Implementation of the below circuit using assembly

GUNA VARDHAN
gunavardhan.nagamalla@gmail.com
FWC22123
IIT Hyderabad-Future Wireless Communication Assignment

March 2023

Contents

1 Problem

GATE EC-2019
Q.25. In the circuit shown,the clock frequency, i.e.,the frequency of the clock signal ,is 12 KHz.The frequency of the signal at Q2 is ..... KHz.

2 Introduction

The aim is to implement the above sequential circuit using D flip-flops (IC 7474) and to find out the frequency of the signal at Q2(it is given that the frequency of the clock signal is 12KHz).IC 7474 is a dual positive edge triggered D type flip flop,which means it has two separate flip-flop that are triggered by the rising edge of a clock signal.

In the above circuit Q_1,Q_2 are inputs and D_1,D_2 are outputs.So,from the circuit the expressions of D_1 and D_2 are:

$D_1 = Q_1'Q_2'$.
$D_2 = Q_1$.

Below is the transition table of the above circuit which is as follows:

INPUT		OUTPUT	
Q_1	Q_2	D_1	D_2
0	0	1	0
1	0	0	1
0	1	0	0

Table 1: Transition table

3 Components

COMPONENTS		
Component	Value	Quantity
Resistor	=220 Ohm	1
Arduino	UNO	1
Seven Segent Display	Common Anode	1
Decoder	7447	1
Flip Flop	7474	1
Jumper Wires		20
Breadboard		1

Table 2: Components

4 Hardware

IC 7474 is a D flip-flop integrated circuit that is commonly used in digital electronics applications.It is a dual positive edge-triggered by the rising edge of a clock signal.Below is the pin diagram of IC 7474:

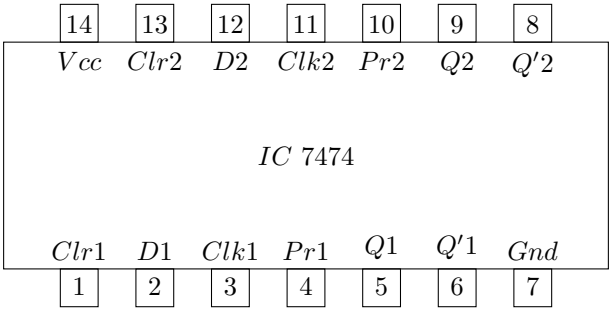

Figure 1: 7474

The connections between the arduino and IC 7474 is as follows:

	INPUT		OUTPUT		CLOCK		VCC			
ARDUINO	D2	D3	D5	D6	D13		5V			
7474	5	9	2	12	3	11	1	4	10	13
7447			1	7				16		

Table 3: connections

5 Software

The code to implement the above circuit is :

.include "/sdcard/Download/FWC/assembly//m328Pdef.inc" ; Initialize registers LDI R16, 0x00 ; Load 00 into R16 LDI R17, 0x00 ; Load 00 into R17 LDI R18, 0x03 ; Load 3 into R18 for comparison later LDI R19, 0x01 ; Load 1 into R19 for toggling

LOOP: ; Wait for falling edge on clock SBIC PIND, 0 ; Skip next instruction if PD0 is low RJMP LOOP ; Jump back to LOOP if clock is high

; Update D1 and D2 flip-flops AND R24, R17 ; (!Q1)*(!Q2) = Q1 Q2 LSL R24 ; Shift left one bit to prepare for D1 OR R24, R16 ; Combine with current state for D1 MOV R16, R17 ; Update $D1_Q - 1 MOV R17, R24; Update D1_Q$

MOV R24, R16 ; Copy $D1_Q to R24 for use in updating D2_Q AND I R24, 0x01; Q1 = D1_Q 0x01 LSL R24; Shift left one bit to prepare for D2 OR R24, R16; Combine with current state for D2 MOV R16, 1 MOV R17, R24; Update D2_Q$

; Output D2 state OUT PORTB, R17 ; Output $D2_Q$

; Check for initial state of 00 and stop the loop CP R16, R18 ; Compare R16 $(D1_Q - 1) with 0x03 BREQ END_L OOP$; $Branch to END_L OOP if equal$

; Wait for rising edge on clock SBIS PIND, 0 ; Skip next instruction if PD0 is high RJMP LOOP ; Jump back to LOOP if clock is low

; Toggle the LED on PORTC EOR R20, R19 ; Toggle R20 (PORTC)

; Jump back to LOOP RJMP LOOP

$END_L OOP$:; $Stop the loop NOP$