DIGITAL DESIGN LAB

MINI - PROJECT

ASCII LETTER TO MORSE CODE CONVERTER

AIM:

To design an ASCII Letter to Morse Code converter using basic sequential and combinational logic.

APPARATUS REQUIRED

S.No.	Component	Specification	Quantity
1	AND - Gate	IC7408	
2	NOT - Gate	IC7404	
3.	OR - Gate	IC 7432	
4.	JK - Flipflop	IC 7476	
5.	NAND - Gate	IC 7400	
6.	16-1 Multiplexer	IC74150	
7	Connecting Wires		
8	IC Trainer Kit		

THEORY

Morse code is a system of sending messages that is used to send telegraphic information using signals and rythm. Morse uses dots and dashes of a specific sequence to represent letters of a message. When messages are sent by sound or radio, dots are signals of a short duration, and dashes are signals of a longer duration.

ASCII is a table of characters for computers used by systems to handle text using the english alphabet, numbers, or certain special characters. ASCII is an abbreviation of American Standard Code for Information Interchange.

In Morse, every letter is represented as a sequence of dots and dashes, with spaces between any two dots or dashes, and in ASCII, every letter is represented as a sequence of 7 binary digits, of which only 5 denote the actual letter.

The longest letters to represent in morse require 2 dashes and two dots. If each dash requires 2 bits, and each dot, or space needs one bit, then the longest morse letters need 10 bits to be represented in a sequential format.

Thus, a converter from ASCII to morse code would take 5 bits as input, and give 10 bits as output.

These 10 bits may be directed to a multiplexer, and the selector bits of this multiplexer may be driven by a MOD-10 [BCD] counter, This will result in the output of the multiplexer transmitting the morse code equivalent of the character at the Input.

SPECIFICATIONS:

1) Draw the truth table for the ASCII (5 bit) to Morse (10 bit) conversion.
   1.1) For each of the output variables, draw the K Map and solve for the boolean expression.
   1.2) Implement the boolean expressions for each of the output variables.
2. Connect the 10 output bits ~~to~~ of the combinational logic circuit to the 10 Least Significant bit inputs of the 16-1 multiplexer.
3. Construct a MOD 10 (BCD) counter using the JK-flipflops.
   3.1) Ensure that the counter may also be reset when the START input of the system is held low.
   3.2) The 4 output bits of this Counter must be connected to the selector lines of the 16-1 MUX.
4. Ensure that signals are inverted when input to active-low inputs, and verify that the morse code of a letter is correctly output when the input is set accordingly, and the START input is held high.

USES FOR CIRCUIT IN REAL WORLD APPLICATIONS:

→ In robust military communication systems, as a failsafe in case other less tedious forms of communication fail.

→ As a teaching aid to help students understand Morse Code.

→ As a teaching aid, to help students understand ASCII.

→ In emergency communication systems.

INPUT AND OUTPUT:

INPUTS:

- A : 5th LSB of ASCII Character (Letter in English)
- B : 4th LSB of ASCII Character (Letter in English)
- C : 3rd LSB of ASCII Character (Letter in English)
- D : 2nd LSB of ASCII Character (Letter in English)
- E : LSB of ASCII character (Letter in English)
- START: Signal to indicate when the circuit should start outputting morse in time with input clock signal.

OUTPUT:

→ M :
- Goes high for 2 clock pulses to indicate a dash '—' and goes high for 1 clock pulse to indicate a dot '•'. Is low for 1 clock pulse to denote space between any two dots or dashes.
- Is low when START input is low.

SSN COLLEGE OF ENGINEERING
RECORD SHEET

Sheet No...................

TRUTH TABLE : ASCII (5 bit) → Morse (10 bit) Conversion.

Char	INPUT					OUTPUT									
	A	B	C	D	E	a	b	c	d	e	f	g	h	i	j
INV	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X
A	0	0	0	0	1	1	0	1	1	0	0	0	0	0	0
B	0	0	0	1	0	1	1	0	1	0	1	0	1	0	0
C	0	0	0	1	1	1	1	0	1	0	1	1	0	1	0
D	0	0	1	0	0	1	1	0	1	0	1	0	0	0	0
E	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0
F	0	0	1	1	0	1	0	1	0	1	1	0	1	0	0
G	0	0	1	1	1	1	1	0	1	1	0	1	0	0	0
H	0	1	0	0	0	1	0	1	0	1	0	1	0	0	0
I	0	1	0	0	1	1	0	1	0	0	0	0	0	0	0
J	0	1	0	1	0	1	0	1	1	0	1	1	0	1	1
K	0	1	0	1	1	1	1	0	1	0	1	1	0	0	0
L	0	1	1	0	0	1	0	1	1	0	1	0	1	0	0
M	0	1	1	0	1	1	1	0	1	1	0	0	0	0	0
N	0	1	1	1	0	1	1	0	1	0	0	0	0	0	0
O	0	1	1	1	1	1	1	0	1	1	0	1	1	0	0
P	1	0	0	0	0	1	0	1	1	0	1	1	0	1	0
Q	1	0	0	0	1	1	1	0	1	1	0	1	0	1	1
R	1	0	0	1	0	1	0	1	1	0	1	0	0	0	0
S	1	0	0	1	1	1	0	1	0	1	0	0	0	0	0
T	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0
U	1	0	1	0	1	1	0	1	0	1	1	0	0	0	0
V	1	0	1	1	0	1	0	1	0	1	0	1	1	0	0

SSN COLLEGE OF ENGINEERING

RECORD SHEET

Sheet No..................

W	1	0	1	1	1		1	0	1	1	0	1	1	0	0	0
X	1	1	0	0	0		1	1	0	1	0	1	0	1	1	0
Y	1	1	0	0	1		1	1	0	1	0	1	1	0	1	1
Z	1	1	0	1	0		1	1	0	1	1	0	1	0	1	0
INV	1	1	0	1	1		x	x	x	x	x	x	x	x	x	x
INV	1	1	1	x	x		x	x	x	x	x	x	x	x	x	x

K-Maps And Output Equations.

SSN COLLEGE OF ENGINEERING
RECORD SHEET
Sheet No...................

a

A'

BC \ DE	00	01	11	10
00	x	1 i	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

A

BC \ DE	00	01	11	10
00	1	1 i	1	1
01	1	1	1	1
11	x	x	x	x
10	1	1	x	1

$a = 1$

b

A'

BC \ DE	00	01	11	10
00	x	0	1 ii	1 v
01	1 vi	0	1	0
11	0	1 iii	1 iv	1
10	0	0	1	0

A

BC \ DE	00	01	11	10
00	0	1 vii	0	0
01	1 vi	0	0	0
11	x	x iii	x	x iv
10	1	1 viii	x i	1

$b = AB + A'DE + BCE + BCD + A'B'C'D + B'CD'E' + AC'D'E$

c

A'

BC \ DE	00	01	11	10
00	x ii	1	0	0
01	0	0	0	1 iv
11	1 vi	0	0	0
10	v 1	ii 1	0	1 v

A

BC \ DE	00	01	11	10
00	vii 1	0	1 i	1 vii
01	0	1 iii	1	1 iv
11	x	x	x	x
10	0	0	x	0

$c = AB'D + A'C'D' + ACE + B'CDE' + A'BCDE' + A'BC'E' + A'BD'E' + AB'C'E'$

d

A'

BC \ DE	00	01	11	10
00	x v	1	1	1 iii
01	1 vii	0	1 vi	0
11	1	ii 1	1	1
10	0	0	1 i	1 iii

A

BC \ DE	00	01	11	10
00	1 iv	1	0	1 iii
01	0	0	1 vi	0
11	x	x ii	x	x
10	1	iv 1	x i	1 iii

$d = BD + BC + C'DE' + AC'D' + A'B'C' + CDE + A'CD'E'$

e

A'

BC \ DE	00	01	11	10
00	vii x	0	0	0
01	0	0	1 iii	1 iv
11	0	1 i	1	0
10	x ii	0	0	0

A

BC \ DE	00	01	11	10
00	0	1	v 1	0
01	0	vi 1	0	1 iv
11	x	x i	x	x
10	0	0	x ii	1

$e = BCE + ABD + A'B'CD + B'CDE' + AB'C'E + AB'D'E + A'C'D'E'$

f

A'

BC \ DE	00	01	11	10
00	X	0	1	1
01	1	0	0	1
11	1	0	0	0
10	0	0	1	1

A

BC \ DE	00	01	11	10
00	1	0	0	1
01	0	1	1	0
11	X	X	X	X
10	1	1	X	0

$F = A'C'D + A'B'E' + B'C'E' + ACE + ABD' + A'CD'E'$

g

A'

BC \ DE	00	01	11	10
00	X	0	1	0
01	0	0	1	0
11	0	0	1	0
10	1	0	1	1

A

BC \ DE	00	01	11	10
00	1	1	0	0
01	0	0	1	1
11	X	X	X	X
10	0	1	X	1

$g = A'DE + BC'D + ACD + ABE + A'BC'E' + AB'C'D'$

h

A'

BC \ DE	00	01	11	10
00	X	0	0	1
01	0	0	0	1
11	1	0	1	0
10	0	0	0	0

A

BC \ DE	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	X	X	X	X
10	1	0	X	0

$h = A'B'DE' + B'CDE' + BCD'E' + BCDE + ABD'E'$

i

A'

BC \ DE	00	01	11	10
00	X	0	1	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	1

A

BC \ DE	00	01	11	10
00	1	1	0	0
01	0	0	0	0
11	X	X	X	X
10	1	1	X	1

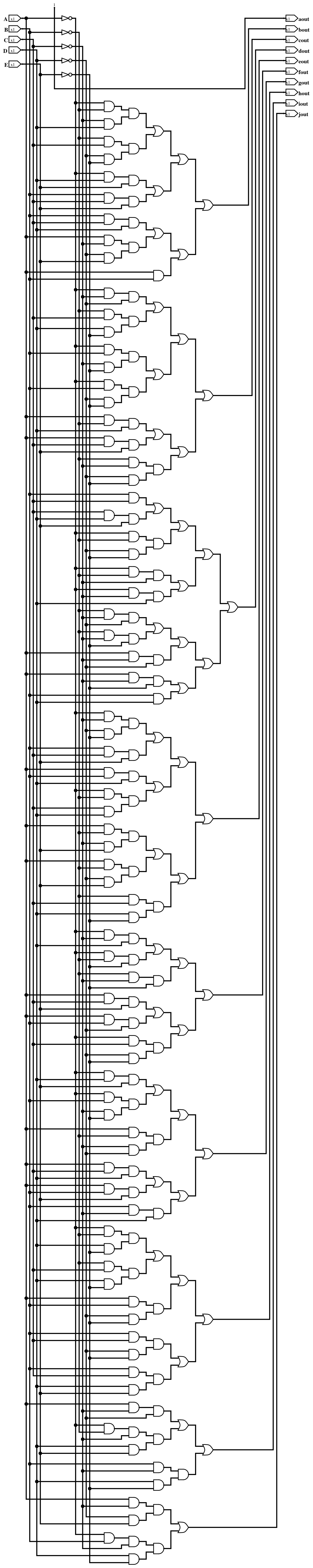
$i = AC'D' + BC'DE' + A'B'C'DE$

j

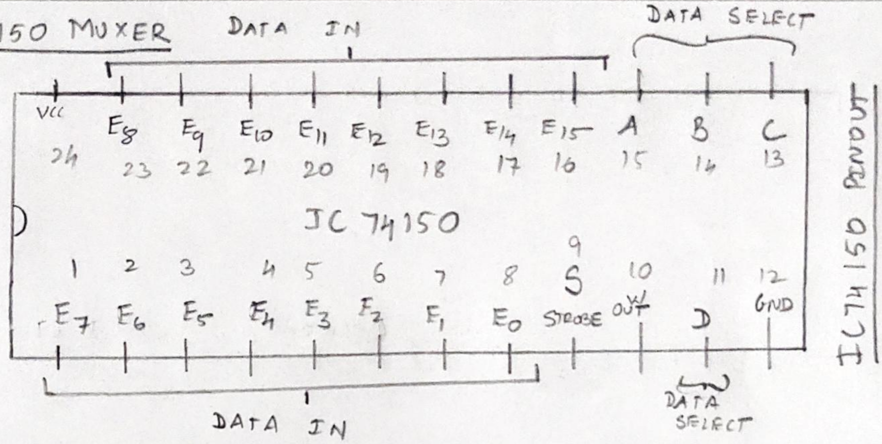
A' BC \ DE	00	01	11	10
00	X	0	0	0
01	0	0	0	0
11	0	0	0	0
10	0	0	0	1

A BC \ DE	00	01	11	10
00	0	1	0	0
01	0	0	0	0
11	X	X	X	X
10	0	1	X	0

$$j = AC'D'E + A'BC'DE'$$

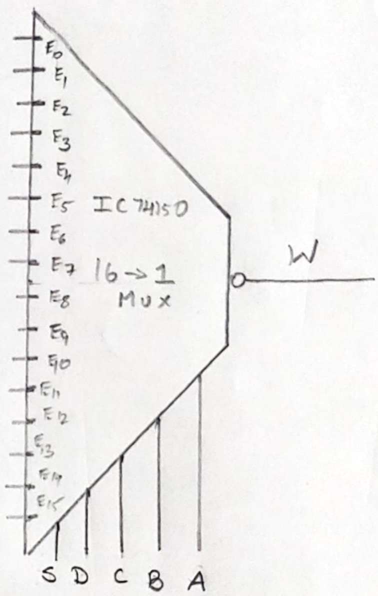

A
B
C
D
E
1
aout
bout
cout
dout
eout
fout
gout
hout
iout
jout

IC 74150 MUXER

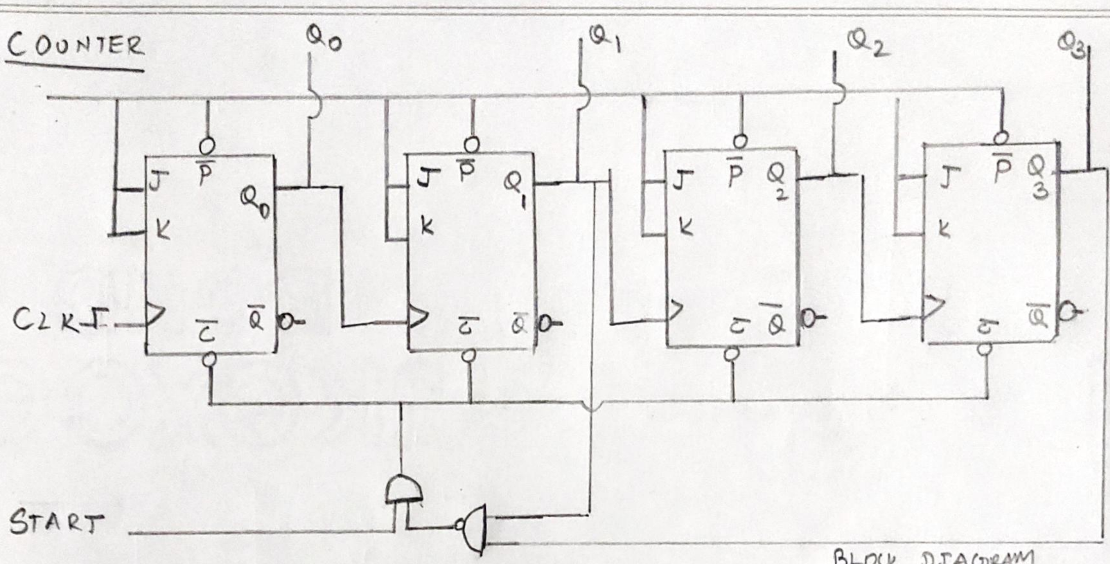


INPUT					OUTPUT
SELECT				STROBE 'S'	W
D	C	B	A		
X	X	X	X	1	1
0	0	0	0	0	$\overline{E_0}$
0	0	0	1	0	$\overline{E_1}$
0	0	1	0	0	$\overline{E_2}$
0	0	1	1	0	$\overline{E_3}$
0	1	0	0	0	$\overline{E_4}$
0	1	0	1	0	$\overline{E_5}$
0	1	1	0	0	$\overline{E_6}$
0	1	1	1	0	$\overline{E_7}$
1	0	0	0	0	$\overline{E_8}$
1	0	0	1	0	$\overline{E_9}$
1	0	1	0	0	$\overline{E_{10}}$
1	0	1	1	0	$\overline{E_{11}}$
1	1	0	0	0	$\overline{E_{12}}$
1	1	0	1	0	$\overline{E_{13}}$
1	1	1	0	0	$\overline{E_{14}}$
1	1	1	1	0	$\overline{E_{15}}$

FUNCTION TABLE

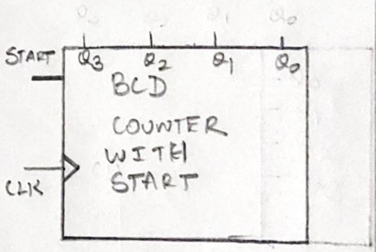


SYMBOL DIAGRAM



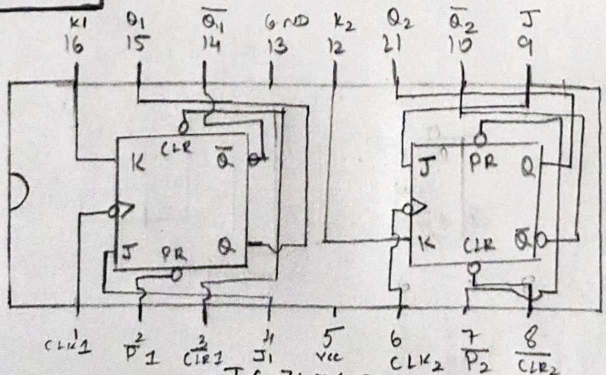
BLOCK DIAGRAM

INPUT	CLK	OUTPUT			
START		Q_3	Q_2	Q_1	Q_0
0	x	0	0	0	0
1	0	0	0	0	0
1	1	0	0	0	1
1	2	0	0	1	0
1	3	0	0	1	1
1	4	0	1	0	0
1	5	0	1	0	1
1	6	0	1	1	0
1	7	0	1	1	1
1	8	1	0	0	0
1	9	1	0	0	1
1	10	1	0	1	0
	11	0	0	0	0



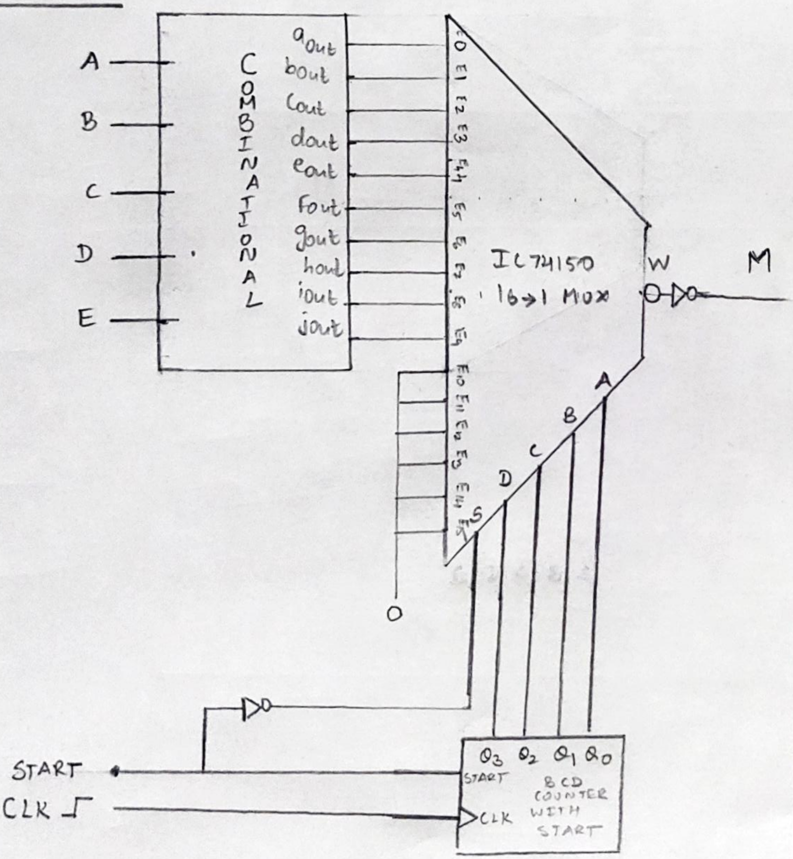
SYMBOL DIAGRAM

STATE TABLE



IC 7476 PINOUT

BLOCK DIAGRAM



SAMPLE INPUT

ASCII 'a' { 1 1 0 0 0 0 1 } a
ASCII 'A' { 1 0 0 0 0 0 1 } A

START	CLK	A	B	C	D	E	M
0	X	X	X	X	X	X	0
1	0						1
1	1						0
1	2						1
1	3						1
1	4						0
1	5	0	0	0	0	1	0
1	6						0
1	7						0
1	8						0
1	9						0
1	10						1

SAMPLE OUTPUT: 1 0 1 1 0 0 0 0 0 0 1 ...
DOT SPACE DASH SPACES

MORSE RESULT: • — } MORSE CODE FOR A/a.