# Verilog

# Mini-Project

# TOPIC: Buzzer Controlled Timer using FPGA

**Course Coordinator:**  **Dr. Kodeeswara Kumaran G**

**Course Code: EEE554**

|  |  |
| --- | --- |
| **NAME** | **USN** |
| **Rohit Shailesh** | **1MS20EE043** |
| **Roopesh GL** | **1MS20EE044** |

# INDEX

|  |  |  |
| --- | --- | --- |
| Sl. NO | Topic | Page No |
| 1 | Abstract | 3 |
| 2 | Introduction | 3 |
| 3 | Theory | 4 |
| 4 | Algorithm | 5 |
| 5 | Conclusion | 8 |

# Buzzer Controlled Timer using FPGA

**Department of Electrical & Electronics Engineering, MS Ramaiah Institute of Technology**

**I. Abstract:**

To design and develop a sixty second timer in DE10-lite Student edition FPGA development board and integrate it with a buzzer which beeps every 59 seconds. The scope of the project is to develop optimal and efficient counters which can be integrated into modules such as a timer.

**II. Introduction:**

A Digital timer is a specialized type of clock that is used for measuring specific time intervals. A timer module has many applications in day-to-day life as they are used in Stopwatches, Oven timer, Digital clocks and even used in rocket launches. Unlike an Analog timer, digital timers can be programmed. This is one of the reasons why digital timer can be considered better than their analog counterpart. They offer flexibility and can be programmed as per needs.

Counter is a sequential circuit that goes through a prescribed sequence of states upon the application of input pulses. A timer module is essentially a counter circuit that is connected to a seven-segment display as output. In our project, we have used an external buzzer which beeps when the timer reaches 59s. The buzzer is controlled using pulse width modulation (PWM). That is, when different frequencies are given, the buzzer beeps differently. PWM, often known as pulse width modulation, is a method for producing analogue effects using digital technology. A square wave is produced via digital control, which is a signal that is toggled on and off.

**III. Theory**

The DE-10 lite FPGA development board has a clock of 50MHz Frequency. We have considered four different seven segment displays, in which the first two represent the seconds and the latter represent Deci and cent seconds respectively. To obtain these clocks we need to use intermediate registers which can create a delay for our requirement. The intermediate registers (say i1, i2, i3, i4) are incremented until a certain value is reached. For a 10s clock, the value is 500M, for a 1s clock 50M and so on. Once this value is reached, the register is reset so that it starts from zero all over again.

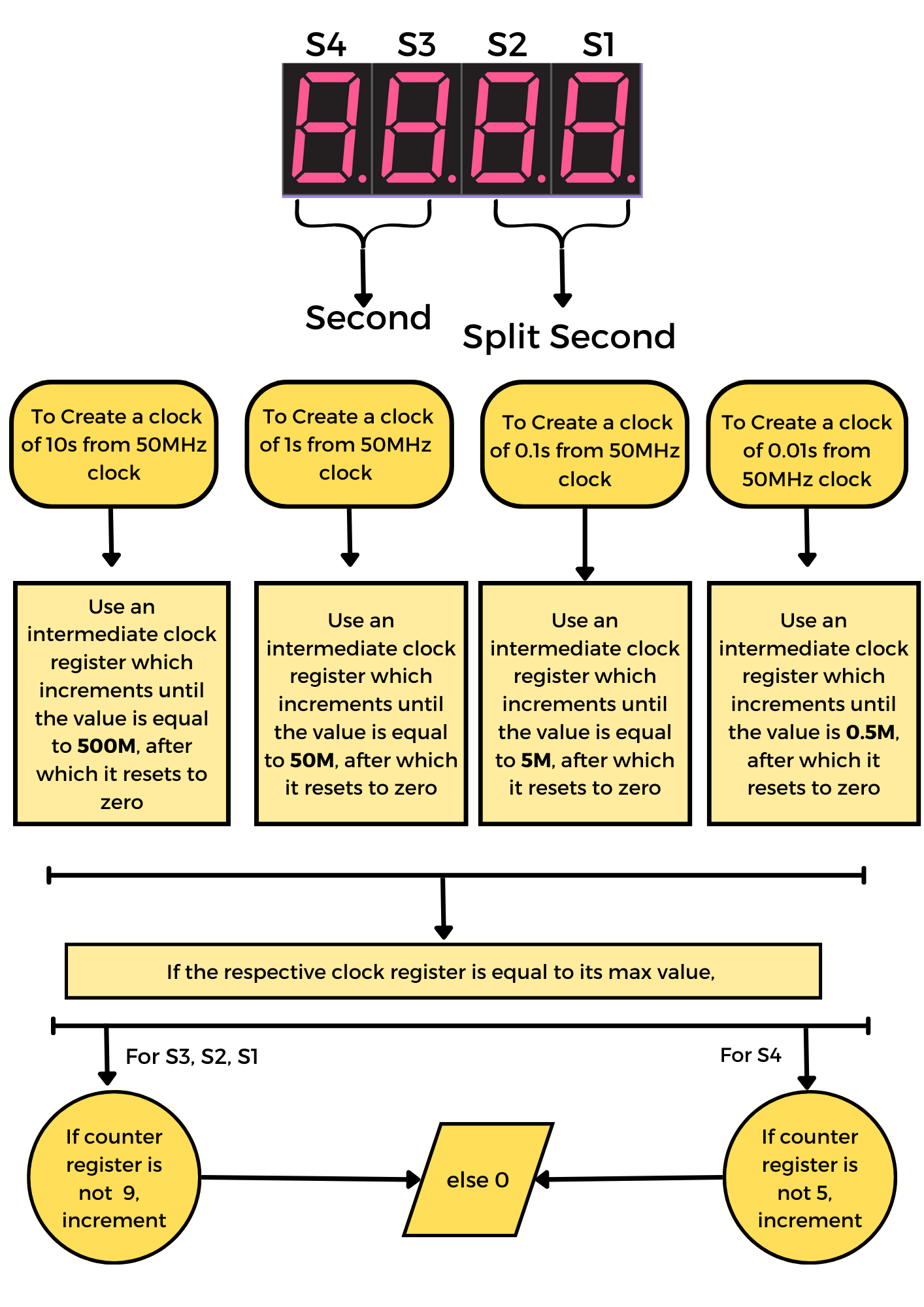
But when the register reaches the required value, we use intermediate counter registers which help in counting the seconds in timer. Since there are 4 seven segment displays, we will need to use four 4-bit registers. For example, let’s consider x1, x2, x3, x4 as the registers, wherein x1 and x2 represent split seconds and x3 and x4 represent the least significant and most significant bit of seconds respectively. The counter registers x1, x2, x3 are incremented if they are not equal to 9. Otherwise, they are assigned to zero. Counter register x4 is incremented if it is not equal to 5 otherwise it is assigned as zero. This is the central logic for counting in timer.

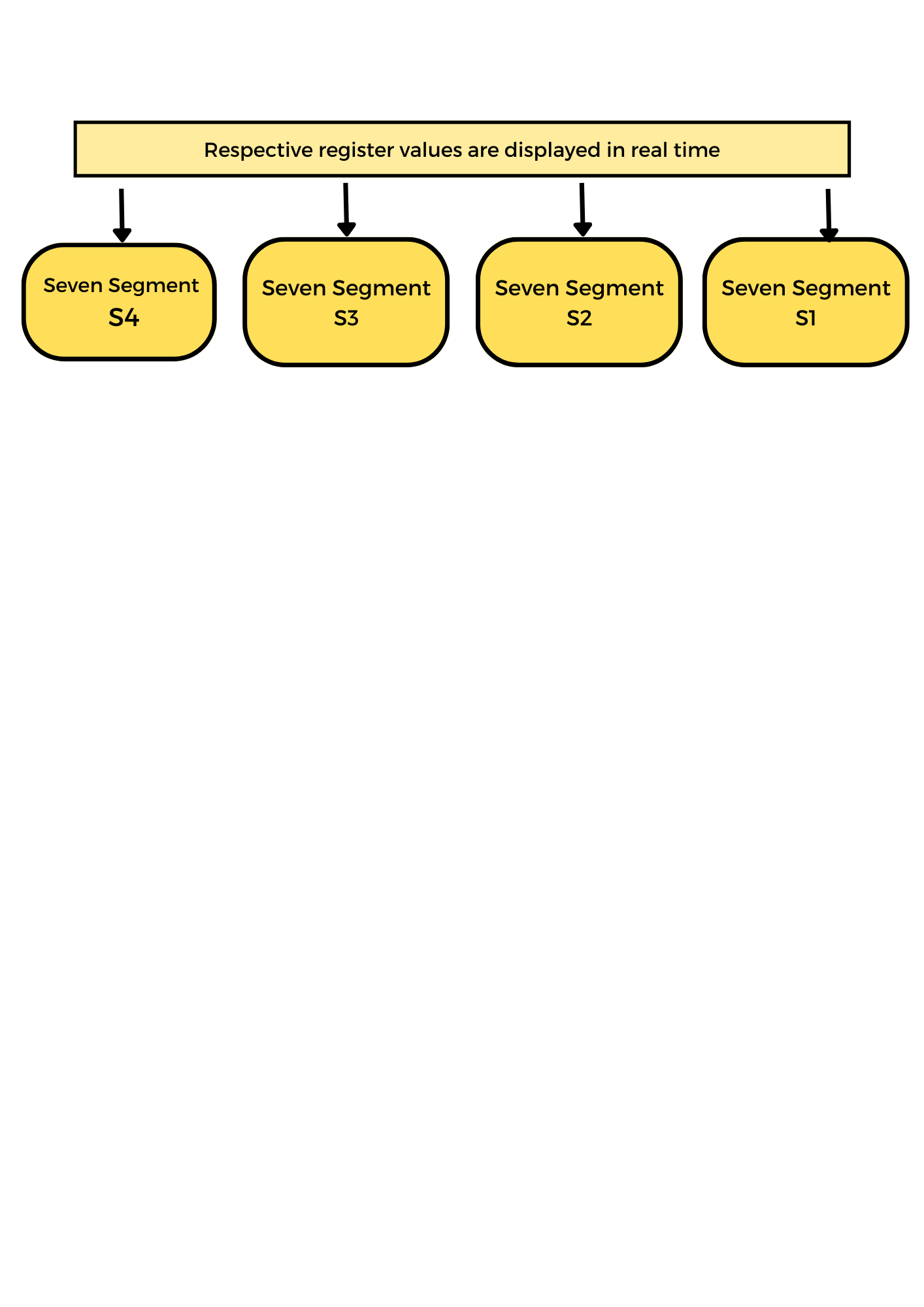
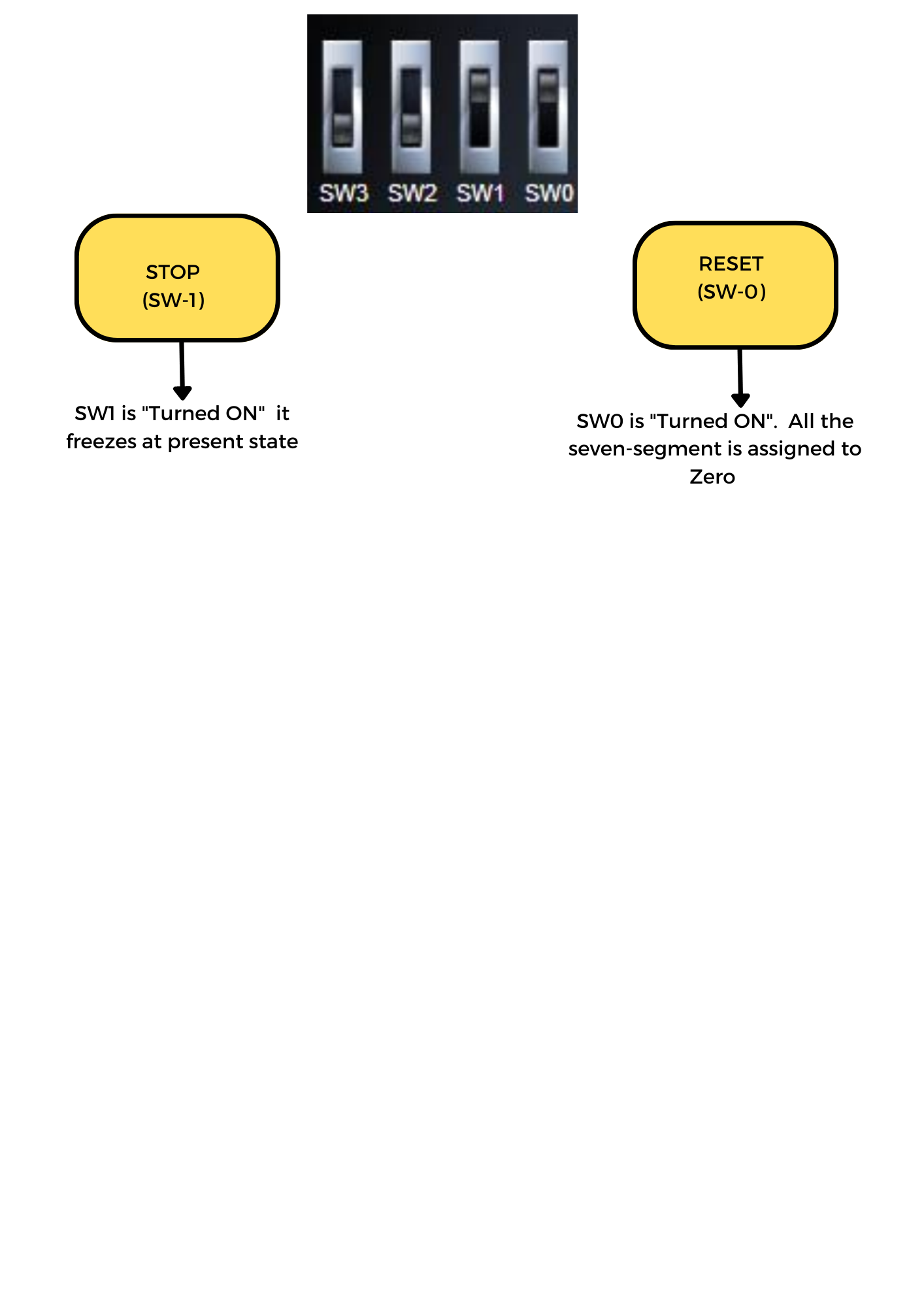
We also need control inputs for the timer such as a reset button and a stop button. The reset function is achieved through a switch. If the switch is turned on, all the counter registers and clock registers are assigned zero. The stop function is also controlled through a switch. If it is turned on all the clock registers and counter registers are frozen.

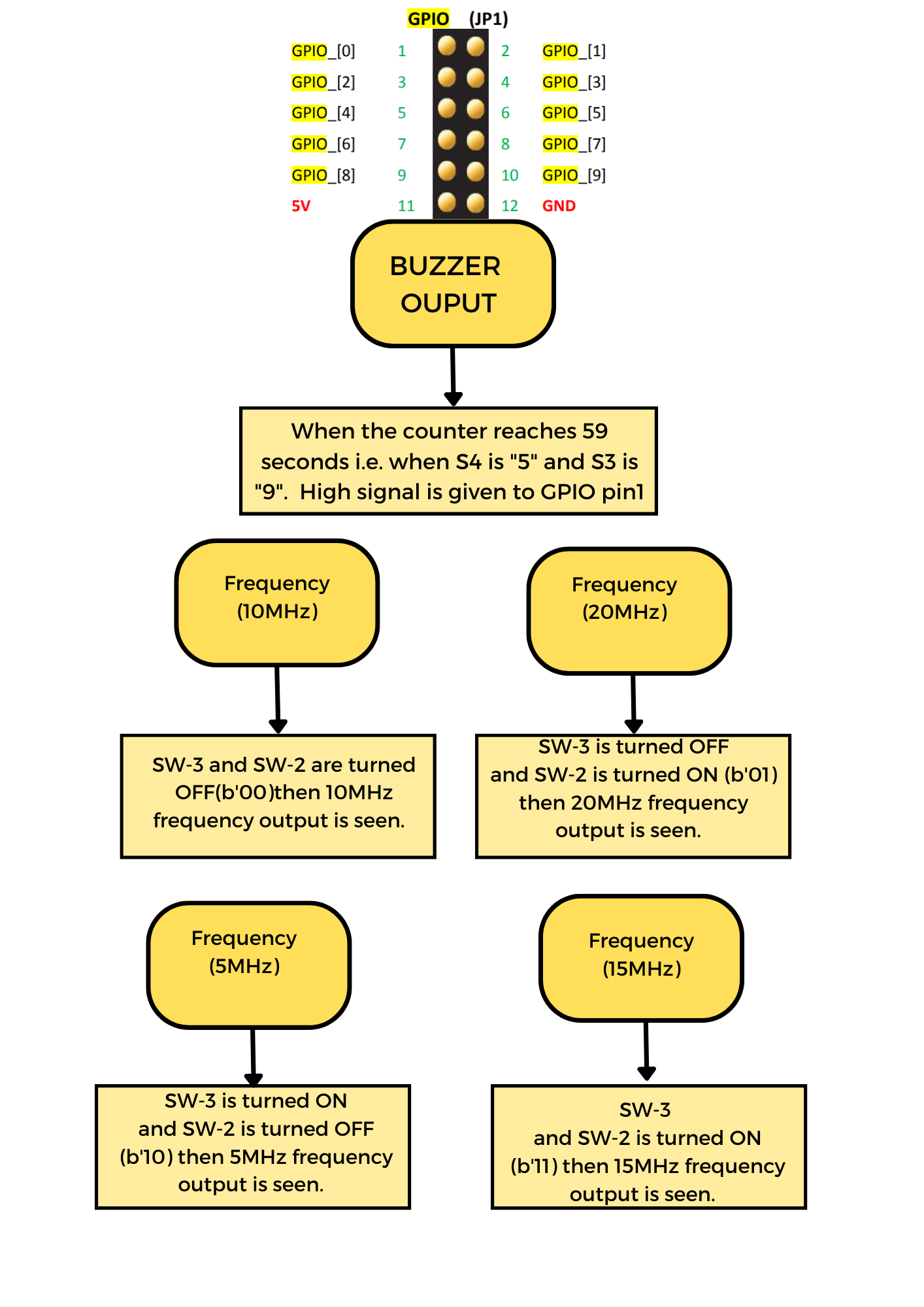
The counter registers are assigned to the seven-segment display via a simple code. An external buzzer is connected to the FPGA GPIO pins, which enable us to observe the analog output for a digital input. One of the buzzer terminals(negative terminal) is given to ground and the positive is given to a pin which can be programmed. In our case, it was given to pin 1.

The client is given four different options for the buzzer control. For the first option, the buzzer toggles between high and low for every 10Mhz frequency.

This is handled by PWM, the output of buzzer is said to be high if it is lower than a certain frequency say, 20MHz. If different frequency are applied, we can notice the difference in output of buzzer.







**CONCLUSION:**  When the timer reached sixty seconds, it was observed that the buzzer buzzed as per the input given by the client. The frequency of the buzzer changed according to the input that was given to the FPGA switch. It was also noted that stop and reset switches were operational.