

**North South University**

**Department of Electrical & Computer Engineering**

**Project Report**

|  |  |
| --- | --- |
| **Experiment Title** | Seven-Segment Display Using Combinational and Sequential Circuits |

|  |  |
| --- | --- |
| **Course Code** | CSE231L |
| **Section** | 17 |
| **Course Name** | Digital Logic Design Lab |

|  |  |
| --- | --- |
| **Lab Group** | 06 |
| **Written By** | Aziza Zaman Sadia |

|  |  |
| --- | --- |
| **Date of Submission** | 04.12.2024 |
| **Due Date of Submission** | 04.12.2024 |

|  |  |
| --- | --- |
| **Group Members ID** | **Group Members Name** |
| 2231603642 | Aziza Zaman Sadia |
| 2132838642 | Ananna Islam |

### ***Introduction –***

### Seven-segment displays are widely used for showing numbers and characters in electronic devices. This project explores the integration of combinational and sequential circuits to automate the display of our chosen string "ALLCOOL".

### The key objectives were:

1. To design a combinational circuit for character selection.
2. To build a counter for automating input changes.
3. To synchronize components efficiently to achieve a seamless display.

### ***Process –***

#### **Step 1: Combinational Circuit Design**

We began by designing the combinational logic circuit using a decoder and IC 7408 (2-input AND gate), which we discussed beforehand.

#### **Step 2: Counter Implementation**

To automate the switching process, we developed a counter using J-K flip-flops. This counter could count from 0 to 7, driven by clock pulses generated by an NE555 Timer

#### **Step 3: Integration of Components**

The counter outputs were connected to the select inputs of the decoder. We, therefore, connected our combinational logic circuit to our sequential circuit for the sequential display of characters.

***Circular State Diagram -***

First of all, we have to create a counter that changes its state based on clock pulses, cycling through a defined sequence of values (e.g., 0 to 7). This will involve using flip-flops to store and toggle the binary states. Finally, the counter's outputs will be connected to the select bits of a decoder to achieve the desired functionality.

***State Table -***

J-K Flip-Flop-

Excitation table for J-K flip-flop -

|  |  |  |  |
| --- | --- | --- | --- |
| Q | QN | J | K |
| 0 | 0 | 0 | ✖ |
| 0 | 1 | 1 | ✖ |
| 1 | 0 | ✖ | 1 |
| 1 | 1 | ✖ | 0 |

# Constructing a counter using J-K flip-flop –

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Present State | | | Next state | | | Flip-flop inputs | | | | | |
| Q2 | Q1 | Q0 | Q2 | Q1 | Q0 | J2 | K2 | J1 | K1 | J0 | K0 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | ✖ | 0 | ✖ | 1 | ✖ |
| 0 | 0 | 1 | 0 | 1 | 0 | 0 | ✖ | 1 | ✖ | ✖ | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 | 0 | ✖ | ✖ | 0 | 1 | ✖ |
| 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 |

Equations Using K-Map –

For J2 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | 0 | 0 | 1 | 0 |
| Q2 | ✖ | ✖ | ✖ | ✖ |

J2 = Q1Q0

For K2 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | ✖ | ✖ | ✖ | ✖ |
| Q2 | 0 | 0 | 1 | 0 |

K2 = Q1Q0

For J1 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | 0 | 1 | ✖ | ✖ |
| Q2 | 0 | 1 | ✖ | ✖ |

J1 = Q0

For K1 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | ✖ | ✖ | 1 | 0 |
| Q2 | ✖ | ✖ | 1 | 0 |

K1 = Q0

For J0 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | 1 | ✖ | ✖ | 1 |
| Q2 | 1 | ✖ | ✖ | 1 |

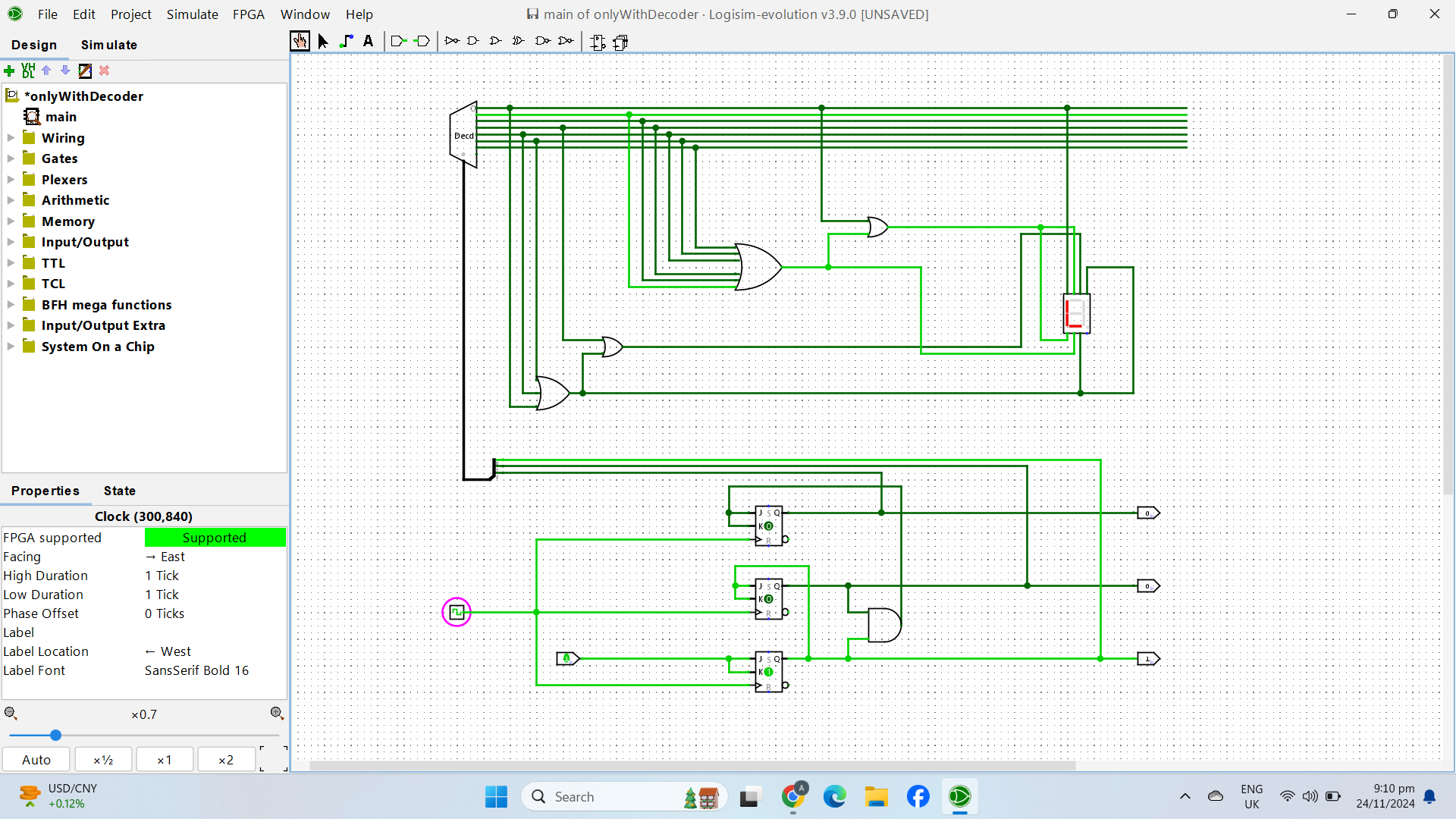
J0 = 1

For K0 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | ✖ | 1 | 1 | ✖ |
| Q2 | ✖ | 1 | 1 | ✖ |

K0 = 1

Logisim Simulation –



T Flip-Flop-

Excitation table for T flip-flop -

|  |  |  |
| --- | --- | --- |
| Q | QN | T |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# Constructing a counter using T flip-flop –

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Present State | | | Next State | | | | Flip-flop inputs | | | |
| Q2 | Q1 | Q0 | | Q2 | Q1 | Q0 | | T2 | T1 | T0 |
| 0 | 0 | 0 | | 0 | 0 | 1 | | 0 | 0 | 1 |
| 0 | 0 | 1 | | 0 | 1 | 0 | | 0 | 1 | 1 |
| 0 | 1 | 0 | | 0 | 1 | 1 | | 0 | 0 | 1 |
| 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 |

Equations Using K-Map –

For T2 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | 0 | 0 | 1 | 0 |
| Q2 | 0 | 0 | 1 | 0 |

T2 = Q1Q0

For T1 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | 0 | 1 | 1 | 0 |
| Q2 | 0 | 1 | 1 | 0 |

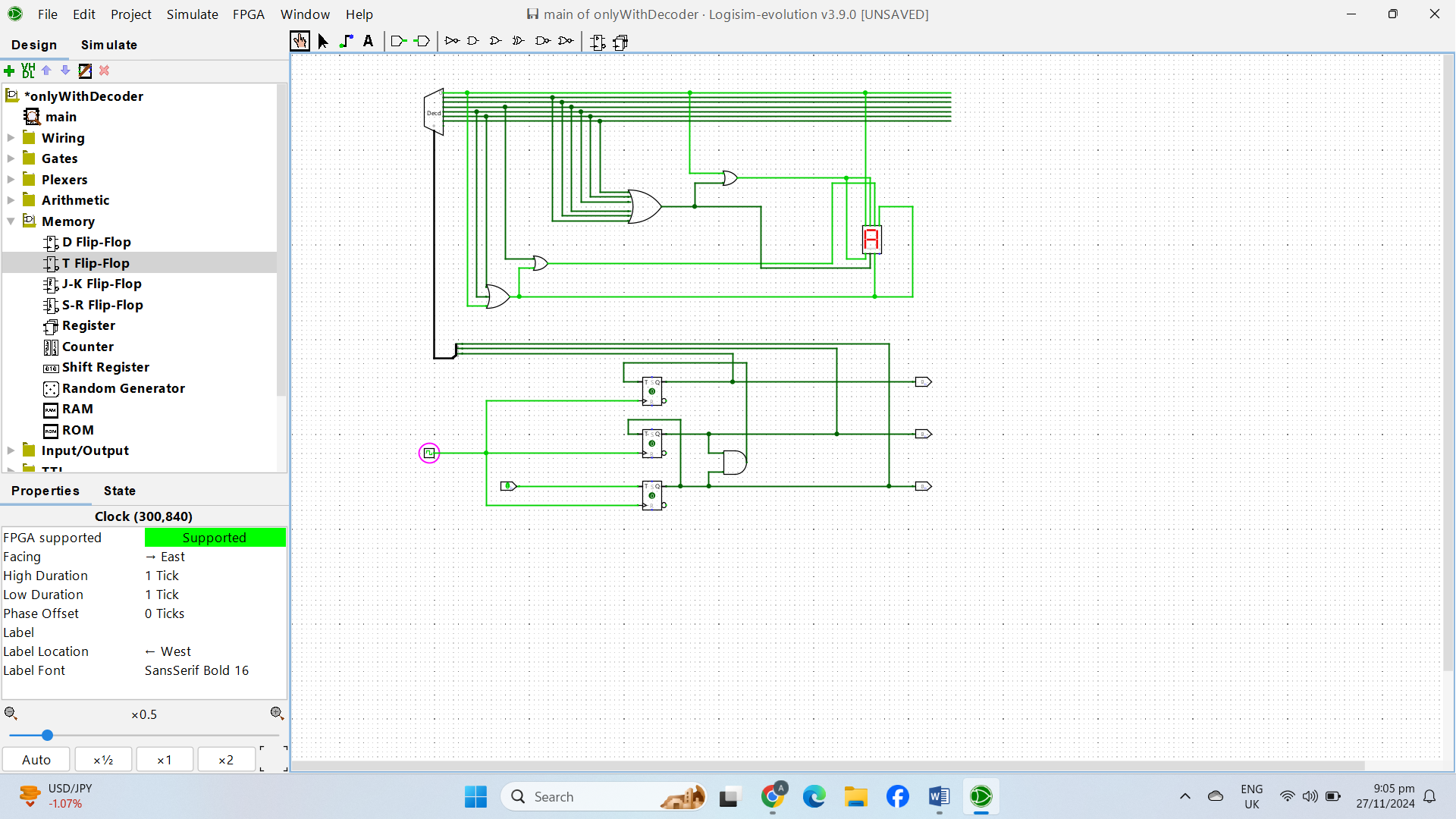
T1 = Q0

For T0 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | 1 | 1 | 1 | 1 |
| Q2 | 1 | 1 | 1 | 1 |

T0 = 1

Logisim Simulation -



D Flip-Flop-

Excitation table for D flip-flop –

|  |  |  |
| --- | --- | --- |
| Q | QN | D |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Constructing a counter using D flip-flop –

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Present State | | | Next State | | | | Flip-flop inputs | | | |
| Q2 | Q1 | Q0 | | Q2 | Q1 | Q0 | | D2 | D1 | D0 |
| 0 | 0 | 0 | | 0 | 0 | 1 | | 0 | 0 | 1 |
| 0 | 0 | 1 | | 0 | 1 | 0 | | 0 | 1 | 0 |
| 0 | 1 | 0 | | 0 | 1 | 1 | | 0 | 1 | 1 |
| 0 | 1 | 1 | | 1 | 0 | 0 | | 1 | 0 | 0 |
| 1 | 0 | 0 | | 1 | 0 | 1 | | 1 | 0 | 1 |
| 1 | 0 | 1 | | 1 | 1 | 0 | | 1 | 1 | 0 |
| 1 | 1 | 0 | | 1 | 1 | 1 | | 1 | 1 | 1 |
| 1 | 1 | 1 | | 0 | 0 | 0 | | 0 | 0 | 0 |

Equations Using K-Map -

For D2 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | 0 | 0 | 1 | 0 |
| Q2 | 1 | 1 | 0 | 1 |

D2 =Q1**'** Q2  **+** Q0**'** Q2 **+** Q0 Q1 Q2**'**

For D1 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | 0 | 1 | 0 | 1 |
| Q2 | 0 | 1 | 0 | 1 |

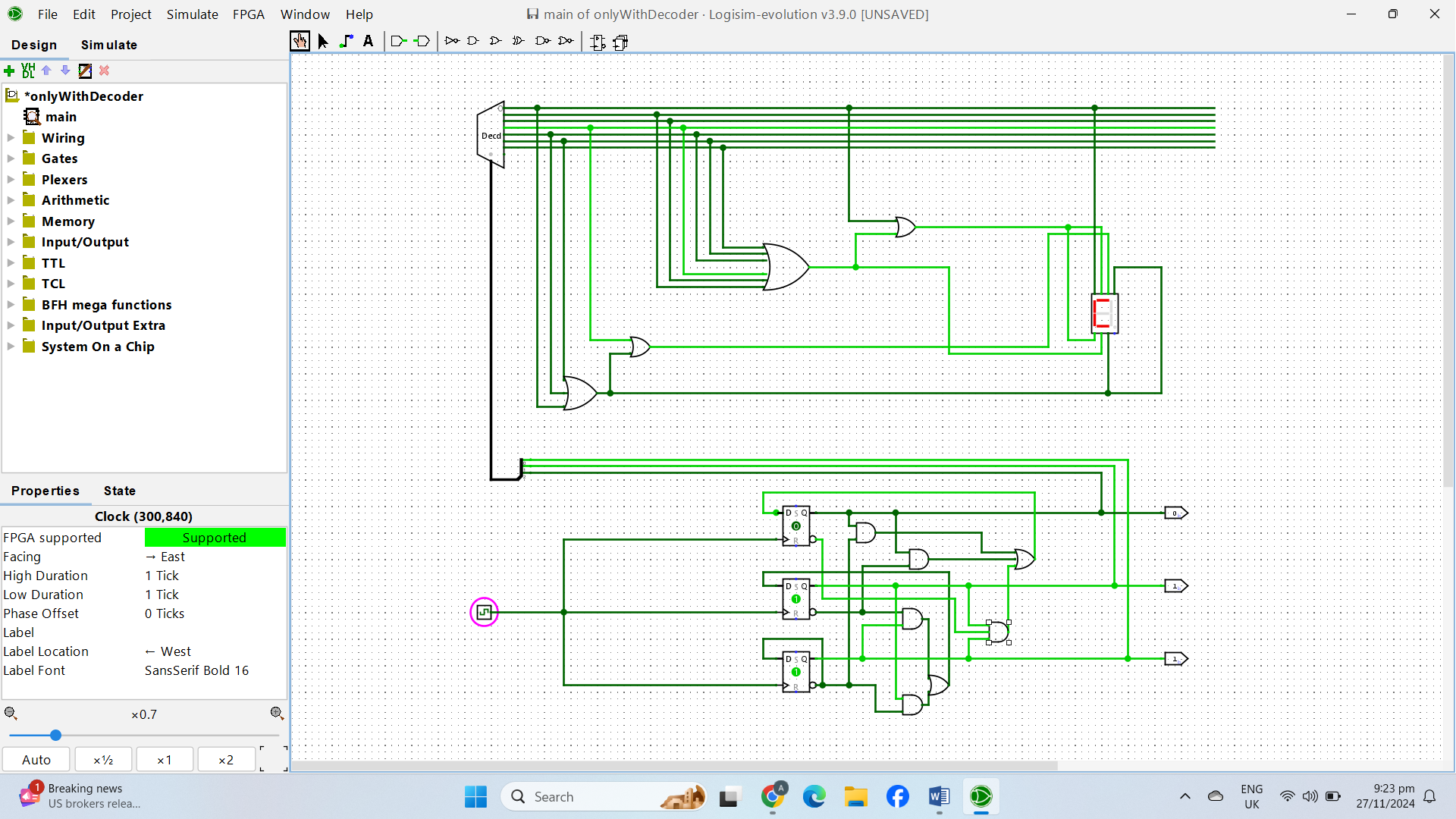
D1 = Q1**'** Q0  **+** Q1 Q0**'**

For D0 , the K-map would be,

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Q1**'** Q0**'** | Q1**'** Q0 | Q1Q0 | Q1 Q0**'** |
| Q2**'** | 1 | 0 | 0 | 1 |
| Q2 | 1 | 0 | 0 | 1 |

D0 = Q0**'**

Logisim Simulation –



***About 555 Timer –***

The 555 timer is a popular integrated circuit used in many electronic projects. It is versatile and can work in three main modes: astable, monostable, and bistable. Inside the 555 timer, there are transistors, flip-flops, and comparators that work together to create timed pulses or oscillations.

To connect a 555 timer, a resistor and capacitor are often used to set the timing. The capacitor is connected between the discharge pin (Pin 7) and ground, while resistors are connected to the threshold (Pin 6) and trigger (Pin 2) pins. The capacitor charges and discharges through the resistors, creating a cycle. The output (Pin 3) switches between HIGH and LOW, creating a square wave signal.

Basically, The 555 timer produces clock pulses, which are sent to the counter's input. The counter changes its state with each pulse, representing numbers from 000 (0) to 111 (7).

***Cost Analysis –***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Flip-Flop Used** | **Required IC’s & Flip-Flops** | **Per Piece Cost** | **Quantity** | **Total Cost for Implementation** |
| **J-K Flip-Flop** | J-K Flip-Flop | 35 taka | 2 | 100 taka |
| IC 7408 (2 Input AND Gate) | 30 taka | 1 |
| **T Flip-Flop** | T Flip-Flop | 35 taka | 2 | 100 taka |
| IC 7408 (2 Input AND Gate) | 30 taka | 1 |
| **D Flip-Flop** | D Flip-Flop | 40 taka | 2 | 170 taka |
| IC 7408 (2 Input AND Gate) | 30 taka | 2 |
| IC 7432 (2 Input OR Gate) | 30 taka | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Equipment** | **Quantity** | **Per Piece Cost** | **Total Cost** |
| NE555 Timer | 1 | 20 taka | 20 taka |
| Register (10 Ω) | 2 | 2 taka | 4 taka |
| Capacitor (100 µF) | 1 | 5 taka | 5 taka |

***Discussion –***

First of all, we completed the combinational part of our project by using decoder and IC 7408 (2 Input AND Gate). At this stage, we relied on switches to change the select bits, allowing us to display the string correctly on the seven-segment display.

Following that, we created a counter using J-K flip-flop which could count from 0 to 7, driven by the clock pulse from the NE555 Timer. We chose J-K flip-flop in this case as it was cost effective and quite straight forward to implement. As a result, this eliminated the need to manually control the switches.

Finally, we connected the counters outputs to the select bits of the decoder ensuring the least significant bit (LSB) matched with LSB and the most significant bit (MSB) with MSB.

With these steps, we successfully completed our exciting project, which used combinational circuits along with sequential circuits to display our chosen string “ALLCOOL” on the seven segment display.