

AVR-GCC Assignment

Mohamed Hamdan

August 2022

Problem Statement - A sequential circuit has a single input x and a single output z . The input signal x can occur in groups of 1, 2 and 3 pulses. If $x = 1$ for one clock period, the output z will be 1 for three clock periods before returning to the starting state. If $x = 1$ for two clock periods, the output z will be 1 for two clock periods before returning to the starting state. If $x = 1$ for three clock periods, the output z will be 1 for a single clock period before returning to the starting state. Construct a state diagram and implement your design with D F F s . The circuit when designed acts as a pulse width adjuster.

Hardware

Components

Component	Value	Count
Arduino	uno	1
Flip Flop	7474	2
LED	Red	1
Resistor	220ohm	1
Jumper wires	-	as required

Connections

The following connections are to be read as **IC-Name-IC-pin no:Arduino-pin no** :

- **IC7447(1)** - (1:5.5V), (2:8), (3:13), (4:5.5V), (5:2), (6:None), (7:Gnd), (8:None), (9:3), (10:5.5V), (11:13), (12:9), (13:5.5V), (14:5.5V)
- **IC7447(2)** - (1:5.5V), (2:10), (3:13), (4:5.5V), (5:4), (6:None), (7:Gnd), (8:None), (9:5), (10:5.5V), (11:13), (12:11), (13:5.5V), (14:5.5V)

Connect LED to pin 12 of arduino with the 220ohm resistor in series. Use pin 6 of arduino to input X .

State Diagram

The state diagram shown in 1 can be understood easily by grouping the states according to the clock pulse number. Consider one cycle of the state machine to contain three clock pulses numbered 1, 2 and 3. The states 0, 1 and 2 in the clock-1 group are the entry states to produce the Z values for the successive 2 clock pulses (clock-2 and clock-3). Distinct entry states are necessary in order to remember the input sequence of the previous cycle. While entering into any of these three states, the Z value for clock-1 is already determined by

the input at clock-3 of the previous cycle. In order to keep track of the number of ones in the input sequence X and to output the proper values at Z (determined by the entry state), the remaining 7 states (3-9) are required.

Since the state diagram uses 10 states, the design requires 4 DFFs. Let the present state be denoted by $\mathbf{P} = P_3P_2P_1P_0$ and the next state as $\mathbf{S} = S_3S_2S_1S_0$ with X as input and Z as output.

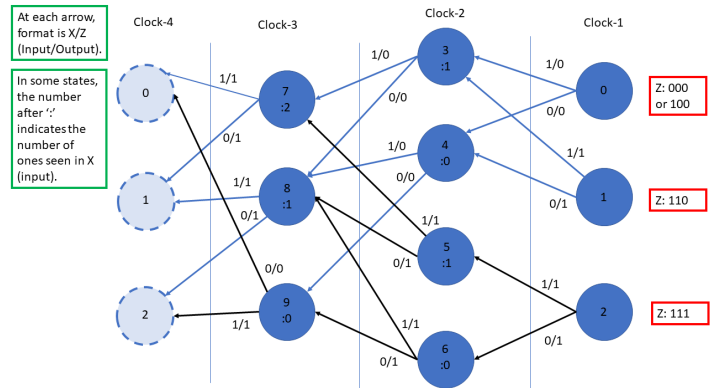


Figure 1: State diagram for pulse width adjuster

Truth table

P_3	P_2	P_1	P_0	X	S_3	S_2	S_1	S_0	Z
0	0	0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	1	1	0
0	0	0	1	0	0	1	0	0	1
0	0	0	1	1	0	0	1	1	1
0	0	1	0	0	0	1	1	0	1
0	0	1	0	1	0	1	0	1	1
0	0	1	1	0	1	0	0	0	0
0	0	1	1	1	0	1	1	1	0
0	1	0	0	0	1	0	0	1	0
0	1	0	0	1	1	0	0	0	0
0	1	0	1	0	1	0	0	0	1
0	1	0	1	1	0	1	1	1	1
0	1	1	0	0	1	0	0	1	1
0	1	1	0	1	1	0	0	0	1
0	1	1	1	0	0	0	0	1	1
0	1	1	1	1	0	0	0	0	1
1	0	0	0	0	0	0	1	0	1
1	0	0	0	1	0	0	0	1	1
1	0	0	1	0	0	0	0	0	0
1	0	0	1	1	0	0	1	0	1
1	0	1	0	0	X	X	X	X	X

P_3	P_2	P_1	P_0	X	S_3	S_2	S_1	S_0	Z
1	0	1	0	1	X	X	X	X	X
1	0	1	1	0	X	X	X	X	X
1	0	1	1	1	X	X	X	X	X
1	1	0	0	0	X	X	X	X	X
1	1	0	0	1	X	X	X	X	X
1	1	0	1	0	X	X	X	X	X
1	1	0	1	1	X	X	X	X	X
1	1	1	0	0	X	X	X	X	X
1	1	1	0	1	X	X	X	X	X
1	1	1	1	0	X	X	X	X	X
1	1	1	1	1	X	X	X	X	X

Minimization using Kmap

$P_3P_2P_1 \backslash P_0X$	00	01	11	10
000	0	0	0	0
001	0	0	0	1
011	1	1	0	0
010	1	1	0	1
110	x	x	x	x
111	x	x	x	x
101	x	x	x	x
100	0	0	0	0

$P_3P_2P_1 \backslash P_0X$	00	01	11	10
000	1	0	0	1
001	1	1	1	0
011	0	0	0	0
010	0	0	1	0
110	x	x	x	x
111	x	x	x	x
101	x	x	x	x
100	0	0	0	0

$P_3P_2P_1 \backslash P_0X$	00	01	11	10
000	0	1	1	0
001	1	0	1	0
011	0	0	0	0
010	0	0	1	0
110	x	x	x	x
111	x	x	x	x
101	x	x	x	x
100	1	0	1	0

$P_3P_2P_1 \backslash P_0X$	00	01	11	10
000	0	1	1	0
001	0	1	1	0
011	1	0	0	1
010	1	0	1	0
110	x	x	x	x
111	x	x	x	x
101	x	x	x	x
100	0	1	0	0

$P_3P_2P_1 \backslash P_0X$		P_0X			
		00	01	11	10
000	0	0	1	1	
001	1	1	0	0	
011	1	1	1	1	
010	0	0	1	1	
110	x	x	x	x	
111	x	x	x	x	
101	x	x	x	x	
100	1	1	1	0	

Boolean expressions

The boolean expressions for **S** and **Z** are:

$$\begin{aligned}
 S_3 &= P_2P'_0 + P_2P'_1X' + P'_2P_1P_0X' \\
 S_2 &= P'_2P_1P'_0 + P'_2P_1X + P'_3P'_2P'_1X' + P_2P'_1P_0X \\
 S_1 &= P'_2P_0X + P'_1P_0X + P_3P'_0X' + P'_3P'_2P'_1X + P'_2P_1P'_0X' \\
 S_0 &= P'_3P'_2X + P'_2P'_0X + P_2P'_0X' + P_2P_1X' + P'_3P'_1P_0X \\
 Z &= P_1P'_0 + P_2P_0 + P_3P'_0 + P_3X + P'_3P'_1P_0
 \end{aligned}$$

Software

Make the connections and connect the arduino to the PC via USB. In the location of choice, type the below commands

1. `svn co https://github.com/Muhammed-Hamdan/iith-fwc-2022-23/trunk/fwc_avr_gcc/avr_gcc_assignment`
2. `cd ide_assignment`
3. `pio run`
4. `pio run t upload`