Clock Implementation on Vaman FPGA using K-Maps and Multiplexing

Siddhanth Yellanki
Department of Electrical Engineering
Indian Institute of Technology Hyderabad
Email: ee24btech11059@iith.ac.in

I. Introduction

The digital clock system described here utilizes Karnaugh maps (K-Maps) for incrementing time units and a multiplexing technique to display time on six seven-segment displays using only a single BCD. This implementation is done in Verilog using a Vaman FPGA.

II. Components

Component	Value	Quantity
Vaman Board		1
USB-UART		1
Seven Segment Display		6
Push Buttons		4
IC 7447		1
Jumper Wires	F-M	30
Wires		
Breadboard		2

Table 1.0

III. Circuit Connections

A. Connections to Vaman

Make the button connections and IC 7447 connections to the Vaman FPGA as per the table below.

Item	Vaman Board	Name
Button 1	PYGMY 1	Pause / Play
Button 2	PYGMY 2	Increment Sec
Button 3	PYGMY 3	Increment Min
Button 4	PYGMY 4	Increment Hour
IC 7447 Pin 7	PYGMY 5	Four Bit Input
IC 7447 Pin 1	PYGMY 6	Four Bit Input
IC 7447 Pin 2	PYGMY 7	Four Bit Input
IC 7447 Pin 6	PYGMY 8	Four Bit Input
Display 1 Power	PYGMY 9	Multiplexing
Display 2 Power	PYGMY 10	Multiplexing
Display 3 Power	PYGMY 11	Multiplexing
Display 4 Power	PYGMY 12	Multiplexing
Display 5 Power	PYGMY 13	Multiplexing
Display 6 Power	PYGMY 14	Multiplexing

Table 2.0

B. Connections from Seven Segment to BCD

Make the seven-segment connections identical for all seven segments. In total, there should only be 7 wires of output coming from the seven-segment display array.

IC 7447	Seven Segment (All)	Name
Pin 13	a	Controls segment a
Pin 12	b	Controls segment b
Pin 11	c	Controls segment c
Pin 10	d	Controls segment d
Pin 9	e	Controls segment e
Pin 15	f	Controls segment f
Pin 14	g	Controls segment g
Pin 8	Ground	Ground Supply
Pin 16	5V	Power Supply

Table 3.0

IV. Multiplexing Technique

Multiplexing is achieved by connecting all inputs of the seven-segment displays to a single BCD. Digital pins are connected to the common cathode/anode of each display, allowing selective activation of each display to show the BCD output. The displays are alternated with a very small time gap, creating the illusion of simultaneous operation.

V. K-Map Incrementing Logic

The incrementing logic for each display is implemented using decade counters. For the unit's place of the seconds, the logic is as follows:

Z	Y	X	W	D	C	B	A
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0

$$A_1 = \overline{W_1}; \tag{1}$$
$$B_1 = (W_1 \wedge \overline{X_1} \wedge \overline{Z_1}) \vee (\overline{W_1} \wedge X_1); \tag{2}$$
$$C_1 = (\overline{X_1} \wedge Y_1) \vee (\overline{W_1} \wedge Y_1) \vee (W_1 \wedge X_1 \wedge \overline{Y_1}); \tag{3}$$
$$D_1 = (\overline{W_1} \wedge Z_1) \vee (W_1 \wedge X_1 \wedge Y_1). \tag{4}$$

For the ten's place of the seconds, which varies from 0 to 5:

Z	Y	X	W	D	C	B	A
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	0	0	0

$$A_2 = \overline{W_2}; \tag{5}$$
$$B_2 = (\overline{Y_2} \wedge \overline{X_2} \wedge W_2) \vee (\overline{W_2} \wedge X_2); \tag{6}$$
$$C_2 = (\overline{W_2} \wedge Y_2) \vee (X_2 \wedge W_2); \tag{7}$$
$$D_2 = 0. \tag{8}$$

To synchronize the ten's place increment with the unit's place reaching 9, an additional variable C is used:

$$C = W_1 \wedge \overline{X_1} \wedge \overline{Y_1} \wedge Z_1 \tag{9}$$
$$A_2 = (A_2 \wedge C) \vee (W_2 \wedge \overline{C}) \tag{10}$$
$$B_2 = (B_2 \wedge C) \vee (X_2 \wedge \overline{C}) \tag{11}$$
$$C_2 = (C_2 \wedge C) \vee (Y_2 \wedge \overline{C}) \tag{12}$$
$$D_2 = (D_2 \wedge C) \vee (Z_2 \wedge \overline{C}). \tag{13}$$

Now, using the above logic, the ten's digit of seconds only updates when the unit's digit previously was 9. This logic can be reapplied for the next display, i.e., the unit's digit of the minutes:

$$A_3 = \overline{W_3}; \tag{14}$$
$$B_3 = (W_3 \wedge \overline{X_3} \wedge \overline{Z_3}) \vee (\overline{W_3} \wedge X_3); \tag{15}$$
$$C_3 = (\overline{X_3} \wedge Y_3) \vee (\overline{W_3} \wedge Y_3) \vee (W_3 \wedge X_3 \wedge \overline{Y_3}); \tag{16}$$
$$D_3 = (\overline{W_3} \wedge Z_3) \vee (W_3 \wedge X_3 \wedge Y_3). \tag{17}$$

The value of C for this case is:

$$C = W_2 \wedge \overline{X_2} \wedge Y_2 \wedge \overline{Z_2} \wedge W_1 \wedge \overline{X_1} \wedge \overline{Y_1} \wedge Z_1. \tag{18}$$

VI. Control Implementation

1) Pressing the first button will pause/play the clock.
2) Pressing the second button while paused will increment the seconds.
3) Pressing the third button while paused will increment the minutes.
4) Pressing the fourth button while paused will increment the hours.

To increment the minutes, the incrementing logic is run 60 times. Similarly, incrementing the hours requires running the loop 3600 times.

VII. Execution

A. Download the repository

```
git clone https://github.com/ysiddhanth/vaman.git
cd vaman
```

B. Locate the folder codes in the Clock folder.

```
cd Clock/codes
```

C. Generate the .bin file

```
ql_symbiflow -compile -src . -d ql-eos-s3 -P pu64 -t main -v ma
```

D. Dump .bin file onto the Vaman FPGA

```
sudo python3 tinyfpgab --port /dev/ttyACM0 --appfpga main.bin -
```

E. Hardware Build

- Connect the seven-segment displays to the breadboard and connect all their outputs together.
- Make connections to the IC7447 and seven-segment display array according to the above table.
- Connect the IC7447 and the buttons to the Vaman FPGA according to the table as well.

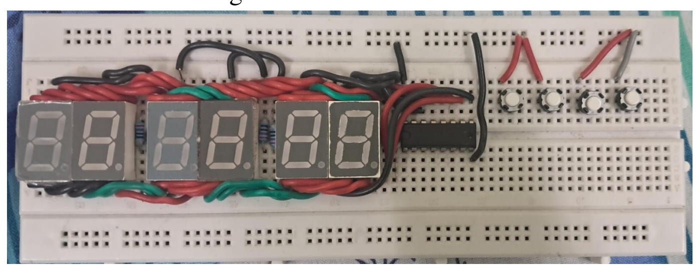

Figure 1 - Final Clock