Seven-Segment Decoder Lab Report

Introduction

In this laboratory experiment, I designed and implemented a digital circuit to display the character set "ICU1211" on a seven-segment display. The project combines sequential and combinational logic circuits to create a system that cycles through the specified characters when a clock pulse is applied.

The design utilizes a 3-bit counter built with JK flip-flops, which provides binary codes that are decoded using a 3-to-8 decoder. The decoded outputs are then processed through OR gates to drive the appropriate segments of the seven-segment display. This approach creates an efficient and modular design that clearly demonstrates the principles of digital sequential circuits and display interfacing.

Procedure

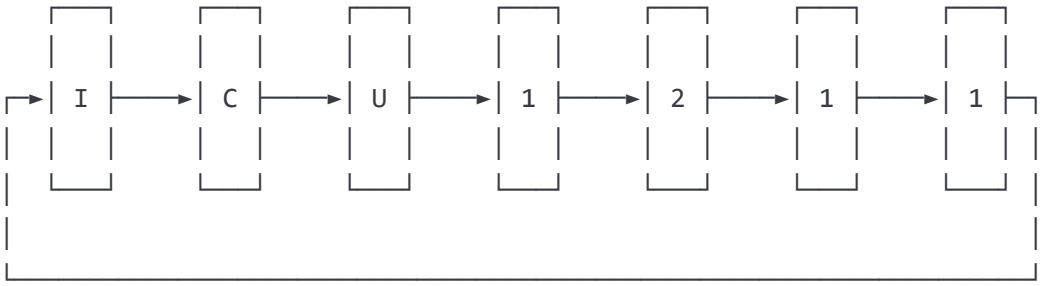
The implementation of the seven-segment decoder for the "ICU1211" character set followed these steps:

1. **Character Analysis**: I analyzed how each character in the set "ICU1211" would appear on the seven-segment display and mapped each segment (a-g) to the corresponding states.
2. **State Assignment**: I assigned 3-bit binary codes to each character:
   - 000: I
   - 001: C
   - 010: U
   - 011: 1
   - 100: 2
   - 101: 1
   - 110: 1
   - 111: (not used/cycles back to 000)
3. **Counter Design**: Implemented a 3-bit counter using JK flip-flops configured in toggle mode.
4. **Decoder Implementation**: Used a 74HC138N 3-to-8 decoder to convert the 3-bit counter output to individual character selection signals.
5. **Output Logic Design**: Created OR gate networks to map the decoder outputs to the seven segments of the display.
6. **Hardware Assembly**: Assembled the circuit on a breadboard using CD4027BE JK flip-flops, 74HC138N decoder, 7404 hex inverter, and 7432 OR gates.

7. **Testing and Verification**: Tested the circuit by applying clock pulses and observing the sequence of characters on the display.

State Diagram

The state diagram for this sequential circuit represents the transitions between the different character states. The system progresses through the states in a fixed sequence with each clock pulse.



The state diagram shows a simple cyclic progression through the seven states corresponding to the character set "ICU1211". Each transition occurs on the rising edge of the clock signal.

State Table

The state table shows the current state, next state, and outputs for each state in the sequence:

Current State	Next State	Segments Activated
000 (I)	001 (C)	b, c
001 (C)	010 (U)	a, d, e, f
010 (U)	011 (1)	b, c, d, e, f
011 (1)	100 (2)	b, c
100 (2)	101 (1)	a, b, d, e, g
101 (1)	110 (1)	b, c
110 (1)	000 (I)	b, c

Equations and K-Maps

For each segment (a-g) of the seven-segment display, I derived Boolean equations based on which states activate each segment.

Segment a Equation

a = ~~A~~BC + A~~B~~C
= ~A(~B)(C) + (A)(~B)(~C)

The K-map for segment a:

```
     BC
    |00|01|11|10|
  ---|--|--|--|--|
  A=0| 0| 1| 0| 0|
  ---|--|--|--|--|
  A=1| 1| 0| 0| 0|
```

Segment b Equation

b = ~B + ~A(C) + A(~C)
= ~B + ~A(C) + A(~C)

The K-map for segment b:

```
     BC
    |00|01|11|10|
  ---|--|--|--|--|
  A=0| 1| 0| 1| 1|
  ---|--|--|--|--|
  A=1| 1| 1| 0| 1|
```

Segment c Equation

c = ~~A + A~~BC + AB~C
= ~A + A(~~BC + B~~C)

The K-map for segment c:

```
     BC
    |00|01|11|10|
  ---|--|--|--|--|
  A=0| 1| 0| 1| 1|
  ---|--|--|--|--|
  A=1| 0| 1| 0| 1|
```

Segment d Equation

d = ~~A~~BC + ~~AB~~C + A~~B~~C
= ~~A(BC + BC) + A~~B~C

The K-map for segment d:

```
     BC
    |00|01|11|10|
  ---|--|--|--|--|
A=0| 0| 1| 0| 1|
  ---|--|--|--|--|
A=1| 1| 0| 0| 0|
```

Segment e Equation

e = ~~A~~BC + ~~AB~~C + A~~B~~C
= ~~A(BC + BC) + A~~B~C

The K-map for segment e:

```
     BC
    |00|01|11|10|
  ---|--|--|--|--|
A=0| 0| 1| 0| 1|
  ---|--|--|--|--|
A=1| 1| 0| 0| 0|
```

Segment f Equation

f = ~~A~~BC + ~~AB~~C
= ~A(~~BC + B~~C)

The K-map for segment f:

```
     BC
    |00|01|11|10|
  ---|--|--|--|--|
A=0| 0| 1| 0| 1|
  ---|--|--|--|--|
A=1| 0| 0| 0| 0|
```

Segment g Equation

g = A~~B~~C

The K-map for segment g:

```
     BC
    |00|01|11|10|
 ---|--|--|--|--|
A=0| 0| 0| 0| 0|
 ---|--|--|--|--|
A=1| 1| 0| 0| 0|
```

Logisim Simulation

[Images of Logisim simulation will be added here]

555 Timer Implementation

To generate the clock signal for the sequential circuit, I implemented a 555 timer in astable mode. This provides a continuous series of pulses at a frequency that allows for comfortable viewing of the character transitions.

The 555 timer circuit uses the following components:

- 1× NE555 Timer IC
- 1× 10kΩ potentiometer (for adjusting frequency)
- 2× 1kΩ resistors
- 1× 10µF electrolytic capacitor
- 1× 0.01µF ceramic capacitor (for stability)

The frequency of oscillation is determined by:
$f = 1.44 / ((R1 + 2R2) \times C)$

With the potentiometer, the frequency can be adjusted from approximately 1 Hz to 10 Hz, providing a visible transition between characters. The 555 timer output connects directly to the clock input of the first JK flip-flop in the counter chain.

Cost Analysis

The implementation of the seven-segment decoder required the following components:

Component	Quantity	Unit Cost (৳)	Total Cost (৳)
CD4027BE (Dual JK Flip-Flop)	2	38.00	76.00
74HC138N (3-to-8 Decoder)	1	25.00	25.00
7404 (Hex Inverter)	1	18.00	18.00
7432 (Quad 2-input OR Gate)	4	20.00	80.00
Seven-Segment Display	1	40.00	40.00
NE555 Timer	1	15.00	15.00
Resistors (330Ω)	7	1.00	7.00
Resistors (10kΩ)	7	1.00	7.00
Potentiometer (10kΩ)	1	12.00	12.00
Capacitors	2	5.00	10.00
Breadboard	1	150.00	150.00
Jumper Wires	1 set	80.00	80.00
Total			**520.00**

Discussion

The implementation of the seven-segment decoder for displaying the "ICU1211" character set was successful, demonstrating the principles of sequential and combinational logic circuits. The design approach using a 3-to-8 decoder proved to be efficient and modular, making the circuit easier to implement and debug compared to a direct Boolean implementation.

Several challenges were encountered during the implementation:

1. **Output Drive Capability**: The active-LOW outputs of the 74HC138N decoder required inverters before connecting to the OR gates. This added an extra layer of complexity but ensured proper signal levels.
2. **Segment Dimming**: Initial testing showed some segments dimming rather than turning completely off. This was resolved by adding 10kΩ pull-down resistors to ensure segments fully turned off when they should be inactive.
3. **Clock Debouncing**: The manual push button used for testing caused multiple transitions due to contact bounce. Implementing a 555 timer as the clock source provided clean clock signals for reliable operation.
4. **Power Distribution**: Ensuring adequate power distribution across the breadboard was crucial for reliable operation of all ICs. Adding bypass capacitors near the power pins of each IC helped maintain stable operation.

The decoder-based approach used in this implementation offers several advantages over direct Boolean implementation:

- More modular and easier to understand
- Simpler to modify for different character sets
- Clearer relationship between binary codes and displayed characters
- Better maintainability and troubleshooting

Future improvements could include:

- Adding binary LED indicators to show the current counter state
- Implementing a reset button to return to the initial state
- Expanding the design to display additional characters
- Adding a multiplexer to allow switching between automatic sequencing and manual character selection

In conclusion, this project successfully demonstrated the application of digital logic concepts to create a practical display system. The combination of sequential and combinational logic, along with the modular decoder-based approach, resulted in an effective implementation of the seven-segment decoder for displaying the "ICU1211" character set.