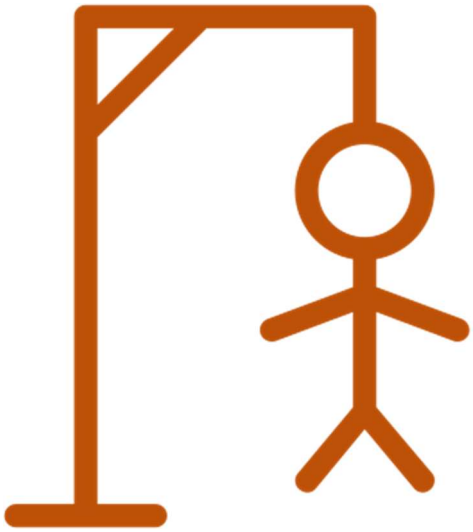
DECEMBER 30, 2021



HANGMAN GAME

DIGITAL LOGIC DESIGN

NAME	CMS
Muhammad Ahmed Mohsin	333060
Muhammad Umer	345834
Tariq Umar	334943
Imran Haider	332569

Table of Contents

List of Figures

Table of Contents

Dedication

We respectfully dedicate this effort to our beloved instructor, Mr. Arshad Nazir, for his unwavering advice and support during digital logic design. This is also dedicated to the Electrical Engineering Department at the School of Electrical Engineering and Computer Sciences. Above all, we devote this to The Almighty, to Whom all wisdom and understanding belong.

Acknowledgements

We would like to thank our lecturer, Sir Arshad Nazir, and the laboratory in-charge, Mr. Mughees Ahmed, for their assistance with this research.

We also acknowledge the School of Electrical Engineering and Computer Sciences' department of Electrical Engineering for equipping us with the resources and facilities we needed to complete this work.

We're also appreciative to all of our individuals who supported us with this endeavor in every way they could.

Abstract

Combinational and sequential circuits play a vital role in daily life. They have applications diversifying from the field of mathematics chemistry and have many applications in daily life. This intriguing notion has also found its way into children's games, such as the well-known Simon Says and Hangman. Both have their advantages and are even used as a memory test in some forms of IQ testing. In this project, we want to create a hangman game where the player may select his own word and compete with others by enabling them to identify the specified word.

To accomplish the required functionality, we leverage latches as memory elements to hold the word to be predicted, as well as comparator circuit and other fundamental digital logic components. For the display of the words, we use Arduino Mega and to limit the number of tries we employ counters in our circuit.

CHAPTER – 1: An Overview

An Overview of the Project

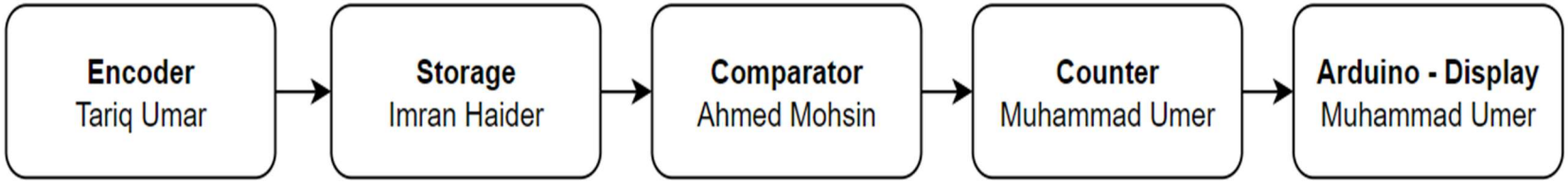
There have been many games over the years that have been developed based on sequential circuits. Our project based on the game Simon-says and Hangman is also one of them. Hangman is a paper and pencil guessing game in which a player thinks of a word phrase, or a sentence and the other player or players must guess that word with a certain number of guesses. Initially the word to be guessed is being shown by several dashes which represent the total characters of the word to be guessed. In most cases the word that is to be guessed contains proper nouns such as names of places, names, brands, and animals representing each letter of the word. Some variants of the game also include slang words which are referred to as informal or short words for a specific language. [1] If the guessed letter is not in the actual word that the player draws one element of the hanged man stick as a tally mark.

Figure 1 Hangman

Our basic goal was to create a digital version of this popular pencil and paper game using the design technique of sequential circuits which we learnt this course. We chose that the word should have maximum of 8 letters and the user will be given a maximum of 16 tries. In our implementation we used D-latches as memory elements. Moreover, we used encoder to encode each character as a 5-bit binary code and use comparator circuits to compare those bits to the stored characters. Moreover, Arduino Mega was used to display the results

Initially we take one out 26 alphabets as inputs (A-Z) with each letter encoded as a five-bit binary word. We will store these five bits in memory elements which in our case are D-latches. The user will be asked to guess the letters stored in the circuit and then by using the magnitude comparator IC we compare the guessed letter with our stored letter in the latches. The circuit also contains a counter with 7 segment display which will count the attempts made by the user. [2]

Work Division



Flowchart of Circuit

Block Diagram of the Complete System

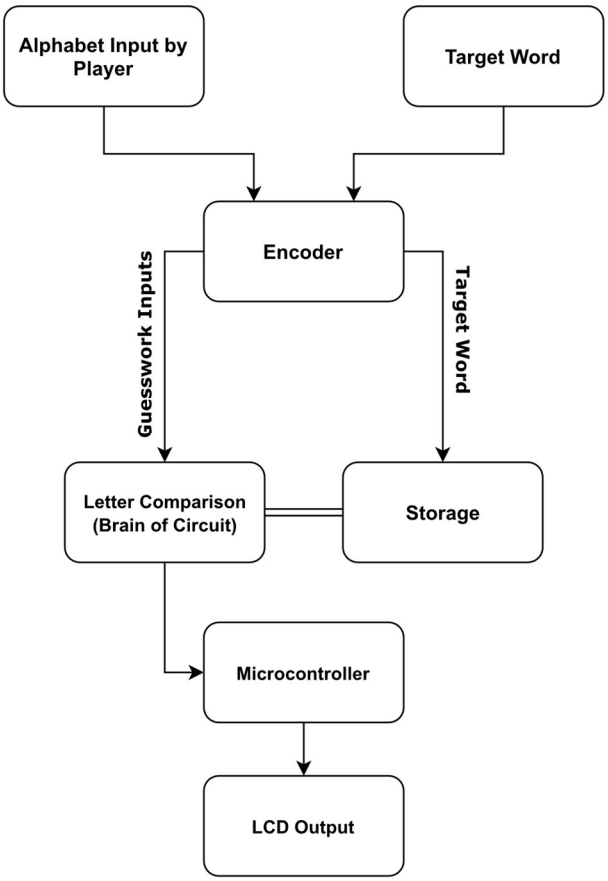


Figure 2 Block Diagram

Work Division

The encoder circuit encoded the 26 English alphabets into 5 bits and these bits are stored in D-latches IC's. The same encoder circuit is used by the player to guess the letter in the game. Upon pressing a key, the contents of the encoded bits are compared across all the comparators and the correct words are displayed on the screens.

The breakdown of our work is given as:

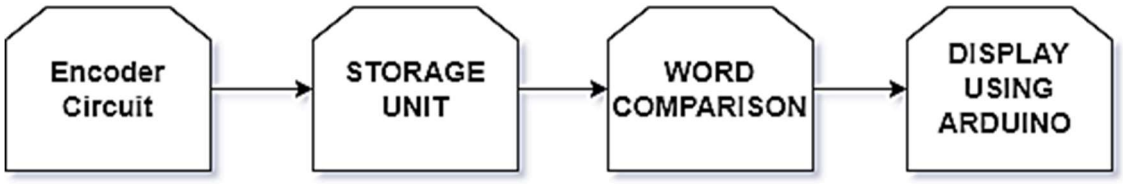


Figure 3 Flow Diagram

CHAPTER – 2: Design

The Problem Statement

Hangman is a guessing game in which the player has to guess the letter of a word. There are pretty limited number of attempts. In our case first our correct word is stored in a D-latch and then after every input the word is compared with stored word through the comparator. The letter is guessed through the 26 switches present before the encoder and the counts of the counter keep increasing after each input. If the guessed word is correct, then the letter is shown on the LCD screen through Arduino.

Truth Table

Table 1 Encoder Bits

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	O1	O2	O3	O4	O5
1																										0	0	0	0	1
	1																									0	0	0	1	0
		1																								0	0	0	1	1
			1																							0	0	1	0	0
				1																						0	0	1	0	1
					1																					0	0	1	1	0
						1																				0	0	1	1	1
							1																			0	1	0	0	0
								1																		0	1	0	0	1
									1																	0	1	0	1	0
										1																0	1	0	1	1
											1															0	1	1	0	0
												1														0	1	1	0	1
													1													0	1	1	1	0
														1												0	1	1	1	1
															1											1	0	0	0	0
																1										1	0	0	0	1
																	1									1	0	0	1	0
																		1								1	0	0	1	1
																			1							1	0	1	0	0
																				1						1	0	1	0	1
																					1					1	0	1	1	0
																						1				1	0	1	1	1
																							1			1	1	0	0	0
																								1		1	1	0	0	1
																									1	1	1	0	1	0

Simplification of Functions / K-Maps and & Equations

By inspection we observe that the derived equations for the encoded bits for the outputs:

Table 2 Expression for Bits

Bit Position	Expression
01	P+Q+R+S+T+U+V+W+X+Y+Z
02	H+I+J+K+L+M+N+O+X+Y+Z
03	D+E+F+G+L+M+N+O+T+U+V+W
04	B+C+F+G+J+K+N+O+R+S+V+W+Z
05	O5=A+C+E+G+I+K+M+O+Q+S+U+W+Y

Complete Proteus Design

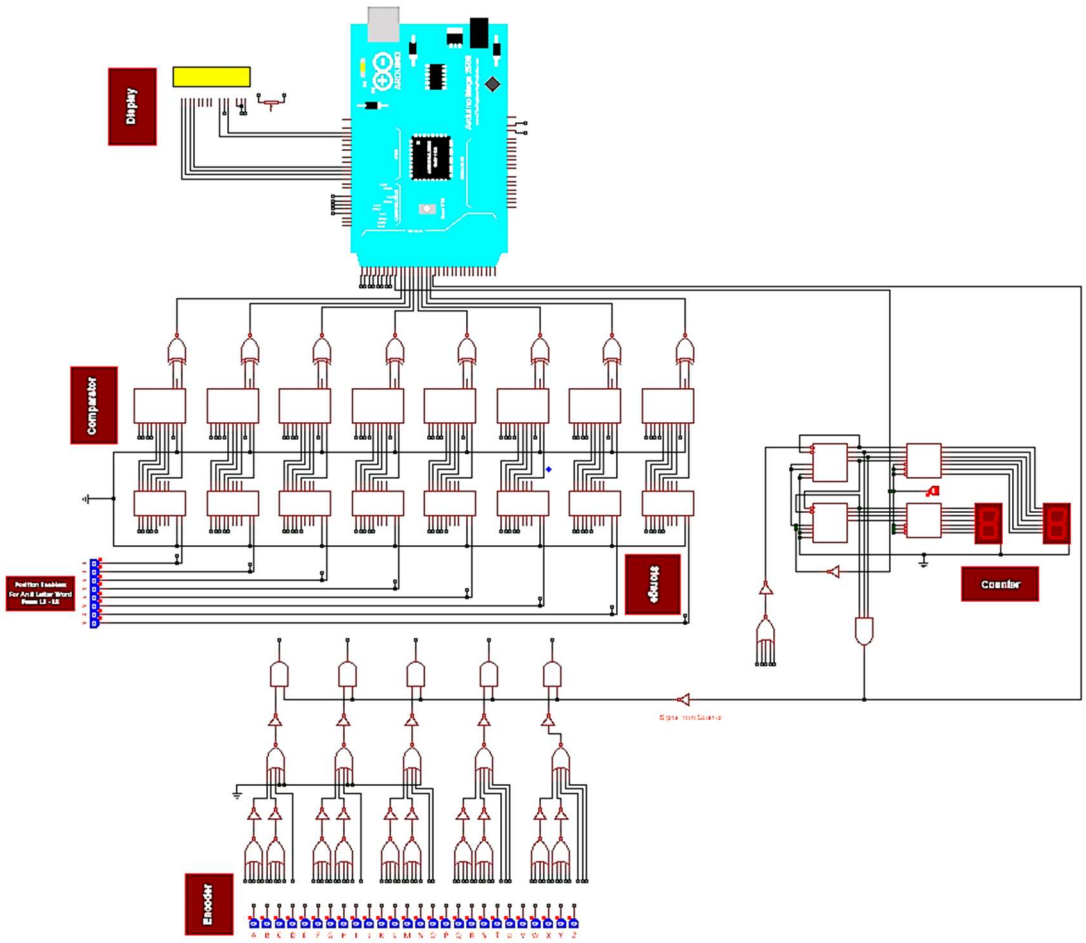


Figure 4 Proteus Design

Simulation of LCD

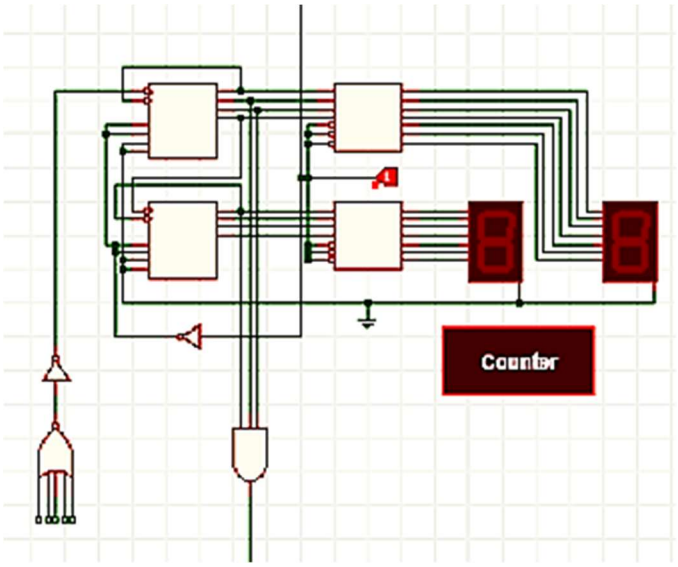


Figure 5 Counter

Complete Logic Diagram

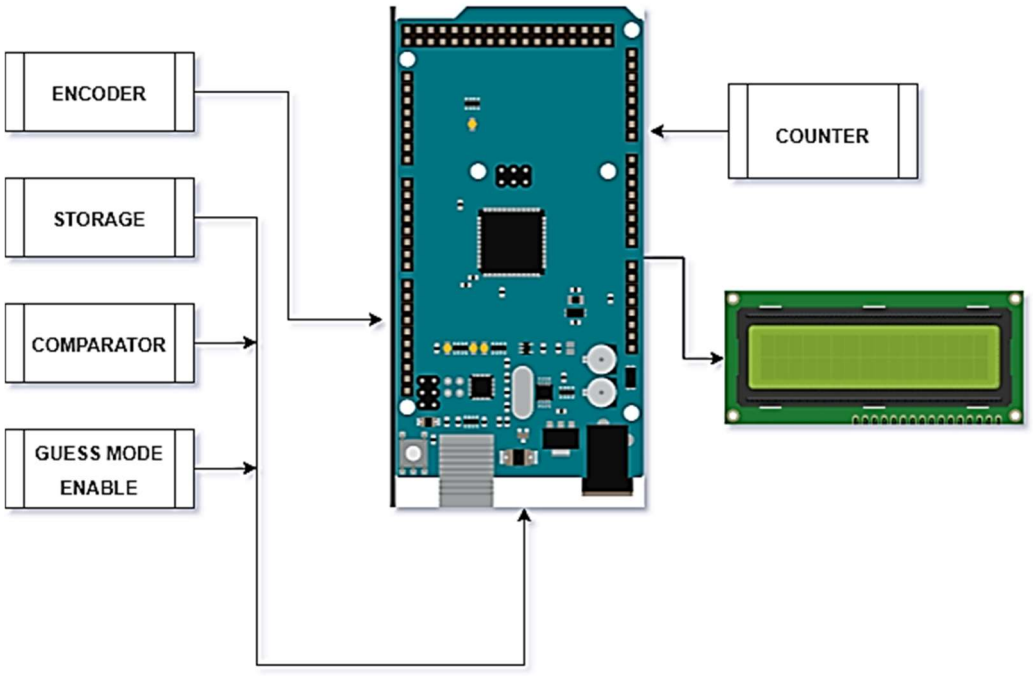
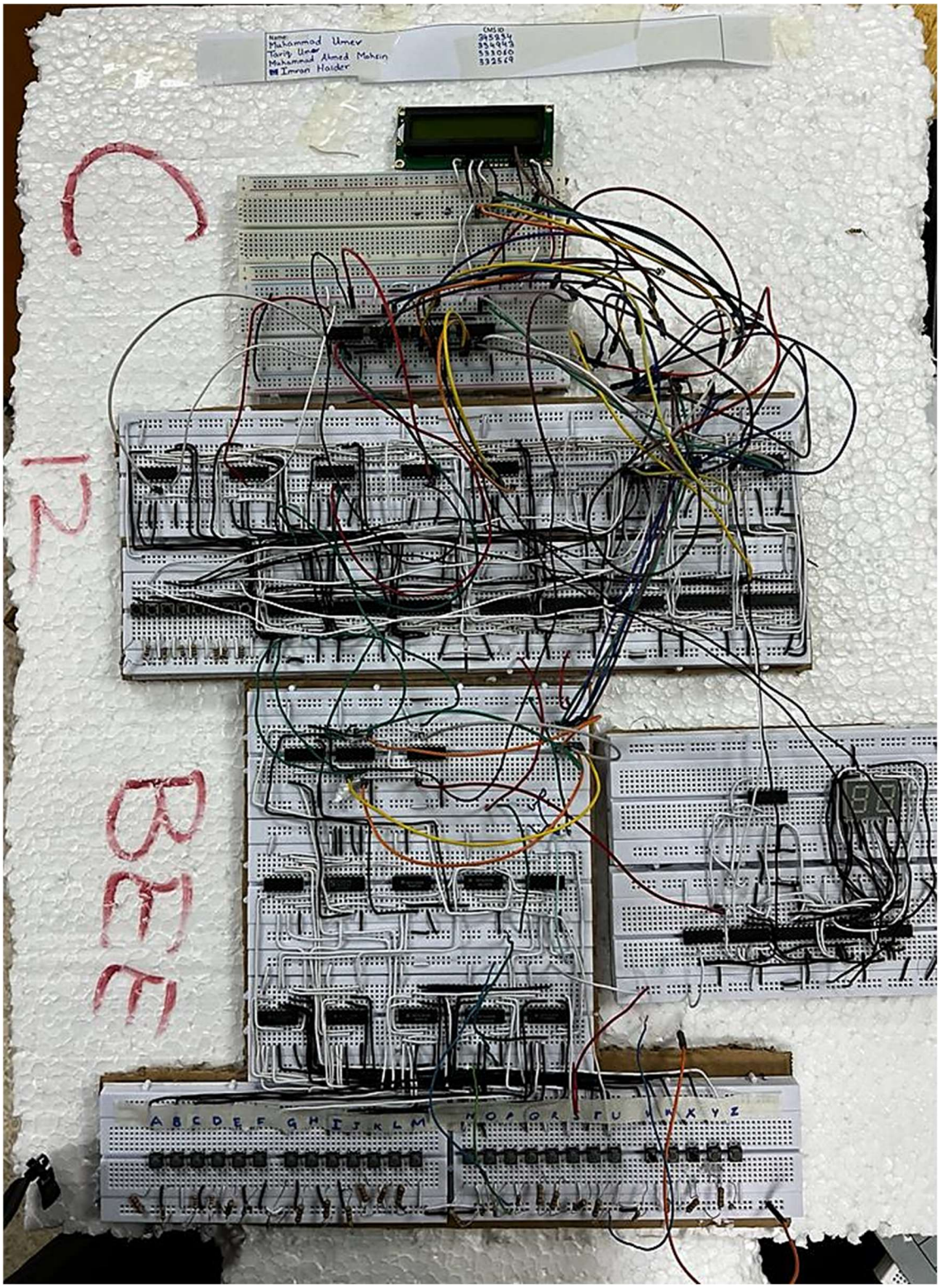


Figure 6 Logic Diagram

CHAPTER – 3: Hardware Implementation



Design of Encoder

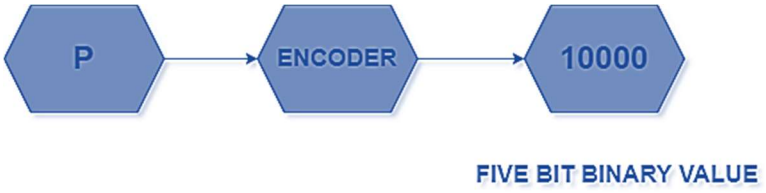


Figure 7 Encoding Flow

Each of the 26 switches are then mapped to encoder parts which encodes the given alphabet into a five-bit binary code. Whenever the button is pressed the corresponding alphabet is encoded. The word is then transmitted to other parts of the circuit to perform further actions. [3]

Table 3 Encoded Alphabets

5 – Bit Binary Number					Alphabet
0	0	0	0	1	A
0	0	0	1	0	B
0	0	0	1	1	C
0	0	1	0	0	D
0	0	1	0	1	E
0	0	1	1	0	F
0	0	1	1	1	G
0	1	0	0	0	H
0	1	0	0	1	I
0	1	0	1	0	J
0	1	0	1	1	K
0	1	1	0	0	L
0	1	1	0	1	M
0	1	1	1	0	N
0	1	1	1	1	O
1	0	0	0	0	P
1	0	0	0	1	Q
1	0	0	1	0	R
1	0	0	1	1	S
1	0	1	0	0	T
1	0	1	0	1	U
1	0	1	1	0	V
1	0	1	1	1	W
1	1	0	0	0	X
1	1	0	0	1	Y
1	1	0	1	0	Z

The hardware implementation of the encoder that we tried previously is shown as:

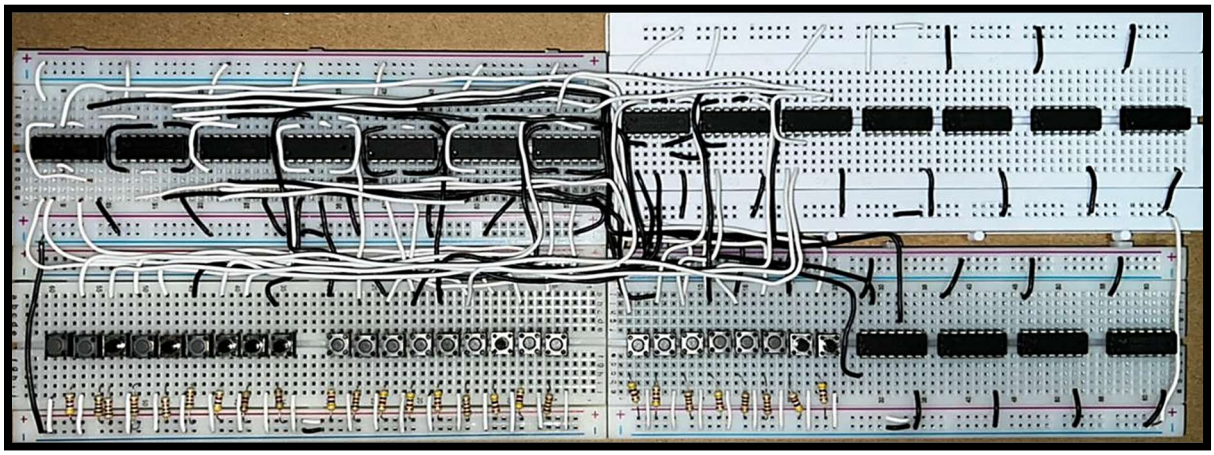


Figure 8 Previous Encoder

The above encoder did not work properly in hardware although it worked perfectly in hardware. The reason was due to some error in switches and the resistors attached to the switches could not pass the low input to the switch.

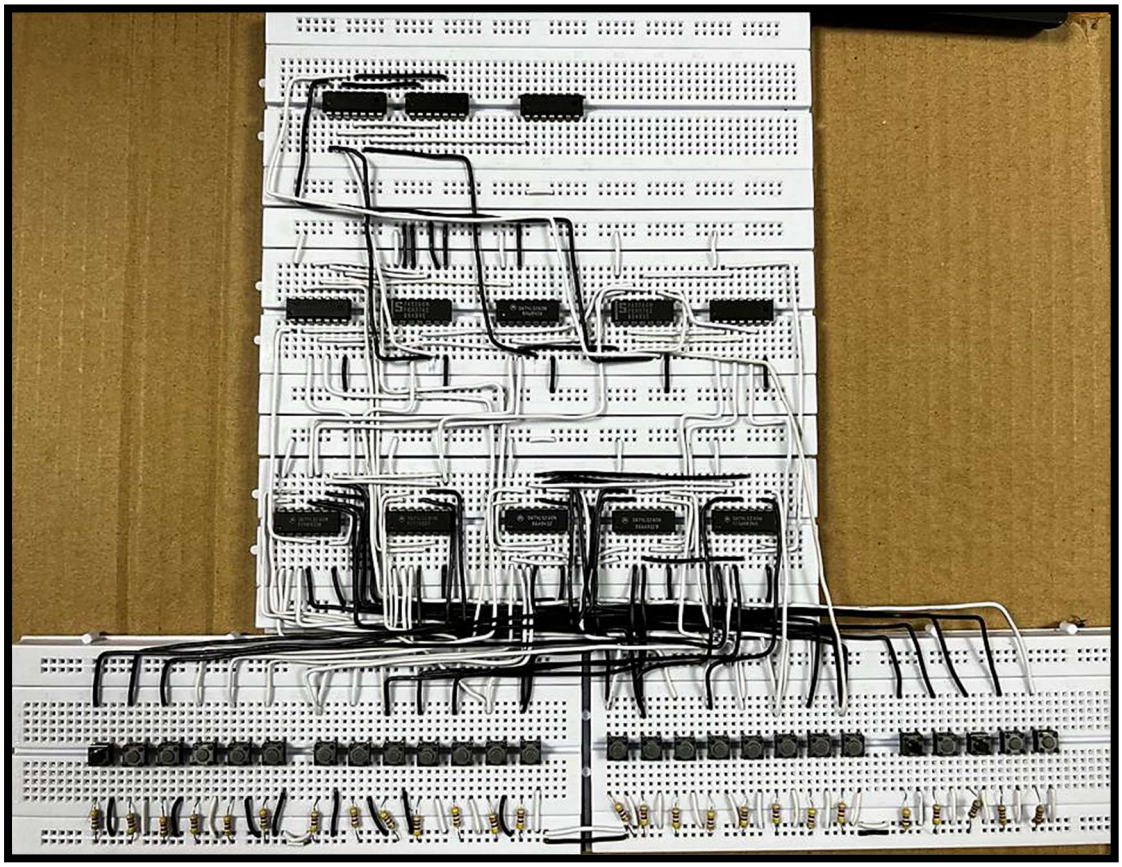


Figure 9 New Encoder

Memory Elements

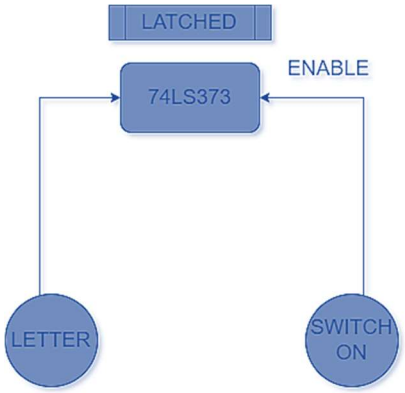


Figure 10 Memory Flow

The memory element in the octal D-latches, the 5 bits that represent each letter of the word are first stored in the latches before the hangman game is started again.

This step is necessary to allow the circuit to have the reference word with which the input is matched:

Word Comparison

After we have stored the words, we now have the next task of comparing the stored words with the words that are input by the user. We have used a four-bit magnitude comparator to compare the inputs with the already stored inputs. We actually have 5 bits to compare so we use cascaded 4-bit comparators for that. The non-standard implementation of 4-bit comparator allowed us to use 4-bit comparator to find the similarity between 5 bits. [4]

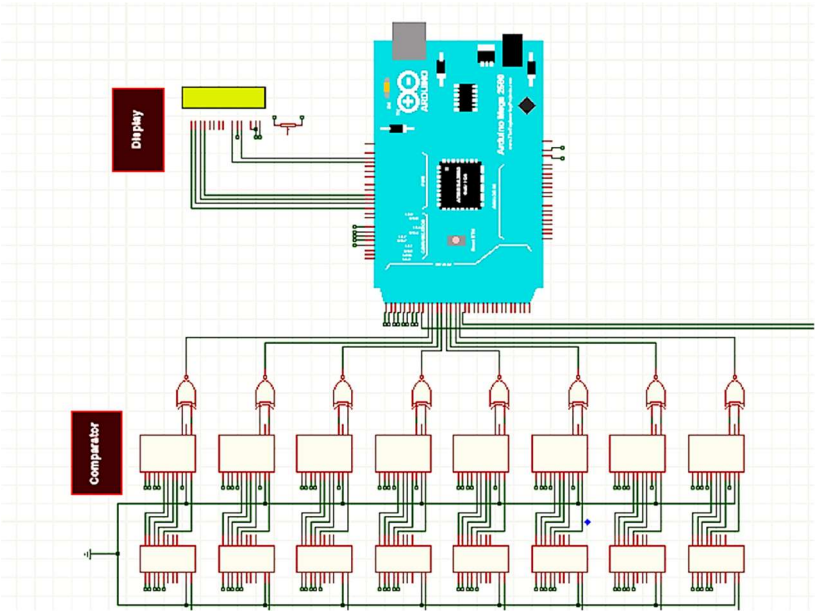


Figure 11 Comparator Circuit

Counter

A timer was available to tally the number of opportunities made by user, allowing users to gain an approximation of the number of tries left before making any more predictions. Because the restricted number of attempts is such an important component of the sport, this is required.

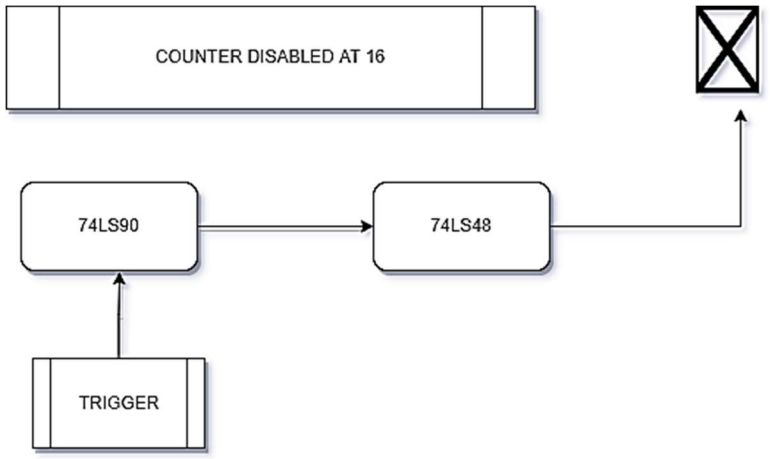


Figure 12 Counter Flow

Proteus Simulation of Counter

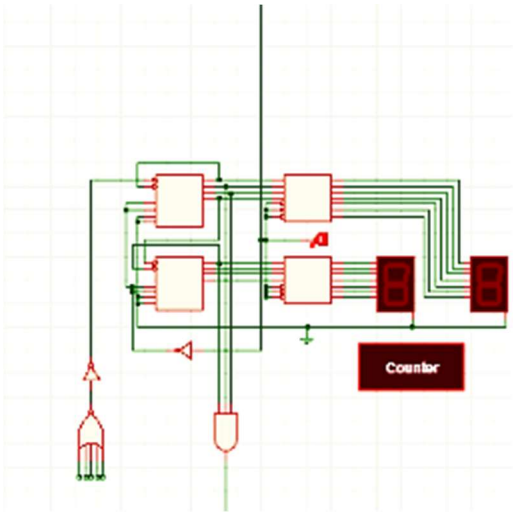


Figure 13 Counter Diagram

Display

Now in order to display the correct guesses made by the user we use a 16x2 LCD that is driven by Arduino mega. The unguessed words are represented to the screen by '_' in their respective positions. The unguessed words are changed by alphabets at their respective positions when the user guesses the correct words.

Arduino Code to Drive LCD

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
  // put your setup code here, to run once:
  lcd.begin(16, 2);
  Serial.begin(9600);

  //For the b 1 - 5 vlaues
  pinMode(14, INPUT);
  pinMode(15, INPUT);
  pinMode(16, INPUT);
  pinMode(17, INPUT);
  pinMode(18, INPUT);

  //For the l 1 - 8 vlaues
  pinMode(22, INPUT);
  pinMode(23, INPUT);
  pinMode(24, INPUT);
  pinMode(25, INPUT);
  pinMode(26, INPUT);
  pinMode(27, INPUT);
  pinMode(28, INPUT);
  pinMode(29, INPUT);

  //For the p 1 - 8 vlaues
  pinMode(31, INPUT);
  pinMode(32, INPUT);
  pinMode(33, INPUT);
  pinMode(34, INPUT);
  pinMode(35, INPUT);
  pinMode(36, INPUT);
  pinMode(37, INPUT);
  pinMode(38, INPUT);

  //For guess mode
  pinMode(30, INPUT);
  pinMode(39, INPUT);

  lcd.print("HANGMAN");
  lcd.setCursor(0,1);
}

int finalVar;

int b1;
int b2;
int b3;
int b4;
int b5;

int l1;
int l2;
int l3;
int l4;
int l5;
int l6;
int l7;
int l8;
```

```
int p1;
int p2;
int p3;
int p4;
int p5;
int p6;
int p7;
int p8;

int r1;
int r2;

char word1 = '_';
char word2 = '_';
char word3 = '_';
char word4 = '_';
char word5 = '_';
char word6 = '_';
char word7 = '_';
char word8 = '_';

void loop() {
  //Reading Bs
  b1 = digitalRead(14);
  b2 = digitalRead(15);
  b3 = digitalRead(16);
  b4 = digitalRead(17);
  b5 = digitalRead(18);

  //Reading Ls
  l1 = digitalRead(29);
  l2 = digitalRead(28);
  l3 = digitalRead(27);
  l4 = digitalRead(26);
  l5 = digitalRead(25);
  l6 = digitalRead(24);
  l7 = digitalRead(23);
  l8 = digitalRead(22);

  //Reading Ps
  p1 = digitalRead(31);
  p2 = digitalRead(32);
  p3 = digitalRead(33);
  p4 = digitalRead(34);
  p5 = digitalRead(35);
  p6 = digitalRead(36);
  p7 = digitalRead(37);
  p8 = digitalRead(38);

  //Reading modes
  r1 = digitalRead(30);
  r2 = digitalRead(39);

  if (r1 == 1) {
    pos1(b1, b2, b3, b4, b5, l1, p1);
    pos2(b1, b2, b3, b4, b5, l2, p2);
    pos3(b1, b2, b3, b4, b5, l3, p3);
    pos4(b1, b2, b3, b4, b5, l4, p4);
    pos5(b1, b2, b3, b4, b5, l5, p5);
    pos6(b1, b2, b3, b4, b5, l6, p6);
    pos7(b1, b2, b3, b4, b5, l7, p7);
    pos8(b1, b2, b3, b4, b5, l8, p8);
  }
```

```
  if (r1 == 0){
      lcd.setCursor(0,1);
      lcd.print("STORAGE!");
      lcd.setCursor(0,1);
  }
  if ((r2 == 1) && ((word1 == '') || (word2 == '') || (word3 == '') || (word4 == '') || (word5 == '')
|| (word6 == '') || (word7 == '') || (word8 == ''))){
      lcd.clear();
      lcd.setCursor(0,1);
      lcd.print("TRY AGAIN!");
  }
}

void pos1(int b1, int b2, int b3, int b4, int b5, int l1, int p1)
{
    if ((b1 == 0) && (b2 == 0) && (b3 == 0) && (b4 == 0) && (b5 == 0)) {
        lcd.setCursor(0, 1);
        lcd.print(word1);
    }
    if ((l1 == 0 && p1 == 1) && ((b1 == 1) || (b2 == 1) || (b3 == 1) || (b4 == 1) || (b5 == 1))) {
        lcd.setCursor(0, 1);
        finalVar = (b1 << 4) | (b2 << 3) | (b3 << 2) | (b4 << 1) | (b5);
        char wordy = 64 + finalVar;
        lcd.print(wordy);
        word1 = wordy;
    }
    else {
        lcd.setCursor(0, 1);
        lcd.print(word1);
    }
}


void pos2(int b1, int b2, int b3, int b4, int b5, int l2, int p2)
{
    if ((b1 == 0) && (b2 == 0) && (b3 == 0) && (b4 == 0) && (b5 == 0)) {
        lcd.setCursor(1, 1);
        lcd.print(word2);
    }
    if ((l2 == 0 && p2 == 1) && ((b1 == 1) || (b2 == 1) || (b3 == 1) || (b4 == 1) || (b5 == 1))) {
        lcd.setCursor(1, 1);
        finalVar = (b1 << 4) | (b2 << 3) | (b3 << 2) | (b4 << 1) | (b5);
        char wordy = 64 + finalVar;
        lcd.print(wordy);
        word2 = wordy;
    }

    else {
        lcd.setCursor(1, 1);
        lcd.print(word2);
    }
}

void pos3(int b1, int b2, int b3, int b4, int b5, int l3, int p3)
{
    if ((b1 == 0) && (b2 == 0) && (b3 == 0) && (b4 == 0) && (b5 == 0)) {
        lcd.setCursor(2, 1);
        lcd.print(word3);
    }
    if ((l3 == 0 && p3 == 1) && ((b1 == 1) || (b2 == 1) || (b3 == 1) || (b4 == 1) || (b5 == 1))) {
        lcd.setCursor(2, 1);
        finalVar = (b1 << 4) | (b2 << 3) | (b3 << 2) | (b4 << 1) | (b5);
        char wordy = 64 + finalVar;
        lcd.print(wordy);
        word3 = wordy;
```

```
    }

    else {
        lcd.setCursor(2, 1);
        lcd.print(word3);
    }
}

void pos4(int b1, int b2, int b3, int b4, int b5, int l4, int p4)
{
    if ((b1 == 0) && (b2 == 0) && (b3 == 0) && (b4 == 0) && (b5 == 0)) {
        lcd.setCursor(3, 1);
        lcd.print(word4);
    }
    if ((l4 == 0 && p4 == 1) && ((b1 == 1) || (b2 == 1) || (b3 == 1) || (b4 == 1) || (b5 == 1))) {
        lcd.setCursor(3, 1);
        finalVar = (b1 << 4) | (b2 << 3) | (b3 << 2) | (b4 << 1) | (b5);
        char wordy = 64 + finalVar;
        lcd.print(wordy);
        word4 = wordy;
    }

    else {
        lcd.setCursor(3, 1);
        lcd.print(word4);
    }
}

void pos5(int b1, int b2, int b3, int b4, int b5, int l5, int p5)
{
    if ((b1 == 0) && (b2 == 0) && (b3 == 0) && (b4 == 0) && (b5 == 0)) {
        lcd.setCursor(4, 1);
        lcd.print(word5);
    }
    if ((l5 == 0 && p5 == 1) && ((b1 == 1) || (b2 == 1) || (b3 == 1) || (b4 == 1) || (b5 == 1))) {
        lcd.setCursor(4, 1);
        finalVar = (b1 << 4) | (b2 << 3) | (b3 << 2) | (b4 << 1) | (b5);
        char wordy = 64 + finalVar;
        lcd.print(wordy);
        word5 = wordy;
    }

    else {
        lcd.setCursor(4, 1);
        lcd.print(word5);
    }
}

void pos6(int b1, int b2, int b3, int b4, int b5, int l6, int p6)
{
    if ((b1 == 0) && (b2 == 0) && (b3 == 0) && (b4 == 0) && (b5 == 0)) {
        lcd.setCursor(5, 1);
        lcd.print(word6);
    }
    if ((l6 == 0 && p6 == 1) && ((b1 == 1) || (b2 == 1) || (b3 == 1) || (b4 == 1) || (b5 == 1))) {
        lcd.setCursor(5, 1);
        finalVar = (b1 << 4) | (b2 << 3) | (b3 << 2) | (b4 << 1) | (b5);
        char wordy = 64 + finalVar;
        lcd.print(wordy);
        word6 = wordy;
    }

    else {
        lcd.setCursor(5, 1);
        lcd.print(word6);
```

```
    }
}

void pos7(int b1, int b2, int b3, int b4, int b5, int l7, int p7)
{
    if ((b1 == 0) && (b2 == 0) && (b3 == 0) && (b4 == 0) && (b5 == 0)) {
        lcd.setCursor(6, 1);
        lcd.print(word7);
    }
    if ((l7 == 0 && p7 == 1) && ((b1 == 1) || (b2 == 1) || (b3 == 1) || (b4 == 1) || (b5 == 1))) {
        lcd.setCursor(6, 1);
        finalVar = (b1 << 4) | (b2 << 3) | (b3 << 2) | (b4 << 1) | (b5);
        char wordy = 64 + finalVar;
        lcd.print(wordy);
        word7 = wordy;
    }

    else {
        lcd.setCursor(6, 1);
        lcd.print(word7);
    }
}

void pos8(int b1, int b2, int b3, int b4, int b5, int l8, int p8)
{
    if ((b1 == 0) && (b2 == 0) && (b3 == 0) && (b4 == 0) && (b5 == 0)) {
        lcd.setCursor(7, 1);
        lcd.print(word8);
    }
    if ((l8 == 0 && p8 == 1) && ((b1 == 1) || (b2 == 1) || (b3 == 1) || (b4 == 1) || (b5 == 1))) {
        lcd.setCursor(7, 1);
        finalVar = (b1 << 4) | (b2 << 3) | (b3 << 2) | (b4 << 1) | (b5);
        char wordy = 64 + finalVar;
        lcd.print(wordy);
        word8 = wordy;
    }

    else {
        lcd.setCursor(7, 1);
        lcd.print(word8);
    }
}
```

Chapter 4: Conclusion

Table 4 Details of ICs Used

Sr.no	Item	Quantity
1	Push Buttons	35
3	74LS373	8
4	74LS85	8
5	74LS86	2
6	7404	6
8	74LS260	8
9	74LS08	5

Details of Other Components used like diodes, transistors, resistors etc.

- Arduino Mega
- LCD Display
- Resistor: 100 ohms
- DIP Switches
- Push Buttons
- Breadboards

Hardware Issues

We faced an issue when we applied voltage source on switches the low voltage wasn't being passed on the ground of the switch. Instead, it was being open circuit.
The project was completed in time with counter and storage, counter and display working as expected. Multiple power supplies were used in order to provide sufficient power to the ICs.
It has been observed that along with voltage, the current also plays a vital role in proper functioning of the IC. Furthermore, it is also observed that an appropriate amount of current is required for the proper functioning of the transistors that are used in the circuit.

Practical Applications of the Project

The project has quite a wide range of practical applications and learning uses. The project is a practical application of how to use asynchronous storage elements and display them on screen.
Moreover, the project demonstrates that extra complexity that helps the individuals to understand and develop interest the complex nature of circuits used in daily life and how to implement them to perform various tasks.
Another major contribution of this game is for the patients suffering from Alzheimer's patients and is used as a brain exercise for them.

Bill of Materials

Table 5 Bill of ICs

Sr.no	Item	Quantity	Per Unit Price	Total
1	Push Buttons	35	2	70
2	74LS373	8	35	280
3	74LS85	8	50	296
4	74LS86	2	30	60
5	74LS04	6	25	150
6	74LS260	8	40	320
7	74LS08	5	30	150
10	Arduino Mega	1	2450	2450
11	100 Ohm Resistors	35	2	70
12	16x2 LCD	1	270	270
13	7 Segment Display	2	20	40
14	Breadboard	14	150	1650

The total cost for the circuit is: 5806 Rs.

References

The references used in the report are as:

1. https://en.wikipedia.org/wiki/Hangman_(game)#:~:text=Hangman%20is%20a%20paper%20and,a%20certain%20number%20of%20guesses.
2. https://www.wikihow.com/Play-Hangman
3. http://www.printactivities.com/Paper-Games/Rules-For-Hangman.shtml
4. https://inventwithpython.com/invent4thed/chapter7.html

Datasheets

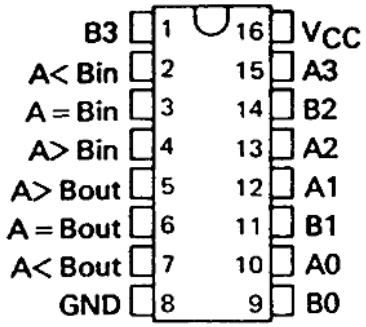
1. 74LS04 NOT Gate - https://www.ti.com/lit/ds/symlink/sn74ls04.pdf
2. 74LS373 Octal D-Latch - https://www.ti.com/lit/gpn/sn54ls373-sp
3. 74LS260 NOR Gate - http://www.datasheet.es/PDF/248180/74260-pdf.html
4. 74LS85 Comparator - https://www.ti.com/lit/gpn/sn54s85
5. 74LS08 AND Gate - https://www.ti.com/lit/gpn/sn54ls08-sp
6. 74LS90 Decade Counter - http://web.sonoma.edu/users/m/marivani/datasheets/74ls_series/74ls90.pdf
7. 74LS48 7 Segment Decoder - https://www.ti.com/lit/gpn/SN5447A

Appendix

SN5485, SN54LS85, SN54S85
SN7485, SN74LS85, SN74S85
4-BIT MAGNITUDE COMPARATORS

SDLS123 – MARCH 1974 – REVISED MARCH 1988

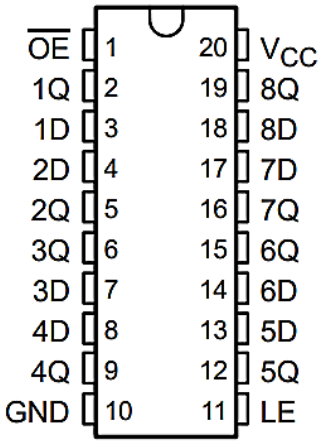
SN5485, SN54LS85, SN54S85 . . . J OR W PACKAGE
SN7485 . . . N PACKAGE
SN74LS85, SN74S85 . . . D OR N PACKAGE
(TOP VIEW)



SN54ACT373, SN74ACT373
OCTAL D-TYPE TRANSPARENT LATCHES WITH 3-STATE OUTPUTS

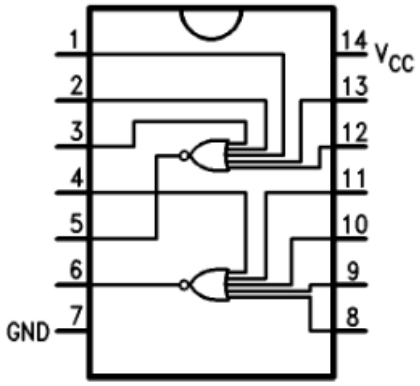
SCAS544E – OCTOBER 1995 – REVISED OCTOBER 2002

SN54ACT373 . . . J OR W PACKAGE
SN74ACT373 . . . DB, DW, N, NS, OR PW PACKAGE
(TOP VIEW)

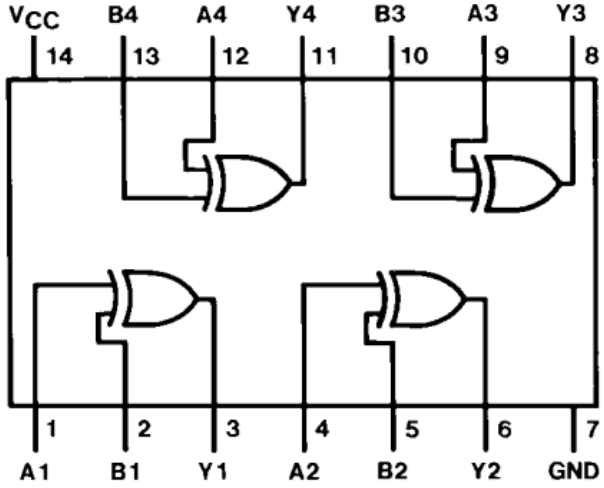


DM54LS260/DM74LS260
Dual 5-Input NOR Gate

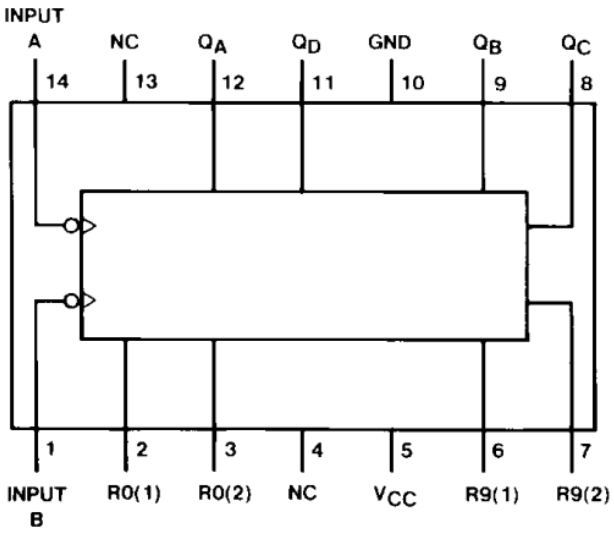
Dual-In-Line Package



Quad 2-Input Exclusive-OR Gate



DM74LS90
Decade and Binary Counters



BCD TO 7-SEGMENT DECODER