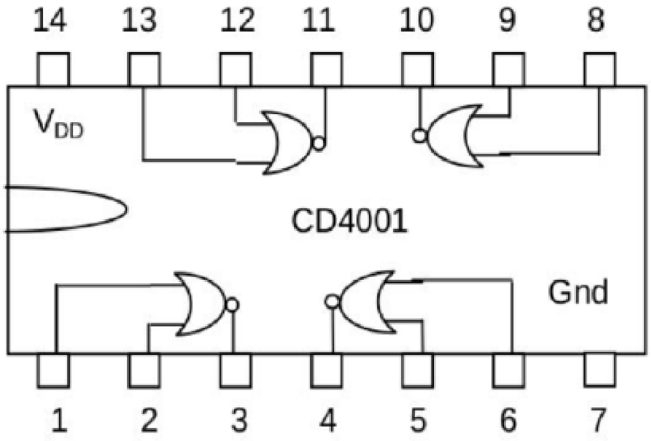
Objective:

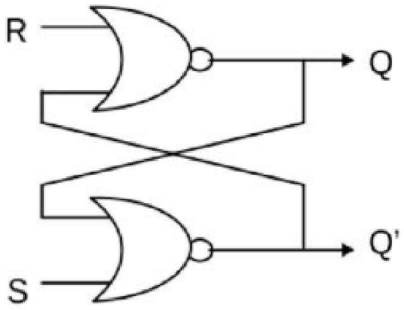
To assemble and test an SR-Latch using its NOR implementation

Components Used:

Digital Test Kit, CD4001 IC and wires

Reference Circuit:



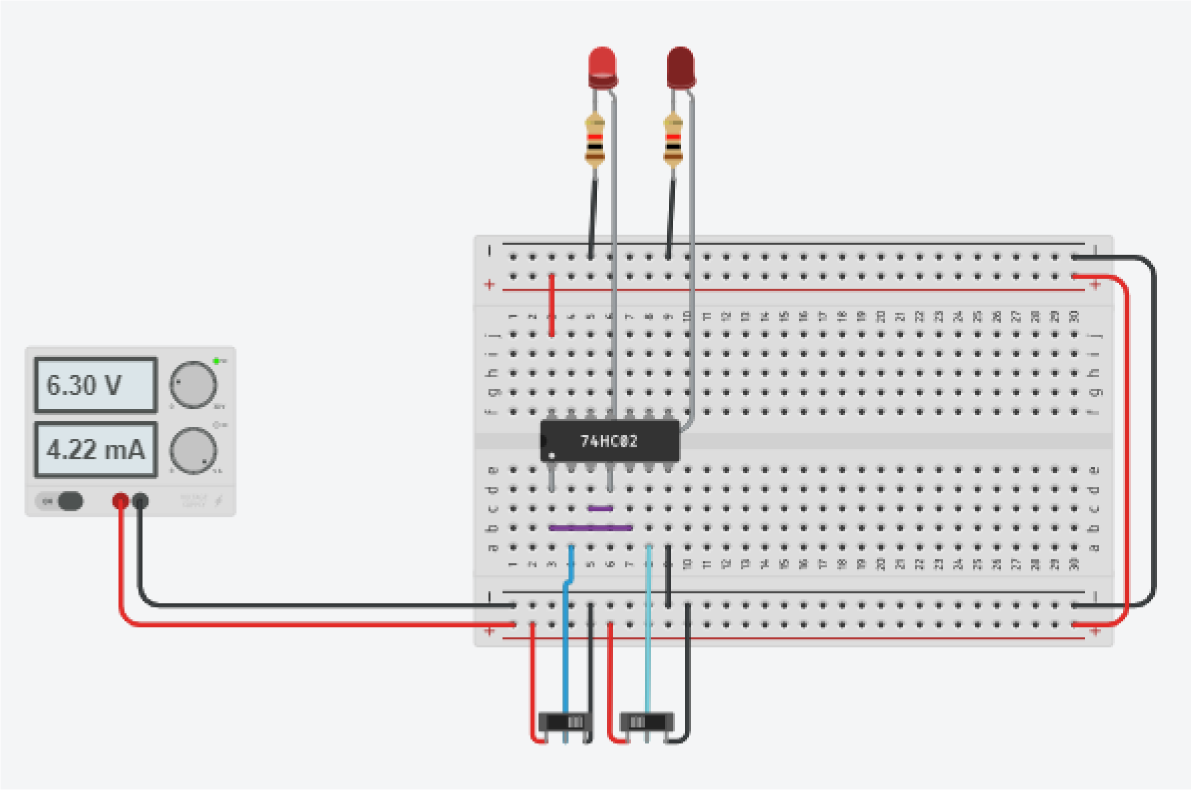


Procedure:

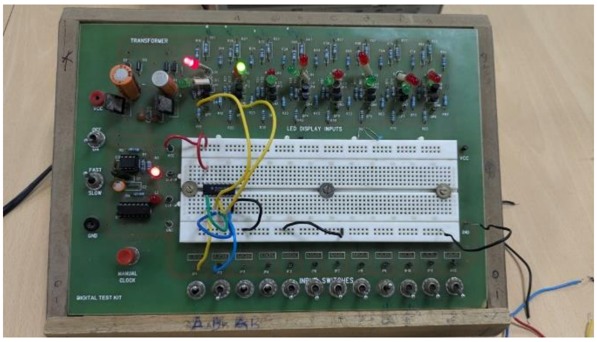
1. Connect the VCC and GND of the Digital Test Kit to the VCC and GND pins of the IC.
2. Using two gates of the IC make connections so as to make a circuit as shown above.

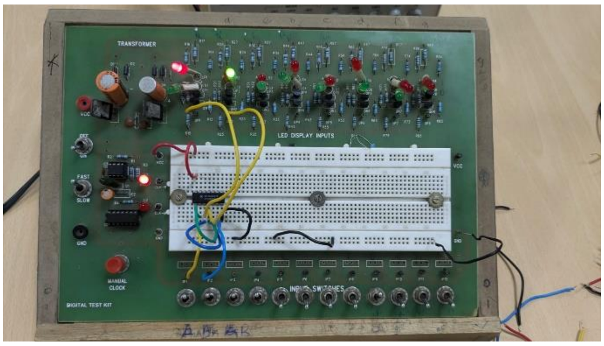
Tinkercad Simulation:

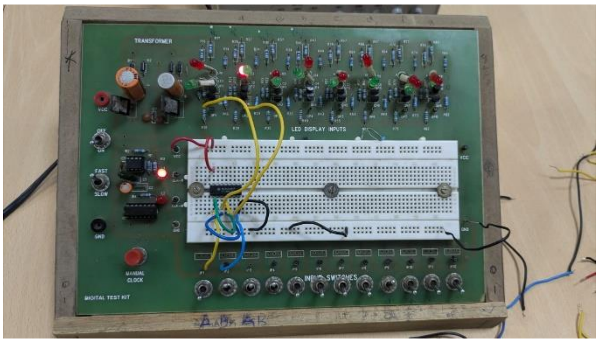
https://www.tinkercad.com/things/iIG49lsVEGX-dsmlab6exp1?sharecode=iGSU0PW0EAgvCBub7nUA2EfoxSnC2XQLFpXK_Xu_33o

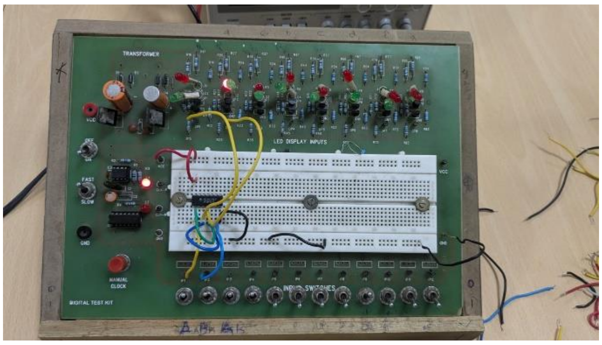


Outputs:









Observations:

S	R	Q	Q’	Action
0	1	0	1	RESET
1	0	1	0	SET
0	0	UNPREDICTABLE		HOLD
1	1	0	0	FORBIDDEN

Conclusion:

NOR-implementation of SR-Latch assembled and tested.

Experiment-2

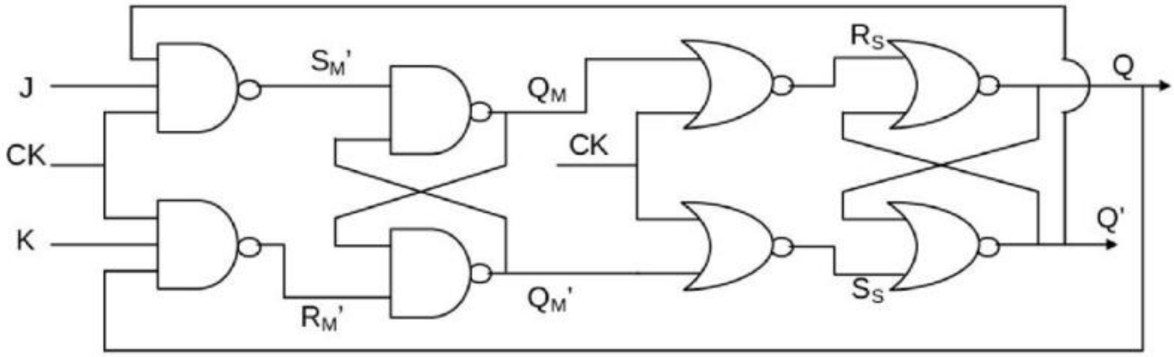
Objective:

To assemble and test a JK Master-Slave Flip Flop

Components Used:

Digital Test Kit, ICs = CD4012, CD4001 and wires

Reference Circuit:

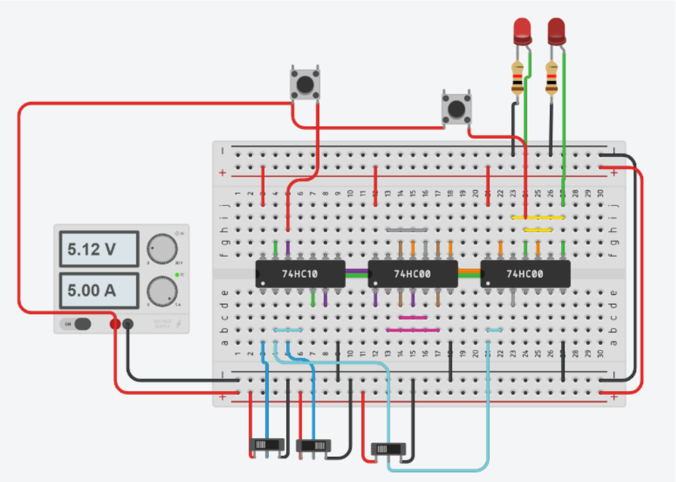


Procedure:

1. Connect the VCC and GND of the Digital Test Kit to the VCC and GND pins of the ICs.
2. Make connections as shown above to form the necessary circuit.

Tinkercad Simulation:

https://www.tinkercad.com/things/iqd6yzODoqe-dsmlab6exp2?sharecode=9lOFSm6O_w44Pu7BucYOmNzR1X6D8LovoZX2b5hBFV4



Output:

JK Master-Slave Flip Flop.mp4

Observations:

Clk	J	K	Q(t+1)
0	X	X	Q(t)
1	0	0	Q(t)
1	0	1	0
1	1	0	1
1	1	1	Q’(t)

Conclusion:

JK Master Slave Flip Flop assembled and tested.

Experiment 3:

Objective:

Using an Arduino to implement a 4-bit up-down ripple counter.

Components Used:

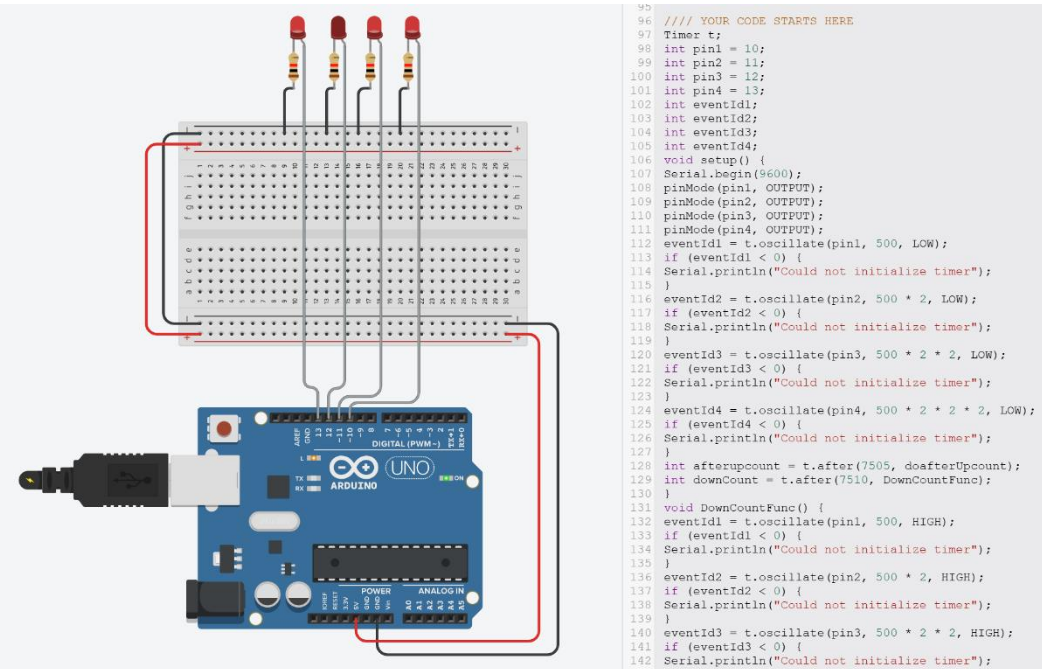
Digital Test Kit, Arduino Uno board and wires

Procedure:

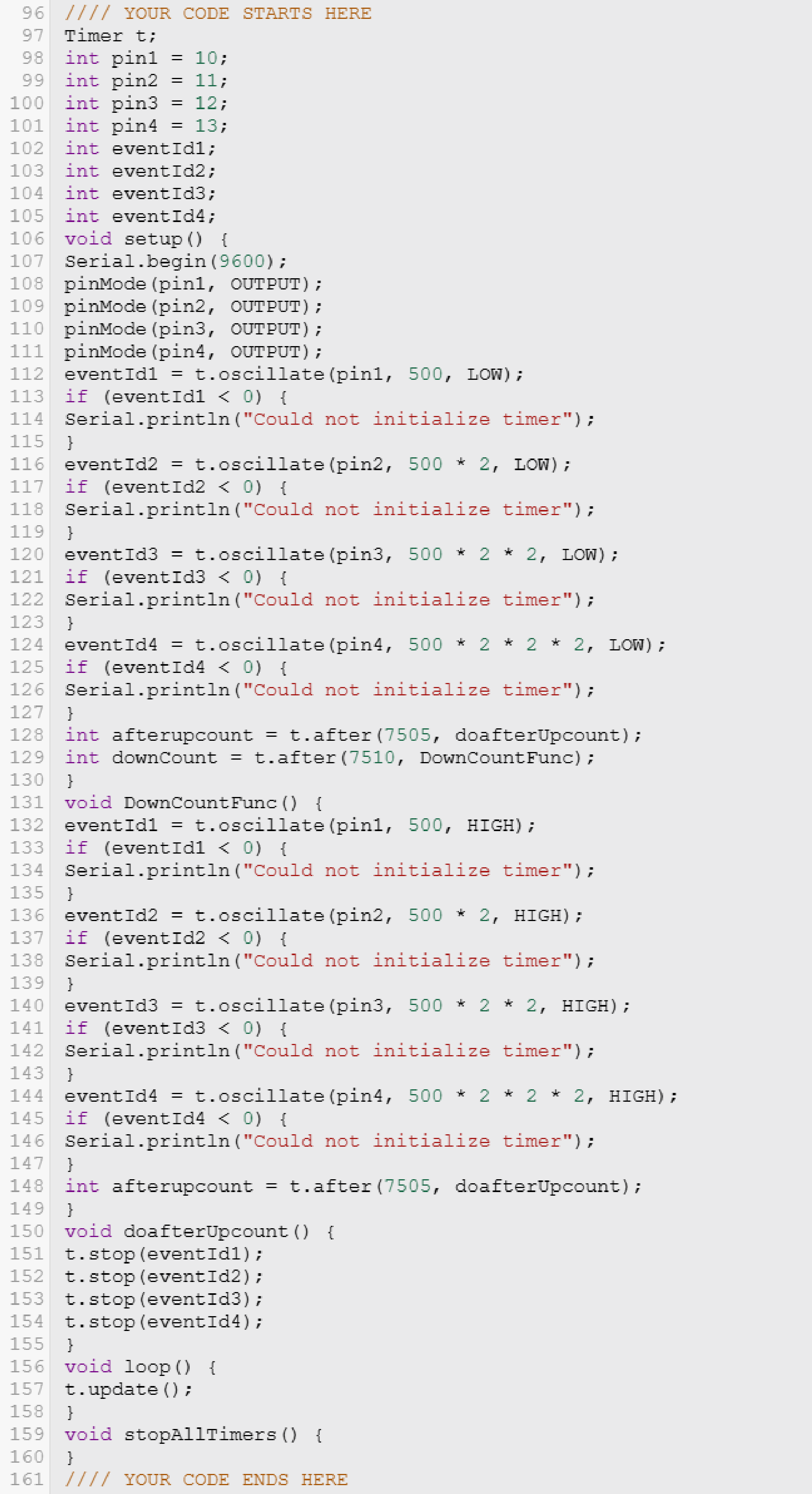
1. Connect the VCC and GND of the Digital Test Kit to the VCC and GND of the Arduino.
2. Connect the Digital Pins of the Arduino to the Display Points of the Digital Test Kit.
3. Write the appropriate code for the up-down counter.

Tinkercad Simulations:

https://www.tinkercad.com/things/4Tkq9BMEozE-dsmlab6exp3?sharecode=huGLRJGLYVb3NPWgBkdN7uYk99QgpbMvQRv4S2gGLXA



```
//// YOUR CODE STARTS HERE
Timer t;
int pin1 = 10;
int pin2 = 11;
int pin3 = 12;
int pin4 = 13;
int eventId1;
int eventId2;
int eventId3;
int eventId4;
void setup() {
Serial.begin(9600);
pinMode(pin1, OUTPUT);
pinMode(pin2, OUTPUT);
pinMode(pin3, OUTPUT);
pinMode(pin4, OUTPUT);
eventId1 = t.oscillate(pin1, 500, LOW);
if (eventId1 < 0) {
Serial.println("Could not initialize timer");
}
eventId2 = t.oscillate(pin2, 500 * 2, LOW);
if (eventId2 < 0) {
Serial.println("Could not initialize timer");
}
eventId3 = t.oscillate(pin3, 500 * 2 * 2, LOW);
if (eventId3 < 0) {
Serial.println("Could not initialize timer");
}
eventId4 = t.oscillate(pin4, 500 * 2 * 2 * 2, LOW);
if (eventId4 < 0) {
Serial.println("Could not initialize timer");
}
int afterupcount = t.after(7505, doafterUpcount);
int downCount = t.after(7510, DownCountFunc);
}
void DownCountFunc() {
eventId1 = t.oscillate(pin1, 500, HIGH);
if (eventId1 < 0) {
Serial.println("Could not initialize timer");
}
eventId2 = t.oscillate(pin2, 500 * 2, HIGH);
if (eventId2 < 0) {
Serial.println("Could not initialize timer");
}
eventId3 = t.oscillate(pin3, 500 * 2 * 2, HIGH);
if (eventId3 < 0) {
Serial.println("Could not initialize timer");
```



```
//// YOUR CODE STARTS HERE
Timer t;
int pin1 = 10;
int pin2 = 11;
int pin3 = 12;
int pin4 = 13;
int eventId1;
int eventId2;
int eventId3;
int eventId4;
void setup() {
Serial.begin(9600);
pinMode(pin1, OUTPUT);
pinMode(pin2, OUTPUT);
pinMode(pin3, OUTPUT);
pinMode(pin4, OUTPUT);
eventId1 = t.oscillate(pin1, 500, LOW);
if (eventId1 < 0) {
Serial.println("Could not initialize timer");
}
eventId2 = t.oscillate(pin2, 500 * 2, LOW);
if (eventId2 < 0) {
Serial.println("Could not initialize timer");
}
eventId3 = t.oscillate(pin3, 500 * 2 * 2, LOW);
if (eventId3 < 0) {
Serial.println("Could not initialize timer");
}
eventId4 = t.oscillate(pin4, 500 * 2 * 2 * 2, LOW);
if (eventId4 < 0) {
Serial.println("Could not initialize timer");
}
int afterupcount = t.after(7505, doafterUpcount);
int downCount = t.after(7510, DownCountFunc);
}
void DownCountFunc() {
eventId1 = t.oscillate(pin1, 500, HIGH);
if (eventId1 < 0) {
Serial.println("Could not initialize timer");
}
eventId2 = t.oscillate(pin2, 500 * 2, HIGH);
if (eventId2 < 0) {
Serial.println("Could not initialize timer");
}
eventId3 = t.oscillate(pin3, 500 * 2 * 2, HIGH);
if (eventId3 < 0) {
Serial.println("Could not initialize timer");
}
eventId4 = t.oscillate(pin4, 500 * 2 * 2 * 2, HIGH);
if (eventId4 < 0) {
Serial.println("Could not initialize timer");
}
int afterupcount = t.after(7505, doafterUpcount);
}
void doafterUpcount() {
t.stop(eventId1);
t.stop(eventId2);
t.stop(eventId3);
t.stop(eventId4);
}
void loop() {
t.update();
}
void stopAllTimers() {
}
//// YOUR CODE ENDS HERE
```

Observations:

The counter displays the numbers 0 to 15 in binary using LEDs. In this representation, a HIGH output of the LED indicates a 1, while a LOW output represents a 0. After reaching 15, the counter begins displaying numbers in reverse order.

Conclusion:

4-bit up-down ripple counter successfully assembled using an Arduino.