Implementation of 4x1 mux in Arduino using Assembly

M Patrick Manohar
patrickmanohar152001@gmail.com
FWC22119
IIT Hyderabad-Future Wireless Communication Assignment

March 2023

Contents

1 Problem

(GATE EC-2022)

Q.19. Consider the 2-bit multiplexer(MUX) shown in the figure.For output to be the XOR of R and S,the values for W, X, Y and Z are ?

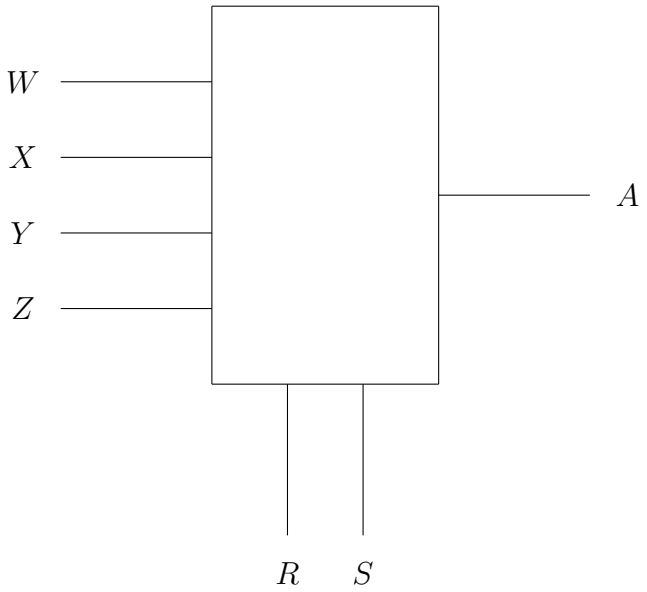

Figure 1: mux

1. $W = 0, X = 0, Y = 1, Z = 1$
2. $W = 1, X = 0, Y = 1, Z = 0$
3. $W = 0, X = 1, Y = 1, Z = 0$
4. $W = 1, X = 1, Y = 0, Z = 0$

2 Introduction

The above diagram is a 4:1 multiplexer where W, X, Y, Z are the inputs of the multiplexer and A is the output of the multiplexer.R, S are the select

lines of the multiplexer,which means:

1. For $R = 0, S = 0$,the first input line W is selected.
2. For $R = 0, S = 1$,the second input line X is selected.
3. For $R = 1, S = 0$,the third input line Y is selected.
4. For $R = 1, S = 1$,the fourth input line Z is selected.

Therefore,the resultant output expression of the multiplexer is $R'S'W + R'SX + RS'Y + RSZ$.

3 Components

COMPONENTS		
Component	Value	Quantity
Resistor	220 ohm	1
Arduino	UNO	1
Seven Segment Display		1
Jumper Wires	M-M	20
Breadboard		1

Table 1: contents

4 Hardware

1. Connect the COM of the seven-segment display to 5V and dot of the seven-segment to the ground.
2. Now connect any one of the pin of the seven-segment to pin no.2(digital).
3. Pin no.s 5,6,7,8 of the arduino should be initially connected to ground.
4. Now move pin no.s 5,6,7,8 accordingly and for the right combination the second pin of the arduino becomes high and the seven segement display glows.

Truth table		
R	S	A
0	0	0
0	1	1
1	0	1
1	1	0

Table 2: truth table

The K-map for this truth table will be a two variable K-map and it will be as follows:

		R	
		0	1
S	0	0	1
	1	1	0

Figure 2: k-map

So,the resultant expression of A is $A = R'S + RS'$.

5 Software

The embedded code for the given circuit is

```
#include <avr/io.h>
#include <util/delay.h>
int A,B,W,X,Y,Z,R,S;
int main (void)
{
        DDRD &= (1<<PD2);
        DDRD &= (1<<PD3);
        DDRD &= (1<<PD4);
        DDRD &= (1<<PD5);
        DDRB &= (1<<PB0);
        DDRB &= (1<<PB1);
        DDRB |= (1<<PB5);
        while(1){
                W = (PIND&(1<<PD2)) == (1<<PIND2);
                X = (PIND&(1<<PD3)) == (1<<PIND3);
                Y = (PIND&(1<<PD4)) == (1<<PIND4);
                Z = (PIND&(1<<PD5)) == (1<<PIND5);
                R = (PINB&(1<<PB0)) == (1<<PINB0);
                S = (PINB&(1<<PB1)) == (1<<PINB1);
                A=(!R&&!S&&W)||(!R&&S&&X)||
                        (R&&!S&&Y)||(R&&S&&Z);
                B=((!R&&S)||(R&&!S));
                if(A==B)
                {
                        PORTB = (1<<5);
                }
                else{
                        PORTB = (0<<5);
                }
        }
        return 0;
}
```