<u>California Polytechnic State University Pomona</u>

DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

Digital Logic LAB
ECE 2300L
Lab#9

Prepared by
Paul Yang
Ryan Balatbat
Jason Nguyen

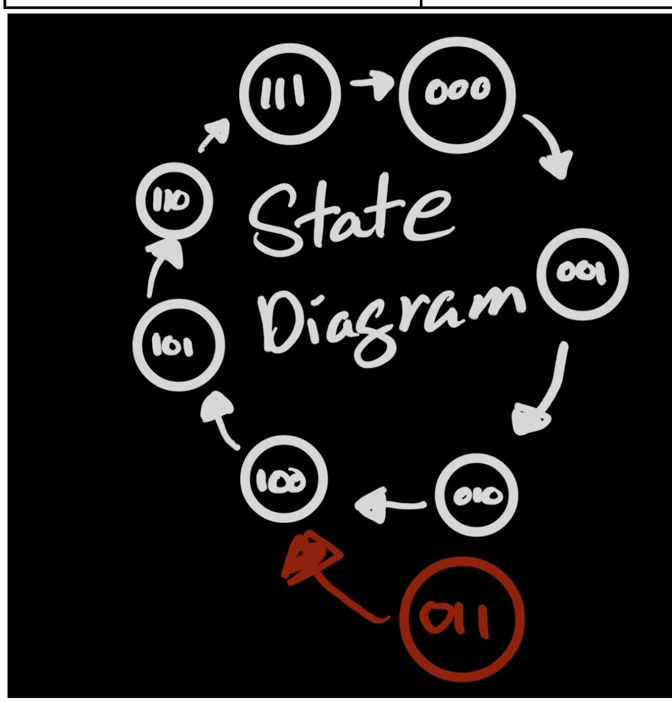
Presented to
Keji Baril

Introduction
Today we will try to use flip flop to create a sequential circuit

Objective
Using T-flip flops we will create a sequential counter that will go through the following numbers in the given order:
0,1,2,4,5,6,7
After 7 the counter will go back to 0 and repeat this sequence

Requirement
Given no input other than a clock input the circuit must cycle through the sequential sequence
State table T-flip flop

T	Q_n	Q_{n+1}
0	0	0
0	1	1
1	0	1
1	1	0



Start			Next			T(input)		
C2	C1	C0	C2	C1	C0	C2	C1	C0
0	0	0	0	0	1	0	0	1

0	0	1	0	1	0	0	1	1
0	1	0	1	0	0	1	1	0
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

Parts List

1. Resistors
2. 7 segment display (common anode)
3. 74ls47
4. And gates
5. Or gates
6. Breadboard
7. Jumpers
8. Power Source
9. T flip flop (in this case we will make a t flip flop out of a j-k flip flop)

Design and Implementation

Not self-correcting

T_{C_2}

$C_2 \backslash C_1C_0$	00	01	11	10
0	0	0	X	1
1	0	0	1	0

$$\boxed{T_{C_2} = C_1C_0 + \bar{C}_2C_1}$$

T_{C_1}

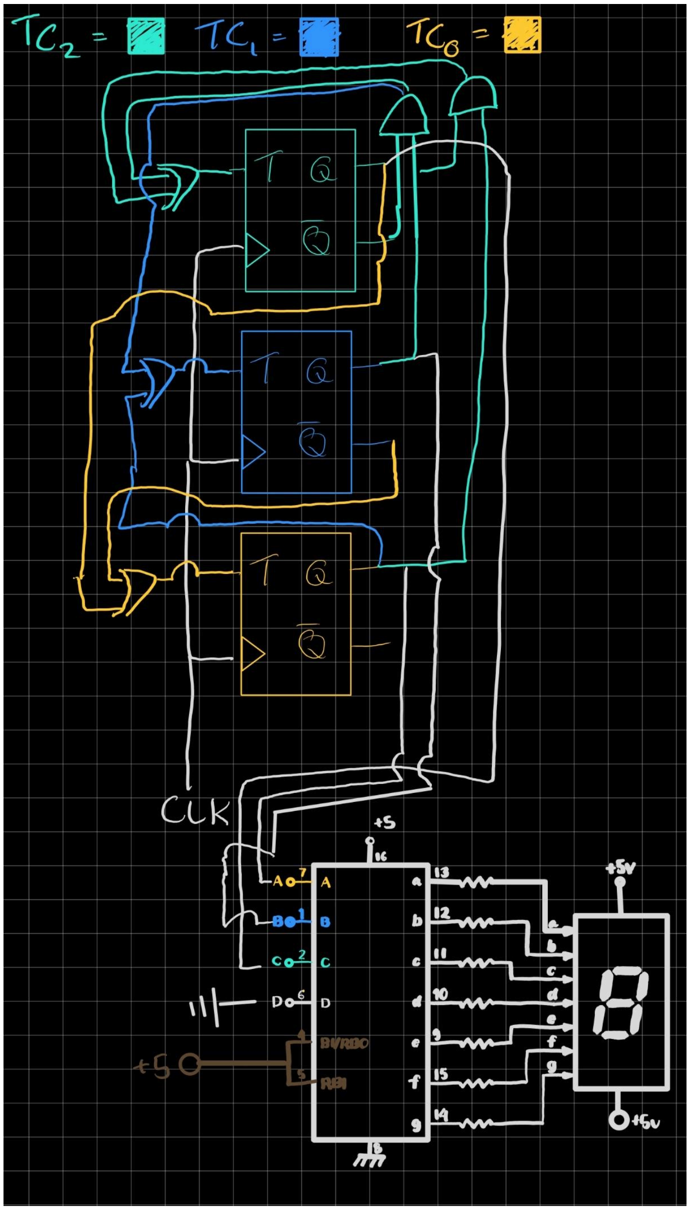
$C_2 \backslash C_1C_0$	00	01	11	10
0	0	1	X	1
1	0	1	1	0

$$\boxed{T_{C_1} = C_0 + \bar{C}_2C_1}$$

T_{C0}

$C_2 \backslash C_1C_0$	00	01	11	10
0	1	1	X	0
1	1	1	1	1

$$T_{C_0} = \bar{C}_1 + C_2$$



Self-Correcting given a 3 input

The difference between self-correcting and non-self-correcting is that in our nonself correcting we use our 3 as an I dont care since we assume it will never occur. However in order to make it self-correcting in the instance it is given a 3 input we just need to replace the dont care values with 1's in our K map. In this case, it will not effect a lot except out last flip-flop input as will be shown in the circuit diagram and is denoted in red.

T_{C_2}

$C_2 \backslash C_1C_0$	00	01	11	10
0	0	0	1	1
1	0	0	1	0

$$\boxed{T_{C_2} = C_1C_0 + \bar{C}_2C_1}$$

T_{C_1}

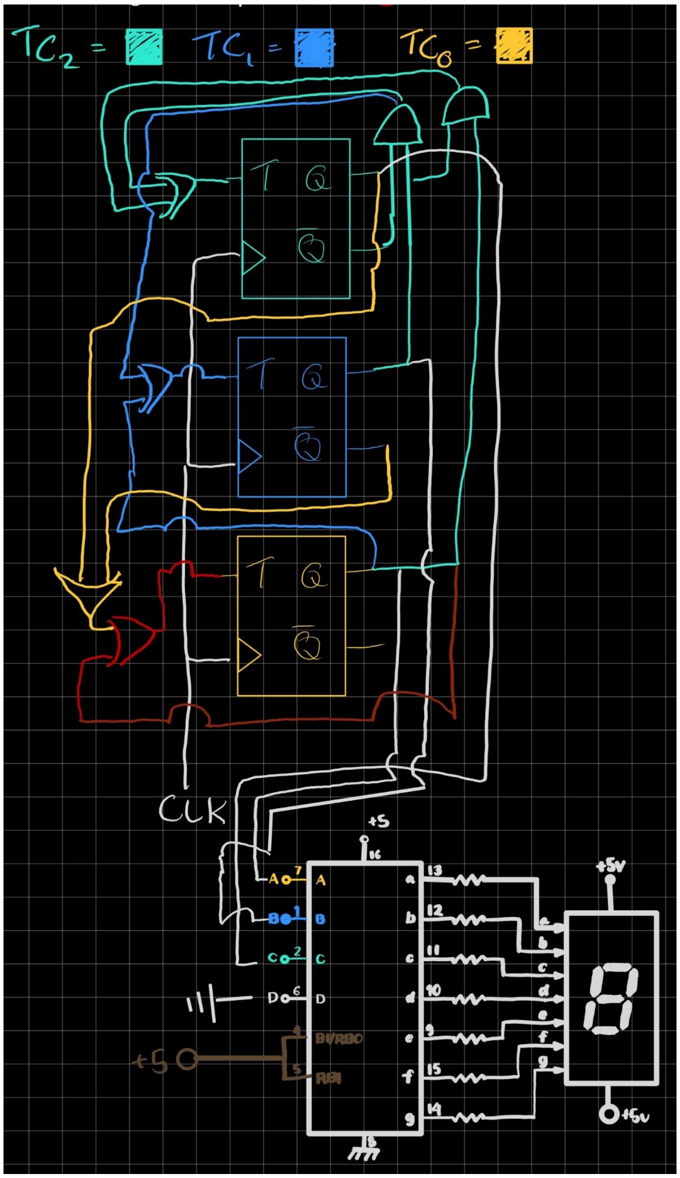
$C_2 \backslash C_1C_0$	00	01	11	10
0	0	1	1	1
1	0	1	1	0

$$\boxed{T_{C_1} = C_0 + \bar{C}_2C_1}$$

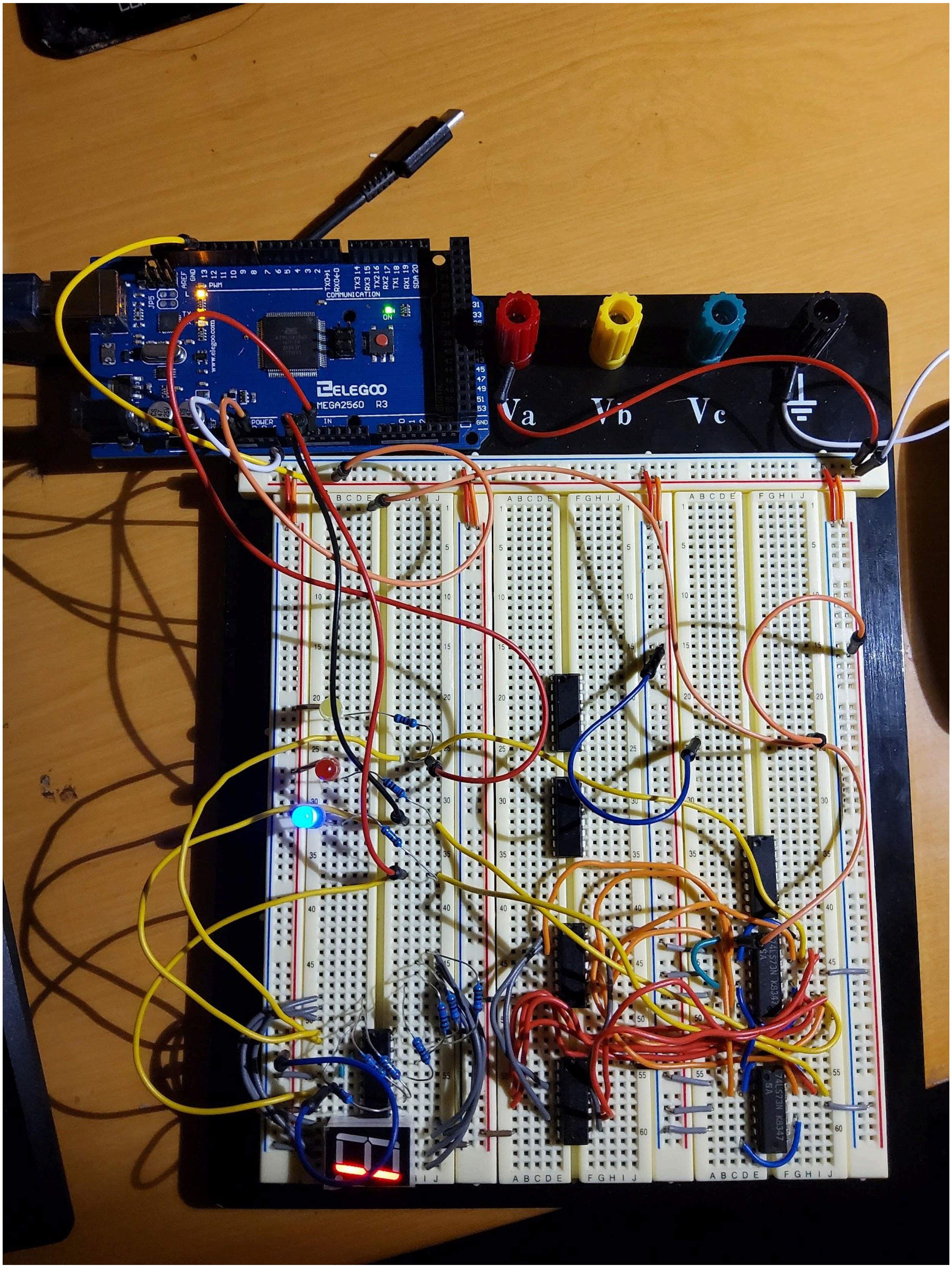
T_{C_0}

$C_2 \backslash C_1C_0$	00	01	11	10
0	1	1	1	0
1	1	1	1	1

$$T_{C_0} = \bar{C}_1 + C_2 + C_0$$



Test and set up



Results and Verification

These are the results of the given display recorded by an Arduino which is also acting as our clock input

```
Binary: 1 1 1 Decimal: 7
Binary: 0 0 0 Decimal: 0
Binary: 0 0 1 Decimal: 1
Binary: 0 1 0 Decimal: 2
Binary: 1 0 0 Decimal: 4
Binary: 1 0 1 Decimal: 5
Binary: 1 1 0 Decimal: 6
Binary: 1 1 1 Decimal: 7
Binary: 0 0 0 Decimal: 0
Binary: 0 0 1 Decimal: 1
Binary: 0 1 0 Decimal: 2
Binary: 1 0 0 Decimal: 4
Binary: 1 0 1 Decimal: 5
Binary: 1 1 0 Decimal: 6
Binary: 1 1 1 Decimal: 7
Binary: 0 0 0 Decimal: 0
Binary: 0 0 1 Decimal: 1
Binary: 0 1 0 Decimal: 2
Binary: 1 0 0 Decimal: 4
Binary: 1 0 1 Decimal: 5
Binary: 1 1 0 Decimal: 6
Binary: 1 1 1 Decimal: 7
Binary: 0 0 0 Decimal: 0
Binary: 0 0 1 Decimal: 1
Binary: 0 1 0 Decimal: 2
Binary: 1 0 0 Decimal: 4
Binary: 1 0 1 Decimal: 5
Binary: 1 1 0 Decimal: 6
Binary: 1 1 1 Decimal: 7
Binary: 0 0 0 Decimal: 0
Binary: 0 0 1 Decimal: 1
Binary: 0 1 0 Decimal: 2
```

CLR	T input	Present state	Next state

H	C2	C1	C0	C2	C1	C0	C2	C1	C0
H	0	0	1	0	0	0	0	0	1
H	0	1	1	0	0	1	0	1	0
H	1	1	0	0	1	0	1	0	0
H	1	1	1	0	1	1	1	0	0
H	0	0	1	1	0	0	1	0	1
H	0	1	1	1	0	1	1	1	0
H	0	0	1	1	1	0	1	1	1
H	1	1	1	1	1	1	0	0	0

Arduino Code

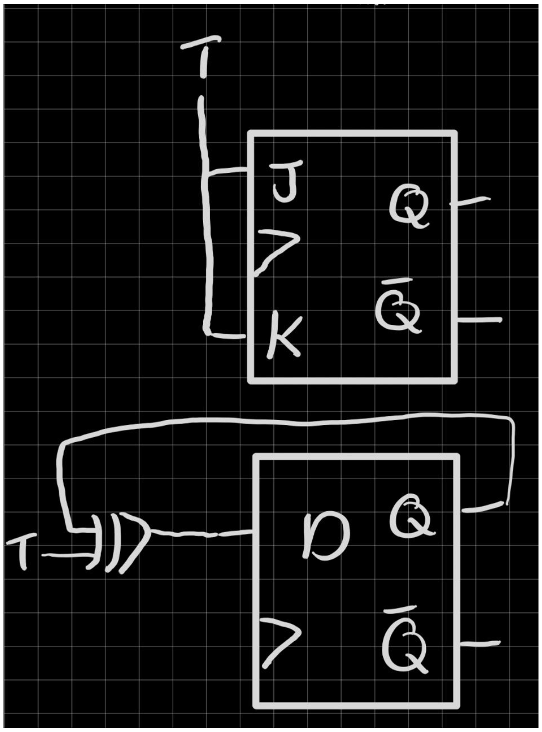
```
int outPort=13;
int lsb=A0;
int nsb=A1;
int msb=A2;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
pinMode(outPort,OUTPUT);
pinMode(lsb, INPUT);
pinMode(nsb, INPUT);
pinMode(msb, INPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(outPort,HIGH);
  delay(500);
  digitalWrite(outPort,LOW);
  delay(500);
  int C2=analogRead(msb);
  int C5=1;
  if(C2<200){
    C5=0;
  }
  int C1=analogRead(nsb);
  int C4=1;
```

```
  if(C1<200){
    C4=0;
  }
  int C0=analogRead(lsb);
  int C3=1;
  if(C0<200){
    C3=0;
  }
  int decimal=0;
  if (C5==1){
    decimal=decimal+4;
  }
  if (C4==1){
    decimal=decimal+2;
  }
  if(C3==1){
    decimal=decimal+1;
  }
  Serial.print("Binary: ");
  Serial.print(C5);
  Serial.print(' ');
  Serial.print(C4);
  Serial.print(' ');
  Serial.print(C3);
  Serial.print(' ');
  Serial.print("Decimal: ");
  Serial.print(decimal);
  Serial.println(' ');
}
```

Post-Lab Questions

1. **Is T- flip-flop commercially available? If so, draw the pin assignments from the Internet. If not, draw block diagrams for obtaining T- flip flops in two different ways**

   T-flip flops are not commercially available. They however can be made from jk or d flip flops if wired a certain way



2. **How many flip flops are needed to design a counter to count in the following sequence:**
   12, 20,1,0, and then repeat?
   This sequence needs 5 flip flops because each flip flop essentially represents 1 bit thus to accommodate for a number as large as 20(the largest number in the sequence) we would need 5 flip flops